The Journal of Machine Learning Research
Print-Archive Edition

Volume 18 (papers from 2017)
Issues 74–149



Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Print-Archive Edition
Volume 18 (papers from 2017)
Issues 74–149

The Journal of Machine Learning Research (JMLR) is an open access journal. All articles published in JMLR are freely available via electronic distribution. This Print-Archive Edition is published annually as a means of archiving the contents of the journal in perpetuity. The contents of this edition are articles published electronically in JMLR in 2017.

JMLR is abstracted in ACM Computing Reviews, INSPEC, and Psychological Abstracts/PsycINFO.

JMLR is a publication of Journal of Machine Learning Research, Inc. For further information regarding JMLR, including open access to articles, visit http://www.jmlr.org/.

JMLR Print-Archive Edition is a publication of Microtome Publishing under agreement with Journal of Machine Learning Research, Inc. For further information regarding the Print-Archive Edition, including subscription and distribution information and background on open-access print archiving, visit Microtome Publishing at http://www.mtome.com/.

Collection copyright © 2017 The Journal of Machine Learning Research, Inc. and Microtome Publishing.
Individual articles copyright © 2017 by their respective authors and distributed under a Creative Commons Attribution 4.0 International License.

ISSN 1532-4435 (print)
ISSN 1533-7928 (online)

JMLR Editorial Board

Editor-in-Chief
Bernhard Schölkopf, MPI for Intelligent Systems, Germany

Editor-in-Chief
Kevin Murphy, Google Research, USA

Managing Editor
Aron Culotta, Illinois Institute of Technology, USA

Production Editor
Charles Sutton, University of Edinburgh, UK

JMLR Web Master
Chiyuan Zhang, Massachusetts Institute of Technology, USA

JMLR Action Editors
Ryan Adams, Princeton University, USA. **Shivani Agarwal**, University of Pennsylvania, USA. **Edoardo M. Airoldi**, Harvard University, USA. **Anima Anandkumar**, California Institute of Technology, USA. **Peter Auer**, University of Leoben, Austria. **David Barber**, University College London, UK. **Samy Bengio**, Google Research, USA. **Yoshua Bengio**, Université de Montréal, Canada. **Jeff Bilmes**, University of Washington, USA. **Karsten Borgwardt**, ETH Zurich, Switzerland. **Léon Bottou**, Facebook AI Research. **François Caron**, University of Oxford, United Kingdom. **Miguel A. Carreira-Perpinan**, University of California, Merced, USA. **Alexander Clark**, King's College London, UK. **Corinna Cortes**, Google Research, USA. **Koby Crammer**, Technion, Israel. **Sanjoy Dasgupta**, University of California, San Diego, USA. **Inderjit S. Dhillon**, University of Texas, Austin, USA. **Jennifer Dy**, Northeastern University, USA. **Gal Elidan**, Hebrew University, Israel. **Charles Elkan**, University of California at San Diego, USA. **Barbara Engelhardt**, Princeton University, USA. **Rob Fergus**, New York University, USA. **Kenji Fukumizu**, The Institute of Statistical Mathematics, Japan **Amir Globerson**, Tel Aviv University, Israel **Moises Goldszmidt**, Microsoft Research, USA. **Russ Greiner**, University of Alberta, Canada. **Arthur Gretton**, University College London, UK. **Maya Gupta**, Google Research, USA. **Isabelle Guyon**, ClopiNet, USA. **Moritz Hardt**, Google Research, USA. **Thomas Hofmann**, ETH Zurich, Switzerland **Bert Huang**, Virginia Tech, USA. **Aapo Hyvärinen**, University of Helsinki, Finland **Alex Ihler**, University of California, Irvine, USA. **Tommi Jaakkola**, Massachusetts Institute of Technology, USA. **Samuel Kaski**, Aalto University, Finland **Sathiya Keerthi**, Microsoft Research, USA. **Emtiyaz Khan**, RIKEN Center for Advanced Intelligence, Japan **George Konidaris**, Duke University, USA. **Andreas Krause**, ETH Zurich, Switzerland **Sanjiv Kumar**, Google Research, USA. **Christoph Lampert**, Institute of Science and Technology, Austria **Daniel Lee**, University of Pennsylvania, USA. **Qiang Liu**, Dartmouth College, USA. **Gábor Lugosi**, Pompeu Fabra University, Spain **Michael Mahoney**, University of California at Berkeley, USA. **Shie Mannor**, Technion, Israel **Jon McAuliffe**, University of California at Berkeley, USA. **Robert E. McCulloch**, University of Chicago, USA. **Chris Meek**, Microsoft Research, USA. **Qiaozhu Mei**, University of Michigan, USA. **Vahab Mirrokni**, Google Research, USA. **Mehryar Mohri**, New York University, USA. **Joris Mooij**, University of Amsterdam, Netherlands **Boaz Nadler**, Weizmann Institute of Science, Israel. **Long Nguyen**, University of Michigan, USA. **Sebastian Nowozin**, Microsoft Research, Cambridge, UK **Una-May O'Reilly**, Massachusetts Institute of Technology, USA. **Manfred**

Opper, Technical University of Berlin, Germany **Laurent Orseau**, Google Deepmind, USA. **Luis Ortiz**, University of Michigan - Dearborn, USA. **Jie Peng**, University of California, Davis, USA. **Jan Peters**, Technische Universitaet Darmstadt, Germany **Avi Pfeffer**, Charles River Analytics, USA. **Joelle Pineau**, McGill University, Canada. **Massimiliano Pontil**, Istituto Italiano di Tecnologia (Italy), University College London (UK) **Luc de Raedt**, Katholieke Universiteit Leuven, Belgium **Alexander Rakhlin**, University of Pennsylvania, USA. **Ben Recht**, University of California, Berkeley, USA. **Lorenzo Rosasco**, Massachusetts Institute of Technology, USA. **Saharon Rosset**, Tel Aviv University, Israel **Ruslan Salakhutdinov**, University of Toronto, Canada. **Sujay Sanghavi**, University of Texas, Austin, USA. **Mark Schmidt**, University of British Columbia, Canada. **Marc Schoenauer**, INRIA Saclay, France **John Shawe-Taylor**, University College London, UK **Xiaotong Shen**, University of Minnesota, USA. **David Sontag**, New York University, USA. **Peter Spirtes**, Carnegie Mellon University, USA. **Nathan Srebro**, Toyota Technical Institute at Chicago, USA. **Ingo Steinwart**, University of Stuttgart, Germany **Amos Storkey**, University of Edinburgh, UK **Csaba Szepesvari**, University of Alberta, Canada. **Olivier Teytaud**, INRIA Saclay, France **Ivan Titov**, University of Amsterdam, Netherlands **Ryan Tibshirani**, Carnegie Mellon University **Ryota Tomioka**, Microsoft Research Cambridge, UK **Koji Tsuda**, National Institute of Advanced Industrial Science and Technology, Japan **Zhuowen Tu**, University of California at San Diego, USA. **Nicolas Vayatis**, ENS Cachan, France **S V N Vishwanathan**, Purdue University, USA. **Manfred Warmuth**, University of California at Santa Cruz, USA. **Kilian Weinberger**, Cornell University, USA. **David Wipf**, Microsoft Research Asia, China **Daniela Witten**, University of Washington. **Stefan Wrobel**, Fraunhofer IAIS and University of Bonn, Germany **Eric Xing**, Carnegie Mellon University, USA. **Tong Zhang**, Baidu Inc, China **Zhihua Zhang**, Peking University, China **Hui Zou**, University of Minnesota, USA.

JMLR MLOSS Editors

Alexandre Gramfort, INRIA, Université Paris-Saclay, France **Antti Honkela**, University of Helsinki, Finland **Balázs Kégl**, CNRS / Université Paris-Saclay, France **Cheng Soon Ong**, Australian National University, Australia

JMLR Editorial Board

Naoki Abe, IBM TJ Watson Research Center, USA **Yasemin Altun**, Google Inc, Switzerland **Jean-Yves Audibert**, CERTIS, France **Jonathan Baxter**, Australia National University, Australia **Richard K. Belew**, University of California at San Diego, USA **Kristin Bennett**, Rensselaer Polytechnic Institute, USA **Christopher M. Bishop**, Microsoft Research, Cambridge, UK **Lashon Booker**, The Mitre Corporation, USA **Henrik Boström**, Stockholm University/KTH, Sweden **Craig Boutilier**, Google Research, USA **John Patrick Cunningham**, Columbia University, USA **Nello Cristianini**, University of Bristol, UK **Peter Dayan**, University College, London, UK **Dennis DeCoste**, eBay Research, USA **Thomas Dietterich**, Oregon State University, USA **Saso Dzeroski**, Jozef Stefan Institute, Slovenia **Ran El-Yaniv**, Technion, Israel **Peter Flach**, Bristol University, UK **Dan Geiger**, Technion, Israel **Claudio Gentile**, Università degli Studi dell'Insubria, Italy **Sally Goldman**, Google Research, USA **Thore Graepel**, Google DeepMind and University College London, UK **Tom Griffiths**, University of California at Berkeley, USA **Carlos Guestrin**, University of Washington, USA **Stefan Harmeling**, University of Düsseldorf, Germany **David Heckerman**, Microsoft Research, USA **Katherine Heller**, Duke University, USA **Philipp Hennig**, MPI for Intelligent Systems, Germany **Larry Hunter**, University of Colorado, USA **Jens Kober**, Delft University of Technology, Netherlands **Risi Kondor**, University of Chicago, USA **Aryeh Kontorovich**, Ben-Gurion University of the Negev, Israel **Samory Kpotufe**,

Princeton University, USA **John Lafferty**, University of Chicago, USA **Erik Learned-Miller**, University of Massachusetts, Amherst, USA **Fei Fei Li**, Stanford University, USA **Yi Lin**, University of Wisconsin, USA **Wei-Yin Loh**, University of Wisconsin, USA **Richard Maclin**, University of Minnesota, USA **Sridhar Mahadevan**, University of Massachusetts, Amherst, USA **Vikash Mansingkha**, Massachusetts Institute of Technology, USA **Yishay Mansour**, Tel-Aviv University, Israel **Jon McAuliffe**, University of California, Berkeley, USA **Andrew McCallum**, University of Massachusetts, Amherst, USA **Raymond J. Mooney**, University of Texas, Austin, USA **Klaus-Robert Muller**, Technical University of Berlin, Germany **Kevin Murphy**, Google, USA **Guillaume Obozinski**, Ecole des Ponts - ParisTech, France **Pascal Poupart**, University of Waterloo, Canada **Konrad Rieck**, University of Göttingen, Germany **Cynthia Rudin**, Massachusetts Institute of Technology, USA **Suchi Saria**, Johns Hopkins University, USA **Robert Schapire**, Princeton University, USA **Fei Sha**, University of Southern California, USA **Shai Shalev-Shwartz**, Hebrew University of Jerusalem, Israel **Padhraic Smyth**, University of California, Irvine, USA **Bharath Sriperumbudur**, Pennsylvania State University, USA **Alexander Statnikov**, New York University, USA **Jean-Philippe Vert**, Mines ParisTech, France **Martin J. Wainwright**, University of California at Berkeley, USA **Chris Watkins**, Royal Holloway, University of London, UK **Max Welling**, University of Amsterdam, Netherlands **Chris Williams**, University of Edinburgh, UK **Alice Zheng**, GraphLab, USA

JMLR Advisory Board

Shun-Ichi Amari, RIKEN Brain Science Institute, Japan **Andrew Barto**, University of Massachusetts at Amherst, USA **Thomas Dietterich**, Oregon State University, USA **Jerome Friedman**, Stanford University, USA **Stuart Geman**, Brown University, USA **Geoffrey Hinton**, University of Toronto, Canada **Michael Jordan**, University of California at Berkeley at USA **Leslie Pack Kaelbling**, Massachusetts Institute of Technology, USA **Michael Kearns**, University of Pennsylvania, USA **Steven Minton**, InferLink, USA **Tom Mitchell**, Carnegie Mellon University, USA **Stephen Muggleton**, Imperial College London, UK **Kevin Murphy**, Google, USA **Nils Nilsson**, Stanford University, USA **Tomaso Poggio**, Massachusetts Institute of Technology, USA **Ross Quinlan**, Rulequest Research Pty Ltd, Australia **Stuart Russell**, University of California at Berkeley, USA **Lawrence Saul**, University of California at San Diego, USA **Terrence Sejnowski**, Salk Institute for Biological Studies, USA **Richard Sutton**, University of Alberta, Canada **Leslie Valiant**, Harvard University, USA

Journal of Machine Learning Research

Volume 18 (papers from 2017)
Volume year 2018

Part A

18(1):1–29 **Averaged Collapsed Variational Bayes Inference**
Katsuhiko Ishiguro, Issei Sato, Naonori Ueda

18(2):1–45 **Scalable Influence Maximization for Multiple Products in Continuous-Time Diffusion Networks**
Nan Du, Yingyu Liang, Maria-Florina Balcan, Manuel Gomez-Rodriguez, Hongyuan Zha, Le Song

18(3):1–35 **Local algorithms for interactive clustering**
Pranjal Awasthi, Maria Florina Balcan, Konstantin Voevodski

18(4):1–5 **SnapVX: A Network-Based Convex Optimization Solver**
David Hallac, Christopher Wong, Steven Diamond, Abhijit Sharang, Rok Sosič, Stephen Boyd, Jure Leskovec

18(5):1–30 **Communication-efficient Sparse Regression**
Jason D. Lee, Qiang Liu, Yuekai Sun, Jonathan E. Taylor

18(6):1–36 **Improving Variational Methods via Pairwise Linear Response Identities**
Jack Raymond, Federico Ricci-Tersenghi

18(7):1–37 **Distributed Sequence Memory of Multidimensional Inputs in Recurrent Networks**
Adam S. Charles, Dong Yin, Christopher J. Rozell

18(8):1–35 **Persistence Images: A Stable Vector Representation of Persistent Homology**
Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, Lori Ziegelmeier

18(9):1–57 **Spectral Clustering Based on Local PCA**
Ery Arias-Castro, Gilad Lerman, Teng Zhang

18(10):1–33 **On Perturbed Proximal Gradient Algorithms**
Yves F. Atchadé, Gersende Fort, Eric Moulines

18(11):1–39 **Differential Privacy for Bayesian Inference through Posterior Sampling**
Christos Dimitrakakis, Blaine Nelson, Zuhe Zhang, Aikaterini Mitrokotsa, Benjamin I. P. Rubinstein

18(12):1–5 **Refinery: An Open Source Topic Modeling Web Platform**
Daeil Kim, Benjamin F. Swanson, Michael C. Hughes, Erik B. Sudderth

18(13):1–43 **Using Conceptors to Manage Neural Long-Term Memories for Temporal Patterns**
Herbert Jaeger

18(14):1–45 **Automatic Differentiation Variational Inference**
Alp Kucukelbir, Dustin Tran, Rajesh Ranganath, Andrew Gelman, David M. Blei

18(15):1–35 **Empirical Evaluation of Resampling Procedures for Optimising SVM Hyperparameters**
Jacques Wainer, Gavin Cawley

18(16):1–49 **A Unified Formulation and Fast Accelerated Proximal Gradient Method for Classification**
Naoki Ito, Akiko Takeda, Kim-Chuan Toh

18(17):1–5 **Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning**
Guillaume Lemaître, Fernando Nogueira, Christos K. Aridas

18(18):1–65 **Information-Geometric Optimization Algorithms: A Unifying Picture via Invariance Principles**
Yann Ollivier, Ludovic Arnold, Anne Auger, Nikolaus Hansen

18(19):1–53 **Breaking the Curse of Dimensionality with Convex Neural Networks**
Francis Bach

18(20):1–32 **Memory Efficient Kernel Approximation**
Si Si, Cho-Jui Hsieh, Inderjit S. Dhillon

18(21):1–38 **On the Equivalence between Kernel Quadrature Rules and Random Feature Expansions**
Francis Bach

18(22):1–40 **Analyzing Tensor Power Method Dynamics in Overcomplete Regime**
Animashree Anandkumar, Rong Ge, Majid Janzamin

18(23):1–5 **JSAT: Java Statistical Analysis Tool, a Library for Machine Learning**
Edward Raff

18(24):1–29 **Identifying a Minimal Class of Models for High–dimensional Data**
Daniel Nevo, Ya'acov Ritov

18(25):1–5 **Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA**
Lars Kotthoff, Chris Thornton, Holger H. Hoos, Frank Hutter, Kevin Leyton-Brown

18(26):1–5 **POMDPs.jl: A Framework for Sequential Decision Making under Uncertainty**
Maxim Egorov, Zachary N. Sunberg, Edward Balaban, Tim A. Wheeler, Jayesh K. Gupta, Mykel J. Kochenderfer

18(27):1–32 **Generalized Pólya Urn for Time-Varying Pitman-Yor Processes**
François Caron, Willie Neiswanger, Frank Wood, Arnaud Doucet, Manuel Davy

18(28):1–39 **Particle Gibbs Split-Merge Sampling for Bayesian Inference in Mixture Models**
Alexandre Bouchard-Côté, Arnaud Doucet, Andrew Roth

18(29):1–53 **Certifiably Optimal Low Rank Factor Analysis**
Dimitris Bertsimas, Martin S. Copenhaver, Rahul Mazumder

18(30):1–52 **Group Sparse Optimization via lp,q Regularization**
Yaohua Hu, Chong Li, Kaiwen Meng, Jing Qin, Xiaoqi Yang

18(31):1–32 **Preference-based Teaching**
Ziyuan Gao, Christoph Ries, Hans U. Simon, Sandra Zilles

18(32):1–40 **Nonparametric Risk Bounds for Time-Series Forecasting**
Daniel J. McDonald, Cosma Rohilla Shalizi, Mark Schervish

18(33):1–39 **Online Bayesian Passive-Aggressive Learning**
Tianlin Shi, Jun Zhu

18(34):1–41 **Asymptotic Analysis of Objectives Based on Fisher Information in Active Learning**
Jamshid Sourati, Murat Akcakaya, Todd K. Leen, Deniz Erdogmus, Jennifer G. Dy

18(35):1–39 **A Spectral Algorithm for Inference in Hidden semi-Markov Models**
Igor Melnyk, Arindam Banerjee

18(36):1–30 **Simplifying Probabilistic Expressions in Causal Inference**
Santtu Tikka, Juha Karvanen

18(37):1–22 **Nearly optimal classification for semimetrics**
Lee-Ad Gottlieb, Aryeh Kontorovich, Pinhas Nisnevitch

18(38):1–39 **Bridging Supervised Learning and Test-Based Co-optimization**
Elena Popovici

18(39):1–5 **GFA: Exploratory Analysis of Multiple Data Sources with Group Factor Analysis**
Eemeli Leppäaho, Muhammad Ammad-ud-din, Samuel Kaski

18(40):1–6 **GPflow: A Gaussian Process Library using TensorFlow**
Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, James Hensman

18(41):1–49 **COEVOLVE: A Joint Point Process Model for Information Diffusion and Network Evolution**
Mehrdad Farajtabar, Yichen Wang, Manuel Gomez-Rodriguez, Shuang Li, Hongyuan Zha, Le Song

18(42):1–60 **Learning Local Dependence In Ordered Data**
Guo Yu, Jacob Bien

18(43):1–29 **Bayesian Learning of Dynamic Multilayer Networks**
Daniele Durante, Nabanita Mukherjee, Rebecca C. Steorts

18(44):1–29 **Time-Accuracy Tradeoffs in Kernel Prediction: Controlling Prediction Quality**
Samory Kpotufe, Nakul Verma

18(45):1–21 **Asymptotic behavior of Support Vector Machine for spiked population model**
Hanwen Huang

18(46):1–22 **Distributed Semi-supervised Learning with Kernel Ridge Regression**
Xiangyu Chang, Shao-Bo Lin, Ding-Xuan Zhou

18(47):1–43 **On Markov chain Monte Carlo methods for tall data**
Rémi Bardenet, Arnaud Doucet, Chris Holmes

18(48):1–33 **Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers**
Abraham J. Wyner, Matthew Olson, Justin Bleich, David Mease

18(49):1–47 **Clustering from General Pairwise Observations with Applications to Time-varying Graphs**
Shiau Hong Lim, Yudong Chen, Huan Xu

18(50):1–41 **Uniform Hypergraph Partitioning: Provable Tensor Methods and Sampling Techniques**
Debarghya Ghoshdastidar, Ambedkar Dukkipati

18(51):1–24 **Reconstructing Undirected Graphs from Eigenspaces**
Yohann De Castro, Thibault Espinasse, Paul Rochet

18(52):1–11 **An Optimal Algorithm for Bandit and Zero-Order Convex Optimization with Two-Point Feedback**
Ohad Shamir

18(53):1–31 **Perishability of Data: Dynamic Pricing under Varying-Coefficient Models**
Adel Javanmard

18(54):1–24 **Two New Approaches to Compressed Sensing Exhibiting Both Robust Sparse Recovery and the Grouping Effect**
Mehmet Eren Ahsen, Niharika Challapalli, Mathukumalli Vidyasagar

18(55):1–35 **On the Consistency of Ordinal Regression Methods**
Fabian Pedregosa, Francis Bach, Alexandre Gramfort

18(56):1–26 **Statistical Inference with Unnormalized Discrete Models and Localized Homogeneous Divergences**
Takashi Takenouchi, Takafumi Kanamori

18(57):1–59 **Density Estimation in Infinite Dimensional Exponential Families**
Bharath Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Aapo Hyvärinen, Revant Kumar

18(58):1–52 **Lens Depth Function and k-Relative Neighborhood Graph: Versatile Tools for Ordinal Data Analysis**
Matthäus Kleindessner, Ulrike von Luxburg

18(59):1–39 **Joint Label Inference in Networks**
Deepayan Chakrabarti, Stanislav Funiak, Jonathan Chang, Sofus A. Macskassy

18(60):1–45 **Achieving Optimal Misclassification Proportion in Stochastic Block Models**
Chao Gao, Zongming Ma, Anderson Y. Zhang, Harrison H. Zhou

18(61):1–33 **On the Propagation of Low-Rate Measurement Error to Subgraph Counts in Large Networks**
Prakash Balachandran, Eric D. Kolaczyk, Weston D. Viles

18(62):1–58 **Dense Distributions from Sparse Samples: Improved Gibbs Sampling Parameter Estimators for LDA**
Yannis Papanikolaou, James R. Foulds, Timothy N. Rubin, Grigorios Tsoumakas

18(63):1–29 **Fundamental Conditions for Low-CP-Rank Tensor Completion**
Morteza Ashraphijuo, Xiaodong Wang

18(64):1–34 **Parallel Symmetric Class Expression Learning**
An C. Tran, Jens Dietrich, Hans W. Guesgen, Stephen Marsland

18(65):1–35 **Learning Partial Policies to Speedup MDP Tree Search via Reduction to I.I.D. Learning**
Jervis Pinto, Alan Fern

18(66):1–42 **Hierarchically Compositional Kernels for Scalable Nonparametric Learning**
Jie Chen, Haim Avron, Vikas Sindhwani

18(67):1–29 **Sharp Oracle Inequalities for Square Root Regularization**
Benjamin Stucky, Sara van de Geer

18(68):1–43 **Soft Margin Support Vector Classification as Buffered Probability Minimization**
Matthew Norton, Alexander Mafusalov, Stan Uryasev

18(69):1–29 **Variational Particle Approximations**
Ardavan Saeedi, Tejas D. Kulkarni, Vikash K. Mansinghka, Samuel J. Gershman

18(70):1–37 **A Bayesian Framework for Learning Rule Sets for Interpretable Classification**
Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, Perry MacNeille

18(71):1–46 **A Robust-Equitable Measure for Feature Ranking and Selection**
A. Adam Ding, Jennifer G. Dy, Yi Li, Yale Chang

18(72):1–32 **Multiscale Strategies for Computing Optimal Transport**
Samuel Gerber, Mauro Maggioni

18(73):1–46 **Non-parametric Policy Search with Limited Information Loss**
Herke van Hoof, Gerhard Neumann, Jan Peters

Part B

18(74):1–40 **Tests of Mutual or Serial Independence of Random Vectors with Applications**
Martin Bilodeau, Aurélien Guetsop Nangue

18(75):1–34 **Recovering PCA and Sparse PCA via Hybrid-(l1,l2) Sparse Sampling of Data Elements**
Abhisek Kundu, Petros Drineas, Malik Magdon-Ismail

18(76):1–61 **Quantifying the Informativeness of Similarity Measurements**
Austin J. Brockmeier, Tingting Mu, Sophia Ananiadou, John Y. Goulermas

18(77):1–36 **Time for a Change: a Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis**
Alessio Benavoli, Giorgio Corani, Janez Demšar, Marco Zaffalon

18(78):1–44 **Relational Reinforcement Learning for Planning with Exogenous Effects**
David Martínez, Guillem Alenyà, Tony Ribeiro, Katsumi Inoue, Carme Torras

18(79):1–31 **Bayesian Tensor Regression**
Rajarshi Guhaniyogi, Shaan Qamar, David B. Dunson

18(80):1–50 **Robust Discriminative Clustering with Sparse Regularizers**
Nicolas Flammarion, Balamurugan Palaniappan, Francis Bach

18(81):1–36 **Making Decision Trees Feasible in Ultrahigh Feature and Label Dimensions**
Weiwei Liu, Ivor W. Tsang

18(82):1–37 **Learning Scalable Deep Kernels with Recurrent Structure**
Maruan Al-Shedivat, Andrew Gordon Wilson, Yunus Saatchi, Zhiting Hu, Eric P. Xing

18(83):1–52 **Convolutional Neural Networks Analyzed via Convolutional Sparse Coding**
Vardan Papyan, Yaniv Romano, Michael Elad

18(84):1–42 **Stochastic Primal-Dual Coordinate Method for Regularized Empirical Risk Minimization**
Yuchen Zhang, Lin Xiao

18(85):1–21 **Angle-based Multicategory Distance-weighted SVM**
Hui Sun, Bruce A. Craig, Lingsong Zhang

18(86):1–47 **Minimax Estimation of Kernel Mean Embeddings**
Ilya Tolstikhin, Bharath K. Sriperumbudur, Krikamol Muandet

18(87):1–28 **The Impact of Random Models on Clustering Similarity**
Alexander J. Gates, Yong-Yeol Ahn

18(88):1–35 **Hierarchical Clustering via Spreading Metrics**
Aurko Roy, Sebastian Pokutta

18(89):1–5 **The MADP Toolbox: An Open Source Library for Planning and Learning in (Multi-)Agent Systems**
Frans A. Oliehoek, Matthijs T. J. Spaan, Bas Terwijn, Philipp Robbel, João V. Messias

18(90):1–50 **A survey of Algorithms and Analysis for Adaptive Online Learning**
H. Brendan McMahan

18(91):1–35 **A distributed block coordinate descent method for training l1 regularized linear classifiers**
Dhruv Mahajan, S. Sathiya Keerthi, S. Sundararajan

18(92):1–31 **Distributed Learning with Regularized Least Squares**
Shao-Bo Lin, Xin Guo, Ding-Xuan Zhou

18(93):1–67 **Identifying Unreliable and Adversarial Workers in Crowdsourced Labeling Tasks**
Srikanth Jagabathula, Lakshminarayanan Subramanian, Ashwin Venkataraman

18(94):1–38 **An Easy-to-hard Learning Paradigm for Multiple Classes and Multiple Labels**
Weiwei Liu, Ivor W. Tsang, Klaus-Robert Müller

18(95):1–32 **Fisher Consistency for Prior Probability Shift**
Dirk Tasche

18(96):1–5 **openXBOW – Introducing the Passau Open-Source Crossmodal Bag-of-Words Toolkit**
Maximilian Schmitt, Björn Schuller

18(97):1–47 **Optimal Rates for Multi-pass Stochastic Gradient Methods**
Junhong Lin, Lorenzo Rosasco

18(98):1–29 **Rank Determination for Low-Rank Data Completion**
Morteza Ashraphijuo, Xiaodong Wang, Vaneet Aggarwal

18(99):1–32 **Bayesian Network Learning via Topological Order**
Young Woong Park, Diego Klabjan

18(100):1–37 **Stability of Controllers for Gaussian Process Dynamics**
Julia Vinogradska, Bastian Bischoff, Duy Nguyen-Tuong, Jan Peters

18(101):1–51 **Harder, Better, Faster, Stronger Convergence Rates for Least-Squares Regression**
Aymeric Dieuleveut, Nicolas Flammarion, Francis Bach

18(102):1–28 **Confidence Sets with Expected Sizes for Multiclass Classification**
Christophe Denis, Mohamed Hebiri

18(103):1–50 **Online Learning to Rank with Top-k Feedback**
Sougata Chaudhuri, Ambuj Tewari

18(104):1–72 **A Unifying Framework for Gaussian Process Pseudo-Point Approximations using Power Expectation Propagation**
Thang D. Bui, Josiah Yan, Richard E. Turner

18(105):1–23 **Accelerating Stochastic Composition Optimization**
Mengdi Wang, Ji Liu, Ethan X. Fang

18(106):1–37 **Distributed Bayesian Learning with Stochastic Natural Gradient Expectation Propagation and the Posterior Server**
Leonard Hasenclever, Stefan Webb, Thibaut Lienart, Sebastian Vollmer, Balaji Lakshminarayanan, Charles Blundell, Yee Whye Teh

18(107):1–28 **Optimal Dictionary for Least Squares Representation**
Mohammed Rayyan Sheriff, Debasish Chatterjee

18(108):1–37 **Computational Limits of A Distributed Algorithm for Smoothing Spline**
Zuofeng Shang, Guang Cheng

18(109):1–67 **Hinge-Loss Markov Random Fields and Probabilistic Soft Logic**
Stephen H. Bach, Matthias Broecheler, Bert Huang, Lise Getoor

18(110):1–49 **Clustering with Hidden Markov Model on Variable Blocks**
Lin Lin, Jia Li

18(111):1–55 **Approximation Vector Machines for Large-scale Online Learning**
Trung Le, Tu Dinh Nguyen, Vu Nguyen, Dinh Phung

18(112):1–29 **Efficient Sampling from Time-Varying Log-Concave Distributions**
Hariharan Narayanan, Alexander Rakhlin

18(113):1–24 **Document Neural Autoregressive Distribution Estimation**
Stanislas Lauly, Yin Zheng, Alexandre Allauzen, Hugo Larochelle

18(114):1–26 **Target Curricula via Selection of Minimum Feature Sets: a Case Study in Boolean Networks**
Shannon Fenn, Pablo Moscato

18(115):1–52 **A General Distributed Dual Coordinate Optimization Framework for Regularized Loss Minimization**
Shun Zheng, Jialei Wang, Fen Xia, Wei Xu, Tong Zhang

18(116):1–40 **Second-Order Stochastic Optimization for Machine Learning in Linear Time**
Naman Agarwal, Brian Bullins, Elad Hazan

18(117):1–49 **Regularized Estimation and Testing for High-Dimensional Multi-Block Vector-Autoregressive Models**
Jiahe Lin, George Michailidis

18(118):1–25 **Learning Theory of Distributed Regression with Bias Corrected Regularization Kernel Network**
Zheng-Chu Guo, Lei Shi, Qiang Wu

18(119):1–59 **Probabilistic Line Searches for Stochastic Optimization**
Maren Mahsereci, Philipp Hennig

18(120):1–49 **Learning Instrumental Variables with Structural and Non-Gaussianity Assumptions**
Ricardo Silva, Shohei Shimizu

18(121):1–34 **Classification of Time Sequences using Graphs of Temporal Constraints**
Mathieu Guillame-Bert, Artur Dubrawski

18(122):1–43 **Distributed Stochastic Variance Reduced Gradient Methods by Sampling Extra Data with Replacement**
Jason D. Lee, Qihang Lin, Tengyu Ma, Tianbao Yang

18(123):1–41 **Kernel Partial Least Squares for Stationary Data**
Marco Singer, Tatyana Krivobokova, Axel Munk

18(124):1–40 **Robust and Scalable Bayes via a Median of Subset Posterior Measures**
Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, David B. Dunson

18(125):1–53 **Statistical and Computational Guarantees for the Baum-Welch Algorithm**
Fanny Yang, Sivaraman Balakrishnan, Martin J. Wainwright

18(126):1–45 **Online but Accurate Inference for Latent Variable Models with Local Gibbs Sampling**
Christophe Dupuy, Francis Bach

18(127):1–45 **Poisson Random Fields for Dynamic Feature Models**
Valerio Perrone, Paul A. Jenkins, Dario Spanò, Yee Whye Teh

18(128):1–33 **Gap Safe Screening Rules for Sparsity Enforcing Penalties**
Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, Joseph Salmon

18(129):1–31 **Minimax Filter: Learning to Preserve Privacy from Inference Attacks**
Jihun Hamm

18(130):1–38 **Knowledge Graph Completion via Complex Tensor Factorization**
Théo Trouillon, Christopher R. Dance, Éric Gaussier, Johannes Welbl, Sebastian Riedel, Guillaume Bouchard

18(131):1–36 **Stabilized Sparse Online Learning for Sparse Data**
Yuting Ma, Tian Zheng

18(132):1–31 **Active-set Methods for Submodular Minimization Problems**
K. S. Sesh Kumar, Francis Bach

18(133):1–33 **A Bayesian Mixed-Effects Model to Learn Trajectories of Changes from Repeated Manifold-Valued Observations**
Jean-Baptiste Schiratti, Stéphanie Allassonnière, Olivier Colliot, Stanley Durrleman

18(134):1–35 **Stochastic Gradient Descent as Approximate Bayesian Inference**
Stephan Mandt, Matthew D. Hoffman, David M. Blei

18(135):1–37 **STORE: Sparse Tensor Response Regression and Neuroimaging Analysis**
Will Wei Sun, Lexin Li

18(136):1–46 **A Survey of Preference-Based Reinforcement Learning Methods**
Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz

18(137):1–50 **Generalized SURE for optimal shrinkage of singular values in low-rank matrix denoising**
Jérémie Bigot, Charles Deledalle, Delphine Féral

18(138):1–35 **Dimension Estimation Using Random Connection Models**
Paulo Serra, Michel Mandjes

18(139):1–58 **Bayesian Inference for Spatio-temporal Spike-and-Slab Priors**
Michael Riis Andersen, Aki Vehtari, Ole Winther, Lars Kai Hansen

18(140):1–30 **Adaptive Randomized Dimension Reduction on Massive Data**
Gregory Darnell, Stoyan Georgiev, Sayan Mukherjee, Barbara E Engelhardt

18(141):1–35 **A Nonconvex Approach for Phase Retrieval: Reshaped Wirtinger Flow and Incremental Algorithms**
Huishuai Zhang, Yingbin Liang, Yuejie Chi

18(142):1–39 **Consistency, Breakdown Robustness, and Algorithms for Robust Improper Maximum Likelihood Clustering**
Pietro Coretto, Christian Hennig

18(143):1–41 **On Computationally Tractable Selection of Experiments in Measurement-Constrained Regression Models**
Yining Wang, Adams Wei Yu, Aarti Singh

18(144):1–46 **Generalized Conditional Gradient for Sparse Estimation**
Yaoliang Yu, Xinhua Zhang, Dale Schuurmans

18(145):1–31 **Following the Leader and Fast Rates in Online Linear Prediction: Curved Constraint Sets and Other Regularities**
Ruitong Huang, Tor Lattimore, András György, Csaba Szepesvári

18(146):1–48 **Regularization and the small-ball method II: complexity dependent error rates**
Guillaume Lecué, Shahar Mendelson

18(147):1–25 **Matrix Completion with Noisy Entries and Outliers**
Raymond K. W. Wong, Thomas C. M. Lee

18(148):1–29 **Faithfulness of Probability Distributions and Graphs**
Kayvan Sadeghi

18(149):1–49 **Community Extraction in Multilayer Networks with Heterogeneous Community Structure**
James D. Wilson, John Palowitch, Shankar Bhamidi, Andrew B. Nobel

Journal of Machine Learning Research 18 (2017) 1-40 Submitted 4/16; Revised 7/17; Published 8/17

Tests of Mutual or Serial Independence of Random Vectors with Applications

Martin Bilodeau bilodeau@dms.umontreal.ca

Aurélien Guetsop Nangue guetsopn@dms.umontreal.ca
Département de mathématiques et de statistique
Université de Montréal
C.P. 6128, Succursale A
Montréal, Canada H3C 3J7

Editor: Arthur Gretton

Abstract

The problem of testing mutual independence between many random vectors is addressed. The closely related problem of testing serial independence of a multivariate stationary sequence is also considered. The Möbius transformation of characteristic functions is used to characterize independence. A generalization to p vectors of distance covariance and Hilbert-Schmidt independence criterion ($HSIC$) tests with the translation invariant kernel of a stable probability distribution is proposed. Both test statistics can be expressed in a simple form as a sum over all elements of a componentwise product of p doubly-centered matrices. It is shown that an $HSIC$ statistic with sufficiently small scale parameters is equivalent to a distance covariance statistic. Consistency and weak convergence of both types of statistics are established. Approximation of p-values is made by randomization tests without recomputing interpoint distances for each randomized sample. The dependogram is adapted to the proposed tests for the graphical identification of sources of dependencies. Empirical rejection rates obtained through extensive simulations confirm both the applicability of the testing procedures in small samples and the high level of competitiveness in terms of power. Applications to meteorological and financial data provide some interesting interpretations of dependencies revealed by dependograms.

Keywords: Distance covariance, Hilbert-Schmidt independence criterion, Möbius transformation, mutual independence, serial independence

1. Introduction

The problem of testing for independence between p components of a random vector has attracted considerable attention in statistics. Many nonparametric procedures exist in the literature. A natural approach is to consider a functional of the difference between the empirical joint distribution and the product of the empirical marginal distributions. This same approach can also use empirical characteristic functions. When the functional of the difference is above a certain threshold, the components are declared dependent. Csörgő (1985), Kankainen (1995), Sejdinovic et al. (2013b) and Fan et al. (2017) considered mutual tests of independence based on empirical characteristic functions. However, when dependence is declared, it is not possible to identify, with their proposed tests, subsets of variables responsible for the dependence. This limitation is similar to that of a global F-test in an analysis of variance model with one fixed factor, as opposed to multiple comparisons procedures, or that of a global chi-square test of independence in a multi-way contingency table, as opposed to log-linear models with interaction terms. For tests of independence, a useful method is the Möbius transformation.

The Möbius transformation defined in (1) of Section 2 has a long history in statistics. The Möbius transformation of distribution functions was first proposed in Blum et al. (1961) for $p = 3$. The general case was treated in Deheuvels (1981), Ghoudi et al. (2001), Genest and Rémillard (2004), Kojadinovic and Holmes (2009), Kojadinovic and Yan (2011), and Duchesne et al. (2012). It can also be defined with characteristic functions as in Bilodeau and Lafaye de Micheaux (2005), with half-space probabilities as in Beran et al. (2007), or with cell probabilities in a contingency table as in Bilodeau and Lafaye de Micheaux (2009). The first appearance of a Möbius transformation, although not stated explicitly, goes back to the work of Lancaster (1951) on contingency tables as explained in Bilodeau and Lafaye de Micheaux (2009). The machine learning community (Sejdinovic et al., 2013a) proposed kernel nonparametric tests for Lancaster three-variable interaction. This test is in fact a test based on the empirical version of the Möbius transformation of the characteristic function when $p = 3$. The general Möbius transformation considered in this paper can be used to build tests for general interactions of any order, as well as tests of mutual and serial independence.

The paper is organized as follows. Section 2 introduces the Möbius transformation of characteristic functions. It presents a characterization of the mutual independence between p random vectors by the Möbius transformation. In Section 3, new tests based on the Möbius transformation of empirical characteristic functions are introduced. They generalize the Hilbert-Schmidt independence criterion ($HSIC$) test (Gretton et al., 2005, 2008) and the distance covariance test (Székely et al., 2007) to the case $p > 2$. This work addresses the case of finite-dimensional Euclidean spaces. $HSIC$ was originally defined more generally using any semimetric space of negative type (as in the distance covariance), or any Borel measurable space on which a kernel is defined (Sejdinovic et al., 2013b). For example, in Gretton et al. (2008), dependence was detected between text in different languages using kernels on strings. On the other hand, this manuscript proposes a criterion that is more general than $HSIC$ in a different respect, via subsets of components in the Möbius transformation, where the criterion coincides with $HSIC$ when the latter is specialized to Euclidean spaces and $p = 2$. The new test statistics have a common form as a sum of elements of a componentwise product of p doubly-centered matrices. An equivalence is established between an $HSIC$ statistic with infinitesimal scale parameters and a distance covariance statistic. The weak convergence of the empirical processes based on the Möbius transformation is proved. The consistency and weak convergence of the $HSIC$ and distance covariance functionals are also established. A difficulty encountered in establishing the asymptotic independence of the collection of distance covariance functionals, over all subsets of components, is described. Other competing nonparametric tests of independence are reviewed in Section 4.

Rather than relying on the limiting distribution to conduct tests, Section 5 describes randomization tests as an approximation to permutation tests. In Section 6, p-values obtained by randomization tests for all possible subsets of components are combined using

©2017 Martin Bilodeau and Aurélien Guetsop Nangue.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-184.html.

the methods of Fisher (1950, pp. 99-101) or Tippett (1952) to obtain a global test of mutual independence. Section 7 describes the dependogram, a graphical device of Genest and Rémillard (2004), in the context of $HSIC$ and distance covariance tests. Section 8 adapts all the results described for the mutual independence situation to the problem of testing for the serial independence of a multivariate stationary sequence. Computational costs are given in Section 9 together with a short description of `R` (R Core Team, 2015) packages for nonparametric independence tests. Simulated models are considered in Section 10 to verify that the proposed tests have empirical significance levels close to the nominal level in small samples and comparable or higher powers in many situations when compared to existing tests such as those of Csörgő (1985), Kojadinovic and Holmes (2009) or Kojadinovic and Yan (2011).

Finally, Section 11 contains an application to real data on variables related to air temperature, soil temperature, humidity, wind, and evaporation. $HSIC$ or distance covariance tests should be preferred to the Gaussian likelihood ratio test since a multivariate Gaussian model is rejected by the test of Henze and Zirkler (1990). According to these tests, wind does not exhibit any dependence with all other variables considered. Another application finds significant serial dependencies in the S&P/TSX composite, DOW JONES, and S&P500 daily percent increasing rates ranging from January 2, 2014 to March 2, 2016. The strongest dependency observed at a lag of 4 days by a distance covariance test is interpreted using a broken line regression model as the tendency of stock markets to recover in the days following a sharp decline.

2. Möbius Transformation

The Möbius transformation of characteristic functions is a powerful tool for the characterization of mutual independence between p random vectors $Z^{(1)}, \ldots, Z^{(p)}$. The dimension of the vector $Z^{(j)}$ is d_j, for $j = 1, \ldots, p$. Let f be the joint characteristic function of these p vectors, and let $f^{(j)}$ be the marginal characteristic function of $Z^{(j)}$. Mutual independence is characterized by the factorization

$$f(t^{(1)}, \ldots, t^{(p)}) = \prod_{j=1}^{p} f^{(j)}(t^{(j)}),$$

for all $t^{(1)}, \ldots, t^{(p)}$. It may also be characterized by the Möbius transformation which is defined as follows. Let $\mathcal{I}_p$ be the family of subsets B of $\{1, \ldots, p\}$ of cardinality $|B| > 1$. The set $\mathcal{I}_p$ has $2^p - p - 1$ elements since the empty set is excluded, as well as all p singletons. For any $B \in \mathcal{I}_p$ and any $t^{(1)}, \ldots, t^{(p)}$, define $t^{(B)} = (t^{(j)} : \ j \in B)$. Similarly, $Z^{(B)} = (Z^{(j)} : j \in B)$, and $f^{(B)}$ is the joint characteristic function of $Z^{(B)}$. The Möbius transformation of the characteristic function f for the set $B \in \mathcal{I}_p$ is given by

$$\mu_B(t^{(B)}) = \sum_{C \subseteq B} (-1)^{|B \setminus C|} f^{(C)}(t^{(C)}) \prod_{j \in B \setminus C} f^{(j)}(t^{(j)}). \tag{1}$$

By convention, for $C = \emptyset$, both $f^{(C)}(t^{(C)})$ and $\prod_{j \in C} f^{(j)}(t^{(j)})$ are equal to one. The following characterization holds: $Z^{(1)}, \ldots, Z^{(p)}$ are mutually independent if and only if, $\mu_B(t^{(B)}) = 0$,

for all $B \in \mathcal{I}_p$, and all vectors $t^{(B)}$. A proof by induction of this characterization using distribution functions is given in Ghoudi et al. (2001) and is immediately applicable to characteristic functions.

3. Dependence Statistics

Consider $Z_k = (Z_k^{(1)}, \ldots, Z_k^{(p)})$, $k = 1, \ldots, n$, an independent and identically distributed sample of size n. The Möbius processes corresponding to μ_B, $B \in \mathcal{I}_p$, are defined as

$$R_{nB}(t^{(B)}) = \sqrt{n} \sum_{C \subseteq B} (-1)^{|B \setminus C|} f_n^{(C)}(t^{(C)}) \prod_{j \in B \setminus C} f_n^{(j)}(t^{(j)}), \tag{2}$$

where

$$f_n^{(C)}(t^{(C)}) = \frac{1}{n} \sum_{k=1}^{n} e^{i \langle t^{(C)}, Z_k^{(C)} \rangle} \tag{3}$$

is the empirical characteristic function. When $C = \{j\}$ is a singleton, the notation used for $f_n^{(C)}$ is simply $f_n^{(j)}$. The empirical processes R_{nB}, for all $B \in \mathcal{I}_p$, are illustrated when $p = 3$ in Table 1. The processes R_{nB} are identical to the empirical characteristic independence processes

$$S_{nB}(t^{(B)}) = \sqrt{n} \left[f_n^{(B)}(t^{(B)}) - \prod_{j \in B} f_n^{(j)}(t^{(j)}) \right]$$

in Csörgő (1985) only for subsets B of cardinality 2. The process S_{nB} appears later in the test statistic $\mathcal{J}_n^2$ in (17) used for testing the hypothesis of mutual independence. For $B = \{1, 2, 3\}$, the process

$$S_{nB}(t^{(B)}) = \sqrt{n} \left[f_n^{(1,2,3)}(t^{(1)}, t^{(2)}, t^{(3)}) - f_n^{(1)}(t^{(1)}) f_n^{(2)}(t^{(2)}) f_n^{(3)}(t^{(3)}) \right]$$

can be contrasted with the process R_{nB} in Table 1. Although, at first sight, R_{nB} may look more complicated than S_{nB} and both processes converge weakly to Gaussian processes, the processes R_{nB} have major advantages which are enunciated in Section 4.

B	$R_{nB}(t^{(B)})/\sqrt{n}$
$\{1,2\}$	$f_n^{(1,2)}(t^{(1)}, t^{(2)}) - f_n^{(1)}(t^{(1)}) f_n^{(2)}(t^{(2)})$
$\{1,3\}$	$f_n^{(1,3)}(t^{(1)}, t^{(3)}) - f_n^{(1)}(t^{(1)}) f_n^{(3)}(t^{(3)})$
$\{2,3\}$	$f_n^{(2,3)}(t^{(2)}, t^{(3)}) - f_n^{(2)}(t^{(2)}) f_n^{(3)}(t^{(3)})$
$\{1,2,3\}$	$-f_n^{(1,2,3)}(t^{(1)}, t^{(2)}, t^{(3)}) + f_n^{(1,2)}(t^{(1)}, t^{(2)}) f_n^{(3)}(t^{(3)}) + f_n^{(1,3)}(t^{(1)}, t^{(3)}) f_n^{(2)}(t^{(2)})$ $+f_n^{(2,3)}(t^{(2)}, t^{(3)}) f_n^{(1)}(t^{(1)}) - 2 f_n^{(1)}(t^{(1)}) f_n^{(2)}(t^{(2)}) f_n^{(3)}(t^{(3)})$

Table 1: Empirical processes R_{nB}, $B \in \mathcal{I}_3$.

The dependence statistic for the subset B is now defined as the Cramér-von Mises functional

$$W_{nB} = \frac{1}{n} \int |R_{nB}(t^{(B)})|^2 dw_B(t^{(B)}), \tag{4}$$

where $dw_B(t^{(B)}) = \prod_{j\in B} dw^{(j)}(t^{(j)})$ is a product measure. The evaluation of this integral is facilitated using another representation of the process. First, recall the multinomial formula (Ghoudi et al., 2001)

$$\sum_{C\subseteq B}\left(\prod_{i\in C} u^{(i)}\right)\left(\prod_{j\in B\setminus C} v^{(j)}\right) = \prod_{i\in B}\left(u^{(i)}+v^{(i)}\right). \tag{5}$$

Then, the empirical process (2) can be written as

$$R_{nB}(t^{(B)}) = \frac{1}{\sqrt{n}}\sum_{k=1}^{n}\prod_{j\in B}\left[e^{i\langle t^{(j)},Z_k^{(j)}\rangle} - f_n^{(j)}(t^{(j)})\right]. \tag{6}$$

The representation (6) is obtained after replacing the expression (3) for the empirical characteristic function in (2) and by applying the multinomial formula (5). The representation given by (6) allows the integral (4) to be evaluated explicitly in some cases and simplifies the proof of theorems to come. Two important cases are now presented.

3.1 Hilbert-Schmidt Independence Criterion

Assume that the measure $dw_B(t^{(B)}) = \prod_{j\in B} dG^{(j)}(t^{(j)})$ is a product of symmetric (around the origin) probability measures. The population Hilbert-Schmidt independence criterion is

$$\mathcal{H}_B^2 = \int |\mu_B(t^{(B)})|^2 \prod_{j\in B} dG^{(j)}(t^{(j)}) \tag{7}$$

which is well defined since the function μ_B is bounded. For the sample version, let $\varphi^{(j)}$ be the (real) characteristic function of $G^{(j)}$. The sample version of the Hilbert-Schmidt independence criterion (7) is denoted $\mathcal{H}_{nB}^2$ and has the following explicit expression.

Theorem 1 *For any $B \in \mathcal{I}_p$, the dependence statistic $\mathcal{H}_{nB}^2$ is given by*

$$\mathcal{H}_{nB}^2 = \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}\prod_{j\in B} A_{kl}^{(j)}, \tag{8}$$

where

$$\begin{aligned} a_{kl}^{(j)} &= \varphi^{(j)}(Z_k^{(j)} - Z_l^{(j)}), \\ A_{kl}^{(j)} &= a_{kl}^{(j)} - \bar{a}_{k.}^{(j)} - \bar{a}_{.l}^{(j)} + \bar{a}_{..}^{(j)}, \\ \bar{a}_{k.}^{(j)} &= \frac{1}{n}\sum_{l=1}^{n} a_{kl}^{(j)},\ \bar{a}_{.l}^{(j)} = \frac{1}{n}\sum_{k=1}^{n} a_{kl}^{(j)},\ \bar{a}_{..}^{(j)} = \frac{1}{n^2}\sum_{k,l=1}^{n} a_{kl}^{(j)}. \end{aligned} \tag{9}$$

By definition, matrices $A^{(j)} = (A_{kl}^{(j)})$, $j = 1,\ldots,p$, are doubly-centered, *i.e.* rows and columns of these matrices sum up to zero. In the special case $p = 2$ of testing the independence between two vectors, the statistic $\mathcal{H}_{nB}^2$, for $B = \{1,2\}$, is the Hilbert-Schmidt independence criterion, or $HSIC$, with translation invariant kernels $\varphi^{(j)}(Z_k^{(j)} - Z_l^{(j)})$, $j = 1,2$. A proof of Theorem 1 for $p = 3$ was provided by Sejdinovic et al. (2013a).

An important special case is when $G^{(j)}$ is the stable distribution of index $\alpha \in (0,2]$ with scale parameter $\beta_j > 0$ first studied by Lévy (1925). Then, the translation invariant kernel is

$$a_{kl}^{(j)} = e^{-\beta_j^{\alpha}|Z_k^{(j)} - Z_l^{(j)}|_{d_j}^{\alpha}}, \tag{10}$$

where $|\cdot|_{d_j}$ is the Euclidean norm in dimension d_j. The corresponding dependence statistic is then denoted $\mathcal{H}_{nB}^{2(\alpha)}$. The case $\alpha = 2$ is the Gaussian kernel, and $\alpha = 1$ is the Cauchy kernel, often referred to as the Laplace kernel in machine learning (Gretton et al., 2005, 2009) because of its similarity to a Laplace density in dimension one.

The following result establishes the consistency of the Hilbert-Schmidt independence criterion $\mathcal{H}_{nB}^2$.

Theorem 2

(i) $\mathcal{H}_{nB}^2 \overset{a.s.}{\to} \mathcal{H}_B^2$, *as* $n \to \infty$.

(ii) *If* $\mu_B(t^{(B)}) \neq 0$ *for some vector* $t^{(B)}$, *then* $n\mathcal{H}_{nB}^2 \overset{a.s.}{\to} \infty$, *as* $n \to \infty$.

3.2 Distance Covariance

Assume that the measure $dw_B(t^{(B)}) = \prod_{j\in B} dw^{(j)}(t^{(j)})$ is a product of non integrable measures. The measure $dw^{(j)}$ of index $0 < \alpha < 2$ is defined as

$$dw^{(j)}(t^{(j)}) = \left[C(d_j,\alpha)|t^{(j)}|_{d_j}^{d_j+\alpha}\right]^{-1} dt^{(j)},$$

with the normalizing constant

$$C(d,\alpha) = 2\pi^{d/2}\Gamma(1-\alpha/2)/\left[\alpha 2^{\alpha}\Gamma((d+\alpha)/2)\right].$$

A similar representation as in (8) also holds. The corresponding dependence statistic is denoted $\mathcal{V}_{nB}^{2(\alpha)}$, which is the usual notation for distance covariance of index α.

Theorem 3 *Let $0 < \alpha < 2$. Then, for any $B \in \mathcal{I}_p$, the dependence statistic $\mathcal{V}_{nB}^{2(\alpha)}$ has the same form as in (8) of Theorem 1 with*

$$a_{kl}^{(j)} = -|Z_k^{(j)} - Z_l^{(j)}|_{d_j}^{\alpha}. \tag{11}$$

In the special case $|B| = 2$, the statistic $\mathcal{V}_{nB}^{2(\alpha)}$ reduces to the distance covariance of index α in Feuerverger (1993) for the case $d_1 = d_2 = 1$, and later generalized by Székely et al. (2007) to the case $d_1 \geq 1$, $d_2 \geq 1$. A very special case requiring a separate analysis is when $|B| = 2$ and $\alpha = 2$. In this case, $\mathcal{V}_{nB}^{2(2)}$ is the numerator of the RV coefficient of Escoufier (1973) as noticed by Josse and Holmes (2016) when $d_1 \geq 1$ and $d_2 \geq 1$, and earlier by Székely et al. (2007) only when $d_1 = d_2 = 1$. It should be noted that the case $\alpha = 2$ leads only to a test of non correlation but not of independence, unless the joint distribution is Gaussian. For this reason, the value $\alpha = 2$ will not be considered for distance covariance in the sequel.

Székely et al. (2007) showed the consistency of distance covariance for $0 < \alpha < 2$ and $|B| = 2$. Consistency of distance covariance for $|B| > 2$ is now established. Only when necessary, the notation $\mathbb{E}_3$ is for the expectation with respect to Z_3, treating the other variable as a constant to avoid using conditional expectations.

Theorem 4 *Let $0 < \alpha < 2$. Assume*

$$\mathbb{E}\prod_{j\in B}|Z_1^{(j)} - Z_2^{(j)}|_{d_j}^{\alpha} < \infty. \tag{12}$$

Define

$$\mathcal{V}_B^{2(\alpha)} = \mathbb{E}\prod_{j\in B}\left[-|Z_1^{(j)} - Z_2^{(j)}|_{d_j}^{\alpha} + \mathbb{E}_3|Z_1^{(j)} - Z_3^{(j)}|_{d_j}^{\alpha} + \mathbb{E}_3|Z_2^{(j)} - Z_3^{(j)}|_{d_j}^{\alpha} - \mathbb{E}|Z_3^{(j)} - Z_4^{(j)}|_{d_j}^{\alpha}\right].$$

Then,

(i) $\mathcal{V}_B^{2(\alpha)} = \int |\mu_B(t^{(B)})|^2 dw_B(t^{(B)}) < \infty$.

(ii) *If $\mu_B(t^{(B)}) \neq 0$ for some vector $t^{(B)}$, then $n\mathcal{V}_{nB}^{2(\alpha)} \overset{a.s.}{\to} \infty$, as $n \to \infty$.*

For $0 < \alpha < 2$, the following limit establishes that $\mathcal{V}_{nB}^{2(\alpha)}$ is, for all practical purpose, equivalent to $\mathcal{H}_{nB}^{2(\alpha)}$ when scale parameters β_j, $j \in B$, are sufficiently small:

$$\lim_{\beta_j\to 0, \forall j\in B} \mathcal{H}_{nB}^{2(\alpha)} / \prod_{j\in B}\beta_j^{\alpha} = \mathcal{V}_{nB}^{2(\alpha)}. \tag{13}$$

This result, proved in the appendix, implies that $\mathcal{H}_{nB}^{2(\alpha)}$, properly normalized with sufficiently small scale parameters, can be as close as desired to $\mathcal{V}_{nB}^{2(\alpha)}$ and thus, $\mathcal{H}_{nB}^{2(\alpha)}$ will have a power function indistinguishable from that of $\mathcal{V}_{nB}^{2(\alpha)}$. For semimetrics generated by kernels, Sejdinovic et al. (2013b, Theorem 24) established an equivalence between distance covariance and $HSIC$. However, for distance covariance defined in terms of a weighted distance between characteristic functions, one can not find a continuous translation invariant kernel for which $HSIC$ coincide with distance covariance (Sejdinovic et al., 2013b, Section 5.3). Nevertheless, (13) provides an equivalence of a different nature: for α-stable distributions, appropriately normalized $HSIC$ and distance covariance are equivalent in the limit, as the scale parameters converge to zero.

As simple as it may seem, this equivalence, for the simplest case $|B| = 2$, has gone unnoticed in the discussions of distance covariance (Székely and Rizzo, 2009; Gretton et al., 2009). In Section 8.2 of Sejdinovic et al. (2013b), for $|B| = 2$, the $HSIC$ test based on $\mathcal{H}_{nB}^{2(2)}$ with Gaussian kernels with scale parameters set at the inverse of median of interpoint distances is compared to distance covariance tests of varying index α. It was found in the independent component analysis benchmark example that $\mathcal{V}_{nB}^{2(1/3)}$ is more powerful than $\mathcal{H}_{nB}^{2(2)}$. From (13), the $HSIC$ test based on $\mathcal{H}_{nB}^{2(1/3)}$, with translation invariant kernels of the stable distribution of index $\alpha = 1/3$ and very small scale parameters, would have a power function indistinguishable from that of $\mathcal{V}_{nB}^{2(1/3)}$. In another example with sinusoidally dependent data, the $HSIC$ test based on $\mathcal{H}_{nB}^{2(2)}$ has a very poor power function compared to $\mathcal{V}_{nB}^{2(1/6)}$. Sejdinovic et al. (2013b) explained: "the exponent in the distance-induced kernel plays a similar role as the bandwidth of the Gaussian kernel, and smaller exponents are able to detect dependencies at smaller lengthscales. Poor performance of the Gaussian kernel with median bandwidth in this example is a consequence of the mismatch between the overall

lengthscale of the marginal distributions (captured by the median interpoint distances) and the lengthscales at which dependencies are present". In fact, the exponent in the distance-induced kernel of a distance covariance test plays the same role as the index of the translation invariant kernel of a stable distribution in an $HSIC$ test. Indistinguishable power functions can be obtained by choosing sufficiently small scale parameters in the translation invariant kernel. This all means that $HSIC$ with sufficiently small scale parameters always match distance covariance in terms of power. But $HSIC$ with scale parameters appropriately selected may, in some cases, improve on distance covariance.

3.3 Asymptotic Distribution

Empirical processes as in (2) have been recently very useful at tackling problems related to mutual independence because of the simplicity of the asymptotic distribution. Let $d_B = \sum_{j\in B} d_j$. Each process R_{nB} is defined on the space $C(\mathbb{R}^{d_B}, \mathbb{C})$ of complex-valued continuous functions defined on $\mathbb{R}^{d_B}$. Let $d = \sum_{j=1}^{p} d_j$ and $t = (t_1, \ldots, t_d) \in \mathbb{R}^d$. The following mild tail condition (Csörgő, 1981, 1985) is assumed

$$\int_0^1 \frac{\bar{\psi}(h)}{h\left(\log\frac{1}{h}\right)^{1/2}} dh < \infty, \tag{14}$$

where

$$\bar{\psi}(h) = \sup\left\{y:\ 0 \le y \le 1,\ \lambda_d\left\{t:\ |t|_\infty < \frac{1}{2},\ \psi(t) < y\right\} < h\right\},$$

with λ_d standing for the Lebesgue measure in $\mathbb{R}^d$ and $|t|_\infty = \max(|t_1|, \ldots, |t_d|)$, is the nondecreasing rearrangement of the function $\psi(t) = [1 - Ref(t)]^{1/2}$. Weak convergence of the collection of processes R_{nB} is now established. For details concerning the metrics on the spaces in Theorem 5, the reader is referred to the appendix. The symbol $\Rightarrow$ stands for weak convergence.

Theorem 5 *If $Z^{(1)}, \ldots, Z^{(p)}$ are mutually independent, then the process $R_{nB} \Rightarrow R_B$ in $C(\mathbb{R}^{d_B}, \mathbb{C})$, where R_B is a zero mean complex Gaussian process with $\bar{R}_B(t^{(B)}) = R_B(-t^{(B)})$ and complex covariance function*

$$\mathbb{E}\left[R_B(t^{(B)})\bar{R}_B(s^{(B)})\right] = \prod_{j\in B}[f^{(j)}(t^{(j)} - s^{(j)}) - f^{(j)}(t^{(j)})f^{(j)}(-s^{(j)})].$$

Moreover, the collection of processes $(R_{nB}:\ B \in \mathcal{I}_p) \Rightarrow (R_B:\ B \in \mathcal{I}_p)$ on the product of spaces $\underset{B\in\mathcal{I}_p}{\times} C(\mathbb{R}^{d_B}, \mathbb{C})$ to a zero mean Gaussian process such that the marginal processes R_B, $B \in \mathcal{I}_p$, are mutually independent.

The convergence of functionals (4) also holds even though they are not defined on the whole space $C(\mathbb{R}^{d_B}, \mathbb{C})$, but only on the space of squared integrable functions. In the next theorem, the asymptotic distribution of $n\mathcal{H}_{nB}^2$ is described. In particular, it provides the asymptotic distribution of $n\mathcal{H}_{nB}^{2(\alpha)}$, for any $\alpha \in (0, 2]$.

Theorem 6 *Let $W_{nB} = \mathcal{H}_{nB}^2$. If $Z^{(1)}, \ldots, Z^{(p)}$ are mutually independent, then $nW_{nB} \Rightarrow W_B$ for each $B \in \mathcal{I}_p$, where $W_B = \int |R_B(t^{(B)})|^2 \prod_{j\in B} dG^{(j)}(t^{(j)})$. Moreover, the collection*

of variables $(nW_{nB} : \ B \in \mathcal{I}_p) \Rightarrow (W_B : \ B \in \mathcal{I}_p)$*, where the variables* W_B*,* $B \in \mathcal{I}_p$*, are mutually independent.*

Theorem 6 holds without moment conditions for $\mathcal{H}^2_{nB}$ since in this case dw_B is a probability measure. Theorems 5 and 6 were proved when $p = 2$ by Zhang et al. (2011). The case for $p = 3$ was covered in Appendix E of Sejdinovic et al. (2013a). The distribution of W_B can be represented using the Karhunen-Loève expansion. Without loss of generality, let $B = \{1, \ldots, k\}$. Then, W_B is distributed as

$$\sum_{i_1=1}^{\infty} \cdots \sum_{i_k=1}^{\infty} \lambda^{(1)}_{i_1} \ldots \lambda^{(k)}_{i_k} Z^2_{i_1 \ldots i_k}, \tag{15}$$

where $Z_{i_1 \ldots i_k}$ are independent standard normal variables, and for $j = 1, \ldots, k$, $\lambda^{(j)}_1, \lambda^{(j)}_2, \ldots$ are eigenvalues depending only on the probability measure $P^{(j)}$ of $Z^{(j)}$ and the weighting probability measure $dG^{(j)}$. From arguments as in Sejdinovic et al. (2013a) and Sejdinovic et al. (2013b), the eigenvalues $\lambda^{(j)}_1, \lambda^{(j)}_2, \ldots$ are those of the integral operator

$$S_{\tilde{k}^{(j)}} g(s^{(j)}) = \int_{\mathbb{R}^{d_j}} \tilde{k}^{(j)}(s^{(j)}, t^{(j)}) g(t^{(j)}) dP^{(j)}(t^{(j)}),$$

where $k^{(j)}(s^{(j)}, t^{(j)}) = \varphi^{(j)}(t^{(j)} - s^{(j)})$ is the translation invariant kernel (9) and

$$\tilde{k}^{(j)}(s^{(j)}, t^{(j)}) = k^{(j)}(s^{(j)}, t^{(j)}) - \mathbb{E}k^{(j)}(Z^{(j)}_1, t^{(j)}) - \mathbb{E}k^{(j)}(s^{(j)}, Z^{(j)}_2) + \mathbb{E}k^{(j)}(Z^{(j)}_1, Z^{(j)}_2), \tag{16}$$

for $Z^{(j)}_1$, $Z^{(j)}_2$ independent and distributed according to $P^{(j)}$, is the corresponding doubly-centered kernel.

The same type of results for $|B| = 2$ have been obtained by Székely et al. (2007) for distance covariance, see also Lyons (2013, Corollary 2.8 and Remark 2.9) for the product structure of eigenvalues. For $|B| > 2$, provided that $\mathbb{E} \prod_{j \in B} |Z^{(j)}_1 - Z^{(j)}_2|^{2\alpha}_{d_j}$ is finite, the V-statistic structure of $W_{nB} = \mathcal{V}^{2(\alpha)}_{nB}$ can be used as in Lyons (2013, Theorem 2.7) or Sejdinovic et al. (2013a) to show that $nW_{nB} \Rightarrow W_B$, where W_B is of the same form as in (15). However, this V-statistic argument does not establish the asymptotic independence of the collection of variables $(W_B : \ B \in \mathcal{I}_p)$. It can not be proven either as in Theorem 6 since the generalization of a result of Kellermeier (1980) used in the proof no longer holds since it is based on Jensen's inequality which is valid only for probability measures. The asymptotic independence of the collection could be concluded if the collection of processes $(R_{nB} : \ B \in \mathcal{I}_p)$ were independent for each n, unfortunately this is not the case. For distance covariance, the asymptotic independence of the collection remains unanswered.

In Section 6, p-values of global tests, such as tests of Fisher in (18) or Tippett in (19), computed by combining individual p-values for each $B \in \mathcal{I}_p$ must be approximated. Simple approximations assume that individual p-values are mutually independent. However, in finite samples with sample size as small as 100, an approximation relying on randomization tests which does not rely on the independence of individual p-values was found to yield better conformity of global p-values to the nominal significance level of 0.05.

4. Other Functionals

The test of Kojadinovic and Holmes (2009, Proposition 13) is based on the Möbius decomposition of the independence empirical copula process and is defined as the Cramér-von Mises functional (4) in which the process R_{nB} in (2) defined with empirical characteristic functions is replaced by an analogous process defined with empirical copulas. Also, the integrating measure dw_B is replaced by the uniform distribution over the hypercube. The dependence statistics W_{nB} thus obtained will be denoted KH^2_{nB}. The statistics KH^2_{nB} can also be represented in the form (8) with appropriately defined terms $a^{(j)}_{kl}$.

The test of Beran et al. (2007) uses the Möbius decomposition of the independence empirical half-space process and is defined as the Kolmogorov statistic obtained by taking the supnorm of the processes. The Kolmogorov statistics do not have an explicit form as in (8). They must be computed by solving a costly optimization problem over discretized unit spheres of dimensions d_j and p-values are approximated by the bootstrap. Distance covariance and $HSIC$ tests are only orthogonally invariant, whereas the test of Beran et al. (2007) is invariant to the general linear group. The heavy computational cost of this test makes it unsuitable for large scale simulations of power functions. For this reason, it will not be considered in the simulations of Section 10.

A global test of mutual independence can also be constructed directly from the empirical characteristic independence process (Csörgő, 1985; Kankainen, 1995; Sejdinovic et al., 2013a; Fan et al., 2017),

$$\begin{aligned} \mathcal{J}^2_n &= \int \left| f_n(t^{(1)}, \ldots, t^{(p)}) - \prod_{j=1}^{p} f^{(j)}_n(t^{(j)}) \right|^2 \prod_{j=1}^{p} dG^{(j)}(t^{(j)}) \\ &= \frac{1}{n^2} \sum_{k=1}^{n} \sum_{l=1}^{n} \prod_{j=1}^{p} a^{(j)}_{kl} - \frac{2}{n^{p+1}} \sum_{k=1}^{n} \prod_{j=1}^{p} \sum_{l=1}^{n} a^{(j)}_{kl} + \frac{1}{n^{2p}} \prod_{j=1}^{p} \sum_{k=1}^{n} \sum_{l=1}^{n} a^{(j)}_{kl}, \end{aligned} \tag{17}$$

where $dG^{(j)}$, $j = 1, \ldots, p$, are probability measures and $a^{(j)}_{kl}$ is defined in (9). The special case of the stable distribution of index α with $a^{(j)}_{kl}$ defined in (10) yields a statistic denoted as $\mathcal{J}^{2(\alpha)}_n$. This choice of weight function, except for $\alpha = 2$, seems to have been overlooked in the literature and no reference to this choice appears in the recent paper by (Fan et al., 2017). In the more general context of random variables taking values in separable metric spaces, Pfister et al. (2017) embedded the joint distribution and the product of the marginals in a reproducing kernel Hilbert space and defined the p-component Hilbert-Schmidt independence criterion as the squared distance between the embeddings. This framework provides a global test of mutual independence which contains (17) as a special case. Also, similar tests to (17) based on empirical distribution functions have been proposed by Blum et al. (1961), Cotterill and Csörgő (1982), and Cotterill and Csörgő (1985) in the univariate case, *i.e.* $d_j = 1$ for $j = 1, \ldots, p$. Kojadinovic and Holmes (2009, Proposition 10) also considered in the multivariate case a test similar to (17) based on empirical copulas.

Simulations in Section 10 will compare $HSIC$ tests $\mathcal{H}^{2(\alpha)}_{nB}$ and distance covariance tests $\mathcal{V}^{2(\alpha)}_{nB}$ to the tests KH^2_{nB} and $\mathcal{J}^{2(2)}_n$. Dependence statistics resulting from the Möbius decomposition have the following advantages over statistics of the type $\mathcal{J}^2_n$ in (17).

1. The statistic W_{nB} in (8) has a simpler structure than that of $\mathcal{J}_n^2$ in (17). The integral operator defining the eigenvalues associated with $\mathcal{J}_n^2$ is an integral of dimension $d = \sum_{j=1}^p d_j$ which can not be written as a product of integral operators as soon as $p > 2$. The integral operator for W_{nB} is always a product of $|B|$ integral operators in smaller dimensions d_j, $j \in B$.

2. The p-values of W_{nB}, $B \in \mathcal{I}_p$, can be combined (see Section 6) to get a global test of mutual independence with a predetermined global significance level.

3. When a global test in item 2 rejects the mutual independence hypothesis, subsets $B \in \mathcal{I}_p$ yielding small p-values can be identified as the possible source of dependence using a dependogram described in Section 7.

5. Approximate p-values

Ways to approximate the null distribution of test statistics are now discussed.

5.1 Spectral Approach

The asymptotic distribution of $n\mathcal{J}_n^2$ in (17) is also an infinite linear combination of chi-squared variables with one degree of freedom with coefficients which are eigenvalues of an integral operator (Fan et al., 2017). Using estimated eigenvalues, they resorted to the algorithm of Imhof (1961), see Duchesne and Lafaye De Micheaux (2010), to compute the cumulative distribution function by a numerical inversion of the characteristic function. For $p = 2$, the spectral approach is also proposed by Zhang et al. (2017). After estimation of eigenvalues, rather than using the algorithm of Imhof (1961), they simulated a large number of values of (15) by generating independent random $N(0, 1)$ variables. The spectral approach to approximate (15) is appropriate for large sample sizes and kernels with an eigenspectrum which decays very rapidly such as the Gaussian kernel. For slowly decaying kernels, the number $(N_\lambda)^k$ of terms in (15), where N_λ is the number of eigenvalues considered, may be too large to apply such methods. The spectral approach for the very simple RV coefficient of Escoufier (1973) is appropriate only in large samples (Josse and Holmes, 2016).

5.2 Resampling Techniques

Another approximation is the *permutation test* which recomputes the statistics for all $(n!)^{p-1}$ permutations. From a theoretical point of view (Hoeffding, 1952), permutation tests are well known to guarantee a non asymptotic control of the significance level (by permutation invariance of the test statistic under the null hypothesis, that is mutual independence here). Since this is not feasible, even for moderate sample sizes, the strategy is to rely on resampling techniques. As an approximation to the permutation test, the *randomization test* simulates the null distribution by permuting (resample without replacement) the observations $Z_1^{(j)}, \ldots, Z_n^{(j)}$, independently for each component, a large number of, say, 1000 permutations. Sejdinovic et al. (2013a) proposed to approximate the null distribution of $n\mathcal{J}_n^2$ in (17) by a randomization test. Another technique is to resample with replacement, independently for each component, a large number of times which is a *bootstrap test*. Bootstrap tests and randomization tests are asymptotically equivalent in the sense that

the resulting critical values and power functions are appropriately close (Romano, 1989; van der Vaart and Wellner, 1996). General references for permutation tests are Efron and Tibshirani (1993), Good (2000), and Pesarin and Salmaso (2010).

5.3 Methods of Moments

Other approximations are based on the method of moments. The distribution obtained by recomputing a statistic for all $(n!)^{p-1}$ permutations is called the permutation distribution of the statistic. The exact first three moments of the permutation distribution of statistics of the general form (8) were obtained by Kazi-Aoual et al. (1995) when $p = 2$ and generalized by Guetsop Nangue (2016) to the case $p \geq 2$. The Pearson type III approximation is a shifted gamma distribution with the same first three moments as the permutation distribution. The Pearson type III distribution is part of the original system of distributions devised by Pearson (1895) in an effort to model visibly skewed observations. The first published paper in which a Pearson type III distribution is used as an approximation to a permutation test is Mielke et al. (1981). For an historical account of the Pearson type III distribution in the context of permutation tests, the reader is referred to Berry et al. (2016, Section 1.2.2). For $p = 2$, Gretton et al. (2008) proposed the approximation by a gamma distribution with the same first two moments as those of the asymptotic distribution, not the permutation distribution, in (15). These first two moments depend on unknown eigenvalues but can be estimated using traces of matrices involving the doubly-centered matrices $A^{(j)} = (A_{kl}^{(j)})$ without having to compute eigenvalues. An empirical failure mode of the gamma approximation was demonstrated in Gretton and Györfi (2010, Figure 1). More recently, Guetsop Nangue (2016, Table 2.9) simulated empirical significance levels of $HSIC$ for meta-Gaussian distributions in the case $p = 2$. The kernel was Gaussian with scaling set at the median of distances. For sample sizes $n = 15$, 25, 50, and 100, the Pearson type III approximation gave rates close to the nominal rate of 0.05 for dimensions up to $d_1 = d_2 = 50$, whereas the gamma approximation gave rates close or equal to 0 as the dimensions were increased. Although the exact permutation test guarantees a non asymptotic control of the significance level, it should be stressed that neither the Pearson type III nor the gamma approximations guarantee an asymptotic control of the significance level. Although p-values of individual statistics W_{nB} can be accurately approximated by the Pearson type III approximation for small samples, the independence of these p-values, guaranteed by the asymptotic distribution, holds to a satisfying degree only in large samples. This is particularly the case as soon as $p > 4$ and $d_j > 1$. Solely for this reason, p-values of global tests which combine individual p-values of tests for sub-hypotheses will be assessed by randomization tests.

5.4 Randomization Tests

Randomization tests are now described to approximate p-values of individual test statistics. Randomization tests for global tests which combine individual p-values are deferred to Section 6. Let Q_n be any test statistic such as $\mathcal{J}_n^2$, $\mathcal{V}_{nB}^{2(\alpha)}$ or $\mathcal{H}_{nB}^{2(\alpha)}$, $B \in \mathcal{I}_p$. Denote a permutation of $\{1, \ldots, n\}$ by $\sigma = (\sigma(1), \ldots, \sigma(n))$. Consider independent random permutations $\sigma_1, \ldots, \sigma_p$ of $\{1, \ldots, n\}$. For $j = 1, \ldots, p$, the permutation σ_j permutes the observations

$(Z_1^{(j)}, \ldots, Z_n^{(j)})$ to yield the permuted data $(Z_{\sigma_j(1)}^{(j)}, \ldots, Z_{\sigma_j(n)}^{(j)})$ of the component j. The statistic Q_n is then recomputed on the permuted data. To this end, note that one need not recompute the elements $a_{kl}^{(j)}$ in (10) or (11). For $\mathcal{J}_n^2$ in (17), one simply permutes according to σ_j the rows and corresponding columns of the $n \times n$ matrix $a^{(j)} = (a_{kl}^{(j)})$. For $\mathcal{V}_{nB}^{2(\alpha)}$ or $\mathcal{H}_{nB}^{2(\alpha)}$, this same argument applies to the matrix $A^{(j)} = (A_{kl}^{(j)})$ in (1).

Approximate p-values are now obtained by recomputing the test statistic N times. Generate Np independent random permutations $\sigma_{i,j}$, $i = 1, \ldots, N$, $j = 1, \ldots, p$.[1] For $i = 1, \ldots, N$, let $Q_{n,i}$ be the test statistic computed from the permuted data $(Z_{\sigma_{i,j}(1)}^{(j)}, \ldots, Z_{\sigma_{i,j}(n)}^{(j)})$, $j = 1, \ldots, p$. An approximate p-value is then obtained as follows.

(1) Let $Q_{n,0}$ be the test statistic computed from the original data.

(2) Generate N randomized samples from the original data and compute $Q_{n,i}$ for $i = 1, \ldots, N$.

(3) An approximate p-value is then given by

$$\frac{1}{N+1}\left[1 + \sum_{i=1}^{N} \mathbb{I}\left\{Q_{n,i} \geq Q_{n,0}\right\}\right].$$

This approximate p-value of the randomization test guaranties (by virtue of exchangeability of the variables $Q_{n,i}$, $i = 0, \ldots, N$) a non-asymptotic control of type I error rates for each individual test (Romano and Wolf, 2005, Lemma 1). The N randomized samples can also be used to compute an approximate quantile of the test statistic Q_n. Let $Q_{n,(1)}, \ldots, Q_{n,(N)}$ be the order statistics of $Q_{n,1}, \ldots, Q_{n,N}$, an approximate π-quantile of Q_n is the order statistic $Q_{n,(\lfloor N\pi \rfloor)}$. The finite sample control of the significance level of randomization tests will be assessed by simulations in Section 10.

6. Combining p-values

If the exact distribution function F_{nB} of W_{nB} was known, then $1 - F_{nB}(W_{nB})$ would be the exact p-value. Since F_{nB} is continuous, this p-value would be exactly uniformly distributed over the interval $(0, 1)$. Moreover, exact p-values obtained by varying B would also be approximately independent in large samples. This holds due to the asymptotic mutual independence of statistics W_{nB} in Theorem 6.

As an approximation, a p-value $\hat{p}_{nB}$ is computed for every dependence statistic W_{nB}, $B \in \mathcal{I}_p$, using randomization tests as in Section 5. Under the mutual independence hypothesis, these $2^p - p - 1$ p-values are, for large samples, approximately independent and also approximately uniformly distributed on $(0, 1)$. The quantity $-2 \log \hat{p}_{nB}$ thus has approximately a χ_2^2 null distribution. Fisher's global test statistic rejects the mutual independence when $-2 \sum_{B \in \mathcal{I}_p} \log \hat{p}_{nB}$ is large. It reports a global p-value of

$$\hat{p}_n = P\left(\chi_\nu^2 > -2 \sum_{B \in \mathcal{I}_p} \log \hat{p}_{nB}\right), \text{ where } \nu = 2(2^p - p - 1). \tag{18}$$

1. Since observations in one component can be held fixed, $N(p-1)$ permutations would suffice. However, treating all components the same way is convenient for purposes of notation and software development.

The number of subsets may be too large for some p or only low order interaction terms may be of interest. In this case, p-values of a test of mutual independence of order q, where $2 \leq q \leq p$, are computed as

$$\hat{p}_n = P\left(\chi_\nu^2 > -2 \sum_{B \in \mathcal{I}_p, |B| \leq q} \log \hat{p}_{nB}\right), \text{ where } \nu = 2 \sum_{i=2}^{q} \binom{p}{i}.$$

Another way of combining independent p-values leads to Tippett's global test with a rejection region consisting of small values of $\min_{B \in \mathcal{I}_p} \hat{p}_{nB}$. Tippett's test of mutual independence of order q reports a combined p-value of

$$\hat{p}_n = 1 - \left(1 - \min_{B \in \mathcal{I}_p, |B| \leq q} \hat{p}_{nB}\right)^r, \text{ where } r = \sum_{i=2}^{q} \binom{p}{i}. \tag{19}$$

In the simulations of Section 10, although each individual p-value is well approximated by a uniform variable on $(0, 1)$, the global p-values obtained by Fisher's (18) or Tippett's (19) method sometimes lead to a global test of significance level which exceeds the nominal level. This was observed especially for $p > 4$ and some $d_j > 1$ and can be attributed to the lack of independence between statistics W_{nB} in finite samples. For a test of mutual independence of order q, $2 \leq q \leq p$, it is preferable to obtain the global p-values using randomized samples as follows (Kojadinovic and Holmes, 2009).

(1) Compute the statistics $W_{nB,0}$, $B \in \mathcal{I}_p$ and $|B| \leq q$, from the original data.

(2) Generate N randomized samples from the original data and let $W_{nB,i}$ be the statistics from the ith randomized sample.

(3) An approximate p-value for the statistic $W_{nB,j}$ based on Romano and Wolf (2005) [2] is

$$\psi(W_{nB,j}) = \frac{1}{N+1}\left[1 + \sum_{i \in \{0,\ldots,N\} \setminus \{j\}} \mathbb{I}\left\{W_{nB,i} \geq W_{nB,j}\right\}\right], \; j = 0, 1, \ldots, N.$$

(4) For $i = 0, 1, \ldots, N$, compute Fisher's and Tippett's statistics

$$F_{n,i} = -2 \sum_{B \in \mathcal{I}_p, |B| \leq q} \log[\psi(W_{nB,i})] \text{ and } T_{n,i} = \min_{B \in \mathcal{I}_p, |B| \leq q} \psi(W_{nB,i}).$$

(5) Approximate p-values for the global tests of Fisher and Tippett are given respectively by

$$\frac{1}{N+1}\left[1 + \sum_{i=1}^{N} \mathbb{I}\left\{F_{n,i} \geq F_{n,0}\right\}\right] \text{ and } \frac{1}{N+1}\left[1 + \sum_{i=1}^{N} \mathbb{I}\left\{T_{n,i} \leq T_{n,0}\right\}\right].$$

2. The non-asymptotic control of type I error rates can not be concluded from the expression $\psi(W_{nB,j}) = \frac{1}{N+1}\left[\frac{1}{2} + \sum_{i=1}^{N} \mathbb{I}\left\{W_{nB,i} \geq W_{nB,j}\right\}\right]$ in Kojadinovic and Holmes (2009, p. 1152).

Note that variables $F_{n,i}$ (and $T_{n,i}$), $i = 0, \ldots, N$, are exchangeable since this is the case of variables $W_{nB,j}$, and thus also $\psi(W_{nB,j})$, $j = 0, \ldots, N$, for each fixed B. The p-values of the global tests of Fisher and Tippett in item (5) thus lead to a non-asymptotic control of type I global error rates (Romano and Wolf, 2005, Lemma 1).

Neither of the two methods is generally more powerful than the other. When only a few of the individual hypotheses are strongly false, Tippett's method is preferable. To detect alternatives for which many of the individual hypotheses are equally false, Fisher's method is likely to be preferable (Westberg, 1985; Loughin, 2004).

7. Dependogram

The dependogram is described for test statistics W_{nB}, $B \in \mathcal{I}_p$, which can be either $\mathcal{V}_{nB}^{2(\alpha)}$ or $\mathcal{H}_{nB}^{2(\alpha)}$. Genest and Rémillard (2004) introduced the dependogram which is a graphical tool, with subsets ordered by size on the horizontal axis, and corresponding values W_{nB} represented by vertical bars on the vertical axis. For each subset $B \in \mathcal{I}_p$, statistics W_{nB} are computed together with corresponding critical values given by the π-quantile, $c_{\pi B}$, obtained from N randomized samples as in Section 5. Critical values are represented by dashes. Dependence in a subset is declared when the vertical bar extends beyond the dash. If the number of subsets is large, a dependogram of order q, $2 \leq q \leq p$, can be constructed by considering only subsets of maximum cardinality q. Assuming statistics W_{nB} are independent, the choice $\pi = (1-\alpha')^{1/r}$ with $r = \sum_{i=2}^{q} \binom{p}{i}$ leads to a global significance level α'. Indeed, under the mutual independence hypothesis

$$P(W_{nB} \leq c_{\pi B} : \ |B| \leq q) = \prod_{|B| \leq q} P(W_{nB} \leq c_{\pi B}) = \pi^r = 1 - \alpha'.$$

The dependogram should be seen as an exploratory tool that can be used when a global test rejects the mutual independence hypothesis. It helps to identify the dependency structures present in the data as in Figure 4. A dependogram built from $HSIC$ statistics will be normalized as in (13) so that, for small scale parameters, it has roughly the same order of magnitude as the dependogram based on distance covariance statistics.

A difficulty in interpreting a dependogram is now illustrated with an example. Consider a model with three dependent variables $Z^{(1)}$, $Z^{(2)}$ and $Z^{(3)}$, where $Z^{(1)}$ and $Z^{(2)}$ are dependent, but the pair $(Z^{(1)}, Z^{(2)})$ is independent of $Z^{(3)}$. Looking at Table 1 with the true characteristic functions, one can verify that $\mu_{12} \not\equiv 0$, $\mu_{13} \equiv 0$, $\mu_{23} \equiv 0$, and $\mu_{123} \equiv 0$. Therefore, only the test for the subset $\{1, 2\}$ is expected to be significant. The test for the subset $\{1, 2, 3\}$ is not expected to be significant even though the three variables are dependent. Of course, the global test is expected to be significant as it will combine tests for all subsets.

8. Tests of Serial Independence

The problem of testing for serial independence of a multivariate stationary ergodic sequence is now addressed. The test statistic in the serial context is very similar. Consider a stationary ergodic sequence $Y_1, Y_2, \ldots$ in $\mathbb{R}^{d_1}$, where Y_1 is distributed according to the characteristic function $f^{(1)}$. Let $p \geq 2$ be a fixed integer. For a sequence of length m, let $n = m - p + 1$

and

$$Z_k = (Z_k^{(1)}, \ldots, Z_k^{(p)}) = (Y_k, Y_{k+1}, \ldots, Y_{k+p-1}), \ k = 1, \ldots, n, \tag{20}$$

where $Z_k^{(j)} = Y_{k+j-1}$, $j = 1, \ldots, p$. Let $f(t^{(1)}, \ldots, t^{(p)})$ be the joint characteristic function of $Z_1 = (Y_1, \ldots, Y_p)$ and $f^{(B)}$ be the joint characteristic function of $Z_1^{(B)} = (Y_j : \ j \in B)$. For example, if $p = 4$ and the indices in the subset $B = \{1, 2, 3\}$ are translated by one to yield $C = B + 1 = \{2, 3, 4\}$ then, $f^{(B)}$ and $f^{(C)}$ are the same characteristic function since the process is stationary. In the serial context, a subset B and its translate, say $B + k$, can be treated as the same subset. The set $\mathcal{I}_p$ is thus reduced to $\mathcal{B}_p = \{B \in \mathcal{I}_p : 1 \in B\}$ and has now cardinality $2^{p-1} - 1$. For a given $B \in \mathcal{B}_p$, the Möbius transformation of the characteristic function f for the set B is given by

$$\mu_{B,s}(t^{(B)}) = \sum_{C \subseteq B} (-1)^{|B \setminus C|} f^{(C)}(t^{(C)}) \prod_{j \in B \setminus C} f^{(1)}(t^{(j)}).$$

The subscript s stands for serial. The Möbius transformation characterizes the serial independence: $Y_1, \ldots, Y_p$ are mutually independent if and only if, $\mu_{B,s}(t^{(B)}) = 0$, for all $B \in \mathcal{B}_p$, and all vectors $t^{(B)}$.

The corresponding process in the serial case is denoted $R_{nB,s}$ and is defined exactly as in (2). Here, the index set of the process $R_{nB,s}$ is the Euclidean space $\mathbb{R}^{d_B}$, where $d_B = d_1|B|$. Note that all the empirical characteristic functions $f_n^{(j)}$, $j = 1, \ldots, p$, are now essentially the same estimate of the unknown characteristic function $f^{(1)}$. They are not replaced by a single estimate based on all n observations to preserve the representation of the functional (8) in terms of doubly-centered matrices. The dependence statistic for the subset B is now defined as the functional

$$W_{nB,s} = \frac{1}{n} \int |R_{nB,s}(t^{(B)})|^2 \prod_{j \in B} dw(t^{(j)}).$$

It can be computed as before in (8). Due to the stationarity, it should be noted that the same kernel is used to define the elements $a_{kl}^{(j)}$, for all $j \in B$. Two types of weighting measures are considered.

1. For $HSIC$, $dw(t^{(j)}) = dG(t^{(j)})$ is a probability measure with characteristic function φ in which case

$$a_{kl}^{(j)} = \varphi(Z_k^{(j)} - Z_l^{(j)}). \tag{21}$$

The dependence statistic is denoted $W_{nB,s} = \mathcal{H}_{nB,s}^2$. As in Theorem 2, the population Hilbert-Schmidt independence criterion

$$\mathcal{H}_{B,s}^2 = \int |\mu_{B,s}(t^{(B)})|^2 \prod_{j \in B} dG(t^{(j)})$$

is consistently estimated by $\mathcal{H}_{nB,s}^2$. The proof is omitted since it can be proven as Theorem 2 using the ergodic theorem. The special case when $dG(t^{(j)})$ is the probability measure of the stable distribution of index α is denoted $\mathcal{H}_{nB,s}^{2(\alpha)}$, for $\alpha \in (0, 2]$.

2. For distance covariance, $dw(t^{(j)}) = \left[C(d_1,\alpha)|t^{(j)}|_{d_1}^{d_1+\alpha}\right]^{-1}$, in which case $a_{kl}^{(j)} = -|Z_k^{(j)} - Z_l^{(j)}|_{d_1}^{\alpha}$. Under an appropriate moment condition as in Theorem 4, the consistency and weak convergence of the functional, which is denoted $W_{nB,s} = \mathcal{V}_{nB,s}^{2(\alpha)}$ for $\alpha \in (0,2)$, should hold.

Pinkse (1998) proposed a non parametric test based on characteristic functions of serial independence against serial dependence of fixed lag k that is consistent against all such alternatives. It is based on a consistent estimator for an upper bound of $\mathcal{H}_{B,s}^2$ for the subset $B = \{1, 1+k\}$. The control of the global significance level when multiple tests are done for several values of k is not addressed by this author. Diks and Panchenko (2007) constructed a test based on divergence measures between distributions using kernel-based quadratic forms to detect dependence at lag k using the subset $B = \{1, 1+k, \ldots, 1+(c-1)k\}$ of given cardinality c. Their criterion is written as $Q = Q_{11} - 2Q_{12} + Q_{22}$. They used U-statistics as estimates for each term. Unfortunately, an error in the estimates of the last two terms on p. 85 leads to an erroneous statistic. If V-statistics are used instead, it is not hard to verify that their test statistic is of the exact form (17).

Under the hypothesis of serial independence, the vectors $Z_k = (Y_k, Y_{k+1}, \ldots, Y_{k+p-1})$, $k = 1, \ldots, n$, are dependent due to some overlapping Y's. The proof of Theorem 5 is not directly applicable. Nevertheless, a very similar weak convergence theorem still holds.

Theorem 7 *Let $Y_1, Y_2, \ldots$ be independent and identically distributed. Then, for any fixed p, the process $R_{nB,s} \Rightarrow S_B$ in $C(\mathbb{R}^{d_B}, \mathbb{C})$ to a zero mean complex Gaussian processes S_B with complex covariance function*

$$\mathbb{E}\left[S_B(t^{(B)})\bar{S}_B(s^{(B)})\right] = \prod_{j\in B}\left[f^{(1)}(t^{(j)} - s^{(j)}) - f^{(1)}(t^{(j)})f^{(1)}(-s^{(j)})\right].$$

Moreover, the collection of processes $(R_{nB,s} : B \in \mathcal{B}_p) \Rightarrow (S_B : B \in \mathcal{B}_p)$ on the product of spaces $\underset{B\in\mathcal{B}_p}{\times} C(\mathbb{R}^{d_B}, \mathbb{C})$ to a zero mean Gaussian process such that the marginal processes S_B, $B \in \mathcal{B}_p$, are mutually independent.

In the next theorem, the asymptotic distribution of $n\mathcal{H}_{nB,s}^2$ is described. The proof is omitted since it is the same as for Theorem 6.

Theorem 8 *Let $W_{nB,s} = \mathcal{H}_{nB,s}^2$. If $Y_1, Y_2, \ldots$ are independent and identically distributed, then $nW_{nB,s} \Rightarrow W_{B,s}$ for each $B \in \mathcal{B}_p$, where $W_{B,s} = \int |S_B(t^{(B)})|^2 \prod_{j\in B} dG(t^{(j)})$. Moreover, the collection of variables $(nW_{nB,s} : B \in \mathcal{B}_p) \Rightarrow (W_{B,s} : B \in \mathcal{B}_p)$, where the variables $W_{B,s}$, $B \in \mathcal{B}_p$, are mutually independent.*

The distribution of $W_{B,s}$ can be also represented using the Karhunen-Loève expansion. Without loss of generality, let $B = \{1, \ldots, k\}$. Then, $W_{B,s}$ is distributed as in (15), where the eigenvalues $\lambda_1^{(j)}, \lambda_2^{(j)}, \ldots$ no longer depend on j, but only on the probability measure $P^{(1)}$ of Y_1 and the weighting probability measure $dG(t^{(1)})$. The eigenvalues $\lambda_1, \lambda_2, \ldots$ are those of the integral operator

$$S_{\tilde{k}}g(s^{(1)}) = \int_{\mathbb{R}^{d_1}} \tilde{k}(s^{(1)}, t^{(1)})g(t^{(1)})dP(t^{(1)}),$$

where $k(s^{(1)}, t^{(1)}) = \varphi(t^{(1)} - s^{(1)})$ is the translation invariant kernel (21) and

$$\tilde{k}(s^{(1)}, t^{(1)}) = k(s^{(1)}, t^{(1)}) - \mathbb{E}k(Y_1, t^{(1)}) - \mathbb{E}k(s^{(1)}, Y_2) + \mathbb{E}k(Y_1, Y_2),$$

for Y_1, Y_2 independent and distributed according to $P^{(1)}$, is the corresponding doubly-centered kernel.

Computations of p-values of individual tests and global tests in Sections 5 and 6 can also be used for testing for serial independence. Subsets B are now restricted to $B \in \mathcal{B}_p$. A modification to the dependogram is necessary since the null distributions of statistics of the same cardinality are now identical. The critical values $c_{\pi B}$ of Section 7 for all B of the same cardinality are replaced by a single critical value as in Figure 6. Since there are $\binom{p-1}{\omega-1}$ subsets B containing 1 and of cardinality ω, the single critical value is taken as the π-quantile of the amalgamated $N\binom{p-1}{\omega-1}$ statistics $W_{nB,i}$, for $i = 1, \ldots, N$ and $|B| = \omega$.

It has been shown how distance covariance and $HSIC$ tests can be adapted for testing for serial independence. The global test $\mathcal{J}_n^2$ in (17) can also be used for serial independence by defining subsequences of length p as in (20).

9. Computational Aspects

The complexity cost of all randomization tests based on the Möbius decomposition in this paper are of the order $O(n^2 rN)$, where n is the sample size, $r = \sum_{i=2}^{q}\binom{p}{i}$ is the number of subsets considered in a global test of order q, and N is the number of randomized samples. Even for the smallest value $q = 2$, the number of subsets of the order $r = O(p^2)$ increases quadratically with p. These tests thus become rapidly infeasible as n or p increases. The serial statistics in Section 8 have a lower complexity cost since in this case $r = \sum_{i=2}^{q}\binom{p-1}{i-1}$ yields a number of subsets of the order $r = O(p)$ increasing linearly with p when $q = 2$.

The statistics $\mathcal{H}_{nB}^{2(\alpha)}$ and $\mathcal{V}_{nB}^{2(\alpha)}$ are combined as in Section 6 to yield global statistics $\mathcal{H}_n^{2(\alpha)}$ and $\mathcal{V}_n^{2(\alpha)}$, respectively. Scale parameters β_j in (10) must be selected for $\mathcal{H}_{nB}^{2(\alpha)}$. They are set to

$$\beta_j = c_j/\text{med}_{k<l}|Z_k^{(j)} - Z_l^{(j)}|_{d_j},\ j = 1, \ldots, p. \tag{22}$$

The choice $c_j = 1$ leads to the so-called heuristic method suggested by Gretton et al. (2008). Selection of very small constants c_j allows to check the equivalence between distance covariance and $HSIC$ tests as described in (13). Unless stated otherwise, the heuristic method will be used for $HSIC$ tests. Distance covariance and $HSIC$ tests will be compared to two other tests: the test of Kojadinovic and Holmes (2009) denoted KH_n^2 for the mutual independence case, or Kojadinovic and Yan (2011) denoted KY_n^2 for the serial case, and the test $\mathcal{J}_n^{2(2)}$ in (17) with heuristic scale parameters (22) as in Sejdinovic et al. (2013a).

When $d_j = 1$ for all $j = 1, \ldots, p$, the statistics KH_n^2 (or KY_n^2) are replaced by the equivalent statistic of Genest and Rémillard (2004) denoted GR_n^2. KH_n^2 differs from GR_n^2 only in the approximation used for p-values. The former uses a randomization test, whereas the latter takes advantage of the fact that for $d_j = 1$, the test is distribution free implying that critical values can be obtained by simulating the null distribution for given values of n and p. Thus, a single set of critical values can be used for all replicates.

The `copula R` package (Kojadinovic and Yan, 2010) functions `multIndepTest` and `multSerialIndepTest` were used for tests based on KH_n^2 and KY_n^2, respectively. Inci-

dentally, the implementation of KH_n^2 unnecessarily recomputes the doubly-centered matrices $A^{(j)}$ for every permutation in their `C` subroutine `bootstrap_MA_I`. The same package also contains the functions `indepTest` and `serialIndepTest` for GR_n^2. An `R` function for distance covariance and $HSIC$ tests proposed in this paper for $p \geq 2$ is available at `dms.umontreal.ca/~bilodeau`. For $p = 2$, it produces, up to a factor of $1/n$ depending on the definitions of statistics, the same result as the function `dcov.test` of the `energy` package (Rizzo and Szekely, 2016). Computations in Section 10 were done on an Intel(R) Core(TM) i7 with a CPU of 3.20 GHz.

10. Simulated Models

All empirical significance levels and empirical powers in this section are evaluated with 1000 tests (replicates) conducted at a global significance level $\alpha' = 0.05$. The p-values of all randomization tests are based on 1000 permutations (the default). Power results are summarized by graphics with a tick mark at 0.05 on the power axis to check for the conformity of empirical significance levels to the nominal level. Six tests are compared: distance covariance tests $\mathcal{V}_n^{2(1/2)}$ and $\mathcal{V}_n^{2(1)}$, $HSIC$ tests $\mathcal{H}_n^{2(1/2)}$ and $\mathcal{H}_n^{2(1)}$, the test KH_n^2 of Kojadinovic and Holmes (2009) and the test $\mathcal{J}_n^{2(2)}$ in (17). When all $d_j = 1$, the test KH_n^2 is replaced by GR_n^2 as explained in Section 9. Also, for the serial case, the test KH_n^2 is replaced by the test KY_n^2 of Kojadinovic and Yan (2011). Unless stated otherwise, p-values of tests based on the Möbius decomposition are combined using the method of Fisher.

10.1 Copula Models

Power comparisons are made for the Gaussian, Student, Frank and Clayton copulas. A general reference for copulas is the book by Nelsen (2006).

10.1.1 Gaussian and Student Copulas with Bivariate Marginals

As in Kojadinovic and Holmes (2009), a correlation matrix is constructed of the form

$$R = \begin{pmatrix} (1-\rho_w)I_2 + \rho_w J_2 J_2' & \rho_b J_2 J_2' & \rho_b J_2 J_2' \\ \rho_b J_2 J_2' & (1-\rho_w)I_2 + \rho_w J_2 J_2' & \rho_b J_2 J_2' \\ \rho_b J_2 J_2' & \rho_b J_2 J_2' & (1-\rho_w)I_2 + \rho_w J_2 J_2' \end{pmatrix}, \tag{23}$$

where I_2 is the identity matrix of dimension 2 and J_2 is the vector of ones of dimension 2. Notations w and b stand for within and between, respectively. Samples of size $n = 100$ are generated from a multivariate distribution of dimension 6 (Gaussian or Student with 2 degrees of freedom) with correlation matrix R in (23). Probability transforms are then applied so that all 6 variables are uniformly distributed on the interval $(0, 1)$. The resulting vector is partitioned into three two-dimensional vectors. The value of ρ_w is set to 0.5. For the Gaussian model, Figure 1 shows that the best test is KH_n^2 followed closely by $\mathcal{V}_n^{2(1)}$. However, for the Student model, KH_n^2 performs poorly compared to all the other tests, the most powerful test being $\mathcal{H}_n^{2(1/2)}$. For the Student model, the three components are always dependent even when $\rho_b = 0$ which explains the power at $\rho_b = 0$ in the right panel of Figure 1. In fact, the only elliptical distribution for which uncorrelatedness implies independence is the Gaussian distribution (Bilodeau and Brenner, 1999, Proposition 4.11).

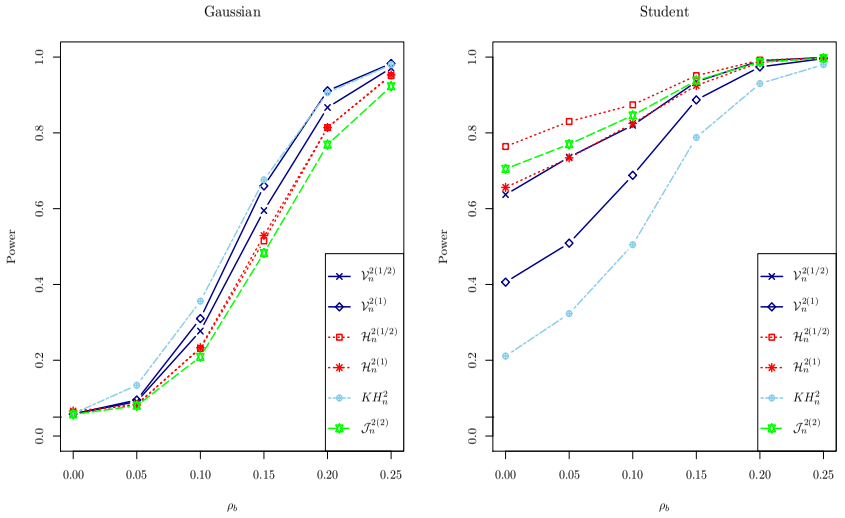


Figure 1: Empirical powers for the Gaussian (left panel) and Student (right panel) copulas with bivariate marginals of Section 10.1.1.

10.1.2 Frank and Clayton Copulas with Univariate Marginals

The Frank and Clayton copulas are now considered for $n = 100$, $p = 3$ and $d_j = 1$ for $j = 1, 2, 3$. These two copulas have a parameter θ with the value $\theta = 0$ corresponding to the independence copula. For the Frank copula, independence is obtained as the limiting case $\theta \to 0$. Figure 2 shows that for both copulas, the tests GR_n^2 and $\mathcal{V}_n^{2(1)}$ have very similar powers and are the most powerful. The least powerful test is $\mathcal{J}_n^{2(2)}$. The $HSIC$ tests are less powerful than their distance covariance counterparts, but their powers could be increased to those of distance covariance by selecting smaller scale parameters as predicted by (13).

10.1.3 Frank Copula with Bivariate Marginals

The Frank copula model is now considered for $p = 3$ and $d_j = 2$ for $j = 1, 2, 3$. The sample size is still 100. A random vector is generated from the Frank copula of dimension six with parameter θ and it is partitioned into three vectors of dimension two. The most powerful test in Figure 3 is $\mathcal{V}_n^{2(1)}$ and the least powerful is KH_n^2.

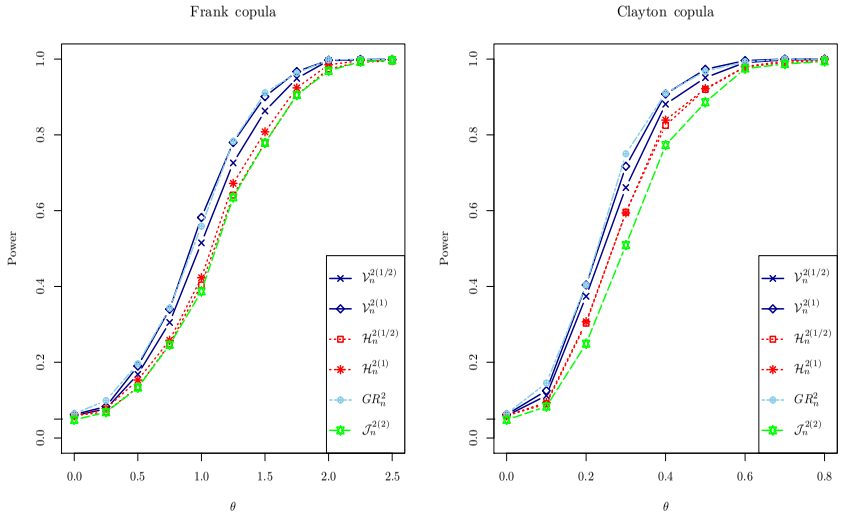


Figure 2: Empirical powers for the Frank (left panel) and Clayton (right panel) copulas with univariate marginals of Section 10.1.2.

10.2 Model of Romano and Siegel

The original model is taken from Kojadinovic and Holmes (2009) which extends an example in Genest and Rémillard (2004). Samples of size $n = 100$ are generated from the distribution of a 12-dimensional random vector as follows.

1. Generate a two-dimensional Gaussian vector $X^{(1)} = (X_1^{(1)}, X_2^{(1)})$ with means 0, variances 1, and covariance 0.5.

2. Generate two independent copies $Z^{(2)}$ and $Z^{(3)}$ of $X^{(1)}$.

3. Define $Z^{(1)} = (Z_1^{(1)}, Z_2^{(1)})$ by $Z_i^{(1)} = |X_i^{(1)}|\text{sign}(Z_1^{(2)} Z_1^{(3)})$, $i = 1, 2$.

4. Generate a three-dimensional Gaussian vector $Z^{(4)}$ with means 0, variances 1, and covariances 0.3.

5. Generate an independent copy $X^{(5)}$ of $Z^{(4)}$.

6. Define $Z^{(5)} = Z^{(4)} + X^{(5)}$.

Following Romano and Siegel (1986), the three two-dimensional vectors $(Z^{(1)}, Z^{(2)}, Z^{(3)})$ are pairwise independent, but not jointly independent. This vector is independent of $(Z^{(4)}, Z^{(5)})$

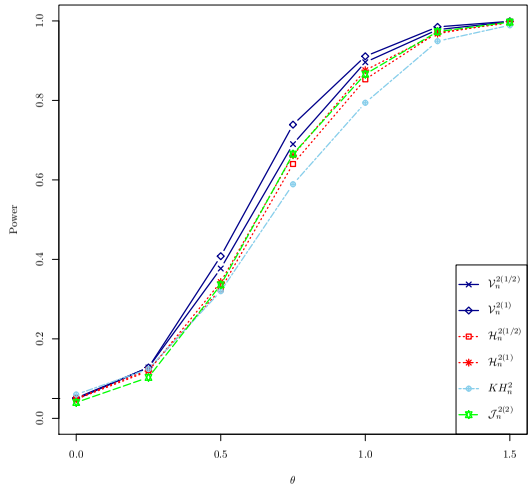


Figure 3: Empirical powers for the Frank copula with bivariate marginals of Section 10.1.3.

in which the two three-dimensional vectors $Z^{(4)}$ and $Z^{(5)}$ are dependent. One can check that the only non null terms μ_B are for the subsets $\{4,5\}$, $\{1,2,3\}$, and $\{1,2,3,4,5\}$.

In order to compare the powers of various tests, a modified model with weaker dependence is introduced. Items 3 and 6 are modified to

3'. Define $Z^{(1)} = (Z_1^{(1)}, Z_2^{(1)})$ by $Z_i^{(1)} = (1-\theta)|X_i^{(1)}| + \theta|X_i^{(1)}|\text{sign}(Z_1^{(2)} Z_1^{(3)})$, $i = 1, 2$,

6'. Define $Z^{(5)} = \theta Z^{(4)} + X^{(5)}$,

for some $\theta \in [0, 0.4]$. Mutual independence now holds among the 5 components for $\theta = 0$ and the dependence increases with θ. The value $\theta = 1$ leading to the original model is not considered since it yields a dependence too easily detected.

Figure 4 is the dependogram based on one simulated sample with $\theta = 0.4$. The statistics computed are $\mathcal{V}_{nB}^{2(1)}$ and $\mathcal{H}_{nB}^{2(1)}$ with very small constants $c_j = .0001$ in (22). It illustrates the equivalence for small scale parameters between $\mathcal{V}_{nB}^{2(\alpha)}$ and $\mathcal{H}_{nB}^{2(\alpha)}$ described precisely in (13). Both dependograms are identical apart from small variations between critical values due to the permutations generated by the randomization tests. The dependence among $Z^{(4)}$ and $Z^{(5)}$, represented by the subset $\{4,5\}$, is significant. Moreover, the third order dependence between $Z^{(1)}$, $Z^{(2)}$ and $Z^{(3)}$, represented by the subset $\{1,2,3\}$, is also significant. The dependence for the subset $\{1,2,3,4,5\}$ is not significant. This can be explained by the powers of tests for false sub-hypotheses corresponding to subsets of small cardinality which

are generally higher because the presence of noise is less important than in subsets of large cardinality. Tests based on the Möbius decomposition combine p-values according to the

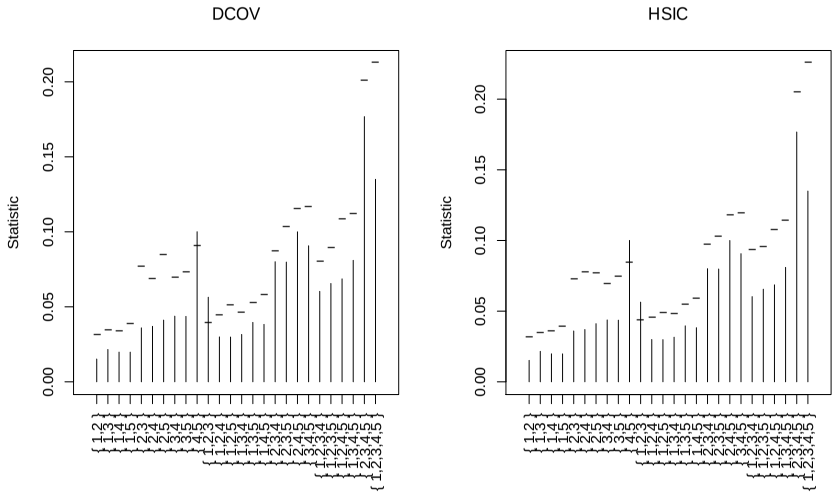


Figure 4: Dependograms for one sample of size $n = 100$ from the modified model of Romano and Siegel with $\theta = 0.4$ in Section 10.2 based on $\mathcal{V}_{nB}^{2(1)}$ (left panel) and $\mathcal{H}_{nB}^{2(1)}$ (right panel) with small constants $c_j = .0001$ in (22).

methods of Tippett and Fisher. Figure 5 shows that KH_n^2 is more powerful than distance covariance and $HSIC$ tests when using the method of Fisher. However, the method of Tippett is markedly more powerful than that of Fisher. This finding should not come as a surprise since only 3 of the 26 sub-hypotheses are false. Using the method of Tippett the most powerful test $\mathcal{V}_n^{2(1)}$ is markedly better than $\mathcal{J}_n^{2(2)}$ and KH_n^2. The popular belief that the method of Fisher is more powerful than that of Tippett should not be given too much consideration. The preferred test depends on the model under consideration.

10.3 Bivariate AR(1) Model

The model considered is the bivariate AR(1) model defined by $Y_k = AY_{k-1} + \epsilon_k$, where the innovations ϵ_k are independently distributed as bivariate Gaussian with mean vector 0 and covariance matrix C. The final specification is made by defining

$$A = \begin{pmatrix} 0 & \theta \\ \theta & 0 \end{pmatrix} \text{ and } C = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

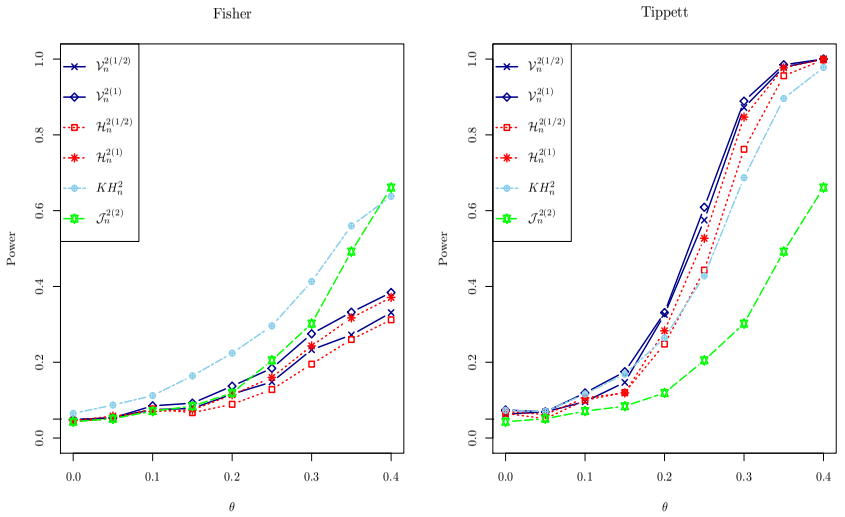


Figure 5: Empirical powers for the modified model of Romano and Siegel in Section 10.2. The methods of Fisher (left panel) and Tippett (right panel) are used to combine p-values.

The serial dependence of the sequence increases with θ and serial independence holds for $\theta = 0$. Tests of Tippett and Fisher are compared. The value $p = 3$ chosen arbitrarily can detect dependencies among three successive observations. In particular, it can detect dependencies at lags one or two. Sequences of length $m = 100$ are generated using the `mAr.sim` function of the `R` package `mAr` (Barbosa, 2012). The dependogram in Figure 6 shows a significant dependence at lag one. It shortly fails to detect the weaker dependence at lag 2 with the short sequence length of 100. Power functions in Figure 7 reveal $\mathcal{V}_n^{2(1)}$ as the most powerful test. Comparisons between Fisher and Tippett tests show comparable powers with a slight advantage for Tippett. Higher powers of $\mathcal{J}_n^{2(2)}$ locally around $\theta = 0$ can be attributed to the higher empirical significance level of this test.

11. Applications

Applications to meteorological and financial data are now provided in the following two sections.

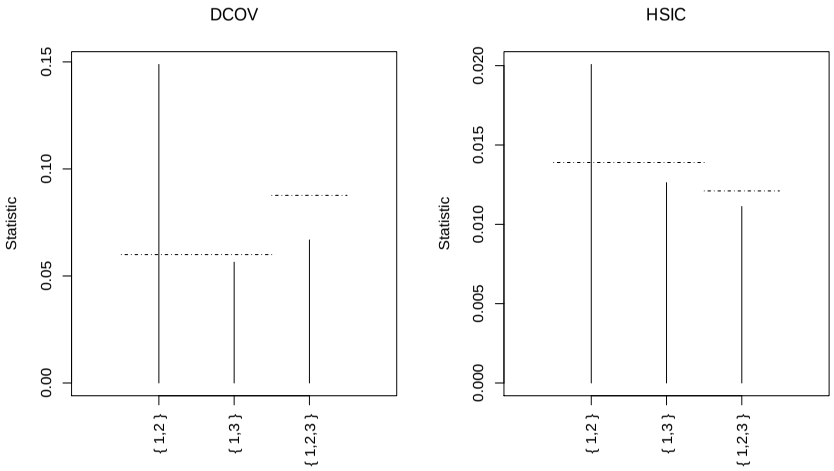


Figure 6: Dependograms for one sequence of length $m = 100$ for the AR(1) model with $\theta = 0.4$ of Section 10.3 based on $\mathcal{V}_{nB}^{2(1)}$ (left panel) and $\mathcal{H}_{nB}^{2(1)}$ (right panel).

11.1 Testing Mutual Independence Between Air Temperature, Soil Temperature, Humidity, Wind and Evaporation

These meteorological data are from Rencher (1995, p. 294). Table 2 describes the data with 46 observations on 11 variables. When a data set consists of variables measured on different scales, the scaling of variables often helps to enhance the appearance of the dependogram. In this application, variables were scaled to zero mean and unit variance. The `R` package `MVN` (Korkmaz et al., 2014) contains the function `hzTest` to perform the test for multivariate normality of Henze and Zirkler (1990). This test applied to the joint distribution of all 11 variables rejected a multivariate Gaussian model with a p-value of 0. Now, five groups of variables are considered as in Table 2. The Gaussian likelihood ratio test found a significant mutual dependence between the five groups. However, this test should not be relied on since it was found that data are not jointly Gaussian and it is well known that Gaussian likelihood ratio tests are not robust (Tyler, 1983; Bilodeau and Brenner, 1999). Mutual independence between these five groups is tested with distance covariance and $HSIC$ tests, both of index 1. Figure 8 contains dependograms only for subsets B of order up to 3. Both dependograms are very similar and lead to the same conclusions. It reveals that air temperature, soil temperature, relative humidity and evaporation are pairwise dependent.

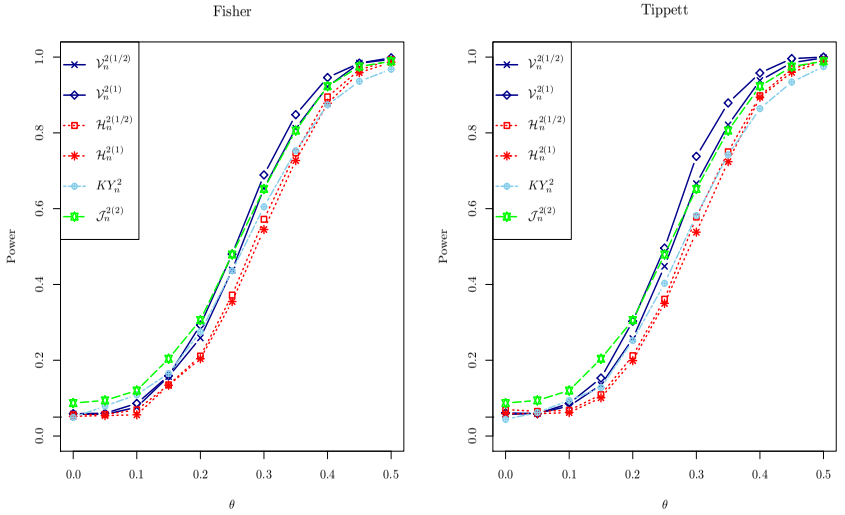


Figure 7: Empirical powers for the AR(1) model of Section 10.3. The methods of Fisher (left panel) and Tippett (right panel) are used to combine p-values.

However, wind does not exhibit any dependence of order 2 or 3 with any of the other 4 groups of variables.

11.2 Testing Serial Independence for Financial Data

Tests of serial independence of the three dimensional sequence formed by the daily percent increasing rates (DPIR) of indices from three stock markets: S&P/TSX composite (TSX), DOW JONES and S&P500. The values of indices are taken at closure. The series of length 534 range from January 2, 2014 to March 2, 2016. Note that five index values are observed weekly since the stock exchanges are not opened on weekends. The top row of Figure 9 shows that the financial series considered are not stationary. It is more appropriate to consider DPIR values. In the bottom row, one may see that DPIR values are more stationary, although still not perfectly stationary.

Tests of serial independence KY_{nB}^2, $\mathcal{V}_{nB}^{2(1)}$, and the test of non serial correlation $\mathcal{V}_{nB}^{2(2)}$ are conducted on the 3 joint series. The value of $p = 10$ allows a maximum lag of 9 days. Figure 10 reveals dependencies at small lags of 1, 2, and 4 in the dependogram of $\mathcal{V}_{nB}^{2(1)}$. The dependogram of KY_{nB}^2 was produced with the `copula` package and does not include critical values. Nevertheless, KY_{nB}^2 and $\mathcal{V}_{nB}^{2(1)}$ agree on the strongest dependency observed

Variables	Labels
$Z^{(1)}$	maximum daily air temperature minimum daily air temperature integrated area under daily air temperature curve
$Z^{(2)}$	maximum daily soil temperature minimum daily soil temperature integrated area under soil temperature
$Z^{(3)}$	maximum daily relative humidity minimum daily relative humidity integrated area under daily humidity curve
$Z^{(4)}$	total wind, measured in miles per day
$Z^{(5)}$	evaporation

Table 2: Variables related to air temperature, soil temperature, humidity, wind and evaporation.

at lag 4. The distance covariance $\mathcal{V}_{nB}^{2(2)}$ of index 2 was also performed on the sequence. One should recall that $\mathcal{V}_{nB}^{2(2)}$ is no longer a test of serial independence, but merely of non serial correlation. Interestingly, this test did not reveal any significant serial correlation. In an attempt to unravel the most significant dependency at lag 4 declared by $\mathcal{V}_{nB}^{2(1)}$, Figure 11 is a scatterplot of DPIR values observed on day k and $k+4$ for the TSX market. The Pearson (p-value of 0.11) and the Kendall (p-value of 0.14) correlation tests applied to this scatterplot are not significant. The test $\mathcal{V}_{nB}^{2(1)}$ for $B = \{1, 5\}$ on the single TSX market is very significant (p-value less than 0.001). A broken line regression with a change point at the origin was fitted to this scatterplot to account for different regimes according to whether DPIR is negative or positive. This regression model has three parameters for the intercept and slopes at the left and right of the origin. The left slope is very significant (p-value of 0.000015) contrary to the right slope (p-value of 0.021). The very significant left slope could be interpreted by the tendency of the TSX market to recover in the days following a decline. The sharper the market declines, the stronger it recovers. Among days such that $\text{DPIR}_k < \xi$, the percentage of days with $\text{DPIR}_{k+4} > 0$ is 60.7% for $\xi = -0.5$. This percentage goes up to 63.4% for $\xi = -1$ and to 68.2% for $\xi = -1.5$. Similar conclusions were found for the DOW JONES and S&P500 stock markets.

12. Conclusion

Generalizations of distance covariance and $HSIC$ tests were done successfully. For both mutual and serial independence hypotheses, the dependence statistics related to distance covariance and $HSIC$ were defined using the Möbius transformation. Simple and explicit expressions for dependence statistics were derived in the explicit form (8) as a sum over all elements of a componentwise product of doubly-centered matrices $A^{(j)} = (A_{kl}^{(j)})$. Computationally efficient approximation of p-values by randomization tests is made possible by this

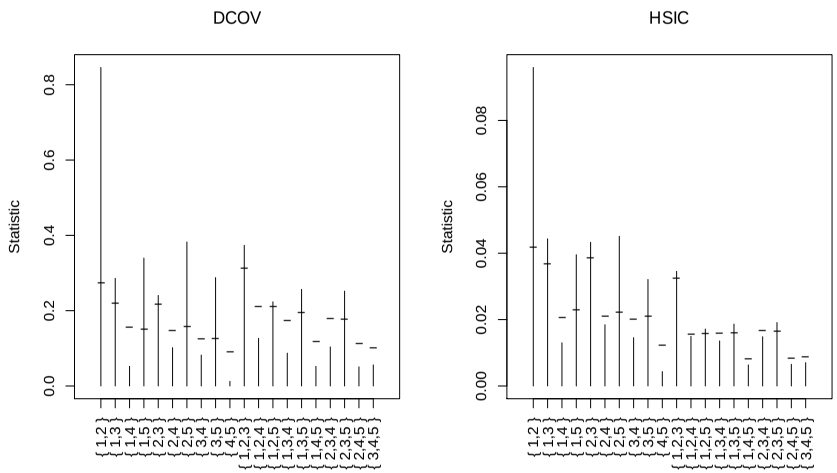


Figure 8: Dependograms of order $q = 3$ of air temperature, soil temperature, relative humidity, wind and evaporation for the meteorological data in Section 11.1 based on $\mathcal{V}_{nB}^{2(1)}$ (left panel) and $\mathcal{H}_{nB}^{2(1)}$ (right panel).

explicit form. Indeed, distances in $A^{(j)}$ do not have to be recomputed since it suffices to permute rows and corresponding columns of $A^{(j)}$ for every randomized sample. The method of combining individual p-values was put forward to construct global tests whose p-values evaluated by randomization tests yielded global significance levels close to the nominal level of 0.05 in all simulated models considered.

Distance covariance tests yielded powers generally very competitive with other tests considered. This paper has presented some advances to the problem of testing independence but some questions remain unanswered. The index α of distance covariance tests has a major influence on power functions. The adaptive selection of this index is a major difficulty which should be the object of future investigations. At the same time, it offers more flexibility and possibilities than tests based on copulas for which the only integrating measure is uniform. $HSIC$ tests can always achieve the same power function as distance covariance tests with the same index simply by selecting very small scale parameters. The additional adaptive selection of scale parameters could possibly yield tests which in certain cases would be more powerful than distance covariance tests. For $p = 2$, Guetsop Nangue (2016) selected the scale parameters which maximize the variance of the permutation distribution. This yielded $HSIC$ tests more powerful than distance covariance tests in two

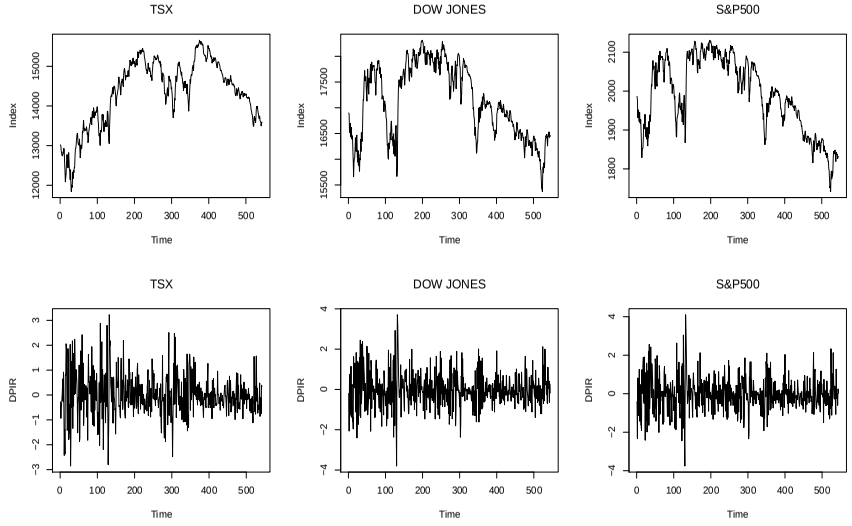


Figure 9: Daily evolution of the S&P/TSX composite (TSX), DOW JONES and S&P500 stock markets. The period of observation ranges from January 2, 2014 to March 2, 2016.

examples form the machine learning community considered in Sejdinovic et al. (2013b): the Independent Component Analysis (ICA) benchmark densities of Bach and Jordan (2002) and the sinusoidally dependent data. The adaptive selection of scale parameters for $p > 2$ is more challenging and is worthy of future research.

Acknowledgments

The authors would like to thank Arthur Gretton and three referees for their careful reading and constructive comments and suggestions which enhanced the quality of the paper. Financial support by the Faculté des arts et des sciences of Université de Montréal during the graduate thesis of the second author was greatly appreciated. This research was done without the financial support from the Natural Sciences and Engineering Research Council of Canada.

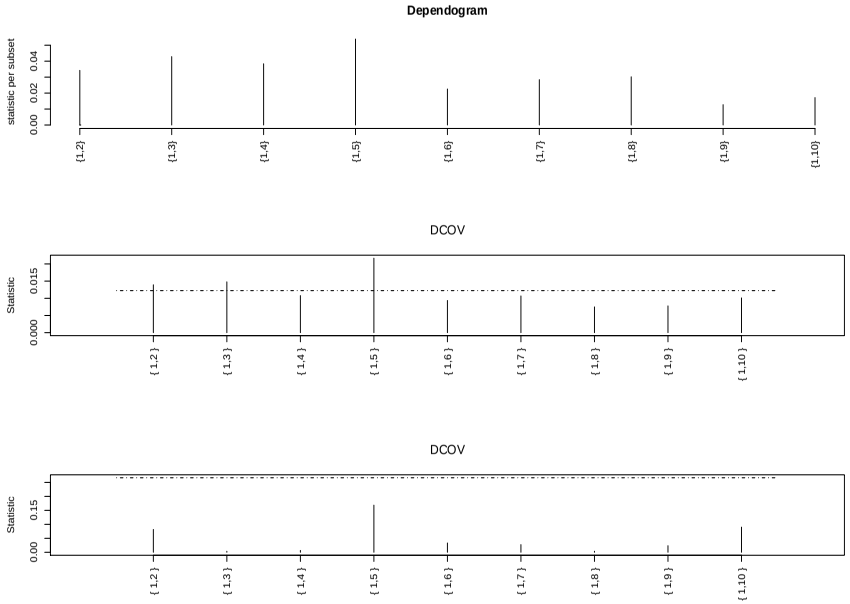


Figure 10: Dependograms for joint DPIR of S&P/TSX composite, DOW JONES, and S&P500. For $p = 10$, the maximum lag is 9. From top to bottom, three tests are considered: KY^2_{nB}, $\mathcal{V}^{2(1)}_{nB}$, and $\mathcal{V}^{2(2)}_{nB}$.

Appendix A: Proofs

Proof [Theorem 1] Upon using the representation (6) of the process,

$$\begin{aligned}\int |R_{nB}(t^{(B)})|^2 \prod_{j\in B} dG^{(j)}(t^{(j)}) &= \frac{1}{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\prod_{j\in B}\int \left[e^{i\langle t^{(j)}, Z_k^{(j)} - Z_l^{(j)}\rangle} - \frac{1}{n}\sum_{v=1}^{n} e^{i\langle t^{(j)}, Z_k^{(j)} - Z_v^{(j)}\rangle}\right. \\ &- \left.\frac{1}{n}\sum_{u=1}^{n} e^{i\langle t^{(j)}, Z_u^{(j)} - Z_l^{(j)}\rangle} + \frac{1}{n^2}\sum_{u=1}^{n}\sum_{v=1}^{n} e^{i\langle t^{(j)}, Z_u^{(j)} - Z_v^{(j)}\rangle}\right] dG^{(j)}(t^{(j)}) \\ &= \frac{1}{n}\sum_{k=1}^{n}\sum_{l=1}^{n}\prod_{j\in B}[a_{kl}^{(j)} - \bar{a}_{k\cdot}^{(j)} - \bar{a}_{\cdot l}^{(j)} + \bar{a}_{\cdot\cdot}^{(j)}],\end{aligned} \tag{24}$$

where $a_{kl}^{(j)} = \varphi^{(j)}(Z_k^{(j)} - Z_l^{(j)})$ and $\varphi^{(j)}$ is the characteristic function of $G^{(j)}$. ■

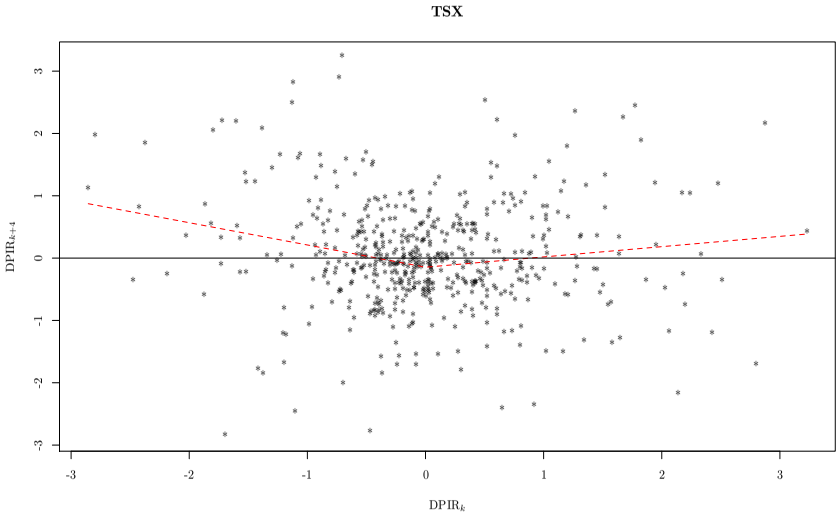


Figure 11: For the TSX market, broken line regression of DPIR on a given day on DPIR four days earlier with a change point at the origin.

Proof [Theorem 2] From (2) and the strong law of large numbers, $R_{nB}(t^{(B)})/\sqrt{n} \overset{a.s.}{\to} \mu_B(t^{(B)})$. Since the number of subsets of B is $2^{|B|}$, it is also clear that $|R_{nB}(t^{(B)})/\sqrt{n}| \leq 2^{|B|}$. Any constant being integrable with respect to the probability measure $\prod_{j\in B} dG^{(j)}(t^{(j)})$, it follows from the dominated convergence theorem that

$$\frac{1}{n}\int |R_{nB}(t^{(B)})|^2 \prod_{j\in B} dG^{(j)}(t^{(j)}) \overset{a.s.}{\to} \int |\mu_B(t^{(B)})|^2 \prod_{j\in B} dG^{(j)}(t^{(j)}),$$

i.e. $\mathcal{H}^2_{nB} \overset{a.s.}{\to} \mathcal{H}^2_B$. This proves (i). To prove (ii), since $\mu_B(t^{(B)}) \neq 0$ and the function μ_B is continuous, it follows that $\mathcal{H}^2_B > 0$. Therefore, $n\mathcal{H}^2_{nB} \overset{a.s.}{\to} \infty$. ■

Proof [Theorem 3] Because of double-centering, the expression between brackets in (24) is unchanged if one is subtracted from all four exponential functions. Then, it suffices in the proof of Theorem 1, for the weight function defining $\mathcal{V}^{2(\alpha)}_{nB}$, to evaluate

$$\int e^{i\langle t^{(j)}, Z^{(j)}\rangle} - 1 dw^{(j)}(t^{(j)}) = -|Z^{(j)}|^\alpha_{d_j} \tag{25}$$

using Lemma 1 of Székely et al. (2007, p. 2771). ■

The representation (6) of the process was unexploited in Székely et al. (2007). It simplifies greatly their derivations.

Proof [Theorem 4] The notation $\mathbb{E}_3$ is for the expectation with respect to Z_3, treating the other variable as a constant to avoid using for conditional expectations. The notation $\mathbb{E}_{12}$ and $\mathbb{E}_{34}$ are defined similarly. Condition (12) implies that

$$\mathbb{E}_{12}\prod_{j\in B}\left||Z_1^{(j)} - Z_2^{(j)}|^\alpha_{d_j} - \mathbb{E}_3|Z_1^{(j)} - Z_3^{(j)}|^\alpha_{d_j} - \mathbb{E}_3|Z_2^{(j)} - Z_3^{(j)}|^\alpha_{d_j} + \mathbb{E}_{34}|Z_3^{(j)} - Z_4^{(j)}|^\alpha_{d_j}\right| < \infty.$$

and that $\mathcal{V}^{2(\alpha)}_B$ is well defined. The integral (25) yields

$$\begin{aligned}
\mathcal{V}^{2(\alpha)}_B &= \mathbb{E}_{12}\prod_{j\in B}\Big[\int e^{i\langle t^{(j)}, Z_1^{(j)} - Z_2^{(j)}\rangle} - 1 dw^{(j)}(t^{(j)}) + \mathbb{E}_3\int 1 - e^{i\langle t^{(j)}, Z_1^{(j)} - Z_3^{(j)}\rangle} dw^{(j)}(t^{(j)}) \\
&\quad + \mathbb{E}_3\int 1 - e^{i\langle t^{(j)}, Z_3^{(j)} - Z_2^{(j)}\rangle} dw^{(j)}(t^{(j)}) + \mathbb{E}_{34}\int e^{i\langle t^{(j)}, Z_3^{(j)} - Z_4^{(j)}\rangle} - 1 dw^{(j)}(t^{(j)})\Big] \\
&= \mathbb{E}_{12}\prod_{j\in B}\mathbb{E}_{34}\int \Big[e^{i\langle t^{(j)}, Z_1^{(j)} - Z_2^{(j)}\rangle} - e^{i\langle t^{(j)}, Z_1^{(j)} - Z_3^{(j)}\rangle} \\
&\quad - e^{i\langle t^{(j)}, Z_3^{(j)} - Z_2^{(j)}\rangle} + e^{i\langle t^{(j)}, Z_3^{(j)} - Z_4^{(j)}\rangle}\Big] dw^{(j)}(t^{(j)}).
\end{aligned}$$

The theorem of Fubini yields

$$\begin{aligned}
\mathcal{V}^{2(\alpha)}_B &= \mathbb{E}_{12}\prod_{j\in B}\int \mathbb{E}_{34}\Big[e^{i\langle t^{(j)}, Z_1^{(j)} - Z_2^{(j)}\rangle} - e^{i\langle t^{(j)}, Z_1^{(j)} - Z_3^{(j)}\rangle} \\
&\quad - e^{i\langle t^{(j)}, Z_3^{(j)} - Z_2^{(j)}\rangle} + e^{i\langle t^{(j)}, Z_3^{(j)} - Z_4^{(j)}\rangle}\Big] dw^{(j)}(t^{(j)}) \\
&= \mathbb{E}_{12}\prod_{j\in B}\int \Big[e^{i\langle t^{(j)}, Z_1^{(j)} - Z_2^{(j)}\rangle} - e^{i\langle t^{(j)}, Z_1^{(j)}\rangle} f^{(j)}(-t^{(j)}) \\
&\quad - e^{-i\langle t^{(j)}, Z_2^{(j)}\rangle} f^{(j)}(t^{(j)}) + f^{(j)}(t^{(j)}) f^{(j)}(-t^{(j)})\Big] dw^{(j)}(t^{(j)}) \\
&= \mathbb{E}_{12}\int \prod_{j\in B}\Big[e^{i\langle t^{(j)}, Z_1^{(j)}\rangle} - f^{(j)}(t^{(j)})\Big]\Big[e^{-i\langle t^{(j)}, Z_2^{(j)}\rangle} - f^{(j)}(-t^{(j)})\Big] dw_B(t^{(B)}) \\
&= \int \mathbb{E}_1\prod_{j\in B}\Big[e^{i\langle t^{(j)}, Z_1^{(j)}\rangle} - f^{(j)}(t^{(j)})\Big]\mathbb{E}_2\prod_{j\in B}\Big[e^{-i\langle t^{(j)}, Z_2^{(j)}\rangle} - f^{(j)}(-t^{(j)})\Big] dw_B(t^{(B)}).
\end{aligned}$$

By a similar development leading to (6),

$$\mu_B(t^{(B)}) = \mathbb{E}_1\prod_{j\in B}\Big[e^{i\langle t^{(j)}, Z_1^{(j)}\rangle} - f^{(j)}(t^{(j)})\Big].$$

Hence, $\mathcal{V}^{2(\alpha)}_B = \int |\mu_B(t^{(B)})|^2 dw_B(t^{(B)})$. This proves (i). For (ii), if $\mu_B(t^{(B)}) \neq 0$, then necessarily $t^{(B)} \neq 0$. Consider a compact ball $\mathfrak{B}$ not containing 0 and but containing the point $t^{(B)}$. The measure dw_B is integrable on $\mathfrak{B}$. Then, arguing as in the proof of Theorem 2

$$\mathcal{V}^{2(\alpha)}_{nB} \geq \frac{1}{n}\int_{\mathfrak{B}} |R_n(t^{(B)})|^2 dw_B(t^{(B)}) \overset{a.s.}{\to} \int_{\mathfrak{B}} |\mu_B(t^{(B)})|^2 dw_B(t^{(B)}) > 0$$

and therefore, $n\mathcal{V}_{nB}^{2(\alpha)} \xrightarrow{a.s.} \infty$. ■

Proof [Equation (13)] The result follows using the invariance by translation, $a_{kl}^{(j)} \mapsto a_{kl}^{(j)} - 1$, and the following limit,

$$\lim_{\beta_j \to 0} \frac{e^{-\beta_j^\alpha |Z_k^{(j)} - Z_l^{(j)}|_{d_j}^\alpha} - 1}{\beta_j^\alpha} = -|Z_k^{(j)} - Z_l^{(j)}|_{d_j}^\alpha.$$

■

Let $d_B = \sum_{j \in B} d_j$. Define the metric (Whitt, 1970)

$$\rho_B(x, y) = \sum_{s=1}^{\infty} 2^{-s} \frac{\rho_{sB}(x, y)}{1 + \rho_{sB}(x, y)},$$

where

$$\rho_{sB}(x, y) = \sup_{|t^{(B)}|_{d_B} \leq s} |x(t^{(B)}) - y(t^{(B)})|,$$

on the linear complete metric space of continuous functions $\mathcal{C}(\mathbb{R}^{d_B}, \mathbb{C})$. The Borel σ-field in $\mathcal{C}(\mathbb{R}^{d_B}, \mathbb{C})$ is generated by the coordinate projections, *i.e.* it is the smallest σ-field with respect to which all coordinate projections are measurable. Weak convergence of random variables in $\mathcal{C}(\mathbb{R}^{d_B}, \mathbb{C})$ is equivalent to weak convergence on any compact subset; see Whitt (1970, Theorem 5) or Kallenberg (2002, Proposition 16.6). Moreover, weak converge of a sequence on a compact subset is equivalent to finite dimensional weak convergence and tightness of that sequence. The metric defined on the product of spaces $\underset{B \in \mathcal{I}_p}{\times} C(\mathbb{R}^{d_B}, \mathbb{C})$ (Whitt, 1970) is

$$\rho\left((x_B,\ B \in \mathcal{I}_p), (y_B,\ B \in \mathcal{I}_p)\right) = \max_{B \in \mathcal{I}_p} \rho_B(x_B, y_B).$$

From Whitt (1970, Corollary 7), weak convergence on this product of spaces is equivalent to finite dimensional weak convergence of the joint process and tightness on compacta of each individual process.

Proof [Theorem 5] The process $R_{nB}(t)$ in (6) is closely related to the process

$$\breve{R}_{nB}(t) = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} \prod_{j \in B} \left[e^{i\langle t^{(j)}, Z_k^{(j)} \rangle} - f^{(j)}(t^{(j)}) \right]. \tag{26}$$

in which marginal characteristic functions are not estimated. The process $\breve{R}_{nB}$ in (26) is a sum of independent and identically distributed random variables. Bilodeau and Lafaye de Micheaux (2005, Theorem 2.1) proved that the collection of processes $\breve{R}_{nB}$ converges as stated in Theorem 5 under the weak condition (14). The independence of the asymptotic processes for $B \neq C$ is verified

$$\mathbb{E}\left[R_B(t^{(B)}) \bar{R}_C(s^{(C)}) \right] = \mathbb{E}\left\{ \prod_{j \in B} \left[e^{i\langle t^{(j)}, Z_k^{(j)} \rangle} - f^{(j)}(t^{(j)}) \right] \prod_{j \in C} \left[e^{i\langle s^{(j)}, -Z_k^{(j)} \rangle} - f^{(j)}(-s^{(j)}) \right] \right\}$$

$$= 0,$$

because there is an index j in B, but not in C, or the converse, for which the corresponding term has expectation zero. Then, it suffices to show $\rho_{sB}(R_{nB}, \breve{R}_{nB}) \xrightarrow{P} 0$, for all $s \geq 1$ and $B \in \mathcal{I}_p$. The representation in Ghoudi et al. (2001, p. 212) holds for characteristic functions

$$R_{nB}(t) = \sum_{C \subseteq B} (-1)^{|C|} \prod_{j \in C} [f_n^{(j)}(t^{(j)}) - f^{(j)}(t^{(j)})] \breve{R}_{n,B\setminus C}(t^{(B\setminus C)}). \tag{27}$$

From (27), it follows that

$$|R_{nB}(t^{(B)}) - \breve{R}_{nB}(t^{(B)})| \leq \sum_{C \subseteq B, C \neq \emptyset} \prod_{j \in C} |f_n^{(j)}(t^{(j)}) - f^{(j)}(t^{(j)})| |\breve{R}_{n,B\setminus C}(t^{(B\setminus C)})|, \tag{28}$$

where the sum has only a finite number of terms. Using the Glivenko-Cantelli convergence in Csörgő (1981, Theorem 2.1) and the fact that the processes $\breve{R}_{n,B\setminus C}$ are tight, it follows that, for any $s \geq 1$, $\rho_s(R_{nB}, \breve{R}_{nB}) \xrightarrow{P} 0$. ■

Proof [Theorem 6] Let

$$|[R_{nB}(t^{(B)})]|^2 = \int |R_{nB}(t^{(B)})|^2 \prod_{j \in B} dG^{(j)}(t^{(j)})$$

be the squared L_2 norm, on the space of squared integrable functions, of the process R_{nB}. Then, $nW_{nB} = |[R_{nB}]|^2$ and $n\breve{T}_{nB} = |[\breve{R}_{nB}]|^2$. Use $\int \tilde{k}(t^{(B)}, t^{(B)}) \prod_{j \in B} dG^{(j)}(t^{(j)}) < \infty$, where $\tilde{k}$ is defined in (16), and Tonelli's theorem to conclude that $W_B := |[R_B]|^2$ is finite almost surely. The proof consists in showing the following two results (Henze and Wagner, 1997): (i) $|[\breve{R}_{nB}]|^2 \Rightarrow |[R_B]|^2$ and (ii) $|[R_{nB} - \breve{R}_{nB}]|^2 \xrightarrow{P} 0$. Note that from (i) and the continuous mapping theorem, $|[\breve{R}_{nB}]| \Rightarrow |[R_B]|$, which, with the triangle inequality $\Big|\, |[R_{nB}]| - |[\breve{R}_{nB}]| \,\Big| \leq |[R_{nB} - \breve{R}_{nB}]|$ and (ii), implies $|[R_{nB}]| \Rightarrow |[R_B]|$ and, thus $|[R_{nB}]|^2 \Rightarrow |[R_B]|^2$. Using a slight generalization of Kellermeier (1980, Theorem 3.3), it suffices for (i) to show the following uniform integrability condition:

$$\lim_{N \to \infty} \limsup_{n \to \infty} \int_{\{|t^{(j)}|_{d_j} > N, \forall j \in B\}} \mathbb{E}|\breve{R}_{nB}(t^{(B)})|^2 \prod_{j \in B} dG^{(j)}(t^{(j)}) = 0. \tag{29}$$

It can be verified that $\mathbb{E}|\breve{R}_{nB}(t^{(B)})|^2 = \prod_{j \in B} \left[1 - |f^{(j)}(t^{(j)})|^2\right]$ does not depend on n. Now, for each $j \in B$,

$$\int 1 - |f^{(j)}(t^{(j)})|^2 dG^{(j)}(t^{(j)}) < \infty$$

since the integrand is bounded. This proves (29). For (ii), since all processes $\breve{R}_{n,B\setminus C}$ are tight, it suffices from (28) to show that for all $j \in B$,

$$\int |f_n^{(j)}(t^{(j)}) - f^{(j)}(t^{(j)})|^2 dG^{(j)}(t^{(j)}) \xrightarrow{P} 0.$$

This follows from the dominated convergence theorem, since $f_n^{(j)}(t^{(j)}) \overset{a.s.}{\to} f^{(j)}(t^{(j)})$ and the integrand is bounded by 4. The weak convergence of the collection $(|[R_{nB}]|^2,\ B \in \mathcal{I}_p)$ follows by the same argument above from (i') $(|[\breve{R}_{nB}]|^2,\ B \in \mathcal{I}_p) \Rightarrow (|[R_B]|^2,\ B \in \mathcal{I}_p)$ and (ii) $|[R_{nB} - \breve{R}_{nB}]|^2 \overset{P}{\to} 0$, for each $B \in \mathcal{I}_p$. Thus, it remains to establish (i'). From Theorem 5 and the continuous mapping theorem, $\sum_{B\in\mathcal{I}_p} c_B|\breve{R}_{nB}(t^{(B)})|^2 \Rightarrow \sum_{B\in\mathcal{I}_p} c_B|R_B(t^{(B)})|^2$, for all constants c_B, $B \in \mathcal{I}_p$. Also, $\sum_{B\in\mathcal{I}_p} c_B|[\breve{R}_{nB}]|^2 \Rightarrow \sum_{B\in\mathcal{I}_p} c_B|[R_B]|^2$ because the following uniform integrability condition

$$\lim_{N\to\infty} \limsup_{n\to\infty} \int_{\{|t^{(j)}|_{d_j}>N, j=1,\ldots,p\}} \mathbb{E}\left|\sum_{B\in\mathcal{I}_p} c_B|\breve{R}_{nB}(t^{(B)})|^2\right| \prod_{j=1}^{p} dG^{(j)}(t^{(j)})$$
$$\leq \sum_{B\in\mathcal{I}_p} |c_B| \lim_{N\to\infty} \limsup_{n\to\infty} \int_{\{|t^{(j)}|_{d_j}>N, \forall j\in B\}} \mathbb{E}|\breve{R}_{nB}(t^{(B)})|^2 \prod_{j\in B} dG^{(j)}(t^{(j)}) = 0$$

is satisfied from (29). Finally, (i') follows from the Cramér-Wold theorem. The mutual independence of $(|[R_B]|^2,\ B \in \mathcal{I}_p)$ follows from that of $(R_B,\ B \in \mathcal{I}_p)$ in Theorem 5. ■

Proof [Theorem 7] Consider the processes

$$\breve{R}_{nB,s}(t^{(B)}) = \frac{1}{\sqrt{n}} \sum_{k=1}^{n} \prod_{j\in B} \left[e^{i\langle t^{(j)}, Z_k^{(j)}\rangle} - f^{(1)}(t^{(j)})\right],\ B \in \mathcal{B}_p.$$

Finite dimensional weak convergence of the processes is proved. Because of overlapping of Y's in consecutive Z_k's, the Z_k's form an $(p-1)$-dependent sequence, see Ferguson (1996, p. 69). Thus, the central limit theorem for such dependent sequences establishes that $\breve{R}_{nB,s}(t^{(B)})$ and $\breve{R}_{nC,s}(s^{(C)})$ are asymptotically and jointly normal with asymptotic covariance $\sigma_{0,0} + 2\sigma_{0,1} + \cdots + 2\sigma_{0,p-1}$, where

$$\sigma_{0,u} = \mathbb{E}\left\{\prod_{j\in B}\left[e^{i\langle t^{(j)}, Z_k^{(j)}\rangle} - f^{(1)}(t^{(j)})\right] \prod_{j\in C}\left[e^{i\langle s^{(j)}, -Z_{k+u}^{(j)}\rangle} - f^{(1)}(-s^{(j)})\right]\right\}.$$

All of the above expectations are null unless $B = C$ (both in $\mathcal{B}_p$) and $u = 0$. Next, to establish weak convergence of the process on any compact, assume without loss of generality that n is a multiple of p, say $n = rp$. This amounts to neglecting at most $p-1$ terms in the sequence. Rewrite the sequence $Z_1, Z_2, \ldots$ as an array with p rows, each consisting of r independent and identically distributed vectors,

$$\begin{array}{cccc} Z_1 & Z_{1+p} & \cdots & Z_{1+(r-1)p} \\ Z_2 & Z_{2+p} & \cdots & Z_{2+(r-1)p} \\ \vdots & \vdots & \ddots & \vdots \\ Z_p & Z_{p+p} & \cdots & Z_{p+(r-1)p}. \end{array}$$

Then, the expression

$$\breve{R}_{nB,s}(t^{(B)}) = \frac{1}{\sqrt{p}} \sum_{h=1}^{p} \sum_{C\subseteq B} (-1)^{|B\setminus C|} \prod_{j\in B\setminus C} f^{(1)}(t^{(j)}) \cdot \frac{1}{\sqrt{r}} \sum_{i=0}^{r-1} \left[e^{i\langle t^{(C)}, Z_{pi+h}^{(C)}\rangle} - \prod_{j\in C} f^{(1)}(t^{(j)})\right]$$

establishes weak convergence since for each pair (h, C) in finite number, the last sum over i is an empirical characteristic function process over a compact. Finally, $R_{nB,s}$ and $\breve{R}_{nB,s}$ are equivalent processes follows from the inequality

$$|R_{nB,s}(t^{(B)}) - \breve{R}_{nB,s}(t^{(B)})| \leq \sum_{C\subseteq B, C\neq\emptyset} \prod_{j\in C} |f_n^{(j)}(t^{(j)}) - f^{(1)}(t^{(j)})||\breve{R}_{nB\setminus C,s}(t^{(B\setminus C)})|,$$

and the same arguments following (28). ■

References

F. R. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48, 2002.

S. M. Barbosa. *mAr: Multivariate AutoRegressive analysis*, 2012. URL `https://CRAN.R-project.org/package=mAr`. R package version 1.1-2.

R. Beran, M. Bilodeau, and P. Lafaye de Micheaux. Nonparametric tests of independence between random vectors. *Journal of Multivariate Analysis*, 98(9):1805–1824, 2007.

K. J. Berry, P. W. Mielke, Jr., and J. E. Johnston. *Permutation Statistical Methods*. Springer International Publishing, 2016.

M. Bilodeau and D. Brenner. *Theory of Multivariate Statistics*. Springer Texts in Statistics. Springer-Verlag, New York, 1999.

M. Bilodeau and P. Lafaye de Micheaux. A multivariate empirical characteristic function test of independence with normal marginals. *Journal of Multivariate Analysis*, 95(2): 345–369, 2005.

M. Bilodeau and P. Lafaye de Micheaux. A-dependence statistics for mutual and serial independence of categorical variables. *Journal of Statistical Planning and Inference*, 139 (7):2407–2419, 2009.

J. R. Blum, J. Kiefer, and M. Rosenblatt. Distribution free tests of independence based on the sample distribution function. *Annals of Mathematical Statistics*, 32(2):485–498, 1961.

D. S. Cotterill and M. Csörgő. On the limiting distribution of and critical values for the multivariate Cramér-von Mises statistic. *The Annals of Statistics*, 10(1):233–244, 1982.

D. S. Cotterill and M. Csörgő. On the limiting distribution of and critical values for the Hoeffding, Blum, Kiefer, Rosenblatt independence criterion. *Statistics & Decisions. International Journal for Statistical Theory and Related Fields*, 3(1-2):1–48, 1985.

S. Csörgő. Multivariate empirical characteristic functions. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 55(2):203–229, 1981.

S. Csörgő. Testing for independence by the empirical characteristic function. *Journal of Multivariate Analysis*, 16(3):290–299, 1985.

P. Deheuvels. An asymptotic decomposition for multivariate distribution-free tests of independence. *Journal of Multivariate Analysis*, 11(1):102–113, 1981.

C. Diks and V. Panchenko. Nonparametric tests for serial independence based on quadratic forms. *Statistica Sinica*, 17:81–98, 2007.

P. Duchesne and P. Lafaye De Micheaux. Computing the distribution of quadratic forms: Further comparisons between the Liu-Tang-Zhang approximation and exact methods. *Computational Statistics and Data Analysis*, 54(4):858–862, 2010.

P. Duchesne, K. Ghoudi, and B. Rémillard. On testing for independence between the innovations of several time series. *The Canadian Journal of Statistics*, 40(3):447–479, 2012.

B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*, volume 57 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, New York, 1993.

Y. Escoufier. Le traitement des variables vectorielles. *Biometrics*, 29(4):751–760, 1973.

Y. Fan, P. Lafaye de Micheaux, S. Penev, and D. Salopek. Multivariate nonparametric test of independence. *Journal of Multivariate Analysis*, 153:189–210, 2017.

T. S. Ferguson. *A Course in Large Sample Theory*. Chapman & Hall Texts in Statistical Science Series. Taylor & Francis, 1996.

A. Feuerverger. A consistent test for bivariate dependence. *International Statistical Review*, 61(3):419–433, 1993.

R. A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, London, 11th edition, 1950.

C. Genest and B. Rémillard. Tests of independence or randomness based on the empirical copula process. *Test*, 13(2):335–370, 2004.

K. Ghoudi, R. J. Kulperger, and B. Rémillard. A nonparametric test of serial independence for time series and residuals. *Journal of Multivariate Analysis*, 79(2):191–218, 2001.

P. Good. *A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer Series in Statistics. Springer-Verlag, New York, second edition, 2000.

A. Gretton and L. Györfi. Consistent nonparametric tests of independence. *Journal of Machine Learning Research*, 11:1391–1423, 2010.

A. Gretton, O. Bousquet, A. Smola, and B. Scholköpf. Measuring statistical dependence with Hilbert-Schmidt norms. In S. Jain, H. Simon, and E. Tomita, editors, *Algorithmic Learning Theory*, volume 3734 of *Lecture Notes in Computer Science*, pages 63–77. Springer Berlin Heidelberg, 2005.

A. Gretton, K. Fukumizu, C.-H. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems 20*, pages 585–592. MIT Press, Cambridge, MA, 2008.

A. Gretton, K. Fukumizu, and B. K. Sriperumbudur. Discussion of: Brownian distance covariance. *The Annals of Applied Statistics*, 3(4):1285–1294, 2009.

A. Guetsop Nangue. *Tests de permutation d'indépendance en analyse multivariée*. PhD thesis, Université de Montréal, 2016.

N. Henze and T. Wagner. A new approach to the BHEP tests for multivariate normality. *Journal of Multivariate Analysis*, 62(1):1–23, 1997.

N. Henze and B. Zirkler. A class of invariant consistent tests for multivariate normality. *Communications in Statistics-Theory and Methods*, 19(10):3595–3617, 1990.

W. Hoeffding. The large-sample power of tests based on permutations of observations. *Annals of Mathematical Statistics*, 23(2):169–192, 1952.

J. P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48(3-4):419–426, 1961.

J. Josse and S. Holmes. Measuring multivariate association and beyond. *Statistics Surveys*, 10:132–167, 2016.

O. Kallenberg. *Foundations of Modern Probability*. Probability and its applications. Springer, New York, 2002.

A. Kankainen. *Consistent Testing of Total Independence Based on the Empirical Characteristic Function*. PhD thesis, University of Jyväskylä, 1995.

F. Kazi-Aoual, S. Hitier, R. Sabatier, and J.-D. Lebreton. Refined approximations to permutation tests for multivariate inference. *Computational Statistics & Data Analysis*, 20(6):643–656, 1995.

J. Kellermeier. The empirical characteristic function and large sample hypothesis testing. *Journal of Multivariate Analysis*, 10(1):78–87, 1980.

I. Kojadinovic and M. Holmes. Tests of independence among continuous random vectors based on Cramér-von Mises functionals of the empirical copula process. *Journal of Multivariate Analysis*, 100(6):1137–1154, 2009.

I. Kojadinovic and J. Yan. Modeling multivariate distributions with continuous margins using the copula R package. *Journal of Statistical Software*, 34(9):1–20, 2010.

I. Kojadinovic and J. Yan. Tests of serial independence for continuous multivariate time series based on a Möbius decomposition of the independence empirical copula process. *Annals of the Institute of Statistical Mathematics*, 63(2):347–373, 2011.

S. Korkmaz, D. Goksuluk, and G. Zararsiz. MVN: An R package for assessing multivariate normality. *The R Journal*, 6(2):151–162, 2014.

H. O. Lancaster. Complex contingency tables treated by the partition of χ^2. *Journal of the Royal Statistical Society. Series B. Methodological*, 13(2):242–249, 1951.

P. Lévy. *Calcul des probabilités*. Gauthier-Villars, Paris, 1925.

T. M. Loughin. A systematic comparison of methods for combining p-values from independent tests. *Computational Statistics & Data Analysis*, 47(3):467–485, 2004.

R. Lyons. Distance covariance in metric spaces. *The Annals of Probability*, 41(5):3284–3305, 2013.

P. W. Mielke, Jr, K. J. Berry, and G. W. Brier. Application of multi-response permutation procedures for examining seasonal changes in monthly mean sea-level pressure patterns. *Monthly Weather Review*, 109:120–126, 1981.

R. B. Nelsen. *An Introduction to Copulas*. Springer Series in Statistics. Springer, New York, second edition, 2006.

K. Pearson. Contributions to the mathematical theory of evolution. II. Skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 186:343–414, 1895.

F. Pesarin and L. Salmaso. *Permutation Tests for Complex Data. Theory, Applications and Software*. Wiley Series in Probability and Statistics. John Wiley & Sons, Chichester, UK, 2010.

N. Pfister, P. Bühlmann, B. Schölkopf, and J. Peters. Kernel-based tests for joint independence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2017. URL `http://dx.doi.org/10.1111/rssb.12235`.

J. Pinkse. A consistent nonparametric test for serial independence. *Journal of Econometrics*, 84(2):205–231, 1998.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL `https://www.R-project.org/`.

A. C. Rencher. *Methods of Multivariate Analysis*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons, New York, 1995.

M. L. Rizzo and G. J. Szekely. *E-Statistics: Multivariate Inference via the Energy of Data*, 2016. URL `https://CRAN.R-project.org/package=energy`. R package version 1.7-0.

J. Romano and A. Siegel. *Counterexamples in Probability and Statistics*. Wadsworth, London, 1986.

J. P. Romano. Bootstrap and randomization tests of some nonparametric hypotheses. *The Annals of Statistics*, 17(1):141–159, 1989.

J. P. Romano and M. Wolf. Exact and approximate stepdown methods for multiple hypothesis testing. *Journal of the American Statistical Association*, 100(469):94–108, 2005.

D. Sejdinovic, A. Gretton, and W. Bergsma. A kernel test for three-variable interactions. In *Advances in Neural Information Processing Systems 26*, pages 1126–1134. Curran Associates, Inc., Red Hook, NY, 2013a.

D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5): 2263–2291, 2013b.

G. J. Székely and M. L. Rizzo. Rejoinder: Brownian distance covariance. *The Annals of Applied Statistics*, 3(4):1303–1308, 2009.

G. J. Székely, M. L. Rizzo, and N. K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

L. H. C. Tippett. *The Methods of Statistics*. John Wiley & Sons, Inc., New York, N. Y.; Williams & Norgate, Ltd., London, 4th edition, 1952.

D. E. Tyler. Robustness and efficiency properties of scatter matrices. *Biometrika*, 70(2): 411–420, 1983.

A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer Series in Statistics. Springer-Verlag, New York, 1996.

M. Westberg. Combining independent statistical tests. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 34(3):287–296, 1985.

W. Whitt. Weak convergence of probability measures on the function space $C[0, \infty)$. *The Annals of Mathematical Statistics*, 41(3):939–944, 1970.

K. Zhang, J. Peters, D. Janzing, and B. Scholköpf. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 804–813. AUAI Press, Corvallis, Oregon, 2011.

Q. Zhang, S. Filippi, A. Gretton, and D. Sejdinovic. Large-scale kernel methods for independence testing. *Statistics and Computing*, pages 1–18, 2017.

Journal of Machine Learning Research 18 (2017) 1-34 Submitted 5/16; Revised 6/17; Published 8/17

Recovering PCA and Sparse PCA via Hybrid-(ℓ_1, ℓ_2) Sparse Sampling of Data Elements

Abhisek Kundu ABHISEKKUNDU@GMAIL.COM
Intel Parallel Computing Labs
Intel Tech (I) Pvt Ltd, Devarabeesanhalli, Outer Ring Road
Bangalore, 560103, India

Petros Drineas PDRINEAS@PURDUE.EDU
Computer Science
Purdue University
West Lafayette, IN 47907, USA

Malik Magdon-Ismail MAGDON@CS.RPI.EDU
Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180, USA

Editor: David Wipf

Abstract

This paper addresses how well we can recover a data matrix when only given a few of its elements. We present a randomized algorithm that element-wise sparsifies the data, retaining only a few of its entries. Our new algorithm independently samples the data using probabilities that depend on both squares (ℓ_2 sampling) and absolute values (ℓ_1 sampling) of the entries. We prove that this hybrid algorithm (i) achieves a near-PCA reconstruction of the data, and (ii) recovers sparse principal components of the data, from a sketch formed by a sublinear sample size. Hybrid-(ℓ_1, ℓ_2) inherits the ℓ_2-ability to sample the important elements, as well as the regularization properties of ℓ_1 sampling, and maintains strictly better quality than either ℓ_1 or ℓ_2 on their own. Extensive experimental results on synthetic, image, text, biological, and financial data show that not only are we able to recover PCA and sparse PCA from incomplete data, but we can speed up such computations significantly using our sparse sketch .

Keywords: element-wise sampling, sparse representation, pca, sparse pca, hybrid-(ℓ_1, ℓ_2)

1. Introduction

We address two fundamental data science problems, namely, (i) a near-PCA reconstruction of the data, and (ii) recovering sparse principal components of the data, in a setting where we can use only a few of the data entries (*element-wise matrix sparsification* problem was pioneered by Achlioptas and McSherry 2001, 2007). This is a situation that one is confronted with all too often in machine learning (say, we have a small sample of data points and those data points have missing features). For example, with user-recommendation data, one does not have all the ratings of any given user. Or in a privacy preserving setting, a client may not want to give us all entries in the data matrix. In such a setting, our goal is to show that if the samples that we get are chosen carefully, the top-k PCA and sparse PCA features of the data can be recovered within some provable error bounds. In fact, we show that solutions of a large class of optimization problems, irrespective of whether they are efficiently solvable (e.g. PCA) or NP-hard (e.g. sparse PCA), can be approximated provably by such element-wise sparse representation of the data.

More formally, the data matrix is $\mathbf{A} \in \mathbb{R}^{m\times n}$ (m data points in n dimensions). Often, real data matrices have low effective rank, so let $\mathbf{A}_k$ be the best rank-k approximation to $\mathbf{A}$ with $\|\mathbf{A} - \mathbf{A}_k\|_2$ being small, where $\|\mathbf{X}\|_2$ is the spectral norm of matrix $\mathbf{X}$. $\mathbf{A}_k$ is obtained by projecting $\mathbf{A}$ onto the subspace spanned by its top-k principal components. In order to approximate this top-k principal subspace, we adopt the following strategy. Select a small number, s, of elements from $\mathbf{A}$ and produce a sparse sketch $\tilde{\mathbf{A}}$; use $\tilde{\mathbf{A}}$ to approximate the top-k principal subspace. Note that both PCA and the sparse PCA problem have the same objective, i.e., to maximize the variance of the data; however, the sparse PCA problem has an additional sparsity constraint on each principal component (and this makes it NP-hard). Sections 2.1 and 2.2 contains the formulations of PCA and sparse PCA, respectively. We give the details of Algorithm 2 to approximate the top-k principal subspace and the corresponding theoretical guarantees in Theorem 3. Theorem 4 shows the quality of approximation for the sparse PCA problem (and this error bound is applicable to a large class of optimization problems as long as the objective function satisfies a generalized notion of Lipschitz continuity to be discussed later). The key quantity that we must control to recover a close approximation to PCA and sparse PCA is how well the sparse sketch approximates the data *in the operator norm*. That is, if $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ is small then we can recover PCA and sparse PCA effectively from a small number of samples.

Problem: Element-wise Matrix Sparsification

Given $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\epsilon > 0$, sample s elements to obtain a sparse sketch $\tilde{\mathbf{A}}$ for which

$$\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 \leq \epsilon \quad \text{and} \quad \|\tilde{\mathbf{A}}\|_0 \leq s. \tag{1}$$

See SECTION 1.1 for notations. Our main result addresses the problem above. In a nutshell, with only partially observed data whose elements have been carefully selected, one can recover an approximation to the top-k principal subspace. An additional benefit is that our approximation to the top-k subspace using iterated matrix multiplication (e.g., power iterations) can benefit computationally from sparsity. To construct $\tilde{\mathbf{A}}$, we use a general randomized approach which independently samples (and rescales) s elements from $\mathbf{A}$ using probability p_{ij} to sample element $\mathbf{A}_{ij}$. We analyze in detail the case $p_{ij} \propto \alpha|\mathbf{A}_{ij}| + (1-\alpha)|\mathbf{A}_{ij}|^2$ to get a bound on $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$. We now make our discussion precise, starting with our notation.

1.1 Notation

We use bold uppercase (e.g., $\mathbf{X}$) for matrices and bold lowercase (e.g., $\mathbf{x}$) for column vectors. The i-th row of $\mathbf{X}$ is $\mathbf{X}_{(i)}$, and the i-th column of $\mathbf{X}$ is $\mathbf{X}^{(i)}$. Let $[n]$ denote the set $\{1, 2, ..., n\}$. $\mathbb{E}(X)$ is the expectation of a random variable X; for a matrix, $\mathbb{E}(\mathbf{X})$ denotes the element-wise expectation. For a matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, the Frobenius norm $\|\mathbf{X}\|_F$ is $\|\mathbf{X}\|_F^2 = \sum_{i,j=1}^{m,n} \mathbf{X}_{ij}^2$, and the spectral (operator) norm $\|\mathbf{X}\|_2$ is $\|\mathbf{X}\|_2 = \max_{\|\mathbf{y}\|_2=1} \|\mathbf{X}\mathbf{y}\|_2$. We also have the ℓ_1 and ℓ_0 norms: $\|\mathbf{X}\|_1 = \sum_{i,j=1}^{m,n} |\mathbf{X}_{ij}|$ and $\|\mathbf{X}\|_0$ is the number of non-zero entries in $\mathbf{X}$. The k-th largest singular value of $\mathbf{X}$ is $\sigma_k(\mathbf{X})$. For symmetric matrices $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{Y} \succeq \mathbf{X}$ if and only if $\mathbf{Y} - \mathbf{X}$ is positive semi-definite. $\mathbf{I}_n$ is the $n \times n$ identity and $\ln x$ is the natural logarithm of x. We use $\mathbf{e}_i$ to denote standard basis vectors whose dimensions will be clear from the context.

©2017 Abhisek Kundu, Petros Drineas and Malik Magdon-Ismail.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-258.html.

Two popular sampling probabilities are ℓ_1, where $p_{ij} = |\mathbf{A}_{ij}|/\|\mathbf{A}\|_1$ (Achlioptas and McSherry 2001; Achlioptas et al. 2013a), and ℓ_2, where $p_{ij} = \mathbf{A}_{ij}^2/\|\mathbf{A}\|_F^2$ (Achlioptas and McSherry 2001; Drineas and Zouzias 2011). We construct $\tilde{\mathbf{A}}$ as follows: $\tilde{\mathbf{A}}_{ij} = 0$ if (i,j)-th entry is not sampled; sampled elements $\mathbf{A}_{ij}$ are rescaled to $\tilde{\mathbf{A}}_{ij} = \mathbf{A}_{ij}/p_{ij}$ which makes the sketch $\tilde{\mathbf{A}}$ an unbiased estimator of $\mathbf{A}$, so $\mathbb{E}[\tilde{\mathbf{A}}] = \mathbf{A}$. The sketch is *sparse* if the number of sampled elements is sublinear in mn, i.e., $s = o(mn)$. Sampling according to element magnitudes is natural in many applications, for example in a recommendation system users tend to rate a product they like (high positive) or dislike (high negative).

Our main sparsification algorithm (Algorithm 1) receives as input a matrix $\mathbf{A}$ and an accuracy parameter $\epsilon > 0$, and samples s elements from $\mathbf{A}$ in s independent, identically distributed trials with replacement, according to a hybrid-(ℓ_1, ℓ_2) probability distribution specified in Equation (3). The algorithm returns $\tilde{\mathbf{A}} \in \mathbb{R}^{m\times n}$, a sparse and unbiased estimator of $\mathbf{A}$, as a solution to (1).

1.2 Prior Work on Element Sampling

Achlioptas and McSherry (2001, 2007) pioneered the idea of ℓ_2 sampling for element-wise sparsification. However, ℓ_2 sampling on its own is insufficient for provably accurate bounds on $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$. As a matter of fact Achlioptas and McSherry (2001, 2007) observed that "small" entries need to be sampled with probabilities that depend on their absolute values only, thus also introducing the notion of ℓ_1 sampling. The underlying reason for the need of ℓ_1 sampling is the fact that if a small element is sampled and rescaled using ℓ_2 sampling, this would result in a huge entry in $\tilde{\mathbf{A}}$ (because of the rescaling). As a result, the variance of ℓ_2 sampling is quite high, resulting in poor theoretical and experimental behavior. ℓ_1 sampling of small entries rectifies this issue by reducing the variance of the overall approach. Arora et al. (2006) proposed a sparsification algorithm that deterministically keeps large entries, i.e., entries of $\mathbf{A}$ such that $|\mathbf{A}_{ij}| \geq \epsilon/\sqrt{n}$ and randomly rounds the remaining entries using ℓ_1 sampling. Formally, entries of $\mathbf{A}$ that are smaller than $\epsilon\sqrt{n}$ are set to sign $(\mathbf{A}_{ij})\,\epsilon/\sqrt{n}$ with probability $p_{ij} = \sqrt{n}\,|\mathbf{A}_{ij}|/\epsilon$ and to zero otherwise. They used an ϵ-net argument to show that $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ was bounded with high probability. Drineas and Zouzias (2011) bypassed the need for ℓ_1 sampling by zeroing-out the small entries of $\mathbf{A}$ (e.g., all entries such that $|\mathbf{A}_{ij}| < \epsilon/2n$ for a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$) and then use ℓ_2 sampling on the remaining entries in order to sparsify the matrix. This simple modification improves Achlioptas and McSherry (2007) and Arora et al. (2006), and comes with an elegant proof using the matrix-Bernstein inequality of Recht (2011). Note that all these approaches need truncation of small entries. Recently, Achlioptas et al. (2013a) showed that ℓ_1 sampling in isolation could be done without any truncation, and argued that (under certain assumptions) ℓ_1 sampling would be better than ℓ_2 sampling, even using the truncation. Their proof is also based on the matrix-valued Bernstein inequality of Recht (2011). Finally, the result that is closest to ours is due to Achlioptas et al. (2013b), where the proposed distribution is

$$p_{ij} = \rho_i \cdot |\mathbf{A}_{ij}|\,/\|\mathbf{A}_{(i)}\|_1 \quad \text{(Bernstein distribution)} \tag{2}$$

where, ρ_i is a distribution over the rows, i.e., $\sum_i \rho_i = 1$. This distribution is derived using matrix-Bernstein inequality for a fixed sampling budget s, and under some assumptions it is shown to be near-optimal in the sense that the error $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ it produces is within a constant factor of the smallest error produced by an optimal distribution over elements $\mathbf{A}_{ij}$. Similar to this distribution we also use a convex combination of probabilities. However, our proposed hybrid distribution only relies on ℓ_1 and ℓ_2 probabilities, thus requiring considerably less prior information about the data

$\mathbf{A}$ comparing to Bernstein distribution (Achlioptas et al. 2013b), while also being more faithful to how matrices in practice are sampled locally based on element properties, and not based on global properties of a matrix.

1.3 Our Contributions

We introduce an intuitive hybrid approach to element-wise matrix sparsification, by combining ℓ_1 and ℓ_2 sampling. We propose to use sampling probabilities of the form

$$p_{ij} = \alpha \cdot \frac{|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_1} + (1-\alpha)\frac{\mathbf{A}_{ij}^2}{\|\mathbf{A}\|_F^2}, \quad \alpha \in (0,1] \tag{3}$$

for all i, j . We retain the good properties of ℓ_2 sampling that bias us towards data elements in the presence of small noise, while *regularizing* smaller entries using ℓ_1 sampling. We summarize the main contributions below.

• We give a parameterized sampling distribution in the variable $\alpha \in (0,1]$ that controls the balance between ℓ_2 sampling and ℓ_1 regularization. This greater flexibility allows us to achieve better accuracy than both ℓ_1 and ℓ_2. Further, we derive the optimal hybrid-(ℓ_1, ℓ_2) distribution, using Lemma 1 for any given $\mathbf{A}$, by computing the optimal parameter α^* which achieves the desired accuracy with the smallest sample size using our theoretical bound.

Setting $\alpha = 1$ in our bounds we reproduce the result of Achlioptas et al. (2013a) who claim that ℓ_1 is typically better than ℓ_2. Moreover, our result shows that $\alpha^* < 1$ which means that the hybrid approach is better than ℓ_1 (and ℓ_2). Thus, our result generalizes the result of Achlioptas et al. (2013a). Note that the Bernstein distribution of Achlioptas et al. (2013b) (for a given s) is a convex combination of 'intra-row' weights $|\mathbf{A}_{ij}|\,/\|\mathbf{A}\|_1$, where as, our distribution (for a given ϵ) is a much simpler and intuitive convex combination of ℓ_1 and ℓ_2 probabilities.

• We propose Algorithm 2 to provably recover PCA by constructing a sparse unbiased sketch of (centered) data from a limited number of samples. Moreover, we show how we can effectively approximate the sparse PCA problem (NP-hard) from incomplete data. In fact, we prove how sampling based sketches can approximate the solutions for a large class of optimization problems (irrespective of their computational hardness) from limited number of observed data points.

For the above problems we want to derive probabilities using little or no global information about the data. The Bernstein distribution in Achlioptas et al. (2013b) requires additional information about row norms $\|\mathbf{A}_{(i)}\|_1$, as well as, optimal convex combinations ρ_i, for each row. In contrast, our distribution assumes only one real number α^*. This makes our optimal hybrid distribution more convenient for the above problems in practice.

• Finally, we propose Algorithm 3 to provably implement hybrid sampling using *only one pass* over the data. This is particularly useful in streaming setting where we receive data as a stream of numbers and we have only limited memory to store them. Note that Bernstein sampling of Achlioptas et al. (2013b) can be implemented in streaming setting as well; however, they need information (or estimates) of those global quantities beforehand. In contrast, we need no prior knowledge of parameter α which governs the sampling distribution. Surprisingly, we can set α^* (or its estimate) at a later stage of the sampling process when the stream terminates.

Extensive experimental results on image, text, biological, and financial data show that sketches using our optimal hybrid-(ℓ_1, ℓ_2) sampling achieve better quality of approximation to PCA and sparse PCA problems than either ℓ_1 or ℓ_2 sampling on their own. Moreover, we can speed up such

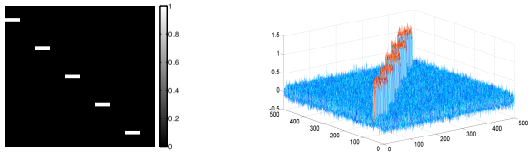


Figure 1: (left) Synthetic 500×500 binary data $\mathbf{D}$; (right) mesh view of noisy data $\mathbf{A}_{0.1}$.

computations significantly using our sparse sketch. Finally, our results are comparable (if not better) than those of Achlioptas et al. (2013b) despite using significantly less information about the data.

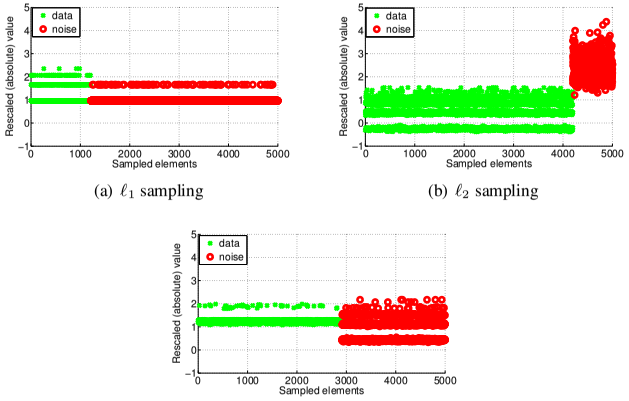
1.4 A Motivating Example for Hybrid Sampling

The main motivation for introducing the idea of hybrid-(ℓ_1, ℓ_2) sampling comes from achieving a tighter bound on s using a simple and intuitive probability distribution on elements of $\mathbf{A}$. For this, we observe certain good properties of both ℓ_1 and ℓ_2 sampling for sparsification of noisy data (in practice, we experience data that are noisy, and it is perhaps impossible to separate "true" data from noise). We illustrate the behavior of ℓ_1 and ℓ_2 sampling on noisy data using the following synthetic example. We construct a 500×500 binary data $\mathbf{D}$ (FIGURE 1), and then perturb it by a random Gaussian matrix $\mathbf{N}$ whose elements $\mathbf{N}_{ij}$ follow Gaussian distribution with mean zero and standard deviation 0.1. We denote this perturbed data matrix by $\mathbf{A}_{0.1}$. First, we note that ℓ_1 and ℓ_2 sampling work *identically* on binary data $\mathbf{D}$. However, FIGURE 2 depicts the change in behavior of ℓ_1 and ℓ_2 sampling to sparsify $\mathbf{A}_{0.1}$. Data elements and noise in $\mathbf{A}_{0.1}$ are the elements with non-zero and zero values in $\mathbf{D}$, respectively. We sample $s = 5000$ indices in i.i.d. trials according to ℓ_1 and ℓ_2 probabilities separately to produce sparse sketch $\tilde{\mathbf{A}}$. FIGURE 2 shows that elements of $\tilde{\mathbf{A}}$, produced by ℓ_1 sampling, have controlled variance but most of them are noise. On the other hand, ℓ_2 sampling is biased towards data elements, although small number of sampled noisy elements create large variance due to rescaling. Such large magnitude noisy elements become outliers in $\tilde{\mathbf{A}}$; consequently, PCA on $\tilde{\mathbf{A}}$ becomes a poor approximation to the true PCA of $\mathbf{A}$. Our hybrid-(ℓ_1, ℓ_2) sampling benefits from this bias of ℓ_2 towards data elements to sample large number of true data. Additionally, the regularization property of ℓ_1 component prevents noisy elements to become outliers in $\tilde{\mathbf{A}}$ helping us to achieve near-PCA reconstruction.

We parameterize our distribution using the variable $\alpha \in (0, 1]$ that controls the balance between ℓ_2 sampling and ℓ_1 regularization. One can view α as a regularization parameter. We derive an expression to compute α^*, the optimal α, corresponding to the smallest sample size that we need to achieve an accuracy ϵ in (1). When $\alpha = 1$ we reproduce the result of Achlioptas et al. (2013a). However, α^* may be smaller than 1, and the bound on sample size, using α^*, is guaranteed to be tighter than that of Achlioptas et al. (2013a) and ℓ_2.

2. Main Results

We present the quality-of-approximation result of our main sampling algorithm (Algorithm 1). We sample some of the non-zero elements of the fixed matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ (and rescale them) to form a sparse matrix $\tilde{\mathbf{A}}$ that is close to $\mathbf{A}$ in spectral norm. It would be useful to interpret the sampling



(a) ℓ_1 sampling (b) ℓ_2 sampling

(c) Hybrid-(ℓ_1, ℓ_2) sampling

Figure 2: Elements of sparse sketch $\tilde{\mathbf{A}}$ produced from $\mathbf{A}_{0.1}$ by (a) ℓ_1, (b) ℓ_2, and (c) hybrid-(ℓ_1, ℓ_2), $\alpha = 0.7$ sampling. In each plot, x-axis is the number of sampled indices $s = 5000$, including both data and noise. y-axis plots the rescaled absolute values (in $\ln$ scale) of $\tilde{\mathbf{A}}$ corresponding to the sampled indices. ℓ_1 sampling produces elements with controlled variance but it mostly samples noise, whereas ℓ_2 samples a lot of data although producing large variance of rescaled elements. Hybrid-(ℓ_1, ℓ_2) sampling uses ℓ_1 as a regularizer while sampling fairly large number of data that helps to preserve the spectral structure of original data. Data and noisy elements are shown as clusters for better visualization.

of elements from $\mathbf{A}$ as follows. We can express $\mathbf{A}$ as a sum of matrices each having at most one non-zero element.

$$\mathbf{A} = \sum_{i,j=1}^{m,n} \mathbf{A}_{ij} \mathbf{e}_i \mathbf{e}_j^T, \quad \forall (i,j) \in [m] \times [n] \tag{4}$$

We sample (in i.i.d. trials) some of the terms in (4) according to some probability distribution $\{p_{ij}\}_{i,j=1}^{m,n}$ defined over the elements of $\mathbf{A}$ to form a (weighted) partial sum that represents our sparse sketch $\tilde{\mathbf{A}}$ of $\mathbf{A}$. More formally, we define the following sampling operator $\mathcal{S}_\Omega : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ in (5) that extracts some of the terms from (4). Let $\Omega \subset [m] \times [n]$ be a multi-set of sampled indices (i_t, j_t), for $t = 1, ..., s$. Then,

$$\tilde{\mathbf{A}} = \mathcal{S}_\Omega(\mathbf{A}) = \frac{1}{s} \sum_{t=1}^{s} \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t} \mathbf{e}_{j_t}^T, \quad (i_t, j_t) \in \Omega \tag{5}$$

Clearly, $\tilde{\mathbf{A}}$ in (5) is a sparse (at most s non-zero elements), unbiased estimator of $\mathbf{A}$. We use the sampling operator in (5) in Algorithm 1 to randomly sample (in i.i.d. trials) s elements of $\mathbf{A}$,

according to p_{ij}'s as in equation (3). In order to bound $\|\mathbf{A}-\tilde{\mathbf{A}}\|_2$ we first express $\mathbf{A}-\tilde{\mathbf{A}}$ as a sum of zero-mean, independent, random matrices, and then use the following matrix-Bernstein inequality in Lemma 1 (due to Recht 2011) that bounds the deviation of spectral norms of such matrices.

Lemma 1 *[Theorem 3.2 of Recht 2011] Let $\mathbf{M}_1, \mathbf{M}_2, ..., \mathbf{M}_s$ be independent, zero-mean random matrices in $\mathbb{R}^{m\times n}$. Suppose $\max_{t\in[s]}\left\{\|\mathbb{E}(\mathbf{M}_t\mathbf{M}_t^T)\|_2, \|\mathbb{E}(\mathbf{M}_t^T\mathbf{M}_t)\|_2\right\} \leq \rho^2$, and $\|\mathbf{M}_t\|_2 \leq \gamma$ for all $t \in [s]$. Then, for any $\epsilon > 0$, $\left\|\frac{1}{s}\sum_{t=1}^{s}\mathbf{M}_t\right\|_2 \leq \epsilon$ holds, subject to a failure probability at most $(m+n)\exp\left(\frac{-s\epsilon^2/2}{\rho^2+\gamma\epsilon/3}\right)$.*

A popular construction of $\mathbf{M}_t \in \mathbb{R}^{m\times n}$, $t \in [s]$, in Lemma 1 is $\mathbf{M}_t = \frac{\mathbf{A}_{i_tj_t}}{p_{i_tj_t}}\mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}$ for $p_{i_tj_t} \neq 0$. Clearly, $\frac{1}{s}\sum_{t=1}^{s}\mathbf{M}_t = \tilde{\mathbf{A}} - \mathbf{A}$. We can see that ρ^2 and γ of Lemma 1 are now dependent on p_{ij} (Lemmas 8 and 9), i.e., different choices of p_{ij} lead to different bounds on ρ^2 and γ. Now, upper bounding the failure probability in Lemma 1 by $\delta > 0$, we can express the sample size s as a function of the key quantities ρ^2 and γ, and consequently as a function of p_{ij}:

$$s \geq \frac{2}{\epsilon^2}\cdot(\rho^2 + \gamma\epsilon/3)\cdot\ln((m+n)/\delta).$$

Since our choice of p_{ij} in (3) is a function of distribution parameter α, we essentially parameterize ρ^2 and γ by α, and express s as a function of α. Let us define $f(\alpha)$ as follows.

$$f(\alpha) = \rho^2(\alpha) + \gamma(\alpha)\epsilon\|\mathbf{A}\|_2/3,$$

where

$$\gamma(\alpha) = \max_{\substack{i,j,\\ \mathbf{A}_{ij}\neq 0}}\left\{\frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\frac{\|\mathbf{A}\|_1\cdot|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2}}\right\} + \|\mathbf{A}\|_2, \tag{6}$$

$$\rho^2(\alpha) = \max\left\{\max_i\sum_{j=1}^{n}\xi_{ij}(\alpha), \max_j\sum_{i=1}^{m}\xi_{ij}(\alpha)\right\} - \sigma_{min}^2, \tag{7}$$

$$\xi_{ij}(\alpha) = \|\mathbf{A}\|_F^2/\Big(\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha)\Big), \mathbf{A}_{ij} \neq 0, \tag{8}$$

σ_{min} is the smallest singular value of $\mathbf{A}$. Such parameterization gives us a flexibility to express the sample size as a function of α: $s(\alpha) \propto f(\alpha)$. Naturally, we want to find $\alpha \in (0, 1]$ that results in the smallest $s(\alpha)$. The optimal α may be less than 1, and setting $\alpha = 1$ (in which case equation 3 coincides with ℓ_1 probabilities of Achlioptas et al. 2013a) may not produce the smallest s.

We can see that $\gamma(\alpha)$ prevents $f(\alpha)$ from blowing up (in case of sampling a tiny element of $\mathbf{A}$) by setting α away from 0 (this is a regularization step). Thus, $\rho^2(\alpha)$ becomes the dominating term in $f(\alpha)$ as $\gamma(\alpha)$ is multiplied by a small constant ϵ. We take a closer look at $\xi_{ij}(\alpha)$ as it is the key quantity in $\rho^2(\alpha)$ which is approximately the larger of the largest row sum or the largest column sum of $\xi_{ij}(\alpha)$. While minimizing $f(\alpha)$, α maintains a trade-off between the two terms $\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1}$ and $(1-\alpha)$ such that the maximum row sum or column sum of $\xi_{ij}(\alpha)$ gets smaller. Along row i or column j, for smaller $|\mathbf{A}_{ij}|$, we typically have $|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1 < \|\mathbf{A}\|_F^2$, and consequently $\alpha \to 1$ reduces $\rho^2(\alpha)$. On the other hand, for larger $|\mathbf{A}_{ij}|$ we have $|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1 > \|\mathbf{A}\|_F^2$ and α away from 1 is preferred to reduce $\rho^2(\alpha)$. Thus α plays an intricate role to reduce the overall sample size

Algorithm 1 Element-wise Matrix Sparsification

1: **Input:** $\mathbf{A} \in \mathbb{R}^{m\times n}$, accuracy $\epsilon > 0$.
2: **Set** s as in eq. (11).
3: **For** $t = 1 \ldots s$ (i.i.d. trials with replacement) **randomly sample** pairs of indices $(i_t, j_t) \in [m] \times [n]$ with $\mathbb{P}\left[(i_t, j_t) = (i, j)\right] = p_{ij}$, where p_{ij} are as in (3), using α as in (10).
4: **Output:** $\mathcal{S}_\Omega(\mathbf{A}) = \frac{1}{s}\sum_{t=1}^{s}\frac{\mathbf{A}_{i_tj_t}}{p_{i_tj_t}}\mathbf{e}_{i_t}\mathbf{e}_{j_t}^T$.

depending on the structure of the data. Note that for $\alpha = 1$ (as in Achlioptas et al. 2013a) we have $\xi_{ij}(\alpha) = |\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1$. Therefore, for larger $|\mathbf{A}_{ij}|$ (true data points), maximum row sum or column sum of $\xi_{ij}(\alpha)$ lose the benefit of the delicate role played by $\alpha < 1$ to reduce those quantities, and they become larger than that for $\alpha < 1$. Thus, parameterization and optimization with α gives us a strictly smaller sample size than that of ℓ_1 or ℓ_2 sampling for the problem in (1).

Our main algorithm Algorithm 1 leads us to the following theorem.

Theorem 2 *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and let $\epsilon > 0$ be an accuracy parameter. Let $\mathcal{S}_\Omega$ be the sampling operator defined in (5), and assume that the multi-set Ω is generated using sampling probabilities $\{p_{ij}\}_{i,j=1}^{m,n}$ as in (3). Then, with probability at least $1-\delta$, $\|\mathcal{S}_\Omega(\mathbf{A}) - \mathbf{A}\|_2 \leq \epsilon\|\mathbf{A}\|_2$, if*

$$s \geq \frac{2}{(\epsilon\|\mathbf{A}\|_2)^2}\cdot f(\alpha)\cdot\ln((m+n)/\delta). \tag{9}$$

Furthermore, we can find α^* (optimal α corresponding to the smallest s) by solving (10):

$$\alpha^* = \min_{\alpha\in(0,1]} f(\alpha), \tag{10}$$

and the corresponding optimal sample size is

$$s^* = \frac{2}{(\epsilon\|\mathbf{A}\|_2)^2}\cdot f(\alpha^*)\cdot\ln((m+n)/\delta). \tag{11}$$

For a given matrix $\mathbf{A}$, we can easily compute $\rho^2(\alpha)$ and $\gamma(\alpha)$ for various values of α. Given an accuracy ϵ and failure probability δ, we can compute α^* corresponding to the tightest bound on s. Note that, for $\alpha = 1$ we reproduce the results of Achlioptas et al. (2013a) (which was expressed using various matrix metrics). However, α^* may be smaller than 1, and is guaranteed to produce tighter s comparing to extreme choices of α (e.g. $\alpha = 1$ for ℓ_1). We illustrate this in FIGURE 3. We give a proof of Theorem 2 in APPENDIX A.

2.1 Approximation of PCA via Element Sampling

The top-k principal components of centered data $\mathbf{A} \in \mathbb{R}^{m\times n}$ (m data points in n dimensions), denoted by $\mathbf{V}_k \in \mathbb{R}^{n\times k}$, can be formulated as the solution to the variance maximization problem:

$$\mathbf{V}_k = \operatorname*{argmax}_{\mathbf{V}\in\mathbb{R}^{n\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}} \operatorname{trace}(\mathbf{V}^T\mathbf{A}^T\mathbf{A}\mathbf{V}). \tag{12}$$

The maximum variance achievable using $\mathbf{V}_k$, denoted by OPT_k, is the sum of squares of the top-k singular values of $\mathbf{A}$. Note that in (12) the optimal solution is computed using the full data $\mathbf{A}$ (which is typically dense).

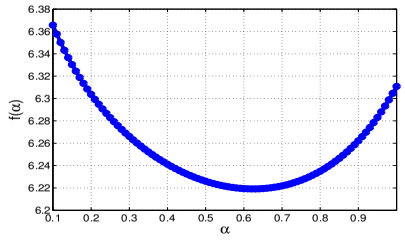


Figure 3: Plot of $f(\alpha)$ in eqn. (10) for data $\mathbf{A}_{0.1}$. We use $\epsilon = 0.05$ and $\delta = 0.1$. x-axis plots α and y-axis is in log scale.

Algorithm 2 Approximation of PCA from Data Samples

1: **Input:** Centered data $\mathbf{A} \in \mathbb{R}^{m\times n}$, sparsity parameter $s > 0$, and rank parameter k.
2: Produce sparse unbiased sketch $\tilde{\mathbf{A}}$ from $\mathbf{A}$, in s i.i.d. trials using Algorithm 1[1].
3: Perform rank truncated SVD on matrix $\tilde{\mathbf{A}}$, i.e., $[\tilde{\mathbf{U}}_k, \tilde{\mathbf{D}}_k, \tilde{\mathbf{V}}_k]$ = SVD($\tilde{\mathbf{A}}$, k).
4: **Output:** $\tilde{\mathbf{V}}_k$ (columns of $\tilde{\mathbf{V}}_k$ are the ordered principal components).

Here we discuss a provable algorithm (Algorithm 2) to approximate PCA by applying element-wise sampling. We produce a sparse unbiased estimator $\tilde{\mathbf{A}}$ of centered data $\mathbf{A}$ by sampling s elements in i.i.d. trials according to our hybrid-(ℓ_1, ℓ_2) distribution in (3). We use this $\tilde{\mathbf{A}}$ instead of $\mathbf{A}$ in (12) to compute the new optimal solution $\tilde{\mathbf{V}}_k$

$$\tilde{\mathbf{V}}_k = \underset{\mathbf{V}\in\mathbb{R}^{n\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}}{\text{argmax}} \text{trace}(\mathbf{V}^T\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\mathbf{V}). \tag{13}$$

The computation of rank-truncated SVD on sparse data requires fewer floating point operations (therefore can be fast), and we consider the right singular vectors $\tilde{\mathbf{V}}_k$ of $\tilde{\mathbf{A}}$ as the approximate principal components of $\mathbf{A}$. Naturally, more samples produce better approximation. However, this reduces sparsity, consequently we may lose the speed advantage. Theorem 3 shows the quality of approximation of principal components produced by Algorithm 2.

Theorem 3 *Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a given matrix, and $\tilde{\mathbf{A}}$ be a sparse sketch produced by Algorithm 1. Let $\tilde{\mathbf{V}}_k$ be the PCA's of $\tilde{\mathbf{A}}$ computed in step 3 of Algorithm 2. Then*

1) $\|\mathbf{A} - \mathbf{A}\tilde{\mathbf{V}}_k\tilde{\mathbf{V}}_k{}^T\|_F^2 \le \|\mathbf{A} - \mathbf{A}_k\|_F^2 + 4\|\mathbf{A}_k\|_F^2 \cdot \|\mathbf{A} - \tilde{\mathbf{A}}\|_2/\sigma_k(\mathbf{A})$,
2) $\|\mathbf{A}_k - \tilde{\mathbf{A}}_k\|_F \le \sqrt{8k}\cdot(\|\mathbf{A} - \mathbf{A}_k\|_2 + \|\mathbf{A} - \tilde{\mathbf{A}}\|_2)$,
3) $\|\mathbf{A} - \tilde{\mathbf{A}}_k\|_F \le \|\mathbf{A} - \mathbf{A}_k\|_F + \sqrt{8k}\cdot(\|\mathbf{A} - \mathbf{A}_k\|_2 + \|\mathbf{A} - \tilde{\mathbf{A}}\|_2)$.

1. Here we avoid computing $\|\mathbf{A}\|_2$ directly and instead upper bound it by $\|\mathbf{A}\|_F$ in $\gamma(\alpha)$ in Lemma 8. In (10) $\rho^2(\alpha)$ is the dominating term (we ignore $\sigma_{\min}$ as it is typically very small) and $\gamma(\alpha)$ is multiplied by a small constant ϵ; thus in reality such relaxation of $\gamma(\alpha)$ and $\rho^2(\alpha)$ has little effect on α^* (we verified it experimentally). However, we refer to the α that we get using the above as near-optimal.

The first inequality of Theorem 3 bounds the approximation of projected data onto the space spanned by top k approximate principal components. The second and third inequalities measure the quality of $\tilde{\mathbf{A}}_k$ as a surrogate for $\mathbf{A}_k$ and the quality of projection of sparsified data onto approximate principal components, respectively. The proofs of first two inequalities of Theorem 3 follow from Theorem 5 and Theorem 8 of Achlioptas and McSherry (2001), respectively. The last inequality follows from the triangle inequality. The last two inequalities above are particularly useful in cases where $\mathbf{A}$ is inherently low-rank and we choose an appropriate k for which $\|\mathbf{A} - \mathbf{A}_k\|_2$ is small.

2.2 Approximating Sparse PCA from Incomplete Data

PCA constructs a low dimensional subspace of the data such that projection of the data onto this subspace preserves as much information as possible. However a shortcoming of PCA is the interpretation of the principal components (or factors) as they can be linear combinations of *all* the original variables. In many cases the original variables have direct physical significance (e.g. genes in biological applications or assets in financial applications). In such cases it is desirable to have factors which have loadings on only a small number of the original variables. These interpretable factors are sparse principal components (SPCA). To derive sparse principal components, we add a sparsity constraint to the optimization problem (see equation 14): every column of $\mathbf{V}$ should have at most r non-zero entries (r is an input parameter),

$$\mathbf{S}_k = \underset{\mathbf{V}\in\mathbb{R}^{n\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}, \|\mathbf{V}^{(i)}\|_0\le r}{\text{argmax}} \text{trace}(\mathbf{V}^T\mathbf{A}^T\mathbf{A}\mathbf{V}). \tag{14}$$

The sparse PCA problem is not only NP-hard, but also inapproximable (Magdon-Ismail 2017). There are many heuristics for obtaining sparse factors (Cadima and Jolliffe, 1995; Trendafilov et al., 2003; Zou et al., 2006; d'Aspremont et al., 2007, 2008; Moghaddam et al., 2006; Shen and Huang, 2008) including some approximation algorithms with provable guarantees Asteris et al. (2014). The existing research typically addresses the task of getting just the top principal component ($k = 1$) (some exceptions are Ma 2013; Cai et al. 2013; Wang et al. 2014; Lei and Vu 2015). While the sparse PCA problem is hard and interesting, it is *not* the focus of this work.

We address the question: What if we do not know $\mathbf{A}$, but only have a sparse sampling of some of the entries in $\mathbf{A}$ (incomplete data)? The sparse sampling is used to construct a *sketch* of $\mathbf{A}$, denoted $\tilde{\mathbf{A}}$. There is not much else to do but solve the sparse PCA problem with the sketch $\tilde{\mathbf{A}}$ instead of the full data $\mathbf{A}$ to get $\tilde{\mathbf{S}}_k$,

$$\tilde{\mathbf{S}}_k = \underset{\mathbf{V}\in\mathbb{R}^{n\times k}, \mathbf{V}^T\mathbf{V}=\mathbf{I}, \|\mathbf{V}^{(i)}\|_0\le r}{\text{argmax}} \text{trace}(\mathbf{V}^T\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\mathbf{V}). \tag{15}$$

We study how $\tilde{\mathbf{S}}_k$ performs as an approximation to $\mathbf{S}_k$ with respective to the objective that we are trying to optimize, namely trace($\mathbf{S}^T\mathbf{A}^T\mathbf{A}\mathbf{S}$) — the quality of approximation is measured with respect to the true $\mathbf{A}$. We show that the quality of approximation is controlled by how well $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$ approximates $\mathbf{A}^T\mathbf{A}$ as measured by the spectral norm of the deviation $\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$. This is a general result that does not rely on how one constructs the sketch $\tilde{\mathbf{A}}$.

Lemma 4 (Sparse PCA from a Sketch) *Let $\mathbf{S}_k$ be a solution to the sparse PCA problem that solves (14), and $\tilde{\mathbf{S}}_k$ a solution to the sparse PCA problem for the sketch $\tilde{\mathbf{A}}$ which solves (15). Then,*

$$\textit{trace}(\tilde{\mathbf{S}}_k{}^T\mathbf{A}^T\mathbf{A}\tilde{\mathbf{S}}_k) \ge \textit{trace}(\mathbf{S}_k{}^T\mathbf{A}^T\mathbf{A}\mathbf{S}_k) - 2k\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2.$$

Lemma 4 says that if we can closely approximate $\mathbf{A}$ with $\tilde{\mathbf{A}}$, then we can compute, from $\tilde{\mathbf{A}}$, sparse components which capture almost as much variance as the optimal sparse components computed from the full data $\mathbf{A}$. We prove that Lemma 4 follows as a corollary of a more general result given in Theorem 10.

In our setting, the sketch $\tilde{\mathbf{A}}$ is computed from a sparse sampling of the data elements in $\mathbf{A}$ (incomplete data). We use probabilities of the form in (3) to determine which elements to sample and how to form the sketch such that the error $\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2$ is small. We can simplify this quantity in terms of $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ as follows. Let $\Delta = \mathbf{A} - \tilde{\mathbf{A}}$.

$$\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2 = \|\mathbf{A}^T\Delta + \Delta^T\mathbf{A} - \Delta^T\Delta\|_2 \leq 2\|\mathbf{A}\|_2\|\Delta\|_2 + \|\Delta\|_2^2. \tag{16}$$

We combine the bound on $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2$ for given accuracy ϵ/k from Theorem 2 with Lemma 4 to derive $k\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2 \leq \epsilon(2 + \epsilon/k)\|\mathbf{A}\|_2^2$ with a sample size $s = k^2 \cdot s^*$ where s^* as in (11). However, we simplify this bound for a better interpretation of s in terms matrix dimensions m and n and stable rank $\tilde{k} = \|\mathbf{A}\|_F^2/\|\mathbf{A}\|_2^2$. Note that $\|\mathbf{A}\|_1 \leq \sqrt{mn}\|\mathbf{A}\|_F$ from Cauchy-Schwartz. Then,

$$\gamma(\alpha)/\|\mathbf{A}\|_2 \leq 1 + \|\mathbf{A}\|_1/(\alpha\|\mathbf{A}\|_2) \leq 1 + \sqrt{mn\tilde{k}}/\alpha = \hat{\gamma}. \tag{17}$$

Also, from $|\mathbf{A}_{ij}| \leq \|\mathbf{A}\|_2$, $\rho^2(\alpha)$ in (7) can be simplified as

$$\xi_{ij}(\alpha) \leq \|\mathbf{A}\|_F^2\Big(\alpha\tilde{k}\|\mathbf{A}\|_2/\|\mathbf{A}\|_1 + (1-\alpha)\Big)^{-1}, \tag{18}$$

$$\rho^2(\alpha)/\|\mathbf{A}\|_2^2 \leq \max\{m,n\}\tilde{k}\Big(\alpha\tilde{k}\|\mathbf{A}\|_2/\|\mathbf{A}\|_1 + (1-\alpha)\Big)^{-1} = \hat{\rho}^2. \tag{19}$$

Using the above relaxations we have the following theorem on sample size complexity.

Theorem 5 (Sampling Complexity for Sparse PCA) *Sample s data-elements from $\mathbf{A} \in \mathbb{R}^{m\times n}$ to form the sparse sketch $\tilde{\mathbf{A}}$ using Algorithm 1. Let $\mathbf{S}_k$ be a solution to the sparse PCA problem that solves (14), and let $\tilde{\mathbf{S}}_k$, which solves (15), be a solution to the sparse PCA problem for the sketch $\tilde{\mathbf{A}}$ formed from the s sampled data elements. Suppose the number of samples s satisfies*

$$s \geq 2k^2\epsilon^{-2}(\hat{\rho}^2 + \epsilon\hat{\gamma}/(3k))\log((m+n)/\delta),$$

($\hat{\rho}^2$ and $\hat{\gamma}$ are dimensionless quantities that depend only on $\mathbf{A}$). Then, with probability at least $1-\delta$

$$\mathit{trace}(\tilde{\mathbf{S}}_k{}^T\mathbf{A}^T\mathbf{A}\tilde{\mathbf{S}}_k) \geq \mathit{trace}(\mathbf{S}_k{}^T\mathbf{A}^T\mathbf{A}\mathbf{S}_k) - 2\epsilon(2+\epsilon/k)\|\mathbf{A}\|_2^2.$$

The dependence of $\hat{\rho}^2$ and $\hat{\gamma}$ on $\mathbf{A}$ are given in (17) and (19), respectively. Roughly speaking, we can ignore the term with $\hat{\gamma}$ since it is multiplied by ϵ/k, and we have $\hat{\rho}^2 = O(\tilde{k}\max\{m,n\})$. To paraphrase Theorem 5, when the stable rank $\tilde{k}$ is a small constant, with $O(k^2\max\{m,n\})$ samples, one can recover almost as good sparse principal components as with all data (a possible price being a small fraction of the optimal variance, since $OPT_k \geq \|\mathbf{A}\|_2^2$). As far as we know, the only prior work related to the problem we consider here is Lounici (2013) which proposed a specific method to construct sparse PCA from incomplete data. However, we develop a general tool that can be used with any existing sparse PCA heuristic. Moreover, we derive much simpler bounds (Theorems 4 and 5) using matrix concentration inequalities, as opposed to ϵ-net arguments in Lounici (2013).

2.2.1 SPARSE SKETCH USING GREEDY THRESHOLD

We also give an application of Lemma 4 to run sparse PCA after "denoising" the data using a greedy thresholding algorithm that sets the small elements to zero (see Theorem 6). Such denoising is appropriate when the observed matrix has been element-wise perturbed by small noise, and the uncontaminated data matrix is sparse and contains large elements. We show that if an appropriate fraction of the (noisy) data is set to zero, one can still recover sparse principal components. This gives a principled approach to regularizing sparse PCA in the presence of small noise when the data is sparse.

We give the simplest scenario of incomplete data where Lemma 4 gives some reassurance that one can compute good sparse principal components. Suppose the smallest data elements have been set to zero. This can happen, for example, if only the largest elements are measured, or in a noisy setting if the small elements are treated as noise and set to zero. So

$$\tilde{\mathbf{A}}_{ij} = \begin{cases} \mathbf{A}_{ij} & |\mathbf{A}_{ij}| \geq \delta, \\ 0 & |\mathbf{A}_{ij}| < \delta. \end{cases}$$

Recall $\tilde{k} = \|\mathbf{A}\|_F^2/\|\mathbf{A}\|_2^2$ (stable rank of $\mathbf{A}$), and define $\|\mathbf{A}_\delta\|_F^2 = \sum_{|\mathbf{A}_{ij}|<\delta}\mathbf{A}_{ij}^2$. Let $\Delta = \mathbf{A} - \tilde{\mathbf{A}}$. By construction, $\|\Delta\|_F^2 = \|\mathbf{A}_\delta\|_F^2$ and $\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2 \leq 2\|\mathbf{A}\|_2\|\Delta\|_2 + \|\Delta\|_2^2$ from (16). Suppose the zeroing of elements only loses a fraction of the energy in $\mathbf{A}$, i.e. δ is selected so that $\|\mathbf{A}_\delta\|_F^2 \leq \epsilon^2\|\mathbf{A}\|_F^2/\tilde{k}$; that is an $\epsilon/\tilde{k}$ fraction of the total variance in $\mathbf{A}$ has been lost in the unmeasured (or zero) data. Then $\|\Delta\|_2 \leq \|\Delta\|_F \leq \epsilon\|\mathbf{A}\|_F/\sqrt{\tilde{k}} = \epsilon\|\mathbf{A}\|_2$.

Theorem 6 *Suppose that $\tilde{\mathbf{A}}$ is created from $\mathbf{A}$ by zeroing all elements that are less than δ, and δ is such that the truncated norm satisfies $\|\mathbf{A}_\delta\|_2^2 \leq \epsilon^2\|\mathbf{A}\|_F^2/\tilde{k}$. Then the sparse PCA solution $\tilde{\mathbf{V}}^*$ satisfies*

$$\mathit{trace}(\tilde{\mathbf{V}}^{*T}\mathbf{A}\mathbf{A}\tilde{\mathbf{V}}^*) \geq \mathit{trace}(\mathbf{V}^{*T}\mathbf{A}\mathbf{A}^T\mathbf{V}^*) - 2k\epsilon\|\mathbf{A}\|_2^2(2+\epsilon).$$

Theorem 6 shows that it is possible to recover sparse PCA after setting small elements to zero. This is appropriate when most of the elements in $\mathbf{A}$ are small noise and a few of the elements in $\mathbf{A}$ contain large data elements. For example if the data consists of sparse $O(\sqrt{nm})$ large elements (of magnitude, say 1) and many $nm - O(\sqrt{nm})$ small elements whose magnitude is $o(1/\sqrt{nm})$ (high signal-to-noise setting), then $\|\mathbf{A}_\delta\|_2^2/\|\mathbf{A}\|_2^2 \to 0$ and with just a sparse sampling of the $O(\sqrt{nm})$ large elements (very incomplete data), we recover near optimal sparse PCA.

Not only do our algorithms preserve the quality of the sparse principal components, but iterative algorithms for sparse PCA, whose running time is proportional to the number of non-zero entries in the input matrix, benefit from the sparsity of $\tilde{\mathbf{A}}$. Our experiments show about five-fold speed gains while producing near-comparable sparse components using less than 10% of the data.

2.3 One-pass Hybrid-(ℓ_1, ℓ_2) Sampling

Here we discuss the implementation of hybrid-(ℓ_1, ℓ_2) sampling in one pass over the input matrix $\mathbf{A}$ using $O(s)$ memory, e.g., a streaming model. Here we propose an algorithm, Algorithm 3, to implement a one-pass version of the hybrid sampling *without a priori knowledge of the regularization parameter* α.

We use SELECT-s algorithm (Algorithm 5 in APPENDIX F) to implement one-pass ℓ_1 and ℓ_2 sampling. We note that steps 2-3 of Algorithm 3 access the elements of $\mathbf{A}$ only once, in parallel,

Algorithm 3 One-pass hybrid sampling

1: **Input:** $\mathbf{A}_{ij}$ for all $(i,j) \in [m] \times [n]$, arbitrarily ordered, and sample size s.
2: Apply SELECT-s algorithm (Algorithm 5) using ℓ_1 probabilities to sample s independent indices (i_{t_1}, j_{t_1}) and corresponding values $\mathbf{A}_{i_{t_1} j_{t_1}}$ to form random multiset S_1 of triples $(i_{t_1}, j_{t_1}, \mathbf{A}_{i_{t_1} j_{t_1}})$, for $t_1 = 1, ..., s$.
3: Apply SELECT-s algorithm using ℓ_2 probabilities to sample s independent indices (i_{t_2}, j_{t_2}) and corresponding values $\mathbf{A}_{i_{t_2} j_{t_2}}$ to form random multiset S_2 of triples $(i_{t_2}, j_{t_2}, \mathbf{A}_{i_{t_2} j_{t_2}})$, for $t_2 = 1, ..., s$.
4: Store $\|\mathbf{A}\|_F^2$ and $\|\mathbf{A}\|_1$ /* these are already computed in SELECT-s algorithms above */
5: /* stream terminates */
6: Set the value of $\alpha \in (0, 1]$ (say, using Algorithm 4).
7: Create empty multiset of triples S.
8: $\mathbf{X} \leftarrow \mathbf{0}_{m \times n}$.
9: **For** $t = 1 \dots s$
10: Generate a uniform random number $x \in [0, 1]$.
11: if $x \leq \alpha$, $S(t) \leftarrow S_1(t)$; otherwise, $S(t) \leftarrow S_2(t)$.
12: $(i_t, j_t) \leftarrow S(t, 1:2)$.
13: $p \leftarrow \alpha \cdot \frac{|S(t,3)|}{\|\mathbf{A}\|_1} + (1-\alpha) \cdot \frac{|S(t,3)|^2}{\|\mathbf{A}\|_F^2}$
14: $\mathbf{X} \leftarrow \mathbf{X} + \frac{S(t,3)}{p \cdot s} e_{i_t} e_{j_t}^T$.
15: **End**
16: **Output:** random multiset S, and sparse matrix $\mathbf{X}$.

to form independent multisets S_1 and S_2. We store $\|\mathbf{A}\|_F^2$ and $\|\mathbf{A}\|_1$ already computed in steps 2-3. Subsequent steps do not need further access to $\mathbf{A}$. Interestingly, we set α in step 6 when the data stream terminates. Steps 9-15 sample s elements from S_1 and S_2 using the α in step 6, and produces a sparse matrix $\mathbf{X}$ using the sampled entries in multiset S. Theorem 7 proves that Algorithm 3 indeed samples elements of $\mathbf{A}$ according to the hybrid-(ℓ_1, ℓ_2) probabilities in (3).

Theorem 7 *Using the same notations as in Algorithm 3, for $\alpha \in (0, 1]$, $t = 1, ..., s$,*

$$P\left[S(t) = (i, j, \mathbf{A}_{ij})\right] = \alpha \cdot \frac{|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_1} + (1-\alpha) \cdot \frac{\mathbf{A}_{ij}^2}{\|\mathbf{A}\|_F^2}.$$

See proof in APPENDIX E. Note that, Theorem 7 holds for any arbitrary $\alpha \in (0, 1]$ in line 7 of Algorithm 3, i.e., Algorithm 4 is not essential for correctness of Theorem 7. We only need α to be independent of the elements of S_1 and S_2. However, we can use Algorithm 4 to get a good estimate of α^* (SECTION 2.3.1). In this case, we need additional independent samples from $\mathbf{A}$ to 'learn' the parameter α^*.

2.3.1 ESTIMATE OF α^* FROM SAMPLES

Here we discuss Algorithm 4, to estimate α^* from samples of $\mathbf{A}$ and few other quantities, such as, $\|\mathbf{A}\|_F$, $\|\mathbf{A}\|_1$, and $\mathbf{A}_{min}$, where $\mathbf{A}_{min}$ is defined as

$$\mathbf{A}_{min} = \min_{\substack{i,j: \\ \mathbf{A}_{ij} \neq 0}} |\mathbf{A}_{ij}|.$$

Algorithm 4 Estimating α^* from Samples

1: **Input:** Ω, the set of triples $\{(i, j, \mathbf{A}_{ij})\}$ where each $\mathbf{A}_{ij}$ is sampled with probability p_{ij} (say $p_{ij} = |\mathbf{A}_{ij}| / \|\mathbf{A}\|_1$), accuracy ε, $\mathbf{A}_{min}$, $\|\mathbf{A}\|_F^2$, $\|\mathbf{A}\|_1$, and matrix dimensions m and n.
2: Compute $\tilde{\gamma}(\alpha)$ as in equation (20) and $\tilde{\xi}_{ij}(\alpha)$ as in equation (21), for a given α.
3: **Output:** $\tilde{\alpha}$ in equation (23).

Let Ω be a set of sampled triples $\{(i, j, \mathbf{A}_{ij})\}$ where each $\mathbf{A}_{ij}$ is sampled with probability p_{ij} (e.g., $p_{ij} = |\mathbf{A}_{ij}| / \|\mathbf{A}\|_1$). Let us define, for a fixed α,

$$\tilde{\gamma}(\alpha) = \max_{\substack{i,j: \\ \mathbf{A}_{ij} \neq 0}} \left\{ \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha) \frac{\|\mathbf{A}\|_1 \cdot |\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2}} \right\} + \|\mathbf{A}\|_F = \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha) \frac{\|\mathbf{A}\|_1}{\|\mathbf{A}\|_F^2} \cdot \mathbf{A}_{min}} + \|\mathbf{A}\|_F. \quad (20)$$

Note that, $\tilde{\gamma}(\alpha)$ can be computed using only one pass over the data. Further, we define the following random variable (parameterized by α)

$$\tilde{\xi}_{ij}(\alpha) \;=\; \frac{\|\mathbf{A}\|_F^2}{\frac{\alpha \cdot \|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1} + (1-\alpha)} \cdot \delta_{ij}, \quad (21)$$

where δ_{ij} is an indicator function defined as $\delta_{ij} := \mathbb{I}((i, j, \mathbf{A}_{ij}) \in \Omega)$. Note that, $\delta_{ij} = 1$ w.p. p_{ij}, and 0 otherwise. For a given column j, we define the following random variable $\tilde{Y}_j(\alpha)$:

$$\tilde{Y}_j(\alpha) = \sum_{i=1}^{m} \tilde{\xi}_{ij}(\alpha).$$

Similarly, for a given row i, we define $\tilde{Y}_i(\alpha) = \sum_{j=1}^{n} \tilde{\xi}_{ij}(\alpha)$. Finally, we define

$$\tilde{\rho}^2(\alpha) = \max\{\max_j \tilde{Y}_j(\alpha), \max_i \tilde{Y}_i(\alpha)\}. \quad (22)$$

Using the above quantities we solve the optimization problem in (23) and use the solution as an estimate for α^*.

$$\tilde{\alpha} : \min_{\alpha \in (0,1]} \quad \tilde{\rho}^2(\alpha) + \tilde{\gamma}(\alpha)\varepsilon/3. \quad (23)$$

Note that, $\tilde{\gamma}(\alpha)$ in equation (20) closely approximates $\gamma(\alpha)$ in equation (6), where the error of approximation is $\|\mathbf{A}\|_F - \|\mathbf{A}\|_2$. This error becomes small in equation (23) as $\tilde{\gamma}(\alpha)$ is multiplied by a small quantity $\varepsilon/3$. Next we analyze how well $\tilde{\xi}_{ij}(\alpha)$ approximates $\xi_{ij}(\alpha)$ using a simplified data model. For this we assume $|\mathbf{A}_{ij}| \in \{L, \epsilon, 0\}$, where $L \gg \epsilon > 0$, and let s_L and s_ϵ be the number of elements with magnitudes L and ϵ, respectively. Also, let the minimum singular value $\sigma_{\min} \approx 0$. For $(i, j, \mathbf{A}_{ij}) \in \Omega$ and $|\mathbf{A}_{ij}| = \epsilon$ (which is unlikely), $|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1 \ll \|\mathbf{A}\|_F^2$; consequently $\tilde{\xi}_{ij}(\alpha)$ is small and does not contribute much to $\tilde{Y}_j(\alpha)$ (or $\tilde{Y}_i(\alpha)$). On the other hand, for $(i, j, \mathbf{A}_{ij}) \in \Omega$ and $|\mathbf{A}_{ij}| = L$ (which is very likely), $|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1 \gg \|\mathbf{A}\|_F^2$, and consequently $\tilde{\xi}_{ij}(\alpha)$ contributes significantly to $\tilde{Y}_j(\alpha)$ (or $\tilde{Y}_i(\alpha)$). So, if we sample s_L number of such large elements $\tilde{\rho}^2(\alpha)$ is close to $\rho^2(\alpha)$, and $\tilde{\alpha} \approx \alpha^*$. Finally, note that Algorithm 4 can be implemented using only one-pass over the data. Experimental results on real data validate the quality of Algorithm 4.

3. Experiments

We perform various element-wise sampling on synthetic and real data to show how well the sparse sketches preserve the structure of the original data, in spectral norm. Also, we show results on the quality and computation time of the principal components and sparse principal components derived from sparse sketches.

3.1 Algorithms for Sparse Sketches

Algorithm 1 is our prototypical algorithm to produce (unbiased) sparse sketches $\tilde{\mathbf{A}}$ from a given matrix via various sampling methods. We construct our sparse sketch using our optimal hybrid-(ℓ_1, ℓ_2) probabilities, and compare its quality with other sketches produced via ℓ_1 and ℓ_2 sampling. We borrow same definitions from (Achlioptas et al. 2013a) for comparing our results with ℓ_1 sampling.

$$\mathrm{nd}(\mathbf{A}) := \frac{\|\mathbf{A}\|_1^2}{\|\mathbf{A}\|_F^2}, \quad \mathrm{rs}_0(\mathbf{A}) := \frac{\max_i \|\mathbf{A}_{(i)}\|_0}{\|\mathbf{A}\|_0/m}, \quad \mathrm{rs}_1(\mathbf{A}) := \frac{\max_i \|\mathbf{A}_{(i)}\|_1}{\|\mathbf{A}\|_1/m},$$

where nd is numeric density and rs is row density skewness. Achlioptas et al. (2013a) argues that ℓ_1 outperforms ℓ_2 sampling when $\mathrm{rs}_0 > \mathrm{rs}_1$. We compute the theoretical optimal mixing parameter α^* by solving (10) for various datasets [2], and compare this α^* with the theoretical condition derived by Achlioptas et al. (2013a). Further, we verify the accuracy of α^* by measuring $\mathcal{E} = \|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \|\mathbf{A}\|_2$ for distributions corresponding to various α, for a given sample size s [3]. We expect $\mathcal{E}$ to be the smallest for $\alpha \approx \alpha^*$ rather than $\alpha = 1$ (ℓ_1) or $\alpha \approx 0$ (ℓ_2).

3.2 Algorithms for PCA from Sparse Sketches

We first compare three algorithms for computing PCA of the centered data $\mathbf{A}$. Let the actual PCA of the original data be $\mathcal{A}$. We use Algorithm 2 to compute approximate PCA from random samples via our optimal hybrid-(ℓ_1, ℓ_2) sampling. Let us denote this approximate PCA by $\mathcal{H}$. Also, we compute PCA of a Gaussian random projection of the original data to compare its quality with $\mathcal{H}$. Let $\mathbf{A}_G = \mathbf{G}\mathbf{A} \in \mathbb{R}^{r \times n}$, where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the original data, and $\mathbf{G}$ is a $r \times m$ standard Gaussian matrix. Let the PCA of this random projection $\mathbf{A}_G$ be $\mathcal{G}$. The quality of various PCA is determined by measuring how much variance of the data they can capture (as in equation 12). For this, let σ_k, σ_h, and σ_g denote the variance preserved by $\mathcal{A}$, $\mathcal{H}$, and $\mathcal{G}$, respectively. Note that, $\mathcal{A}$ and $\mathcal{G}$ require access to the full data, while $\mathcal{H}$ is computed from only few samples of $\mathbf{A}$. Also, we compare the computation time (in milliseconds) t_a, t_h, and t_G for $\mathcal{A}$, $\mathcal{H}$, and $\mathcal{G}$, respectively.

Finally, we compare the quality of our optimal hybrid sampling with ℓ_1 sampling and Bernstein sampling (Achlioptas et al. 2013b) by quantifying the variance, σ_{ℓ_1} and σ_b, preserved by the corresponding sparse sketches.

3.3 Algorithms for Approximate Sparse PCA from Sketches

We show the experimental results for sparse PCA from a sketch using several real data matrices. As we mentioned, sparse PCA is NP-Hard, and so we use heuristics. These heuristics are discussed next, followed by the data, the experimental design and finally the results.

2. we find α^* from the plot of $f(\alpha)$ for $\alpha \in [0.1, 1]$.
3. here we use $\mathbf{A}$, s, and α as input to Algorithm 1 to form $\tilde{\mathbf{A}}$ by s i.i.d samples from p_{ij} in (3) for given α.

Let $\mathcal{G}$ (ground truth) denote the algorithm which computes the principal components (which may not be sparse) of the full data matrix $\mathbf{A}$; the optimal variance is OPT_k. We consider six heuristics for getting sparse principal components.

$\mathcal{G}_{\mathrm{max},r}$	The r largest-magnitude entries in each principal component generated by $\mathcal{G}$.
$\mathcal{G}_{\mathrm{sp},r}$	r-sparse components using the *Spasm* toolbox of Sjstrand et al. (2012) with $\mathbf{A}$.
$\mathcal{H}_{\mathrm{max},r}$	The r largest entries of the principal components for the (ℓ_1, ℓ_2)-sampled sketch $\tilde{\mathbf{A}}$.
$\mathcal{H}_{\mathrm{sp},r}$	r-sparse components using *Spasm* with the (ℓ_1, ℓ_2)-sampled sketch $\tilde{\mathbf{A}}$.
$\mathcal{U}_{\mathrm{max},r}$	The r largest entries of the principal components for the *uniformly* sampled sketch $\tilde{\mathbf{A}}$.
$\mathcal{U}_{\mathrm{sp},r}$	r-sparse components using *Spasm* with the uniformly sampled sketch $\tilde{\mathbf{A}}$.

Output of an algorithm $\mathcal{Z}$ is sparse principal components $\mathbf{V}$, and we consider the metric $f(\mathcal{Z}) = \mathrm{trace}(\mathbf{V}^T\mathbf{A}^T\mathbf{A}\mathbf{V})$, where $\mathbf{A}$ is the original centered data. We consider the following statistics.

$\frac{f(\mathcal{G}_{\mathrm{max},r})}{f(\mathcal{G}_{\mathrm{sp},r})}$	Relative loss of greedy thresholding versus *Spasm*, illustrating the value of a good sparse PCA algorithm. Our sketch based algorithms *do not* address this loss.
$\frac{f(\mathcal{H}_{\mathrm{max/sp},r})}{f(\mathcal{G}_{\mathrm{max/sp},r})}$	Relative loss of using the (ℓ_1, ℓ_2)-sketch $\tilde{\mathbf{A}}$ instead of complete data $\mathbf{A}$. A ratio close to 1 is desired.
$\frac{f(\mathcal{U}_{\mathrm{max/sp},r})}{f(\mathcal{G}_{\mathrm{max/sp},r})}$	Relative loss of using the uniform sketch $\tilde{\mathbf{A}}$ instead of complete data $\mathbf{A}$. A benchmark to highlight the value of a good sketch.

We also report the computation time for the algorithms. We show results to confirm that sparse PCA algorithms using the (ℓ_1, ℓ_2)-sketch are nearly comparable to those same algorithms on the complete data; and, gain in computation time from sparse sketch is proportional to the sparsity.

Also, we evaluate the quality of Algorithm 4 to estimate α^* from a small number of samples from various data.

3.4 Other Sketching Probabilities

We consider p_{ij} proportional to the sum of row and column *leverage scores* of $\mathbf{A}$ (Chen et al. 2014) to produce sparse sketch $\tilde{\mathbf{A}}$ using Algorithm 1. Let $\mathbf{A}$ be an $m \times n$ matrix of rank ϱ, and its SVD is given by $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ contains ϱ orthonormal left and right singular vectors, respectively, and $\mathbf{\Sigma}$ is the diagonal matrix of singular values. We define leverage scores as follows (as in Mahoney and Drineas 2009)

$$\begin{aligned} \text{(row leverage)} \quad \mu_i &= \|\mathbf{U}_{(i)}\|_2^2, \quad \forall i \in [m] \\ \text{(column leverage)} \quad \nu_j &= \|\mathbf{V}_{(j)}\|_2^2, \quad \forall j \in [n] \end{aligned}$$

and element-wise leverage score (similar to Chen et al. 2014)

$$p_{lev} = \frac{1}{2}\frac{\mu_i + \nu_j}{(m+n)\varrho} + \frac{1}{2mn}, \quad i \in [m], j \in [n].$$

Note that p_{lev} is a probability distribution on the elements of $\mathbf{A}$. At a high level, the leverage scores of an element (i, j) is proportional to the squared norms of the i-th row of the left singular matrix and the j-th row of the right singular matrix. The constant term in p_{lev} prevents the rescaled values from blowing up in case of sampling an element with tiny leverage score. Such leverage score sampling is distinct from uniform sampling only for *low-rank* matrices or *low-rank approximations* to matrices. So, we consider sparsification of low-rank and rank-truncated data, and compare the quality of sparse sketches produced from such low-rank matrices using our optimal hybrid sampling with that of using p_{lev}.

	m	n	$\|\mathbf{A}\|_0$	nd	$\text{rs}_0 > \text{rs}_1$	α^*
$\mathbf{A}_{0.1}$	500	500	2.5e+5	9.2e+4	no	.63
TTC1	139	15170	37831	12204	yes	1
TTC2	138	11859	29334	9299	yes	1
TTC3	125	15485	47304	14201	yes	1
TTC4	125	14392	35018	10252	yes	1
Digit	2313	256	5.9e+5	5.1e+5	no	.20
Stock	1218	7956	5.5e+6	6.5e+3	no	.74
Gene	107	22215	2.4e+6	2.25e+6	yes	.99

Table 1: α^* for various $m \times n$ data sets ($\epsilon = .05$ is the desired relative-error accuracy). $\text{rs}_0 > \text{rs}_1$ implies ℓ_1 is better than ℓ_2 (Achlioptas et al. 2013a), and we reproduce this ($\alpha^* = 1$). But, for $\text{rs}_0 \leq \text{rs}_1$, $\alpha^* < 1$ and our hybrid sampling is strictly better than both ℓ_1 and ℓ_2.

3.5 Description of Data

Synthetic Data: $\mathbf{A}_{0.1}$ as described in SECTION 1.4 (FIGURE 1).

TechTC Datasets (TTC): (Gabrilovich and Markovitch 2004) These datasets are bag-of-words features for document-term data describing two topics (ids). We choose four such datasets. Rows represent documents and columns are the words. We preprocessed the data by removing all the words of length four or smaller, and then divide each row by its Frobenius norm.

Digit Data: (Hull 1994) A data set of three handwritten digits: six (664 samples), nine (644 samples), and one (1005 samples). Pixels are treated as features, and pixel values are normalized in [-1,1]. Each 16×16 digit image is encoded to form a row in the data matrix (2313 rows and 256 columns).

Stock Data (S&P): We use a temporal data containing 1218 stocks (rows) collected between 1983 and 2011. This temporal data set has with 7056 snapshots (columns) of prices.

Gene Expression Data: We use GSE10072 gene expression data for lung cancer from NCBI Gene Expression Omnibus database. There are 107 samples (58 lung tumor cases and 49 normal lung controls) forming the rows of the matrix, with 22,215 probes (features) from GPL96 platform annotation table.

3.6 Results

All the random results are based on mean of several independent trials (small variance observed).

3.6.1 QUALITY OF SPARSE SKETCH

Table 1 summarizes α^* for various data sets. Achlioptas et al. (2013a) argued that, for $\text{rs}_0 > \text{rs}_1$, ℓ_1 sampling is better than ℓ_2 (even with truncation). Our results on α^* in Table 1 reproduce this condition ($\alpha^* = 1$ implies ℓ_1). Moreover, our method can derive the right blend of ℓ_1 and ℓ_2 sampling even when the above condition fails. In this sense, we generalize the results of Achlioptas et al. (2013a).

	$s/(k(m+n))$	$\hat{\alpha}$
$\mathbf{A}_{0.1}$, $k = 5$	2	0.7
	3	0.7
Digit, $k = 3$	3	0.3
	5	0.3
Stock, $k = 1$	2	0.7
	3	0.7

Table 2: $\hat{\alpha}$ corresponding to the minimum relative error $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \|\mathbf{A}\|_2$, for a fixed budget of samples size s. $\hat{\alpha} \approx \alpha^*$ in Table 1, indicating that our optimal hybrid sampling produces strictly better sparse sketch than both ℓ_1 and ℓ_2. $k \approx \|\mathbf{A}\|_F^2 / \|\mathbf{A}\|_2^2$ (stable rank).

Also, we note the value of α [4] for the minimum relative error $\frac{\|\mathbf{A}-\tilde{\mathbf{A}}\|_2}{\|\mathbf{A}\|_2}$, for a fixed sample size s in Table 2. Table 2 shows that the quality of the sparse sketch produced by our optimal hybrid sampling is strictly better than that of both ℓ_1 and ℓ_2.

• *Comparison with ℓ_2-with-truncation:* We also compare our optimal hybrid sampling with ℓ_2 sampling with truncation. We use two predetermined truncation parameters, $\varepsilon = 0.1$ and $\varepsilon = 0.01$. First, ℓ_2 sampling without truncation turns out to produce the worst quality sparse sketches for all datasets. For real datasets, hybrid-(ℓ_1, ℓ_2) sampling using α^* outperforms ℓ_2 with truncation $\varepsilon = 0.1$ and $\varepsilon = 0.01$. For $\mathbf{A}_{0.1}$, ℓ_2 with $\varepsilon = 0.01$ appears to produce $\tilde{\mathbf{A}}$ that is as bad as ℓ_2 without truncation. However, ℓ_2 with $\varepsilon = 0.1$ shows better quality than optimal hybrid sampling (for $\mathbf{A}_{0.1}$ only) as this ε filters out the noisy elements. We should point out that we know a good ε for $\mathbf{A}_{0.1}$ as we control the noise in this case. However, in reality, we have no control over the noise, and choosing the right ε with no prior knowledge of the data, is an improbable outcome.

3.6.2 QUALITY OF PCA FROM SPARSE SKETCHES

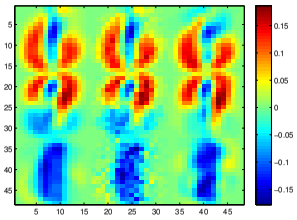
We investigate the quality of Algorithm 2 for Digit data and $\mathbf{A}_{0.1}$. We set $r = 30k$ for the random projection matrix $\mathbf{A}_G$ to achieve a comparable runtime of $\mathcal{G}$ with $\mathcal{H}$ (k is 3 and 5 for Digit and $\mathbf{A}_{0.1}$, respectively). FIGURE 4 shows that $\mathcal{H}$ has better quality than $\mathcal{G}$ for Digit data. Also, Table 3 lists the gain in computation time for Algorithm 2 due to sparsification (using hybrid sampling with α^*). Finally, our optimal hybrid sampling outperforms ℓ_1 and Bernstein sampling (Achlioptas et al. 2013b) by preserving more variance of data via PCA (Table 4).

3.6.3 QUALITY OF SPARSE PCA ALGORITHMS

We report results for primarily the top principal component ($k = 1$) which is the case most considered in the literature. When $k > 1$, our results do not qualitatively change. We note the optimal mixing parameter α^* using Algorithm 1 for various datasets in Table 5.

Handwritten Digits. We sample approximately 7% of the elements from the centered data using (ℓ_1, ℓ_2)-sampling, as well as uniform sampling. The quality of solution for small r is shown in Table 5, including the running time τ. For this data, $f(\mathcal{G}_{\text{max},r})/f(\mathcal{G}_{\text{sp},r}) \approx 0.23$ ($r = 10$), so it is important to use a good sparse PCA algorithm. We see from Table 5 that the (ℓ_1, ℓ_2)-sketch

4. We vary α from 0.1 to 1 with step size 0.1.



Hybrid-(ℓ_1, ℓ_2) σ_h/σ_k	Random Projection σ_g/σ_k
97.94%	96.89%

Figure 4: [Digit data] Approximation quality of PCA (Algorithm 2) from samples of data. (Left) Visualization of principal components as 16×16 image. Principal components are ordered from the top row to the bottom. First column of PCA's are exact ($\mathcal{A}$). Second column of PCA's ($\mathcal{H}$) are computed on sparse sketch with 7% non-zeros of all the elements via hybrid sampling using optimal α. Third column of PCA's ($\mathcal{G}$) are computed on $\mathbf{A}_G$. (Right) Variance preserved by PCA algorithms. Hybrid sampling based PCA, despite using less information, is better than PCA of random projection.

	Sparsified Digit	Sparsified $\mathbf{A}_{0.1}$
# non-zeros	7%	6%
$t_a/t_h/t_G$	151/30/36	73/18/36

Table 3: Computational gain of Algorithm 2 comparing to exact PCA. We report the computation time of MATLAB function 'svds($\mathbf{A}$,k)' for actual data $\mathbf{A}$ (t_a), sparsified data $\tilde{\mathbf{A}}$ (t_h), and random projection data $\mathbf{A}_G$ (t_G). We use only 7% and 6% of all the elements of Digit data and $\mathbf{A}_{0.1}$, respectively, to construct respective sparse sketches. $k = 3$ for Digit and $k = 5$ for $\mathbf{A}_{0.1}$.

significantly outperforms the uniform sketch. A more extensive comparison of recovered variance is given in FIGURE 6(a). We also observe a speed-up of a factor of about 6 for the (ℓ_1, ℓ_2)-sketch. We point out that the uniform sketch is reasonable for the digits data because most data elements are close to either $+1$ or -1, since the pixels are either black or white.

We show a visualization of the principal components in FIGURE 5. We observe that the sparse components from the (ℓ_1, ℓ_2)-sketch are almost identical to that of from the complete data.

TechTC Data. We sample approximately 5% of the elements from the centered data using our (ℓ_1, ℓ_2)-sampling, as well as uniform sampling. For this data, $f(\mathcal{G}_{\text{max},r})/f(\mathcal{G}_{\text{sp},r}) \approx 0.84$ ($r = 10$). We observe a very significant quality difference between the (ℓ_1, ℓ_2)-sketch and uniform sketch. A more extensive comparison of recovered variance is given in FIGURE 6(b). We also observe a speed-up of a factor of about 6 for the (ℓ_1, ℓ_2)-sketch. Unlike the digits data which is uniformly near ± 1, the text data is "spikey" and now it is important to sample with a bias toward larger elements, which is why the uniform-sketch performs very poorly.

	ℓ_1 σ_{ℓ_1}/σ_k	Bernstein σ_b/σ_k	Hybrid-(ℓ_1, ℓ_2) σ_h/σ_k
Digit ($\alpha^* = 0.42$)	98.47 %	98.46 %	**98.51**%
Stock ($\alpha^* = 0.1$)	85.09 %	95.61 %	**97.07**%

Table 4: Variance preserved by PCA of various sparse sketches. Our optimal hybrid sampling outperforms other sampling methods including Bernstein sampling (Achlioptas et al. 2013b). We sample 9% non-zeros from digit data and 2% non-zeros from stock data.

Centered Data	α^*	r	$\frac{f(\mathcal{H}_{\text{max/ sp},r})}{f(\mathcal{G}_{\text{max/ sp},r})}$	$\frac{\tau(\mathcal{G})}{\tau(\mathcal{H})}$	$\frac{f(\mathcal{U}_{\text{max/ sp},r})}{f(\mathcal{G}_{\text{max/ sp},r})}$	$\frac{\tau(\mathcal{G})}{\tau(\mathcal{U})}$
Digit	.42	40	0.99/ 0.90	6.21	1.01/ 0.70	5.33
TTC1	1	40	0.94/ 0.99	5.70	0.41/ 0.38	5.96
Stock	.10	40	1.00/ 1.00	3.72	0.66/ 0.66	4.76
Gene	.82	40	0.82/ 0.88	3.61	0.65/ 0.15	2.53

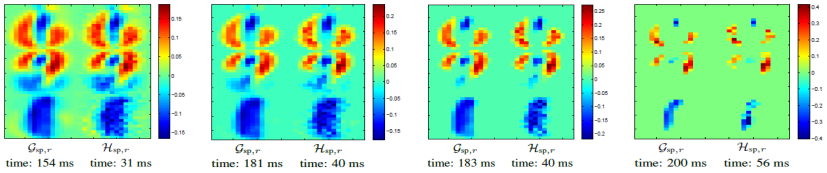
Table 5: Comparison of sparse principal components from the (ℓ_1, ℓ_2)-sketch and uniform sketch. Recall, $\mathcal{G}$ is the ground truth.

As a final comparison, we look at the actual sparse top component with sparsity parameter $r = 10$. The topic IDs in the TechTC data are 10567="*US: Indiana: Evansville*" and 11346="*US: Florida*". The top-10 features (words) in the full PCA on the complete data are shown in Table 6. In Table 7 we show which words appear in the top sparse principal component with sparsity $r = 10$ using various sparse PCA algorithms. We observe that the sparse PCA from the (ℓ_1, ℓ_2)-sketch with only 5% of the data sampled matches quite closely with the same sparse PCA algorithm using the complete data ($\mathcal{G}_{\text{max/sp},r}$ matches $\mathcal{H}_{\text{max/sp},r}$).

Stock Data. We sample about 2% of the non-zero elements from the centered data using the (ℓ_1, ℓ_2)-sampling, as well as uniform sampling. For this data, $f(\mathcal{G}_{\text{max},r})/f(\mathcal{G}_{\text{sp},r}) \approx 0.96$ ($r = 10$). We observe a very significant quality difference between the (ℓ_1, ℓ_2)-sketch and uniform sketch. A more extensive comparison of recovered variance is given in FIGURE 6(c). We also observe a speed-up of a factor of about 4 for the (ℓ_1, ℓ_2)-sketch. Similar to TechTC data this data set is also "spikey", so biased sampling toward larger elements significantly outperforms the uniform-sketch.

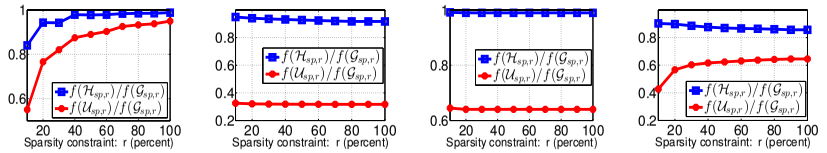
We now look at the actual sparse top component with sparsity parameter $r = 10$. The top-10 features (stocks) in the full PCA on the complete data are shown in Table 8. In Table 9 we show which stocks appear in the top sparse principal component using various sparse PCA algorithms. We observe that the sparse PCA from the (ℓ_1, ℓ_2)-sketch with only 2% of the non-zero elements sampled matches quite closely with the same sparse PCA algorithm using the complete data ($\mathcal{G}_{\text{max/sp},r}$ matches $\mathcal{H}_{\text{max/sp},r}$).

Gene Expression Data. We sample about 9% of the elements from the centered data using the (ℓ_1, ℓ_2)-sampling, as well as uniform sampling. For this data, $f(\mathcal{G}_{\text{max},r})/f(\mathcal{G}_{\text{sp},r}) \approx 0.05$ ($r = 10$) which means a good sparse PCA algorithm is imperative. We observe a very significant quality difference between the (ℓ_1, ℓ_2)-sketch and uniform sketch. A more extensive comparison of re-



(a) $r = 100\%$ (b) $r = 50\%$ (c) $r = 30\%$ (d) $r = 10\%$

Figure 5: [Digits] Visualization of top-3 sparse principal components. In each figure, left panel shows $\mathcal{G}_{\text{sp},r}$ and right panel shows $\mathcal{H}_{\text{sp},r}$. r is the maximum number of non-zeros allowed.



(a) Digit (b) TechTC (c) Stock (d) Gene

Figure 6: Quality of sparse PCA for (ℓ_1, ℓ_2)-sketch and uniform sketch over an extensive range for the sparsity constraint r. The variance preserved by the uniform sketch is significantly lower, highlighting the importance of a good sketch.

covered variance is given in FIGURE 6(d). We also observe a speed-up of a factor of about 4 for the (ℓ_1, ℓ_2)-sketch. Similar to TechTC data this data set is also "spikey", and consequently biased sampling toward larger elements significantly outperforms the uniform-sketch.

Also, we look at the actual sparse top component with sparsity parameter $r = 10$. The top-10 features (probes) in the full PCA on the complete data are shown in Table 10. In Table 11 we show which probes appear in the top sparse principal component with sparsity $r = 10$ using various sparse PCA algorithms. We observe that the sparse PCA from the (ℓ_1, ℓ_2)-sketch with only 9% of the elements sampled matches reasonably with the same sparse PCA algorithm using the complete data ($\mathcal{G}_{\text{max/sp},r}$ matches $\mathcal{H}_{\text{max/sp},r}$).

Finally, we validate the genes corresponding to the top probes in the context of lung cancer. Table 12 lists the top twelve gene symbols in Table 10. Note that a gene can occur multiple times in principal component as genes can be associated with different probes. Genes like SFTPC, AGER, WIF1, and FABP4 are down-regulated in lung cancer, while SPP1 is up-regulated (see the functional gene grouping: `www.sabiosciences.com/rt_pcr_product/HTML/PAHS-134Z.html`). Co-expression analysis on the set of eight genes for $\mathcal{H}_{\text{max},r}$ and $\mathcal{H}_{\text{sp},r}$ using the tool ToppFun (`toppgene.cchmc.org/`) shows that all eight genes appear in a list of selected probes characterizing non-small-cell lung carcinoma (NSCLC) in (Hou et al., 2010, Table

ID	Top 10 in $\mathcal{G}_{\text{max},r}$	ID	Other words
1	**evansville**	11	service
2	**florida**	12	small
3	**south**	13	frame
4	**miami**	14	tours
5	**indiana**	15	faver
6	**information**	16	transaction
7	**beach**	17	needs
8	**lauderdale**	18	commercial
9	**estate**	19	bullet
10	**spacer**	20	inlets
		21	producer

Table 6: [TechTC] Top ten words in top principal component of the complete data (the other words are discovered by some of the sparse PCA algorithms).

$\mathcal{G}_{\text{max},r}$	$\mathcal{H}_{\text{max},r}$	$\mathcal{U}_{\text{max},r}$	$\mathcal{G}_{\text{sp},r}$	$\mathcal{H}_{\text{sp},r}$	$\mathcal{U}_{\text{sp},r}$
1	1	6	1	1	6
2	2	14	2	2	14
3	3	15	3	3	15
4	4	16	4	4	16
5	5	17	5	5	17
6	7	7	6	7	7
7	6	18	7	8	18
8	8	19	8	6	19
9	11	20	9	12	20
10	12	21	13	11	21

Table 7: [TechTC] Relative ordering of the words (w.r.t. $\mathcal{G}_{\text{max},r}$) in the top sparse principal component with sparsity parameter $r = 10$.

ID	Top 10 in $\mathcal{G}_{\text{max},r}$	ID	Other stocks
1	**T.2**	11	HET.
2	**AIG**	12	ONE.1
3	**C**	13	MA
4	**UIS**	14	XOM
5	**NRTLQ**	15	PHA.1
6	**S.1**	16	CL
7	**GOOG**	17	WY
8	**MTLQQ**		
9	**ROK**		
10	**EK**		

Table 8: [Stock data] Top ten stocks in top principal component of the complete data (the other stocks are discovered by some of the sparse PCA algorithms).

$\mathcal{G}_{\text{max},r}$	$\mathcal{H}_{\text{max},r}$	$\mathcal{U}_{\text{max},r}$	$\mathcal{G}_{\text{sp},r}$	$\mathcal{H}_{\text{sp},r}$	$\mathcal{U}_{\text{sp},r}$
1	1	2	1	1	2
2	2	11	2	2	11
3	3	12	3	3	12
4	4	13	4	4	13
5	5	14	5	5	14
6	6	3	6	7	3
7	7	15	7	6	15
8	9	9	8	8	9
9	8	16	9	9	16
10	11	17	10	11	17

Table 9: [Stock data] Relative ordering of the stocks (w.r.t. $\mathcal{G}_{\text{max},r}$) in the top sparse principal component with sparsity parameter $r = 10$.

S1). Further, AGER and FAM107A appear in the *top five* highly discriminative genes in (Hou et al., 2010, Table S3). Additionally, AGER, FCN3, SPP1, and ADH1B appear among the 162 most differentiating genes across two subtypes of NSCLC and normal lung cancer in (Dracheva et al., 2007, Supplemental Table 1). Such findings show that our method can identify, from incomplete data, important genes for complex diseases like cancer. Also, notice that our sampling-based method is able to identify additional important genes, such as, FCN3 and FAM107A in top ten genes.

• *Quality of Other Sketches:* We briefly report on other options for sketching $\mathbf{A}$. We consider suboptimal α (not α^* from (10) to construct a suboptimal hybrid distribution, and use this in proto-Algorithm 1 to construct a sparse sketch. FIGURE 7 reveals that a good sketch using the α^* is important.

ID	Top 10 in $\mathcal{G}_{\max,r}$	ID	Other probes
1	**210081_at**	11	205866_at
2	**214387_x_at**	12	209074_s_at
3	**211735_x_at**	13	205311_at
4	**209875_s_at**	14	216379_x_at
5	**205982_x_at**	15	203571_s_at
6	**215454_x_at**	16	205174_s_at
7	**209613_s_at**	17	204846_at
8	**210096_at**	18	209116_x_at
9	**204712_at**	19	202834_at
10	**203980_at**	20	209425_at
		21	215356_at
		22	221805_at
		23	209942_x_at
		24	218450_at
		25	202508_s_at

Table 10: [Gene data] Top ten probes in top principal component of the complete data (the other probes are discovered by some of the sparse PCA algorithms).

$\mathcal{G}_{\max,r}$	$\mathcal{H}_{\max,r}$	$\mathcal{U}_{\max,r}$	$\mathcal{G}_{\text{sp},r}$	$\mathcal{H}_{\text{sp},r}$	$\mathcal{U}_{\text{sp},r}$
1	4	13	1	4	13
2	1	14	2	1	16
3	11	3	3	2	15
4	2	15	4	11	19
5	3	5	5	3	20
6	8	16	6	8	21
7	7	6	7	7	22
8	9	17	8	9	23
9	5	4	9	5	24
10	12	18	10	12	25

Table 11: [Gene data] Relative ordering of the probes (w.r.t. $\mathcal{G}_{\max,r}$) in the top sparse principal component with sparsity parameter $r = 10$.

$\mathcal{G}_{\max,r}$	ν	$\mathcal{H}_{\max,r}$	ν	$\mathcal{H}_{\text{sp},r}$	ν
SFTPC	4	SFTPC	3	SFTPC	3
AGER	1	SPP1	1	SPP1	1
SPP 1	1	AGER	1	AGER	1
ADH1B	1	FCN3	1	FCN3	1
CYP4B1	1	CYP4B1	1	CYP4B1	1
WIF1	1	ADH1B	1	ADH1B	1
FABP4	1	WIF1	1	WIF1	1
		FAM107A	1	FAM107A	1

Table 12: [Gene data] Gene symbols corresponding to top probes in Table 11. One gene can be associated with multiple probes. Here ν is the frequency of occurrence of a gene in top ten probes of their respective principal component.

3.6.4 ESTIMATE OF α^* FROM SAMPLES

Table 13 shows the quality of estimated α^* computed by Algorithm 4 using a small number of observed data elements. Note that most of the elements of Digit data have magnitude close to 1; consequently we need more samples to estimate α^* accurately.

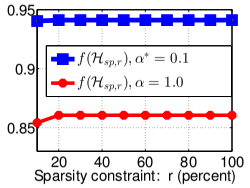


Figure 7: [Stock data] Quality of sketch using *suboptimal* α to illustrate the importance of the optimal mixing parameter α^*.

	Non-zeros sampled	Estimated α^*
Digit, $\alpha^* = 0.42$	6 %	0.80
	19 %	0.67
Stock, $\alpha^* = 0.1$	2 %	0.1
	10 %	0.1

Table 13: Estimated α^* using Algorithm 4 from small number of non-zeros of Digit and Stock data. Most of elements of Digit data are close to either $+1$ or -1, thus requiring more samples to accurately estimate α^*.

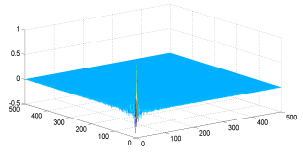
3.6.5 OTHER SKETCHING PROBABILITIES

We compare the quality of sparse sketches produced from low-rank and rank-truncated synthetic and real data using our optimal hybrid sampling with p_{lev}.

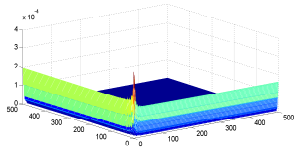
• *Power-law Matrix:* First, we construct a 500×500 low-rank *power-law* matrix, similar to Chen et al. (2014), as follows: $\mathbf{A}_{pow} = \mathbf{D}\mathbf{X}\mathbf{Y}^T\mathbf{D}$, where, matrices $\mathbf{X}$ and $\mathbf{Y}$ are 500×5 i.i.d. Gaussian $\mathcal{N}(0,1)$ and $\mathbf{D}$ is a diagonal matrix with power-law decay, $\mathbf{D}_{ii} = i^{-\gamma}$, $1 \le i \le 500$. The parameter γ controls the 'incoherence' of the matrix (larger γ makes the data more 'spiky').

Further, we construct low-rank approximation by projecting a data set onto a low dimensional subspace. We notice that the datasets projected onto the space spanned by top few principal components preserve the linear structure of the data. For example, Digit data show good separation of digits when projected onto the top three principal components. For TechTC and Gene data the top two respective principal components are good enough to form a low-dimensional subspace where the datasets show reasonable separation of two classes of samples. For the stock data we use top three PCAs because the stable rank is close to 2.

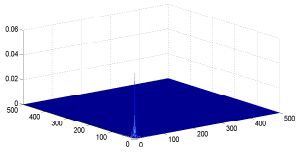
Quality of Sparse Sketch: We measure $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \|\mathbf{A}\|_2$ ($\mathbf{A}$ is the low-rank data) for the above-mentioned two sampling methods using various sample size. Table 14 lists the quality of sparse sketches produced from $\mathbf{A}_{pow}$ via the two sampling methods. Similarly, for all other datasets our optimal hybrid sampling (with α^*) outperforms the leverage score sampling (Table 15) for (low-rank) matrix sparsification.



(a) Low-rank data $\mathbf{A}_{pow}$



(b) Element-wise leverage scores for $\mathbf{A}_{pow}$



(c) Optimal hybrid distribution for $\mathbf{A}_{pow}$

Figure 8: Comparing optimal hybrid-(ℓ_1, ℓ_2) distribution with leverage scores p_{lev} for data $\mathbf{A}_{pow}$ for $\gamma = 1.0$. (a) Structure of $\mathbf{A}_{pow}$, (b) distribution p_{lev}, (c) optimal hybrid-(ℓ_1, ℓ_2) distribution. Our optimal hybrid distribution is more aligned with the structure of the data, requiring much smaller sample size to achieve a given accuracy of sparsification. This is supported by Table 14.

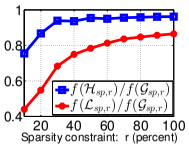
	$\frac{s}{k(m+n)}$	hybrid-(ℓ_1, ℓ_2)	p_{lev}
$\gamma = 0.5$	3	42%	58%
	5	31%	43%
$\gamma = 0.8$	3	15%	43%
	5	12%	40%
$\gamma = 1.0$	3	8%	42%
	5	6%	39%

Table 14: Sparsification quality $\|\mathbf{A}_{pow} - \tilde{\mathbf{A}}_{pow}\|_2 / \|\mathbf{A}_{pow}\|_2$ for low-rank 'power-law' matrix $\mathbf{A}_{pow}$ $(k = 5)$. We compare the quality of hybrid-(ℓ_1, ℓ_2) sampling and leverage score sampling for two sample size. We note (mean) α^* of hybrid-(ℓ_1, ℓ_2) distribution for data $\mathbf{A}_{pow}$ using $\epsilon = 0.05, \delta = 0.1$. For $\gamma = 0.5, 0.8, 1.0$, we have mean $\alpha^* = 0.11, 0.72, 0.8$, respectively, with very small variance.

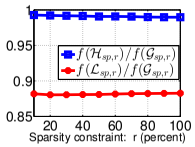
We notice that with increasing γ leverage scores get more aligned with the structure of the data $\mathbf{A}_{pow}$ resulting in gradually improving approximation quality, for the same sample size. Larger γ produces more variance in data elements. ℓ_2 component of our hybrid distribution bias us towards the larger data elements, while ℓ_1 works as a regularizer to maintain the variance of the sampled

	$\frac{s}{k(m+n)}$	Hybrid-(ℓ_1, ℓ_2)	p_{lev}
Digit, $k = 3$	3	44%	61%
	5	34%	47%
$\mathbf{A}_{0.1}$, $k = 5$	3	25%	80%
	5	21%	62%

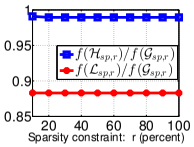
Table 15: Sparsification of rank-truncated data. We compare the quality of $\|\mathbf{A} - \tilde{\mathbf{A}}\|_2 / \|\mathbf{A}\|_2$ for hybrid-(ℓ_1, ℓ_2) sampling and leverage score sampling using two different sample size.



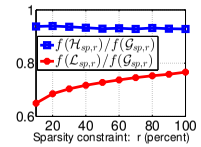
(a) Digit (rank 3)



(b) TechTC (rank 2)



(c) Stock (rank 3)



(d) Gene (rank 2)

Figure 9: [Low-rank data] Quality of sparse PCA of low-rank data for optimal (ℓ_1, ℓ_2)-sketch and leverage score sketch over an extensive range for the sparsity constraint r. The quality of the optimal hybrid sketch is considerably better highlighting the importance of a good sketch.

(and rescaled) elements. With increasing γ we need more regularization to counter the problem of rescaling. Interestingly, our optimal parameter α^* adapts itself with this changing structure of data, e.g. for $\gamma = 0.5, 0.8, 1.0$, we have $\alpha^* = 0.11, 0.72, 0.8$, respectively. This shows the benefit of our parameterized hybrid distribution to achieve a superior approximation quality. FIGURE 8 shows the structure of the data $\mathbf{A}_{pow}$ for $\gamma = 1.0$ along with the optimal hybrid-(ℓ_1, ℓ_2) distribution and leverage score distribution p_{lev}. The figure suggests our optimal hybrid distribution is better aligned with the structure of the data, resulting in a superior sparse sketch of the data. Similar results are observed for other datasets as well.

Quality of Sparse PCA: Let $\mathcal{L}_{\text{sp},r}$ be the r-sparse components using *Spasm* for the leverage score sampled sketch $\tilde{\mathbf{A}}$. FIGURE 9 shows that leverage score sampling is not as effective as the optimal hybrid (ℓ_1, ℓ_2)-sampling for sparse PCA of low-rank data.

3.7 Conclusion

It is possible to use a sparse sketch (incomplete data) to recover nearly as good principal components and sparse principal components as one would have gotten with the complete data. We mention that, while $\mathcal{G}_{\text{max}}$ which uses the largest weights in the unconstrained PCA does not perform well with respect to the variance, it does identify good features. A simple enhancement to $\mathcal{G}_{\text{max}}$ is to recalibrate the sparse component after identifying the features - this is an unconstrained PCA problem on just

the columns of the data matrix corresponding to the features. This method of recalibrating can be used to improve any sparse PCA algorithm.

Our algorithms are simple and efficient, and many interesting avenues for further research remain. Can the sampling complexity for the top-k sparse PCA be reduced from $O(k^2)$ to $O(k)$. We suspect that this should be possible by getting a better bound on $\sum_{i=1}^{k} \sigma_i(\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}})$; we used the crude bound $k\|\mathbf{A}^T\mathbf{A} - \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\|_2$. We also presented a general surrogate optimization bound which may be of interest in other applications. In particular, it is pointed out in Magdon-Ismail and Boutsidis (2016) that though PCA optimizes variance, a more natural way to look at PCA is as the linear projection of the data that minimizes the *information loss*. Magdon-Ismail and Boutsidis (2016) gives efficient algorithms to find sparse linear dimension reduction that minimizes information loss – the information loss of sparse PCA can be considerably higher than optimal. To minimize information loss, the objective to maximize is $f(\mathbf{V}) = \text{trace}(\mathbf{A}^T\mathbf{A}\mathbf{V}(\mathbf{A}\mathbf{V})^\dagger\mathbf{A})$. It would be interesting to see whether one can recover sparse low-information-loss linear projectors from incomplete data.

Overall, the experimental results demonstrate the quality of the algorithms presented here, indicating the superiority of our approach to other extreme choices of element-wise sampling methods, such as, ℓ_1 and ℓ_2 sampling. Also, we demonstrate the theoretical and practical usefulness of hybrid-(ℓ_1, ℓ_2) sampling for fundamental data analysis tasks such as recovering PCA and sparse PCA from sketches. Finally, our sampling scheme outperforms element-wise leverage scores for the sparsification of various *low-rank* synthetic and real data.

Acknowledgments

PD was supported by NSF IIS-1661760. AK was partially supported by NSF IIS-1661760. MMI was supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (the ARL Network Science CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

Appendix A. Proof of Theorem 2

In this section we provide a proof of Theorem 2 following the proof outline of Drineas and Zouzias (2011); Achlioptas et al. (2013a). We use Lemma 1 as our main tool. For all $t \in [s]$ we define the matrix $\mathbf{M}_t \in \mathbb{R}^{m\times n}$ as follows:

$$\mathbf{M}_t = \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}} \mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}.$$

It now follows that $\frac{1}{s}\sum_{t=1}^{s} \mathbf{M}_t = S_\Omega(\mathbf{A}) - \mathbf{A}$. We can bound $\|\mathbf{M}_t\|_2$ for all $t \in [s]$. We define the following quantity:

$$\lambda = \frac{\|\mathbf{A}\|_1 \cdot |\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2}, \text{ for } \mathbf{A}_{ij} \neq 0 \tag{24}$$

Lemma 8 *Using our notation, and using probabilities of the form (3), for all $t \in [s]$,*

$$\|\mathbf{M}_t\|_2 \leq \max_{\substack{i,j:\\ \mathbf{A}_{ij}\neq 0}} \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\lambda} + \|\mathbf{A}\|_2.$$

Proof Using probabilities of the form (3), and because $\mathbf{A}_{ij} = 0$ is never sampled,

$$\begin{aligned}\|\mathbf{M}_t\|_2 &= \|(\mathbf{A}_{i_t j_t}/p_{i_t j_t})\mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}\|_2 \\ &\leq \max_{\substack{i,j:\\ \mathbf{A}_{ij}\neq 0}} \left\{ \left(\frac{\alpha}{\|\mathbf{A}\|_1} + \frac{(1-\alpha)\cdot|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_F^2} \right)^{-1} \right\} + \|\mathbf{A}\|_2,\end{aligned}$$

Using (24), we obtain the bound. ■

Next we bound the spectral norm of the expectation of $\mathbf{M}_t\mathbf{M}_t^T$.

Lemma 9 *Using our notation, and using probabilities of the form (3), for all $t \in [s]$,*

$$\|\mathbb{E}(\mathbf{M}_t\mathbf{M}_t^T)\|_2 \leq \|\mathbf{A}\|_F^2\beta_1 - \sigma^2,$$

$$\beta_1 = \max_i \sum_{j=1}^{n} \left(\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1},$$

for $\mathbf{A}_{ij} \neq 0$.

Proof Recall that $\mathbf{A} = \sum_{i,j=1}^{m,n} \mathbf{A}_{ij}\mathbf{e}_i\mathbf{e}_j^T$ and $\mathbf{M}_t = \frac{\mathbf{A}_{i_t j_t}}{p_{i_t j_t}}\mathbf{e}_{i_t}\mathbf{e}_{j_t}^T - \mathbf{A}$. We derive

$$\mathbb{E}[\mathbf{M}_t\mathbf{M}_t^T] = \sum_{i,j=1}^{m,n} (\mathbf{A}_{ij}^2/p_{ij})\mathbf{e}_i\mathbf{e}_i^T - \mathbf{A}\mathbf{A}^T.$$

Sampling according to probabilities of (3), and because $\mathbf{A}_{ij} = 0$ is never sampled, we get,

$$\begin{aligned}\sum_{i,j=1}^{m,n} \frac{\mathbf{A}_{ij}^2}{p_{ij}} &= \|\mathbf{A}\|_F^2 \sum_{i,j=1}^{m,n} \left(\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1}, \\ &\leq \|\mathbf{A}\|_F^2 \sum_{i=1}^{m} \max_i \sum_{j=1}^{n} \left(\frac{\alpha\cdot\|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}|\cdot\|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1}.\end{aligned}$$

for $\mathbf{A}_{ij} \neq 0$. Thus,

$$\mathbb{E}[\mathbf{M}_t \mathbf{M}_t^T] \;\preceq\; \|\mathbf{A}\|_F^2 \beta_1 \sum_{i=1}^{m} \mathbf{e}_i \mathbf{e}_i^T - \mathbf{A}\mathbf{A}^T = \|\mathbf{A}\|_F^2 \beta_1 \mathbf{I}_m - \mathbf{A}\mathbf{A}^T.$$

Note that, $\|\mathbf{A}\|_F^2 \beta_1 \mathbf{I}_m$ is diagonal with all entries non-negative, and $\mathbf{A}\mathbf{A}^T$ is postive semi-definite. Therefore,

$$\|\mathbb{E}[\mathbf{M}_t \mathbf{M}_t^T]\|_2 \leq \|\mathbf{A}\|_F^2 \beta_1 - \sigma^2.$$

■

Similarly, we can obtain

$$\left\|\mathbb{E}[\mathbf{M}_t^T \mathbf{M}_t]\right\|_2 \leq \|\mathbf{A}\|_F^2 \beta_2 - \sigma^2,$$

where, for $\mathbf{A}_{ij} \neq 0$,

$$\beta_2 = \max_j \sum_{i=1}^{m} \left(\frac{\alpha \cdot \|\mathbf{A}\|_F^2}{|\mathbf{A}_{ij}| \cdot \|\mathbf{A}\|_1} + (1-\alpha) \right)^{-1}.$$

We can now apply Lemma 1 with

$$\rho^2(\alpha) = \|\mathbf{A}\|_F^2 \max\{\beta_1, \beta_2\} - \sigma^2,$$

and

$$\gamma(\alpha) = \frac{\|\mathbf{A}\|_1}{\alpha + (1-\alpha)\lambda} + \|\mathbf{A}\|_2,$$

to conclude that $\|\mathcal{S}_\Omega(\mathbf{A}) - \mathbf{A}\|_2 \leq \varepsilon$ holds subject to a failure probability at most

$$(m+n) \exp\left(\frac{-s\varepsilon^2/2}{\rho^2(\alpha) + \gamma(\alpha)\varepsilon/3} \right).$$

Bounding the failure probability by δ, and setting $\varepsilon = \epsilon \cdot \|\mathbf{A}\|_2$, we complete the proof.

Appendix B. Proof of Lemma 4

Lemma 4 will be a corollary of a more general result, for a class of optimization problems involving a Lipschitz-like objective function over an arbitrary (not necessarily convex) domain. Let $f(\mathbf{V}, \mathbf{X})$ be a function that is defined for a matrix variable $\mathbf{V}$ and a matrix parameter $\mathbf{X}$. The optimization variable $\mathbf{V}$ is in some feasible set $\mathcal{S}$ which is arbitrary. The parameter $\mathbf{X}$ is also arbitrary. We assume that f is locally Lipschitz in $\mathbf{X}$, that is

$$|f(\mathbf{V}, \mathbf{X}) - f(\mathbf{V}, \tilde{\mathbf{X}})| \leq \gamma(\mathbf{X}) \|\mathbf{X} - \tilde{\mathbf{X}}\|_2, \qquad \forall \mathbf{V} \in \mathcal{S}.$$

(Note we allow the "Lipschitz constant" to depend on the fixed matrix $\mathbf{X}$ but not the variables $\mathbf{V}, \tilde{\mathbf{X}}$; this is more general than a globally Lipshitz objective) The next lemma is the key tool we need to prove Lemma 4 and it may be of independent interest in other optimization settings. We are interested in maximizing $f(\mathbf{V}, \mathbf{X})$ with respect to $\mathbf{V}$ to obtain $\mathbf{V}^*$. But, we only have an approximation $\tilde{\mathbf{X}}$ for $\mathbf{X}$, and so we maximize $f(\mathbf{V}, \tilde{\mathbf{X}})$ to obtain $\tilde{\mathbf{V}}^*$, which will be a suboptimal solution with respect to $\mathbf{X}$. We wish to bound $f(\mathbf{V}^*, \mathbf{X}) - f(\tilde{\mathbf{V}}^*, \mathbf{X})$ which quantifies how suboptimal $\tilde{\mathbf{V}}^*$ is with respect to $\mathbf{X}$.

Lemma 10 (Surrogate optimization bound) *Let $f(\mathbf{V}, \mathbf{X})$ be γ-locally Lipschitz w.r.t. $\mathbf{X}$ over the domain $\mathbf{V} \in \mathcal{S}$. Define* $\mathbf{V}^* = \arg\max_{\mathbf{V} \in \mathcal{S}} f(\mathbf{V}, \mathbf{X})$; $\tilde{\mathbf{V}}^* = \arg\max_{\mathbf{V} \in \mathcal{S}} f(\mathbf{V}, \tilde{\mathbf{X}})$. *Then,*

$$f(\mathbf{V}^*, \mathbf{X}) - f(\tilde{\mathbf{V}}^*, \mathbf{X}) \leq 2\gamma(\mathbf{X}) \|\mathbf{X} - \tilde{\mathbf{X}}\|_2.$$

In the lemma, the function f and the domain $\mathcal{S}$ are arbitrary. In our setting, $\mathbf{X} \in \mathbb{R}^{n \times n}$, the domain $\mathcal{S} = \{\mathbf{V} \in \mathbb{R}^{n \times k}; \mathbf{V}^T\mathbf{V} = \mathbf{I}_k; \|\mathbf{V}^{(j)}\|_0 \leq r\}$, and $f(\mathbf{V}, \mathbf{X}) = \text{trace}(\mathbf{V}^T\mathbf{X}\mathbf{V})$. We first show that f is Lipschitz w.r.t. $\mathbf{X}$ with $\gamma = k$ (a constant independent of $\mathbf{X}$). Let the representation of $\mathbf{V}$ by its columns be $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_k]$. Then,

$$|\text{trace}(\mathbf{V}^T\mathbf{X}\mathbf{V}) - \text{trace}(\mathbf{V}^T\tilde{\mathbf{X}}\mathbf{V})| = |\text{trace}((\mathbf{X} - \tilde{\mathbf{X}})\mathbf{V}\mathbf{V}^T)| \leq \sum_{i=1}^{k} \sigma_i(\mathbf{X} - \tilde{\mathbf{X}}) \leq k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2,$$

where, $\sigma_i(\mathbf{A})$ is the i-th largest singular value of $\mathbf{A}$ (we used Von-neumann's trace inequality and the fact that $\mathbf{V}\mathbf{V}^T$ is a k-dimensional projection). Now, by Lemma 10, $\text{trace}(\mathbf{V}^{*T}\mathbf{X}\mathbf{V}^*) - \text{trace}(\tilde{\mathbf{V}}^{*T}\mathbf{X}\tilde{\mathbf{V}}^*) \leq 2k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2$. Lemma 4 follows by setting $\mathbf{X} = \mathbf{A}^T\mathbf{A}$ and $\tilde{\mathbf{X}} = \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$.

Appendix C. Proof of Lemma 10

In this section we provide a proof of Lemma 10. First, we need the following lemma.

Lemma 11 *Let f and g be functions on a domain $\mathcal{S}$. Then,*

$$\sup_{x \in \mathcal{S}} f(x) - \sup_{y \in \mathcal{S}} g(y) \leq \sup_{x \in \mathcal{S}} (f(x) - g(x)).$$

Proof

$$\sup_{x \in \mathcal{S}} (f(x) - g(x)) \geq f(x) - g(x) \geq f(x) - \sup_{y \in \mathcal{S}} g(y), \quad \forall x \in \mathcal{S}.$$

Since RHS holds for all x, $\sup_{x \in \mathcal{S}}(f(x) - g(x))$ is an upper bound for $f(x) - \sup_{y \in \mathcal{S}} g(\mathbf{U})$, and hence

$$\sup_{x \in \mathcal{S}} (f(x) - g(x)) \;\geq\; \sup_{x \in \mathcal{S}} \left(f(x) - \sup_{y \in \mathcal{S}} g(y) \right) = \sup_{x \in \mathcal{S}} f(x) - \sup_{y \in \mathcal{S}} g(y).$$

■

Proof (Lemma 10) Suppose that $\max_{\mathbf{V} \in \mathcal{S}} f(\mathbf{V}, \mathbf{X})$ is attained at $\mathbf{V}^*$ and $\max_{\mathbf{V} \in \mathcal{S}} f(\mathbf{V}, \tilde{\mathbf{X}})$ is attained at $\tilde{\mathbf{V}}^*$, and define $\epsilon = f(\mathbf{V}^*, \mathbf{X}) - f(\tilde{\mathbf{V}}^*, \mathbf{X})$. We have that

$$\begin{aligned} \epsilon &= f(\mathbf{V}^*, \mathbf{X}) - f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) + f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) - f(\tilde{\mathbf{V}}^*, \mathbf{X}) \\ &= \max_{\mathbf{V}} f(\mathbf{V}, \mathbf{X}) - \max_{\mathbf{U}} f(\mathbf{U}, \tilde{\mathbf{X}}) + f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) - f(\tilde{\mathbf{V}}^*, \mathbf{X}) \\ &\leq \max_{\mathbf{V}} \left(f(\mathbf{V}, \mathbf{X}) - f(\mathbf{V}, \tilde{\mathbf{X}}) \right) + f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) - f(\tilde{\mathbf{V}}^*, \mathbf{X}), \end{aligned}$$

where the last step follows from Lemma 11. Therefore,

$$\begin{aligned} |\epsilon| &\leq \max_{\mathbf{V}} \left| f(\mathbf{V}, \mathbf{X}) - f(\mathbf{V}, \tilde{\mathbf{X}}) \right| + |f(\tilde{\mathbf{V}}^*, \tilde{\mathbf{X}}) - f(\tilde{\mathbf{V}}^*, \mathbf{X})| \\ &\leq \max_{\mathbf{V}} \gamma(\mathbf{X}) \|\mathbf{X} - \tilde{\mathbf{X}}\|_2 + \gamma(\mathbf{X}) \|\mathbf{X} - \tilde{\mathbf{X}}\|_2 = 2\gamma(\mathbf{X}) \|\mathbf{X} - \tilde{\mathbf{X}}\|_2. \end{aligned}$$

(We used the Lipschitz condition in the second step.) ■

Appendix D. An alternative proof of Lemma 4

Let $\varepsilon = \text{trace}(\mathbf{V}^T\mathbf{X}\mathbf{V}) - \text{trace}(\tilde{\mathbf{V}}^T\mathbf{X}\tilde{\mathbf{V}})$.

$$\begin{aligned}
\varepsilon &= \text{trace}(\mathbf{V}^T\mathbf{X}\mathbf{V}) - \text{trace}(\mathbf{V}^T\tilde{\mathbf{X}}\mathbf{V}) + \text{trace}(\mathbf{V}^T\tilde{\mathbf{X}}\mathbf{V}) - \text{trace}(\tilde{\mathbf{V}}^T\mathbf{X}\tilde{\mathbf{V}}) \\
&\leq k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2 + \text{trace}(\mathbf{V}^T\tilde{\mathbf{X}}\mathbf{V}) - \text{trace}(\tilde{\mathbf{V}}^T\mathbf{X}\tilde{\mathbf{V}}) \\
&\leq k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2 + \text{trace}(\tilde{\mathbf{V}}^T\tilde{\mathbf{X}}\tilde{\mathbf{V}}) - \text{trace}(\tilde{\mathbf{V}}^T\mathbf{X}\tilde{\mathbf{V}}) \\
&\leq 2k\|\mathbf{X} - \tilde{\mathbf{X}}\|_2.
\end{aligned}$$

Setting $\mathbf{X} = \mathbf{A}^T\mathbf{A}$ and $\tilde{\mathbf{X}} = \tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$ we derive the result.

Appendix E. Proof of Theorem 7

Proof Here we use the notations in Algorithm 3. Let, $p_1 = \frac{|\mathbf{A}_{ij}|}{\|\mathbf{A}\|_1}$, and $p_2 = \frac{\mathbf{A}_{ij}^2}{\|\mathbf{A}\|_F^2}$.
Note that t-th elements of S_1 and S_2 are sampled independently with ℓ_1 and ℓ_2 probabilities, respectively. We consider the following disjoint events:

$$\begin{aligned}
&\mathcal{E}_1 : S_1(t) = (i, j, \mathbf{A}_{ij}) \wedge S_2(t) \neq (i, j, \mathbf{A}_{ij}) \\
&\mathcal{E}_2 : S_1(t) \neq (i, j, \mathbf{A}_{ij}) \wedge S_2(t) = (i, j, \mathbf{A}_{ij}) \\
&\mathcal{E}_3 : S_1(t) = (i, j, \mathbf{A}_{ij}) \wedge S_2(t) = (i, j, \mathbf{A}_{ij}) \\
&\mathcal{E}_4 : S_1(t) \neq (i, j, \mathbf{A}_{ij}) \wedge S_2(t) \neq (i, j, \mathbf{A}_{ij})
\end{aligned}$$

Let us denote the events $x_1 : x \leq \alpha$ and $x_2 : x > \alpha$. Clearly, $P[x_1] = \alpha, P[x_2] = 1 - \alpha$.
The elements $S_1(t)$ and $S_2(t)$ are sampled independently, we have

$$\begin{aligned}
P[\mathcal{E}_1] &= P[S_1(t) = (i, j, \mathbf{A}_{ij})]P[S_2(t) \neq (i, j, \mathbf{A}_{ij})] = p_1(1 - p_2). \\
P[\mathcal{E}_2] &= (1 - p_1)p_2, \quad P[\mathcal{E}_3] = p_1p_2, \quad P[\mathcal{E}_4] = (1 - p_1)(1 - p_2).
\end{aligned}$$

We note that α is independent of elements of S_1 and S_2. Therefore, events x_1 and x_2 are independent of the events $\mathcal{E}_j$, $j = 1, 2, 3, 4$. Thus,

$$\begin{aligned}
P[S(t) = (i, j, \mathbf{A}_{ij})] &= P[(\mathcal{E}_1 \wedge x_1) \vee (\mathcal{E}_2 \wedge x_2) \vee \mathcal{E}_3] \\
&= P[\mathcal{E}_1 \wedge x_1] + P[\mathcal{E}_2 \wedge x_2] + P[\mathcal{E}_3] \\
&= P[\mathcal{E}_1]P[x_1] + P[\mathcal{E}_2]P[x_2] + P[\mathcal{E}_3] \\
&= p_1(1 - p_2)\alpha + (1 - p_1)p_2(1 - \alpha) + p_1p_2 \\
&= \alpha \cdot p_1 + (1 - \alpha) \cdot p_2
\end{aligned}$$

■

Algorithm 5 SELECT-s: One-pass SELECT Algorithm to Sample s Elements

Input: $\{a_1, ..., a_N\}$, $a_i \geq 0$, read in one pass sequentially over the data; number of samples s.
$I[1, ..., s] \leftarrow 0$; $V[1, ..., s] \leftarrow 0$ /* arrays to hold s indices and values */
$p_i = 0$.
/* Stream begins */
For $i = 1, ..., N$
  $p_i = p_i + a_i$.
  (in parallel) generate s independent uniform random numbers $r_j \in [0, 1]$, $j = 1, ..., s$.
  (in parallel) $I[j] = i$ and $V[j] = a_i$, if $r_j \leq a_i/p_i$.
End
Output: random indices I and corresponding values V.

Appendix F. SELECT-s Algorithm

Here we extend the SELECT algorithm appeared on p. 137 of Drineas et al. (2006) to select s elements in parallel. For one-pass ℓ_1 and ℓ_2 sampling on elements of $\mathbf{A}$ we call SELECT-s algorithm with inputs $\{|\mathbf{A}_{ij}|\}$ and $\{\mathbf{A}_{ij}^2\}$, respectively, for all i, j.

References

D. Achlioptas and F. McSherry. Fast computation of low rank matrix approximations. In *Symposium on the Theory of Computing*, pages 611–618, 2001.

D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, page 54(2):9, 2007.

D. Achlioptas, Z. Karnin, and E. Liberty. Matrix entry-wise sampling : Simple is best. In `https://pdfs.semanticscholar.org/aa64/b8fb3382e42f90ee93a1dd0c78f13833963f.pdf`, 2013a.

D. Achlioptas, Z. Karnin, and E. Liberty. Near-optimal entrywise sampling for data matrices. In *Neural Information Processing Systems*, pages 1565–1573, 2013b.

S. Arora, E. Hazan, and S. Kale. A Fast Random Sampling Algorithm for Sparsifying Matrices. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 272–279, vol 4110. Springer, 2006.

M. Asteris, D. Papailiopoulos, and A. Dimakis. Non-negative sparse PCA with provable guarantees. In *International Conference on Machine Learning*, 2014.

J. Cadima and I. Jolliffe. Loadings and correlations in the interpretation of principal components. *Applied Statistics*, 22:203–214, 1995.

T. T. Cai, Z. Ma, and Y. Wu. Sparse pca: Optimal rates and adaptive estimation. *The Annals of Statistics*, 41(6):3074–3110, 2013.

Y. Chen, S. Bhojanapalli, S. Sanghavi, and R. Ward. Coherent Matrix Completion. *International Conference on Machine Learning*, pages 674–682, 2014.

A. d'Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

A. d'Aspremont, F. Bach, and L. E. Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, June 2008.

T. Dracheva, R. Philip, W. Xiao, AG Gee, and et al. Distinguishing Lung Tumours From Normal Lung Based on a Small Set of Genes. In *Lung Cancer*, pages 157–164, 55(2), 2007.

P. Drineas and A. Zouzias. A note on element-wise matrix sparsification via a matrix-valued Bernstein inequality. In *Information Processing Letters*, pages 385–389, 111(8), 2011.

P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices I: approximating matrix multiplication. In *SIAM Journal on Computing*, pages 132–157, 36(1), 2006.

E. Gabrilovich and S. Markovitch. Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5. In *International Conference on Machine Learning*, 2004.

J. Hou, J. Aerts, B. den Hamer, and et al. Gene Expression-Based Classification of Non-Small Cell Lung Carcinomas and Survival Prediction. In *PLoS One*, page 5(4):e10312, 2010.

J. J. Hull. A database for handwritten text recognition research. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 550–554, 16(5), 1994.

J. Lei and V. Q. Vu. Sparsistency and agnostic inference in sparse pca. *The Annals of Statistics*, 43 (1):299–322, 2015.

K. Lounici. Sparse principal component analysis with missing observations. *High Dimensional Probability VI: The Banff Volume*, pages 327–356, 2013.

Z. Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41 (2):772–801, 2013.

M. Magdon-Ismail. NP-hardness and inapproximability of sparse PCA. *Information Processing Letters*, 126:35–38, 2017. Earlier version: arXiv:1502.05675, 2015.

M. Magdon-Ismail and C. Boutsidis. Optimal sparse linear encoders and sparse pca. In *Neural Information Processing Systems*, 2016.

M.W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. In *National Academy of Sciences*, pages 697–702, 106 (3), 2009.

B. Moghaddam, Y. Weiss, and S. Avidan. Generalized spectral bounds for sparse LDA. In *International Conference on Machine Learning*, 2006.

B. Recht. A simpler approach to matrix completion. In *The Journal of Machine Learning Research*, pages 3413–3430, 12, 2011.

H. Shen and Z. J. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, 99:1015–1034, July 2008.

K. Sjstrand, L.H. Clemmensen, R. Larsen, and B. Ersbll. Spasm: A matlab toolbox for sparse statistical modeling. In *Journal of Statistical Software (Accepted for publication)*, 2012.

N. Trendafilov, I. T. Jolliffe, and M. Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12:531–547, 2003.

Z. Wang, H. Lu, and H. Liu. Nonconvex statistical optimization: Minimax-optimal sparse pca in polynomial time. *arXiv preprint arXiv:1408.5352*, 2014.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational & Graphical Statistics*, 15(2):265–286, 2006.

Journal of Machine Learning Research 18 (2017) 1-61 Submitted 6/16; Revised 4/17; Published 7/17

Quantifying the Informativeness of Similarity Measurements

Austin J. Brockmeier AJBROCKMEIER@GMAIL.COM
Department of Computer Science
University of Liverpool
Liverpool L69 3BX, UK

Tingting Mu TINGTING.MU@MANCHESTER.AC.UK
Sophia Ananiadou SOPHIA.ANANIADOU@MANCHESTER.AC.UK
School of Computer Science
University of Manchester
Manchester M1 7DN, UK

John Y. Goulermas J.Y.GOULERMAS@LIVERPOOL.AC.UK
Department of Computer Science
University of Liverpool
Liverpool L69 3BX, UK

Editor: Christoph Lampert

Abstract

In this paper, we describe an unsupervised measure for quantifying the 'informativeness' of correlation matrices formed from the pairwise similarities or relationships among data instances. The measure quantifies the heterogeneity of the correlations and is defined as the distance between a correlation matrix and the nearest correlation matrix with constant off-diagonal entries. This non-parametric notion generalizes existing test statistics for equality of correlation coefficients by allowing for alternative distance metrics, such as the Bures and other distances from quantum information theory. For several distance and dissimilarity metrics, we derive closed-form expressions of informativeness, which can be applied as objective functions for machine learning applications. Empirically, we demonstrate that informativeness is a useful criterion for selecting kernel parameters, choosing the dimension for kernel-based nonlinear dimensionality reduction, and identifying structured graphs. We also consider the problem of finding a maximally informative correlation matrix around a target matrix, and explore parameterizing the optimization in terms of the coordinates of the sample or through a lower-dimensional embedding. In the latter case, we find that maximizing the Bures-based informativeness measure, which is maximal for centered rank-1 correlation matrices, is equivalent to minimizing a specific matrix norm, and present an algorithm to solve the minimization problem using the norm's proximal operator. The proposed correlation denoising algorithm consistently improves spectral clustering. Overall, we find informativeness to be a novel and useful criterion for identifying non-trivial correlation structure.

Keywords: correlation matrices, similarity information, kernel methods, information theory, quantum information theory, clustering

©2017 Austin J. Brockmeier, Tingting Mu, Sophia Ananiadou, and John Y. Goulermas.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-296.html.

1. Introduction

Shannon's entropy measures the dispersion of a sample of objects among the possible elements of a discrete space. A sample consisting solely of repeated instances of the same element has minimal entropy, and a sample wherein each object is distinct from the rest has maximum entropy. If these samples correspond to the labels assigned by a clustering algorithm, then neither the minimal entropy sample corresponding to a single cluster, nor the maximal entropy sample, where each instance is in its own cluster, is very informative about any underlying organization within the sample. In this work, we investigate and introduce univariate measures of *informativeness* that are minimized for both the maximal and minimal entropy samples. Informativeness is based on the heterogeneity of the similarity measurements for a sample and is applicable to samples from any space that has a positive semidefinite correlation measure. The positive definiteness ensures that the similarity measurements can be represented as inner-products in a Hilbert-space, or more concretely, that a given sample can be embedded in Euclidean space regardless of the original space. The restriction to correlation measurements ensures each instance is represented by a vector with equal magnitude, which distinguishes the effect of differences in similarity from differences in scale or variance, and ensures that each object is equally represented. Any positive semidefinite matrix formed from pairwise similarity measurements can be normalized to obtain a correlation matrix.

The notion of informativeness coincides with hypothesis tests for the homogeneity of correlation coefficients, that is, how uniform the off-diagonal elements of a correlation matrix are (Bartlett, 1954; Anderson, 1963; Lawley, 1963; Gleser, 1968; Aitkin et al., 1968; Steiger, 1980; Brien et al., 1984). According to this notion, a *non-informative* correlation matrix has equal off-diagonal elements, which indicates that no pair of objects is more similar than any other pair. For example, a constant matrix of all ones implies the objects are indistinguishable, and an identity matrix implies the objects are all distinct with no similarity. While observing an identity correlation matrix may be informative itself, because the objects have no dependence, no hierarchical grouping or clustering of the objects is appropriate, and any reordering of the objects yields the same correlation matrix. This last property—invariance to permutations—provides another definition of a non-informative correlation matrix.

The null hypothesis of the aforementioned tests is that the expected value of the correlation matrix has equal off-diagonal elements, where the parameter of the off-diagonal elements is left unspecified. Tests of equality of correlation coefficients are a specific case of hypothesis tests for patterns within sets of correlation coefficients (Steiger, 1980). Instead of hypothesis testing, we are interested in using informativeness as an objective function for unsupervised learning and for making relative comparisons between correlation matrices of equal size—in particular, between matrices corresponding to the same sample, e.g., kernel matrices formed with different kernel functions, or correlation matrices obtained by applying different dimensionality reduction techniques. For instance, evaluating a Gaussian kernel on a sample of distinct points yields a non-informative correlation matrix when the kernel bandwidth is chosen either too large (resulting in a nearly constant matrix) or too small (resulting in an identity matrix). These extremes correspond to the minimal and maximal cases of matrix entropy as defined by von Neumann (1955) for quantum density matrices. While many heuristics can be used to select an appropriate kernel bandwidth or

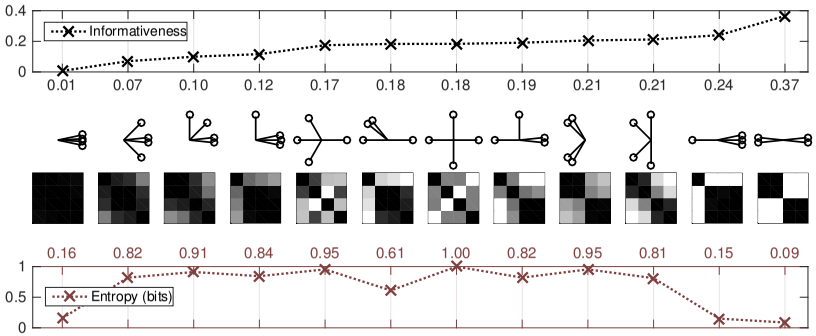


Figure 1: Informativeness versus von Neumann entropy for correlation matrices obtained from various configurations of four unit vectors. Both measures are minimal when the vectors are configured in a single cluster. Informativeness is higher for nontrivial clusterings, whereas entropy is maximized when the vectors are maximally separated.

the dimension of a lower-rank approximation of a kernel matrix, informativeness provides an unsupervised objective function for these selections.

Informativeness can be used to rank different samples and identify samples that have more structure. An example comparing informativeness to entropy for different samples of two-dimensional vectors is shown in Figure 1. Informativeness can also be applied to graphs, since the normalized graph Laplacian is a correlation matrix, and informativeness can be used to identify graphs that exhibit more structure, for example, regularity or clustering. As an objective function, informativeness can be used for enhancing the observed structure in a sample by searching for a matrix with more exaggerated differences in correlation coefficients around a target matrix.

While there are existing statistical tests to measure the heterogeneity of correlation coefficients, we define informativeness based on the distance between a given correlation matrix and the nearest non-informative matrix. In other words, informativeness is proportional to the distance between a given correlation matrix and the set of non-informative matrices. This definition is essential, since defining the distance to a particular non-informative matrix, either the identity matrix or a constant matrix, is insufficient as these distances will be proportional or inversely proportional to the von Neumann entropy. Even if two equal-sized correlation matrices are closest to distinct non-informative matrices, the distance to the set can be used to determine which matrix is more informative.

As defined, informativeness will be dependent upon the choice of the distance metric. Among other distances, we explore some from quantum information theory (Fuchs and van de Graaf, 1999; Nielsen and Chuang, 2000). These distances are applicable since any positive semidefinite matrix can be scaled to be a valid quantum density matrix. In particular, we find the Bures distance (Bures, 1969; Uhlmann, 1976), which generalizes the

Hellinger distance to matrices (Nielsen and Chuang, 2000; Koltchinskii and Xia, 2015), to have useful properties. Unlike many distance or similarity functions that are based on the Hilbert-Schmidt inner product, the similarity function for the Bures distance (known as fidelity (Jozsa, 1994)) relies on the trace norm. The Bures distance is both contractive and locally Riemannian, and can be considered the best-suited distance metric for the space of quantum density matrices (Bromley et al., 2014).

For several choices of distance metrics or dissimilarity functions, the minimum distance to the set of non-informative matrices has a closed-form expression, but this is not the case for the Bures distance. Nonetheless, using the sub-additive property of the trace norm, we derive a closed-form expression of the lower bound which is tight in most cases. Interestingly, the resulting measure leads to a convex cost function that is inversely proportional to the informativeness of a correlation matrix in terms of its Euclidean-space embedding. Besides this unique property, we find the measures of informativeness based on the Bures distance, and the other quantum distances, to perform better in machine learning applications than those based on the Hilbert-Schmidt inner product or the existing test statistics for homogeneous correlation coefficients (Bartlett, 1954; Lawley, 1963).

We organize the exposition of the paper as follows: In Section 2 we introduce some preliminaries regarding positive semidefinite, correlation, and quantum density matrices. In Section 3 we introduce the general definition of informativeness and specific measure instances based on various distance and dissimilarity functions. The derivations are included in Appendix B. We conclude the section with an illustration of the various measures on patterned correlation matrices corresponding to different partitions. In Section 4 we introduce the problem of finding a maximally informative correlation matrix nearby a target matrix, a problem we refer to as correlation matrix denoising. We also propose first-order optimizations of informativeness for a sample's correlation matrix parameterized in terms of a kernel function applied to the sample coordinates. For correlation matrix denoising, we also parameterize the optimization in terms of Euclidean-space embeddings and detail the cost functions that arise from maximizing the informativeness of the embeddings. Moreover, we show that the Bures-based informativeness is inversely proportional to a specific matrix norm. Finally, using the norm's proximal operator, we propose an algorithm to solve a relaxation of the correlation denoising problem. In Section 5 we explore different applications of informativeness for machine learning algorithms including correlation matrix denoising. A discussion of the results and general conclusions are given in Section 6.

2. Preliminaries

For a positive integer m, we use $[m]$ to denote the set $\{1, \ldots, m\}$. We denote a scalar variable as x or X, a set as $\mathcal{X}$, a real-valued column vector as $\mathbf{x}$, a matrix as $\mathbf{X}$, and its jth column as $\mathbf{x}_j$. Vectorizing the diagonal of a matrix is denoted by $\operatorname{diag}(\mathbf{X})$, and $\operatorname{diag}(\mathbf{x})$ denotes a diagonal matrix with $[\operatorname{diag}(\mathbf{x})]_{i,i} = x_i$. Inner-product notation $\langle \cdot, \cdot \rangle$ is used for both vectors and matrices, according to $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_i x_i y_i$ and $\langle \mathbf{X}, \mathbf{Y} \rangle = \operatorname{tr}(\mathbf{X}^\top \mathbf{Y}) = \sum_{i,j} X_{i,j} Y_{i,j}$, respectively. The Euclidean norm of a vector is denoted by $\|\mathbf{x}\|_2$, and $\|\mathbf{X}\|_F^2 = \langle \mathbf{X}, \mathbf{X} \rangle$ denotes the square of the Frobenius norm, which is the sum of the squared entries of $\mathbf{X}$ and equals the sum of the squares of its singular values. We denote the trace norm (also called

the nuclear norm or Schatten 1-norm) by $\|\mathbf{X}\|_*$, defined as the sum of all the singular values of $\mathbf{X}$, and by $\|\mathbf{X}\|_2$ the spectral (operator) norm which is the largest singular value of $\mathbf{X}$.

2.1 Similarity, Correlation, and Embedding Matrices

We assume that we are provided with measurements of the similarity between n objects. This similarity corresponds to either a non-negative symmetric bivariate measure, for which larger positive values indicate similarity between objects and values near zero indicate weak relationships, or if the concept of negative correlation is applicable, a symmetric bivariate measure where negative values indicate inversely correlated objects. The similarity values are represented by an $n \times n$ real symmetric matrix $\mathbf{A}$, where $A_{i,j}$ denotes the similarity between objects i and j. Assuming that the similarities of all objects to themselves are greater than zero, the similarity matrix can be rescaled such that all the self-similarities are one. That is, any symmetric matrix $\tilde{\mathbf{A}}$ with $\tilde{A}_{i,i} > 0$ can be symmetrically normalized via

$$A_{i,j} = \frac{\tilde{A}_{i,j}}{\sqrt{\tilde{A}_{i,i}\tilde{A}_{j,j}}}, \tag{1}$$

such that $A_{i,i} = 1, i \in [n]$, or equivalently as $\mathbf{A} = \mathbf{D}\tilde{\mathbf{A}}\mathbf{D}$, where $\mathbf{D}$ is a diagonal matrix with entries $D_{i,i} = \frac{1}{\sqrt{\tilde{A}_{i,i}}}$. We assume that we have no further information about the objects or their original representations.

Many similarity measures, such as positive definite kernel functions, will yield a positive semidefinite similarity matrix. Positive semidefinite matrices can be represented by the embeddings of the objects in a Hilbert space for which the theory of reproducing kernel Hilbert spaces (Aronszajn, 1950) applies. That is, if a given similarity matrix is positive semidefinite, then there exists a set of n points in a Hilbert space $\mathcal{H}$, such that the inner product between any pair of points quantifies their similarity. Specifically, $A_{i,j} = \langle \mathbf{z}_i, \mathbf{z}_j \rangle$ where $\mathbf{z}_i \in \mathcal{H}$ with $i \in [n]$, denotes the point corresponding to the ith object. For finite dimensional embeddings in $\mathbb{R}^b$, we refer to the $b \times n$ matrix of concatenated vectors $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$ as an embedding. This is not unique as any orthogonal transformation in $\mathbb{R}^b$ yields an equally valid representation.

We restrict our attention to *correlation matrices*,[1] which are matrices that are both positive semidefinite and normalized such that the diagonal elements are equal to one. Applying the symmetric normalization from Equation 1 to a positive semidefinite similarity matrix yields a correlation matrix, and in fact, the symmetric normalization yields the closest correlation matrix in terms of the Bures distance, which we show in Lemma 2 in Appendix A. A correlation matrix can be represented by a Hilbert space embedding where each object is represented by a point on a unit sphere. The convex set of all the $n \times n$ correlation matrices is known as the elliptope and it is defined as $\mathcal{E} = \left\{\mathbf{X} \in \mathbb{R}^{n \times n} : \mathbf{X} = \mathbf{X}^\top \succcurlyeq 0, X_{i,i} = 1, \forall i \in [n]\right\}$. It contains 2^{n-1} vertices that have full dimensional normal cones and correspond to rank-1 cut matrices of the form $\mathbf{v}\mathbf{v}^\top$ with $\mathbf{v} \in \{\pm 1\}^n$ (Laurent and Poljak, 1995, 1996). The convex hull of the cut matrices defines the cut polytope, and a strict subset of correlation matrices are members of this polytope. This relationship between the elliptope and the cut polytope is exploited in semidefinite programming relaxations (Saunderson et al., 2012) of combinatorial problems associated with dividing objects, such as finding the maximum cut of a graph (Goemans and Williamson, 1995). The set of embeddings of correlation matrices is the oblique manifold $\mathcal{OB} = \left\{\mathbf{Z} \in \mathbb{R}^{b \times n} : \|\mathbf{z}_i\|_2 = 1, \forall i \in [n]\right\}$ (Trendafilov and Lippert, 2002; Absil and Gallivan, 2006), where any correlation matrix can be written as $\mathbf{Z}^\top\mathbf{Z}$. When $b = 1$, the oblique manifold is simply the set of vectors with elements ± 1. The elements of a correlation matrix necessarily lie within $[-1, +1]$ since from the Cauchy-Bunyakovsky-Schwarz inequality we have $|\langle \mathbf{z}_i, \mathbf{z}_j \rangle| \leq \|\mathbf{z}_i\| \cdot \|\mathbf{z}_j\|$.

1. Their name is due to the fact they coincide with the set of Pearson correlation matrices for multivariate samples.

When the underlying objects are real-valued vectors, similarity measures that yield correlation matrices include the cosine similarity; various correlation coefficients, such as Pearson's, Spearman's, or Kendall's; and positive definite shift-invariant kernel functions, such as the Laplacian or Gaussian kernels. Efficient positive semidefinite kernels, which can generally be normalized, have also been developed for non-vectorial objects, such as strings (Lodhi et al., 2002), point sets (Kondor and Jebara, 2003), and the univariate point sets known in neuroscience as spike trains (Park et al., 2013). Nonetheless, some similarity measures yield normalized but indefinite matrices. To obtain a correlation matrix, one approach is to apply a non-negative threshold to the eigenvalues by setting any negative eigenvalues to zero (Higham, 1988) and then apply the symmetric normalization. An alternative is to find the nearest, in terms of Euclidean distance, correlation matrix (Higham, 2002; Qi and Sun, 2006). Both of these approaches, however, may distort the measures of similarity. Another approach is to treat each row of the similarity matrix as a vector-space representation of each object and compute a second-order similarity matrix using cosine similarity between these vectors. The result will be a correlation matrix. A final possibility motivated by spectral graph theory (Chung, 1997), is to treat the similarity matrix as the weighted affinity matrix of a graph, and then compute the normalized graph Laplacian, which is always a correlation matrix with non-positive off-diagonal entries.

2.2 Density Matrices

Quantum density matrices represent the probabilities associated with the outcomes of measurements on quantum states (Nielsen and Chuang, 2000; Hayashi, 2006). Ignoring their physical interpretations, density matrices are trace-normalized positive semidefinite Hermitian matrices. This means that statistical tools developed to analyze quantum density matrices can be applied to positive semidefinite matrices, including correlation matrices, after rescaling them by their trace.

Quantum information theory is based on the fact that a quantum density matrix is itself a non-commutative generalization of a probability mass function (Nielsen and Chuang, 2000). The positive semidefiniteness and unit-trace restrictions ensure that a matrix's eigenvalues are strictly positive and sum to 1; these are the same properties that a probability mass function has. This allows extensions of information theoretic quantities, such as entropy and mutual information to be defined for density matrices, and subsequently, trace-normalized positive semidefinite matrices (Sanchez Giraldo et al., 2015).

Our motivation for introducing density matrices is to leverage quantum information-based distance measures, which have unique properties compared to more familiar measures. Any correlation matrix can be converted to a density matrix by dividing by n. The set of

rescaled correlation matrices forms a subset of symmetric density matrices with constant diagonals. In the next section, we introduce our distance-based framework for measures of informativeness that can use the quantum distance measures.

3. Defining and Measuring Informativeness

In this section, we propose a univariate measure of how informative a given correlation matrix is, based on its distance from the nearest non-informative correlation matrix. This distance can be measured using various distance/dissimilarity measures and for some of these measures we show that we can obtain closed-form expressions of informativeness.

3.1 Non-informative Correlation Matrices

A set of pairwise correlation coefficients is non-informative if it indicates that no pair of objects is any more similar than any other pair, i.e., there are no groups of more (or less) related objects. Two exemplary cases of non-informativeness correspond to

1. all objects being indistinguishable, and
2. all objects being distinct and equally dissimilar.

By utilizing correlation matrices, these cases can be represented by the constant matrix $\mathbf{J} = \mathbf{1}\mathbf{1}^\top$ and the identity matrix $\mathbf{I}$, respectively. In general, any correlation matrix with constant off-diagonal elements is non-informative. Reordering the objects, i.e., simultaneously permuting the rows and columns of the correlation matrix, will not change a non-informative matrix. The set of non-informative matrices forms the null hypothesis for tests of homogeneous correlation coefficients (Bartlett, 1954; Lawley, 1963). Like these test statistics, informativeness essentially measures how heterogeneous the off-diagonal entries of the correlation matrix are.

If we let ρ be the value of the off-diagonal elements, correlation matrices with constant off-diagonal elements can be expressed as $\mathbf{A}_\rho = \rho\mathbf{J} + (1-\rho)\mathbf{I}$, where the range of ρ must be restricted to ensure $\mathbf{A}_\rho$ is a valid $n \times n$ correlation matrix. Specifically, since $\mathbf{J}\mathbf{1} = n\mathbf{1}$, the spectrum of $\mathbf{A}_\rho$ is $\{\rho n + 1 - \rho, 1 - \rho\}$. For both eigenvalues to be non-negative, we must have $\frac{-1}{n-1} \leq \rho \leq 1$.

It it more convenient, however, to substitute ρ with a variable $a \in [0, 1]$, so that it parameterizes the expression of a non-informative matrix as a convex combination of two other non-informative matrices. Linearly mapping $\rho \in \left[\frac{-1}{n-1}, 1\right]$ to a, gives $a = \frac{\rho(n-1)+1}{n}$ or equivalently $\rho = \frac{an-1}{n-1}$. Substituting the latter within the expression for $\mathbf{A}_\rho$ gives $\frac{na-1}{n-1}\mathbf{J} + \frac{n(1-a)}{n-1}\mathbf{I}$, which is also equal to the expression $a\mathbf{J} + (1-a)\frac{n}{n-1}\mathbf{H}$, where $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{J}$ is the centering matrix. The parameter a is the mean of the non-informative matrix since $a = \frac{\mathbf{1}^\top \mathbf{A}_\rho \mathbf{1}}{n^2}$, whereas $\rho = \frac{\mathbf{1}^\top(\mathbf{A}_\rho - \mathbf{I})\mathbf{1}}{n(n-1)}$ is the mean of its off-diagonal elements.

In this work, we define the set of non-informative matrices to be

$$\mathcal{N} \equiv \left\{ \mathbf{N}_a = a\mathbf{J} + (1-a)\frac{n}{n-1}\mathbf{H} : 0 \leq a \leq 1 \right\}. \tag{2}$$

Under this parameterization, $\mathbf{N}_1 = \mathbf{J}$ is the rank-1 constant matrix, $\mathbf{N}_{\frac{1}{n}} = \mathbf{I}$ is the identity matrix, and $\mathbf{N}_0 = \frac{n}{n-1}\mathbf{H}$ is a scaled version of the centering matrix with rank $n-1$.

Geometrically, $\mathbf{J}$ is a vertex of the elliptope corresponding to the cut matrix $\mathbf{1}\mathbf{1}^\top$, $\mathbf{I}$ is the barycenter, $\frac{n}{n-1}\mathbf{H}$ is an extreme point with a one-dimensional normal cone, and all the non-informative matrices lie along the line connecting them.

From the characteristic polynomial of $\mathbf{N}_a$, it can be seen that it has one simple eigenvalue $\lambda_1 = an$ and one semisimple (due to the symmetry and, thus, diagonalizability of $\mathbf{N}_a$) eigenvalue $\lambda_2 = \frac{(1-a)n}{n-1}$ of multiplicity $n-1$. The corresponding eigenspaces $\mathrm{Null}(\mathbf{N}_a - \lambda_1\mathbf{I})$ and $\mathrm{Null}(\mathbf{N}_a - \lambda_2\mathbf{I})$ are orthogonal complementary subspaces and, since $(\lambda_1, \mathbf{1})$ is an eigenpair, they are common to the entire family of $\mathbf{N}_a$. Any vector within the eigenspace of λ_2 is an eigenvector. The existence of common eigenvectors is also supported by the fact that $\mathcal{N}$ is a commuting matrix family, which follows from the facts that $\mathbf{J}\mathbf{H} = \mathbf{0}$, and $\frac{1}{n}\mathbf{J}$ and $\mathbf{H}$ are symmetric and idempotent. Furthermore, for any given a, $\mathbf{N}_a$ can be expressed in terms of its eigenvalues as $\frac{\lambda_1}{n}\mathbf{J} + \lambda_2\mathbf{H}$, and since it is diagonalizable, for any function f defined at each λ_i, we have $f(\mathbf{N}_a) = \frac{f(\lambda_1)}{n}\mathbf{J} + f(\lambda_2)\mathbf{H}$. A particular instance of this, which we will use in the subsequent section, is the matrix square root given by $\sqrt{\mathbf{N}_a} = \sqrt{\frac{a}{n}}\mathbf{J} + \sqrt{\frac{(1-a)n}{n-1}}\mathbf{H}$.

3.2 Measuring the Distance Between Correlation Matrices

We propose to quantify the informativeness of a given correlation matrix $\mathbf{K}$ by its distance to the nearest non-informative correlation matrix. This measure of informativeness can be defined as

$$d_{\mathcal{N}}(\mathbf{K}) \equiv \min_{\mathbf{N}\in\mathcal{N}} d(\mathbf{K}, \mathbf{N}) = \min_{0\leq a\leq 1} d(\mathbf{K}, \mathbf{N}_a), \tag{3}$$

and can rely on various distance or dissimilarity measures applicable to matrices. To ensure informativeness is invariant to the ordering of the objects, the distance metric must be invariant to symmetric permutations of both arguments: $d(\mathbf{A}, \mathbf{B}) = d(\mathbf{\Pi}\mathbf{A}\mathbf{\Pi}^\top, \mathbf{\Pi}\mathbf{B}\mathbf{\Pi}^\top)$, where $\mathbf{\Pi}$ is a permutation matrix. For any non-informative matrix $\mathbf{N}$, we have $\mathbf{N} = \mathbf{\Pi}\mathbf{N}\mathbf{\Pi}^\top$. To ensure the informativeness measures can all be bounded between 0 and 1, we restrict our attention to bounded distance/dissimilarity measures.[2]

Table 1 contains distance metrics and dissimilarity functions—along with any underlying similarity measure—that are applicable to positive semidefinite matrices and can be used in the definition of $d_{\mathcal{N}}$. The table includes measures that have been widely applied in kernel-based machine learning, such as kernel alignment (Cristianini et al., 2002), which corresponds to the cosine similarity between the vectorized forms of matrices, the Hilbert-Schmidt independence criterion (Gretton et al., 2005), and the centered kernel alignment (Cortes et al., 2012), also known as distance correlation (Székely et al., 2007; Sejdinovic et al., 2013). The table also includes distances/dissimilarities used in quantum information theory, such as the trace distance, which is a matrix version of the Kolmogorov or total variation distance (Fuchs and van de Graaf, 1999); the quantum Jensen-Shannon divergence (Majtey et al., 2005); a distance based on the quantum Chernoff bound (Audenaert et al.,

2. This restriction excludes the family of log-determinant divergences (Lee and Lim, 2008; Cherian et al., 2011; Chebbi and Moakher, 2012; Cichocki et al., 2015) defined on the set of symmetric positive definite matrices, e.g., the affine invariant Reimannian metric (AIRM) and other symmetric log-det divergences. In preliminary experiments, we found these metrics to have unique behavior; however, they do not perform consistently as measures of informativeness, primarily because they are unbounded and the input matrix needs to be regularized to ensure it is strictly positive definite.

Table 1: Possible distance metrics/dissimilarity functions for the realization of $d_{\mathcal{N}}(\cdot)$ in Equation 3. The first five are applicable to any conformable symmetric matrices $\mathbf{A}$ and $\mathbf{B}$, whereas the second five additionally assume positive semidefiniteness and unity trace.

Distance metric/dissimilarity function $d(\mathbf{A},\mathbf{B})$:	Underlying similarity function:
Euclidean distance $\|\mathbf{A}-\mathbf{B}\|_F$	Matrix dot-product $\langle \mathbf{A},\mathbf{B}\rangle$
Cosine distance $\sqrt{2-2\frac{\langle \mathbf{A},\mathbf{B}\rangle}{\|\mathbf{A}\|_F\,\|\mathbf{B}\|_F}}$	Cosine similarity, or kernel alignment $\frac{\langle \mathbf{A},\mathbf{B}\rangle}{\|\mathbf{A}\|_F\,\|\mathbf{B}\|_F}$
Hilbert-Schmidt independence criterion (HSIC) $\|\mathbf{HAH}-\mathbf{HBH}\|_F$	$\langle \mathbf{HAH},\mathbf{HBH}\rangle$
Centered kernel alignment (CKA) $\sqrt{2-2\frac{\langle \mathbf{HAH},\mathbf{HBH}\rangle}{\|\mathbf{HAH}\|_F\,\|\mathbf{HBH}\|_F}}$	$\frac{\langle \mathbf{HAH},\mathbf{HBH}\rangle}{\|\mathbf{HAH}\|_F\,\|\mathbf{HBH}\|_F}$
Trace distance $\frac{1}{2}\mathrm{tr}\sqrt{(\mathbf{A}-\mathbf{B})^2}=\frac{1}{2}\|\mathbf{A}-\mathbf{B}\|_*$	
Quantum Jensen-Shannon (QJS) divergence $H(\frac{\mathbf{A}+\mathbf{B}}{2})-\frac{1}{2}H(\mathbf{A})-\frac{1}{2}H(\mathbf{B})$, where $H(\mathbf{A})\triangleq-\langle \mathbf{A},\ln\mathbf{A}\rangle$ is the von Neumann entropy	
Chernoff bound $\sqrt{2-2\min_{0\le s\le 1}\langle \mathbf{A}^s,\mathbf{B}^{1-s}\rangle}$	$\min_{0\le s\le 1}\langle \mathbf{A}^s,\mathbf{B}^{1-s}\rangle$
Quantum Hellinger (QH) distance $\|\sqrt{\mathbf{A}}-\sqrt{\mathbf{B}}\|_F=\sqrt{2-2\langle\sqrt{\mathbf{A}},\sqrt{\mathbf{B}}\rangle}$	Affinity $\langle\sqrt{\mathbf{A}},\sqrt{\mathbf{B}}\rangle$
Bures distance (d_B) $\sqrt{2-2\|\sqrt{\mathbf{A}}\sqrt{\mathbf{B}}\|_*}$	Fidelity $\|\sqrt{\mathbf{A}}\sqrt{\mathbf{B}}\|_*^2$
Sub-Bures dissimilarity $\sqrt{2-2\sqrt{\langle \mathbf{A},\mathbf{B}\rangle+\sqrt{(1-\|\mathbf{A}\|_F^2)(1-\|\mathbf{B}\|_F^2)}}}$	Super-fidelity $\langle \mathbf{A},\mathbf{B}\rangle+\sqrt{(1-\|\mathbf{A}\|_F^2)(1-\|\mathbf{B}\|_F^2)}$

2007); the quantum Hellinger divergence (Luo and Zhang, 2004); the Bures distance (Bures, 1969; Uhlmann, 1976; Braunstein and Caves, 1994); and the sub-Bures dissimilarity[3] measure, which is a lower bound on the Bures distance (Mendonça et al., 2008; Miszczak et al., 2009).

Of the quantum information theoretic distances, the Bures distance can be considered the natural metric between trace-normalized positive semidefinite matrices (Bromley et al., 2014). It is contractive, locally Riemannian, invariant to the choice of embedding coordinates (Uhlmann, 1976), and in its infinitesimal form it coincides with the quantum Fisher information metric for quantum densities (Braunstein and Caves, 1994). Various elements

3. The corresponding similarity measure, super-fidelity, can be used to define a metric distance, but it is not a lower bound on the Bures distance. The dissimilarity measure violates the triangle inequality for certain cases (Mendonça et al., 2008).

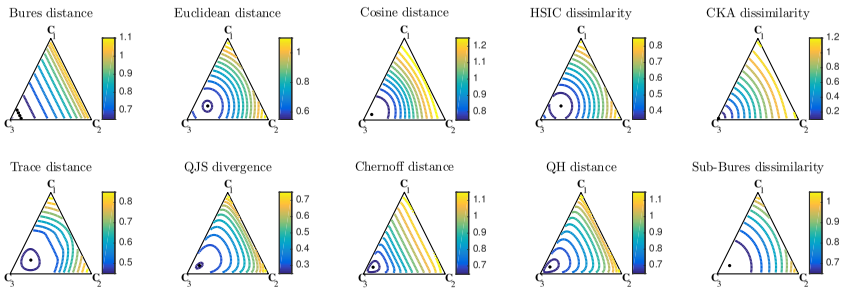


Figure 2: Equidistant contour lines for the measures in the marginal simplex of cut matrices. The Bures distance is the only distance invariant to the relative proportion of $\mathbf{C}_1$ and $\mathbf{C}_2$, which are not in the convex basis of the target matrix.

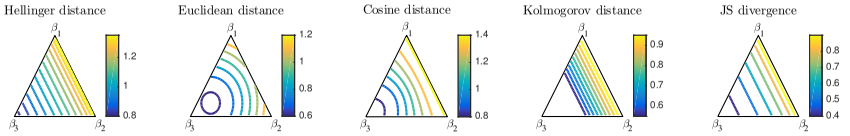


Figure 3: Equidistant contour lines for distances between points on the marginal simplex defined by $[\beta_1,\beta_2,\beta_3,0]$ to the target point $[0,0,0.5,0.5]$. The shape of the contours illustrates that most measures, besides the Euclidean and cosine distances, are invariant to the relative proportion of β_1 and β_2.

of the formulation and computation of the Bures distance and the associated fidelity similarity measure, which are relevant to the study of correlation matrices and informativeness, are presented in Appendix A.

Example 1 (Comparison of Distance/Dissimilarity Measures) *We compare the distance measures from Table 1 on correlation matrices with* $n=3$.

In particular, we consider the set of correlation matrices $\mathbf{X}_{\boldsymbol{\beta}}=\beta_1\mathbf{C}_1+\beta_2\mathbf{C}_2+\beta_3\mathbf{C}_3+\beta_4\mathbf{J}$ that can be represented as the convex combination of the elliptope vertices $\mathbf{C}_i=\mathbf{v}_i\mathbf{v}_i^\top$ and $\mathbf{J}=\mathbf{1}\mathbf{1}^\top$, with $\mathbf{v}_1=[-1,1,1]^\top$, $\mathbf{v}_2=[1,-1,1]^\top$ and $\mathbf{v}_3=[1,1,-1]^\top$. Since $\sum_i\beta_i=1$, each 3×3 matrix $\mathbf{X}_{\boldsymbol{\beta}}$ corresponds to a point of a 3-simplex. For visualization purposes, we calculate the distance between sets of correlation matrices $\mathbf{X}_{\boldsymbol{\beta}}$ and a target matrix $\mathbf{A}$. As an example, we set $\mathbf{A}$ to have as its convex basis (Brøndsted, 2012) the vertices $\mathbf{C}_3$ and $\mathbf{J}$, according to $\mathbf{A}=0.5\,\mathbf{C}_3+0.5\,\mathbf{J}$, and restrict $\mathbf{X}_{\boldsymbol{\beta}}$ to lie on one face of the tetrahedron by fixing $\beta_4=0$. Figure 2 shows the equidistant contours for various measures.

The key distinction is how convex combinations of $\mathbf{C}_1$ and $\mathbf{C}_2$ affect the calculated distances $d(\mathbf{X}_{\boldsymbol{\beta}},\mathbf{A})$. We can observe that the Bures distance is only a function of β_3, or

equivalently of $\beta_1 + \beta_2$. That is, the Bures distance is invariant to the relative proportion of β_1 and β_2, since along lines where β_3 is fixed, every correlation matrix is equidistant from the target matrix $\mathbf{A}$. This invariance is desirable since neither $\mathbf{C}_1$ nor $\mathbf{C}_2$ is in the convex basis of the $\mathbf{A}$. An equivalent invariance occurs when the corresponding vector-based distances[4] are computed between the vector of coefficients $\boldsymbol{\beta} = [\beta_1, \beta_2, \beta_3, 0]$, which defines a discrete probability distribution, and the target $[0, 0, 0.5, 0.5]$. Figure 3 shows the equidistant contours for various measures. Except for the Euclidean and cosine distances, the distance metrics are invariant to the relative proportion of β_1 and β_2, since these coefficients are not in the support of the target. In comparison, this behavior is not exhibited by any of the other matrix measures as evidenced by the nonlinear contours. For example, for fixed β_3 values, the Euclidean distance yields shorter distances to $\mathbf{A}$ when $\beta_1 = \beta_2$. The same applies to the other measures, although the effect is less obvious for the Chernoff-based measure, which behaves similar to the Bures distance for larger distances, but for shorter distances it is also susceptible to the same issue.

3.3 Derivation of Measures of Informativeness

We proceed to analytically derive closed-form expressions of informativeness for a subset of the distance/dissimilarity measures from Table 1. In the derivations we use the parametric form of the nearest non-informative matrix $\mathbf{N}_{a^\star}$, where $a^\star = \arg\min_{0\le a\le 1} d(\mathbf{K}, \mathbf{N}_a)$ and $\mathbf{K} \in \mathcal{E}$ is the input correlation matrix. Given the non-informative matrix $\mathbf{N}_{a^\star}$, we define the informativeness measures as $d_{\mathcal{N}}(\mathbf{K}) = d(\mathbf{K}, \mathbf{N}_{a^\star})$ or $\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = \frac{1}{2}d^2(\mathbf{K}, \mathbf{N}_{a^\star})$ for chordal distances defined on a unit sphere whose range is intrinsically $[0, \sqrt{2}]$; this ensures the informativeness measures all range between 0 and 1. The analytic expressions for $a^\star$ and $d_{\mathcal{N}}(\mathbf{K})$ are derived in Appendix B, and the expressions of informativeness are in Table 2.

These expressions assume the input matrix is a correlation matrix $\mathbf{K} \in \mathcal{E}$. If $\mathbf{K} \notin \mathcal{E}$ then the Euclidean, cosine, HSIC, and CKA measures compute the distance/dissimilarity between $\mathbf{K}$ and the set of matrices with a diagonal of all ones and constant off-diagonal elements, i.e., the set $\{\mathbf{N}_a\}$, where a is unconstrained. In this case, the nearest matrix $\mathbf{N}_{a^\star}$ is not necessarily a correlation matrix, e.g., $a^\star < 0$ or $a^\star > 1$.

We note that for the Chernoff bound the expression for informativeness does not have a closed form, but only requires the same computational complexity as a single evaluation of the Chernoff bound. Also, the sub-Bures informativeness is not strictly closed form, requiring the maximum of two algebraic expressions. We did not find a closed-form expression of informativeness for neither the trace distance nor the quantum Jensen-Shannon divergence. Consequently, computing these measures requires a search over the family of non-informative matrices, which is costly since each search iteration requires computing either the singular values or the eigenvalues of an $n \times n$ matrix.

Each measure in Table 2 is bounded between 0 and 1, and will necessarily reach the lower bound of 0 for non-informative matrices. However, the upper limit of 1 is a loose upper bound and may not be attainable. Because of this and the fact there are no units associated with informativeness;[5] the nominal value of informativeness is not meaningful for

4. The Hellinger distance is the vector equivalent of the Bures and QH distances, and the Kolmogorov distance is the vector equivalent of the trace distance.
5. In comparison, the units of entropy (bit, nat, or hartley) are gauged to the probability of an event.

Table 2: Informativeness measures based on the distance and dissimilarity measures from Table 1. All measures, apart from the Bures-based one, are derived in Appendix B. $\mathbf{K} \in \mathcal{E}$ is a correlation matrix and $\bar{k} = \frac{1}{n^2}\mathbf{1}^\top\mathbf{K}\mathbf{1}$ is the average of its entries. All measures are bounded within the range $[0, 1]$.

Measure:	Informativeness:	Simplified expression (=), or bound (≥):	Relevant equations:
Euclidean	$d_{\mathcal{N}}(\mathbf{K}) = \min_{0\le a\le 1} \frac{1}{n}\|\mathbf{K} - \mathbf{N}_a\|_F$	$= \sqrt{\frac{1}{n^2}\|\mathbf{K}\|_F^2 - \frac{n\bar{k}^2 - 2\bar{k} + 1}{n-1}}$	(B.1),(B.2)
Cosine	$\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = 1 - \max_{0\le a\le 1} \frac{\langle \mathbf{K}, \mathbf{N}_a\rangle}{\|\mathbf{K}\|_F\|\mathbf{N}_a\|_F}$	$= 1 - \frac{n}{\|\mathbf{K}\|_F}\sqrt{\frac{n\bar{k}^2 - 2\bar{k} + 1}{n-1}}$	(B.3),(B.4)
HSIC	$d_{\mathcal{N}}(\mathbf{K}) = \min_{0\le a\le 1} \frac{1}{n}\|\mathbf{H}\mathbf{K}\mathbf{H} - \mathbf{H}\mathbf{N}_a\mathbf{H}\|_F$	$= \sqrt{\frac{1}{n^2}\|\mathbf{H}\mathbf{K}\mathbf{H}\|_F^2 - \frac{(1-\bar{k})^2}{n-1}}$	(B.5),(B.6)
CKA	$\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = 1 - \max_{0\le a\le 1} \frac{\langle \mathbf{H}\mathbf{K}\mathbf{H}, \mathbf{H}\mathbf{N}_a\mathbf{H}\rangle}{\|\mathbf{H}\mathbf{K}\mathbf{H}\|_F\|\mathbf{H}\mathbf{N}_a\mathbf{H}\|_F}$	$= 1 - \frac{n(1-\bar{k})}{\|\mathbf{H}\mathbf{K}\mathbf{H}\|_F\sqrt{n-1}}$	(B.7)
Chernoff	$d_{\mathcal{N}}(\mathbf{K}) = 1 - \max_{0\le a\le 1} \min_{0\le s\le 1} \frac{1}{n}\langle \mathbf{K}^s, \mathbf{N}_a^{1-s}\rangle$	$= 1 - q(\mathbf{K})$	(B.8–B.11)
QH	$\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = 1 - \max_{0\le a\le 1} \frac{1}{n}\langle \sqrt{\mathbf{K}}, \sqrt{\mathbf{N}_a}\rangle$	$= 1 - \sqrt{\frac{(\mathbf{1}^\top\sqrt{\mathbf{K}}\mathbf{1})^2}{n^3} + \frac{\mathrm{tr}^2(\mathbf{H}\sqrt{\mathbf{K}})}{n(n-1)}}$	(B.12),(B.13)
Bures	$\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = 1 - \max_{0\le a\le 1} \frac{1}{n}\|\sqrt{\mathbf{K}}\sqrt{\mathbf{N}_a}\|_*$	$\ge i(\mathbf{K}) = 1 - \sqrt{\bar{k} + \frac{\mathrm{tr}^2\sqrt{\mathbf{H}\mathbf{K}\mathbf{H}}}{n(n-1)}}$	(B.16),(B.17)
Sub-Bures	$\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = 1 - \max_{0\le a\le 1} \sqrt{G(\frac{1}{n}\mathbf{K}, \frac{1}{n}\mathbf{N}_a)}$	$= 1 - \sqrt{g(\mathbf{K})}$	(B.21–B.24)

comparing differently-sized correlation matrices; nonetheless, informativeness can be used for relative comparisons between correlation matrices of equal size, even if they are closest to different non-informative matrices.

For comparison, the test statistics for equality of correlation coefficients proposed by Bartlett (1954) and Lawley (1963) are included in Appendix C. The null hypothesis for these tests is that the multivariate sample, corresponding to the observed correlation matrix, is drawn from a distribution whose true correlation structure is a non-informative matrix. The main benefit of using Lawley's test statistic is that, asymptotically, its distribution under the null hypothesis has a χ^2 distribution regardless of which particular non-informative matrix corresponds to the true correlation matrix. The null distribution for the other measures (including Bartlett's) depends on the underlying non-informative matrix, which is in practice unknown.

Example 2 (Informativeness for 3×3 Correlation Matrices) *To gain a perspective of informativeness measures, we evaluate them for 3×3 correlations matrices.*

Since the Bures-based informativeness is a convex function (Theorem 4 in Appendix B.7), it is necessarily maximized at the elliptope's extreme points, which are correlation matrices with rank-r where $r(r+1) \le 2n$ (Li and Tam, 1994). Figure 4 shows the Bures informativeness $i(\mathbf{K})$ across the parametric surface representing the surface of the elliptope. The measure is maximized at the three rank-1 correlation matrices besides $\mathbf{J}$. In general, the Bures-based informativeness is maximal for centered, rank-1 correlation matrices, as noted in Theorem 5 in Appendix B.7.

To compare the informativeness measures, we consider the subset of 3×3 correlation matrices $\mathbf{X}_{\boldsymbol{\beta}} = \beta_1\mathbf{C}_1 + \beta_2\mathbf{J} + \beta_3\frac{\mathbf{C}_2+\mathbf{C}_3}{2}$ with $\sum_i \beta_i = 1$, which are convex combinations

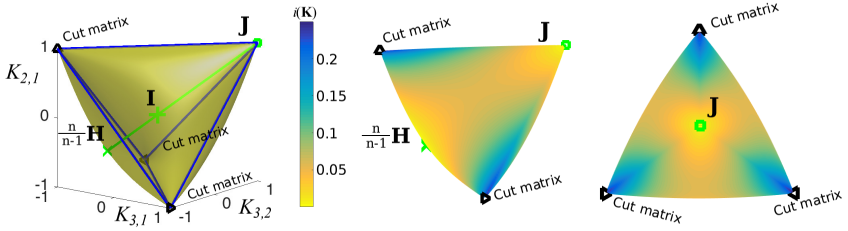


Figure 4: The Bures-based informativeness measure evaluated on the boundary of the elliptope. For $n = 3$, each correlation matrix $\mathbf{K}$ can be represented by a three-dimensional point that defines the three lower-triangular elements $K_{2,1}$, $K_{3,1}$, and $K_{3,2}$. (Left:) The family of non-informative matrices $\mathcal{N}$ is shown as a line across the elliptope passing through $\mathbf{N}_1 = \mathbf{J}$ and $\mathbf{N}_{1/n} = \mathbf{I}$ (discussed in Section 3.1). The outline of the cut polytope is shown as the wireframe connecting the four vertices. (Center, Right:) Informativeness peaks at the three extremal points corresponding to the three cut matrices.

of the elliptope vertices $\mathbf{C}_i = \mathbf{v}_i\mathbf{v}_i^\top$ and $\mathbf{J} = \mathbf{1}\mathbf{1}^\top$, with $\mathbf{v}_1 = [-1, 1, 1]^\top$, $\mathbf{v}_2 = [1, -1, 1]^\top$ and $\mathbf{v}_3 = [1, 1, -1]^\top$. Figure 5 illustrates the measures, along with Bartlett's and Lawley's test statistics, evaluated for each $\mathbf{X}_\beta$ across the 2-simplex defined by $\boldsymbol{\beta}$. All measures are maximized when $\mathbf{X}_\beta = \mathbf{C}_1$, but the shape of the level sets highlights the differences among the distance/dissimilarity measures underlying the informativeness measures.

Example 3 (Patterned Correlation Matrices) *We use* 16×16 *rank-1 and block diagonal correlation matrices to compare the measures of informativeness.*

The value of the informativeness measures for different partition sizes are shown in Figure 6. For binary partitions, the HSIC and Bures measures are uniquely maximized by balanced rank-1 matrices, and the measures—excepting the Euclidean, trace, and CKA measures—are maximized by balanced block-diagonal when there are two blocks. In general, balanced rank-1 matrices have higher informativeness than any of the block diagonal matrices (except for the CKA measure). For block-diagonal matrices, the Euclidean, HSIC, CKA, and sub-Bures measures are maximal for two equal-sized blocks, but the other measures (excepting the trace distance) are maximized when there are $\sqrt{n}$ equal-sized blocks. This latter disparity means that these two groups of measures will behave differently when evaluating correlation matrices with cluster structure.

On these patterned correlation matrices, we see that the Bures-based measure is a tight bound on the exact Bures informativeness measure. Additionally, the quantum Hellinger-based measure is a tight upper bound on the exact Bures measure except in the rank-1 case. The cosine-based measure also tracks the QH measure on these patterned matrices. The sub-Bures bound appears to be quite loose, except in the rank-1 case for which it is equal to the exact Bures measure.

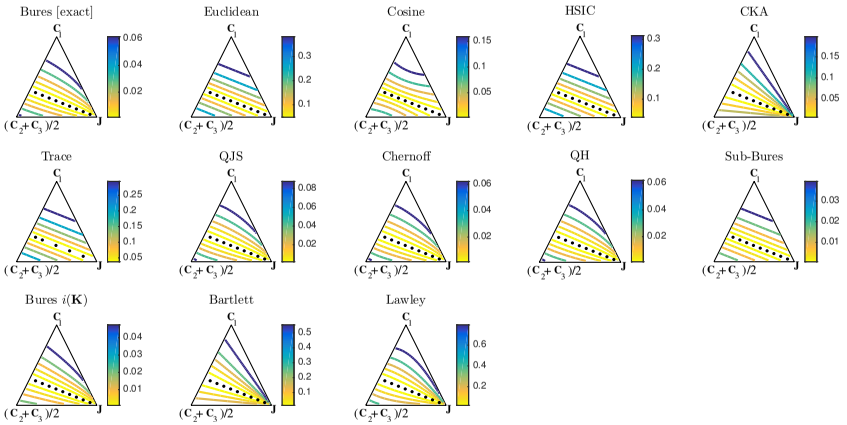


Figure 5: Equidistant contour lines for the measures on a 2-simplex of correlation matrices. The dotted line represents the non-informative matrix family $\mathcal{N}$. Informativeness is maximal at extremal points of the elliptope for convex distance metrics, but the choice of distance/dissimilarity measure affects the geometry of the level sets.

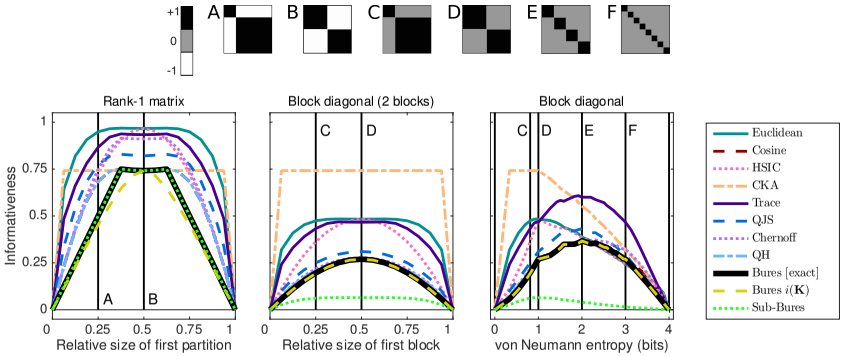


Figure 6: Comparison of informativeness measures for 16×16 correlation matrices with partition structure. Examples include rank-1 matrices (A, B); block diagonal matrices with two blocks (C, D); and block diagonal with increasing number of blocks (D, E, F) as indicated by the von Neumann entropy. (Note: The curves for the QH and cosine measures overlap on all plots, and the Bures curve covers the QH curve in the block diagonal plots.)

4. Optimization of Informativeness

In this section, we investigate constrained optimization problems based on finding a maximally informative correlation matrix close to a target matrix. We cover two types of cases; those optimizations parameterized in terms of a correlation matrix, and those in terms of a Euclidean embedding. In the latter case, we derive the analytic forms of cost functions for some measures of informativeness that can be applied directly to an embedding without requiring the explicit calculation of a correlation matrix. In particular, we show that when applied to an embedding, the Bures-based cost function is convex and corresponds to a novel matrix norm, which is a combination of trace seminorms on orthogonal subspaces. As this norm is non-differentiable, we describe its proximal operator, which requires the definition of the proximal operator for the squared trace norm. Using the Bures-based measure of informativeness as a cost function, we detail the optimization problem of finding a maximally informative correlation matrix nearby a target matrix, where nearness is assessed via the Euclidean distance between the embeddings. Although constraining the embedding to ensure it corresponds to a correlation matrix is non-convex, we relax this constraint to yield a completely convex optimization problem, the solutions of which can satisfy the original constraint. We propose an alternating direction method of multiplier (ADMM) algorithm that uses the proximal operator for the Bures-based cost function to solve this problem.

4.1 Maximally Informative Correlation Matrices

Firstly, we consider finding a maximally informative matrix that lies within an ϵ-radius ball of an arbitrary target matrix $\mathbf{A}$. This optimization can be written as

$$\begin{aligned} &\underset{\mathbf{K}\in\mathcal{E}}{\arg\max}\ d_{\mathcal{N}}(\mathbf{K}) \\ &\text{s.t. } \|\mathbf{A}-\mathbf{K}\|_F \le \epsilon, \end{aligned} \tag{4}$$

where $d_{\mathcal{N}}(\mathbf{K})$ is the distance to closest member of the family of $\mathbf{N}_a$ of non-informative matrices (or a lower bound on this distance in the case of the Bures distance). As noted in Theorem 4, the Bures-based informativeness is a convex function, as are the Euclidean and HSIC-based measures (their convexity can be verified by the properties of the Frobenius norm). Consequently, for these measures maximizing the informativeness is a convex maximization problem with multiple optima necessarily occurring at the extreme points of the elliptope, as evidenced by Figures 4. In particular, as stated in Theorem 5 in Appendix B.7, the Bures-based informativeness is maximized by the vertices of the elliptope corresponding to centered, rank-1 correlation matrices. Although there is multiple maxima, for appropriate choices of ϵ there may be a unique maximum within the local neighborhood.

For convenience, we instead define a minimization problem—replacing the ϵ-ball constraint with a penalty term scaled by ρ (where a large ρ corresponds to a small ϵ)—as

$$\underset{\mathbf{K}\in\mathcal{E}}{\arg\min}\ \rho\,\|\mathbf{A}-\mathbf{K}\|_F^2 + F\left(d_{\mathcal{N}}(\mathbf{K})\right), \tag{5}$$

where F is a monotonically decreasing function of the form $F : d \mapsto 1-\nu d^2$ with $\nu \in \{\frac{1}{2}, 1\}$ ($\nu = \frac{1}{2}$ for chordal distances in order to restrict the range of F to $[0,1]$). The

Table 3: Expressions for finding the most informative correlation matrices via cost functions of the form $J_\rho(\mathbf{K}) = \rho\,\|\mathbf{A}-\mathbf{K}\|_F^2 + F\left(d_{\mathcal{N}}(\mathbf{K})\right)$. The forms are applicable to all positive semidefinite measures, even though informativeness is only defined for correlation matrices. Because of this generality, the Euclidean and cosine measures are written in terms of $\mathbf{N}_{a^\star} = a^\star\mathbf{J} + (1-a^\star)\frac{n}{n-1}\mathbf{H}$.

Measure:	$F(d)$:	$F\left(d_{\mathcal{N}}(\mathbf{K})\right)$:	$J_\rho(\mathbf{K})$ convex?
Euclidean	$1-d^2$	$1-\frac{1}{n^2}\|\mathbf{K}-\mathbf{N}_{a^\star}\|_F^2, \quad a^\star = \frac{1}{n^2}\langle\mathbf{K},\mathbf{J}-\mathbf{I}\rangle + \frac{1}{n}$	$\rho \ge \frac{1}{n^2}$
Cosine	$1-\frac{1}{2}d^2$	$\frac{\langle\mathbf{K},\mathbf{N}_{a^\star}\rangle}{\|\mathbf{K}\|_F\|\mathbf{N}_{a^\star}\|_F}, \quad a^\star = \frac{1}{n}\frac{\langle\mathbf{K},\mathbf{J}\rangle}{\langle\mathbf{K},\mathbf{I}\rangle}$	—
HSIC	$1-d^2$	$1+\frac{1}{n^2(n-1)}\langle\mathbf{K},\mathbf{H}\rangle^2 - \frac{1}{n^2}\|\mathbf{HKH}\|_F^2$	$\rho \ge \frac{1}{n^2}$
CKA	$1-\frac{1}{2}d^2$	$\frac{\langle\mathbf{K},\mathbf{H}\rangle}{\|\mathbf{HKH}\|_F\sqrt{n-1}}$	—
QH	$1-\frac{1}{2}d^2$	$\sqrt{\frac{1}{n^3}\langle\sqrt{\mathbf{K}},\mathbf{J}\rangle^2 + \frac{1}{n(n-1)}\langle\sqrt{\mathbf{K}},\mathbf{H}\rangle^2}$	no
Bures-based	$1-\frac{1}{2}d^2$	$\sqrt{\frac{1}{n^2}\langle\mathbf{K},\mathbf{J}\rangle + \frac{1}{n(n-1)}\|\sqrt{\mathbf{K}}\mathbf{H}\|_*^2}$	no

function $F\left(d_{\mathcal{N}}(\mathbf{K})\right)$ simplifies for the measures of informativeness in Table 2 with closed-form expressions. The choice of F and the resulting expressions of $F\left(d_{\mathcal{N}}(\mathbf{K})\right)$ are listed in Table 3.

Since $1-\nu d^2$ is concave and nonincreasing with respect to d^2, when $d_{\mathcal{N}}^2(\mathbf{K})$ is convex, $\rho\,\|\mathbf{A}-\mathbf{K}\|_F^2 + F\left(d_{\mathcal{N}}(\mathbf{K})\right)$ is a difference of convex functions. By inspection we see that the cost functions for the Euclidean and HSIC-based measures of informativeness are quadratic functions of $\mathbf{K}$ and convex[6] when $\rho \ge \frac{1}{n^2}$, i.e., this choice of ρ corresponds to a sufficiently small ϵ-radius ball, such that it contains a unique maximally informative correlation matrix. In contrast, the convexity of the cost function for the cosine and CKA-based measures depends on both the target matrix $\mathbf{A}$ and the regularization ρ; in certain cases the cost function may be convex. Due to the matrix square root, neither the QH nor the Bures-based measures are convex. In particular, $\frac{1}{n^2}\langle\mathbf{K},\mathbf{J}\rangle + \frac{1}{n(n-1)}\|\sqrt{\mathbf{K}}\mathbf{H}\|_*^2$ is concave, as $\|\sqrt{\mathbf{K}}\mathbf{H}\|_*^2$ is proportional to the squared fidelity between $\frac{1}{n-1}\mathbf{H}$ and $\frac{1}{\mathrm{tr}\mathbf{K}}\mathbf{K}$ and squared fidelity is a concave function (Uhlmann, 1976) as noted in Theorem 3 in Appendix A. The function $\langle\sqrt{\mathbf{K}},\mathbf{C}\rangle$ is also concave for any positive semidefinite $\mathbf{C}$, as the function $-\langle\sqrt{\mathbf{K}},\mathbf{C}\rangle$ is convex (Borwein and Lewis, 2010, Section 3.1, Exercise 25c).

First-order gradient approaches may be applied to the expressions in Table 3 with some adjustments to ensure they are well-defined and smooth. Firstly, the cost functions corresponding to chordal distances (cosine, CKA, QH, and Bures) are squared to remove the scalar square roots from the expressions. Secondly, since the CKA measure is undefined at $\mathbf{K} = \mathbf{J}$, a log barrier of the form $-\log(1-\bar{k})$ should be added to the cost function to prevent this case. Thirdly, for the QH measure to be differentiable, $\mathbf{K}$ must be strictly

6. Letting $\mathbf{x} = \mathrm{vec}(\mathbf{K})$, these two cost functions can be written as $\mathbf{x}^\top\mathbf{M}\mathbf{x} + \mathbf{x}^\top\mathbf{u} + c$, which is a convex function if $\mathbf{M}$ is positive semidefinite. For the Euclidean measure, $\mathbf{M} = (\rho - \frac{1}{n^2})\mathbf{I} + \frac{1}{n^3(n-1)}\mathbf{v}\mathbf{v}^\top$ where $\mathbf{v} = \mathrm{vec}(\mathbf{J}-\mathbf{I})$. For the HSIC measure, $\mathbf{M} = \rho\mathbf{I} - \frac{1}{n^2}\mathbf{H}\otimes\mathbf{H}$, where $\otimes$ indicates the Kronecker product. In both cases, $\mathbf{M}$ is positive semidefinite if $\rho \ge \frac{1}{n^2}$.

Table 4: Gradients of cost functions derived from the closed-form informativeness measures with respect to $\mathbf{K}$, where $\mathbf{K} \in \mathcal{E}$ is a correlation matrix and $\bar{k} = \frac{1}{n^2}\mathbf{1}^\top\mathbf{K}\mathbf{1}$ is the average of its entries. The gradient for the QH measure is only valid if $\mathbf{K}$ is positive definite, which ensures $\sqrt{\mathbf{K}}$ and $-\sqrt{\mathbf{K}}$ have disjoint eigenvalues—guaranteeing a unique solution to the Sylvester equations (Bartels and Stewart, 1972).

Measure:	$\tilde{F}(\mathbf{K})$:	$\nabla_{\mathbf{K}}\tilde{F}(\mathbf{K})$:
Euclidean	$\frac{1}{2}\left(1 - d_{\mathcal{N}}(\mathbf{K})^2\right)$	$\frac{1}{n^2}\left(\frac{n\bar{k}-1}{n-1}\mathbf{J} - \mathbf{K}\right)$
Cosine	$\frac{1}{2}\left(1 - \frac{1}{2}d_{\mathcal{N}}(\mathbf{K})^2\right)^2$	$\frac{1}{\|\mathbf{K}\|_F^2}\left(\frac{n\bar{k}-1}{n^2}\mathbf{J} - \frac{n\bar{k}^2-2\bar{k}+1}{\|\mathbf{K}\|_F^2}\mathbf{K}\right)$
HSIC	$\frac{1}{2}\left(1 - d_{\mathcal{N}}(\mathbf{K})^2\right)$	$\frac{1}{n^2}\left(-\mathbf{HKH} - \frac{1-\bar{k}}{n-1}\mathbf{J}\right)$
CKA	$\frac{1}{2}\left(1 - \frac{1}{2}d_{\mathcal{N}}(\mathbf{K})^2\right)^2$	$\frac{\langle\mathbf{K},\mathbf{H}\rangle}{\|\mathbf{HKH}\|_F^2\sqrt{n-1}}\left(\mathbf{H} - \frac{\langle\mathbf{K},\mathbf{H}\rangle}{\|\mathbf{HKH}\|_F^2}\mathbf{HKH}\right)$
QH	$\frac{1}{2}\left(1 - \frac{1}{2}d_{\mathcal{N}}(\mathbf{K})^2\right)^2$	$\frac{\langle\sqrt{\mathbf{K}},\mathbf{J}\rangle}{n^3}\mathbf{S}_1 + \frac{\langle\sqrt{\mathbf{K}},\mathbf{H}\rangle}{n(n-1)}\mathbf{S}_2$, where $\sqrt{\mathbf{K}}\mathbf{S}_1 + \mathbf{S}_1\sqrt{\mathbf{K}} = \mathbf{J}, \quad \sqrt{\mathbf{K}}\mathbf{S}_2 + \mathbf{S}_2\sqrt{\mathbf{K}} = \mathbf{H}$
Bures-based	$\left(1 - \frac{1}{2}\tilde{d}_{\mathcal{N}}(\mathbf{K})^2\right)^2$	$\frac{1}{n^2}\mathbf{J} + \frac{S_\gamma(\mathbf{HKH})}{n(n-1)}\mathbf{S}'_\gamma(\mathbf{HKH})$
	$\frac{1}{2}\tilde{d}_{\mathcal{N}}(\mathbf{K})^2 = 1 - \sqrt{\frac{1}{n^2}\langle\mathbf{K},\mathbf{J}\rangle + \frac{1}{n(n-1)}S_\gamma^2(\mathbf{HKH})} \approx i(\mathbf{K})$	
	$S_\gamma(\mathbf{X}) = \sum_{i=1}^n s_\gamma(\lambda_i)$, where $\lambda_1, \ldots, \lambda_n$ are eigenvalues of $\mathbf{X}$	
	$s_\gamma(x) = \begin{cases} \sqrt{x} & x > \gamma \\ \frac{1}{4}\gamma^{-\frac{3}{2}}x^2 + 3\sqrt{\gamma} & x \leq \gamma \end{cases}$	
	$\mathbf{S}'_\gamma(\mathbf{X}) = \sum_{i=1}^n s'_\gamma(\lambda_i)\mathbf{u}_i\mathbf{u}_i^\top$, where $\mathbf{X} = \sum_{i=1}^n \lambda_i\mathbf{u}_i\mathbf{u}_i^\top$	

positive definite. Finally, although the Bures-based measure is not differentiable, a smooth function operating on the eigenvalues of $\mathbf{HKH}$, i.e., a spectral function (Lewis, 1996), can be used as an approximation to replace the trace of the matrix square root, which is not differentiable for any $\mathbf{K}$ since $\mathbf{HKH}$ has an eigenvalue of 0 corresponding to the constant eigenvector. For the smoothing function, we use a Huber-type approximation (Huber, 1964; Hintermüller and Wu, 2014) of the square root. Using standard identities for the differentiation of scalar-matrix functions (Lewis, 1996; Olsen et al., 2012), we derive the gradients of cost functions proportional to those in Table 3. The simplified expressions of the gradients are given in Table 4.

As informativeness is only meaningful for correlation matrices, the optimization variable must be constrained to lie within the elliptope. Restriction to correlation matrices can be enforced via a combination of a linear constraint (for the diagonal) and a semidefinite constraint. Finding the maximally informative correlation matrix is a type of semidefinite optimization (convex for the Euclidean and HSIC measures and appropriate choice of ρ). For the convex cases, the unique optimum can be sought using either a projected-gradient or a conditional gradient (Frank-Wolfe) algorithm. In the former case, the Euclidean projection to the elliptope can be used, but is non-trivial (Higham, 2002; Malick, 2004). The

latter case requires solving a semidefinite program for the approximation step at each iteration. Furthermore, both cases require an order of n^2 variables. An alternative is to factorize the correlation matrix into a lower-dimensional Euclidean embedding (Journée et al., 2010). Factorizations have been used for solving the nearest correlation matrix problem (Grubišić and Pietersz, 2007) and other semidefinite programs (Burer and Monteiro, 2003). In Section 4.2, we pursue this factorization approach and express the cost functions corresponding to the measures of informativeness in terms of a Euclidean embedding.

When the correlation matrix is a function of parameters, informativeness can be used as an objective for optimizing these parameters. For instance, the correlation matrix can be defined parametrically in terms of the squared distances among a sample of points, passed through a differentiable function κ,

$$K_{i,j} = \kappa\left((\mathbf{x}_i - \mathbf{x}_j)^\top\mathbf{U}^\top\mathbf{U}(\mathbf{x}_i - \mathbf{x}_j)\right),$$

where $\mathbf{U}^\top\mathbf{U}$ defines a Mahalanobis distance metric as $d^2_{\mathbf{U}^\top\mathbf{U}}(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{U}\mathbf{x}_i - \mathbf{U}\mathbf{x}_j\|_2$ and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ is the sample of points. If $\kappa(|x-y|^2) = k(x,y) \quad \forall x, y \in \mathbb{R}$, where k is a positive semidefinite kernel function and $\kappa(0) = k(x,x) = 1 \quad \forall x$, then $\mathbf{K}$ is always a correlation matrix. Using this form, optimization of either $\mathbf{U}$ or $\mathbf{X}$ can be performed without additional constraints to ensure $\mathbf{K}$ is a correlation matrix. Optimizing informativeness with respect to $\mathbf{X}$ will shift the points in order to increase the informativeness, and optimization in terms of $\mathbf{U}$ corresponds to an unsupervised form of metric learning: a task that is typically supervised (Lowe, 1995; Xing et al., 2003; Fukumizu et al., 2004).

The gradient in terms of the coordinates of the sample $\mathbf{X}$ is

$$\nabla_{\mathbf{X}}\tilde{F}(\mathbf{K}) = -4\mathbf{U}^\top\mathbf{U}\mathbf{X}\left(\mathbf{B} - \operatorname{diag}(\mathbf{B}\mathbf{1})\right),$$

where $\mathbf{B} = \mathbf{K}' \circ \left[\nabla_{\mathbf{K}}\tilde{F}(\mathbf{K})\right]$, $\circ$ denotes the Hadamard product, $K'_{i,j} = \kappa'_{i,j}\left(d^2_{\mathbf{U}^\top\mathbf{U}}(\mathbf{x}_i, \mathbf{x}_j)\right)$, and κ' is the derivative of the scalar kernel function, e.g., for the Gaussian kernel, $k(x,y) = \kappa\left(|x-y|^2\right) = \exp\left(-|x-y|^2\right)$ and $\kappa'\left(|x-y|^2\right) = -\kappa\left(|x-y|^2\right)$. Similarly, the gradient with respect to $\mathbf{U}$ is $\nabla_{\mathbf{U}}\tilde{F}(\mathbf{K}) = -4\mathbf{U}\mathbf{X}\left(\mathbf{B} - \operatorname{diag}(\mathbf{B}\mathbf{1})\right)\mathbf{X}^\top$.

4.2 Maximally Informative Embeddings

When the corresponding embeddings are available and in cases when the dimensionality is lower than the number of objects, the informativeness can be evaluated with lower storage and computational requirements without unnecessarily forming the full correlation matrix. This can be realized as follows. Let $\mathbf{Z} \in \mathbb{R}^{b\times n}$ be a Euclidean space embedding, such that $\mathbf{Z}^\top\mathbf{Z} = \mathbf{K} \in \mathcal{E}$. Since we assume $\mathbf{K}$ is a correlation matrix, then $\mathbf{Z}$ belongs to the oblique manifold $\mathcal{OB}$. For each expression in Table 3, an equivalent expression of the cost function $F\left(d_{\mathcal{N}}(\mathbf{Z}^\top\mathbf{Z})\right)$ is listed in Table 5. It is noteworthy that the Bures-based measure yields a convex cost function, which we investigate in the next section. Empirically, the quantum Hellinger-based cost function appears convex, but we have not been able to prove its convexity. Besides these two, the other cost functions are not convex.

Table 5: Cost functions for finding a maximally informative Euclidean embedding $\mathbf{Z} \in \mathbb{R}^{b\times n}$ of the form $F\left(d_{\mathcal{N}}(\mathbf{Z}^\top\mathbf{Z})\right)$, where F is a monotonically decreasing function of the form $F : d \mapsto 1 - \nu d^2$ with $\nu \in \{\frac{1}{2}, 1\}$, and $d_{\mathcal{N}}(\mathbf{Z}^\top\mathbf{Z})$ is the distance or dissimilarity between the matrix $\mathbf{Z}^\top\mathbf{Z}$ and the family of non-informative matrices.

Measure:	$F\left(d_{\mathcal{N}}(\mathbf{Z}^\top\mathbf{Z})\right)$:	Convexity:
Euclidean	$1 - \frac{1}{n^2}\left\|\mathbf{Z}^\top\mathbf{Z} - \mathbf{N}_a\right\|_F^2, \quad a = \frac{1}{n^2}(\|\mathbf{Z1}\|_2^2 - \|\mathbf{Z}\|_2^2 + n)$	no
Cosine	$\frac{\langle \mathbf{Z}^\top\mathbf{Z}, \mathbf{N}_a\rangle}{\|\mathbf{Z}^\top\mathbf{Z}\|_F\|\mathbf{N}_a\|_F}, \quad a = \frac{1}{n}\|\mathbf{Z1}\|_2^2/\|\mathbf{Z}\|_F^2$	no
HSIC	$1 + \frac{1}{n^2(n-1)}\|\mathbf{ZH}\|_F^4 - \frac{1}{n^2}\|\mathbf{ZHZ}^\top\|_F^2$	no
CKA	$\frac{\|\mathbf{ZH}\|_F^2}{\|\mathbf{ZHZ}^\top\|_F\sqrt{n-1}}$	no
QH	$\sqrt{\frac{1}{n^3}\langle \mathbf{J}, \sqrt{\mathbf{Z}^\top\mathbf{Z}}\rangle^2 + \frac{1}{n(n-1)}\langle \mathbf{H}, \sqrt{\mathbf{Z}^\top\mathbf{Z}}\rangle^2}$	—
Bures-based	$\sqrt{\frac{1}{n^2}\|\mathbf{Z1}\|_2^2 + \frac{1}{n(n-1)}\|\mathbf{ZH}\|_*^2}$	yes

Given these cost functions, we can now pose the problem of finding a maximally informative embedding $\mathbf{Z}$ that lies within an ϵ-radius ball of a target embedding $\mathbf{T}$, as

$$\min_{\mathbf{Z}\in\mathcal{OB}} F\left(d_{\mathcal{N}}(\mathbf{Z}^\top\mathbf{Z})\right) \quad \text{s.t. } \|\mathbf{T} - \mathbf{Z}\|_F \leq \epsilon. \tag{6}$$

The target embedding could arise from a low-rank approximation $\mathbf{T}^\top\mathbf{T} \approx \mathbf{A}$ of a target similarity matrix $\mathbf{A}$. In general, this optimization is non-convex because of the constraint that $\mathbf{Z} \in \mathcal{OB}$. In Section 4.2.2 we present a convex relaxation and optimization algorithm for the Bures-based measure of informativeness. The optimization uses properties of the Bures-based measure explored in Section 4.2.1.

4.2.1 Optimizing the Bures-based Measure of Informativeness for Embeddings

Due to its convexity, we concentrate on the Bures-based informativeness measure for embeddings and define

$$h(\mathbf{Z}) \equiv F^2\left(d_{\mathcal{N}}(\mathbf{Z}^\top\mathbf{Z})\right) = \left(1 - i(\mathbf{Z}^\top\mathbf{Z})\right)^2 = \tfrac{1}{n^2}\|\mathbf{Z1}\|_2^2 + \frac{1}{n(n-1)}\|\mathbf{ZH}\|_*^2, \tag{7}$$

where $d_{\mathcal{N}}(\mathbf{Z}^\top\mathbf{Z}) = \sqrt{2i(\mathbf{Z}^\top\mathbf{Z})}$ is the lower-bound on the Bures distance to the nearest non-informative matrix as defined in Equation B.17. The cost function $h(\cdot)$ has some interesting properties. Firstly, it is well-defined for an arbitrary matrix $\mathbf{X} \in \mathbb{R}^{b\times n}$, not necessarily on the oblique manifold. Secondly, it is convex, since it is the sum of two squared seminorms. Used as a penalty function, the two seminorms simultaneously penalize high-rank embeddings and embeddings which have a large mean, i.e., $h(\cdot)$ is minimal for rank-1,

centered embeddings. Furthermore, since $\|\mathbf{X1}\|_2^2 = \frac{1}{n}\|\mathbf{X11}^\top\|_F^2 = n\|\mathbf{X}\frac{1}{n}\mathbf{J}\|_F^2 = n\|\mathbf{X}\frac{1}{n}\mathbf{J}\|_*^2$ both seminorms correspond to trace norms on the orthogonal subspaces defined by the projections $\mathbf{H}$ and $\frac{1}{n}\mathbf{J}$. We provide a proof that $\sqrt{h(\mathbf{X})} = \sqrt{\frac{1}{n}\|\mathbf{X}\frac{1}{n}\mathbf{J}\|_*^2 + \frac{1}{n(n-1)}\|\mathbf{XH}\|_*^2}$ is itself a matrix norm in Appendix D.

Due to the underlying trace norm, $h(\cdot)$ is not differentiable everywhere. Therefore, to enable the optimization we consider its proximal operator (Moreau, 1965)

$$\mathbf{prox}_{\lambda h}(\mathbf{Z}) \equiv \arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Z} - \mathbf{X}\|_F^2 + \lambda h(\mathbf{X}) \tag{8}$$

with $\lambda \geq 0$. Proximal operators enable first-order optimization involving non-smooth penalty functions (Nesterov, 2007; Wright et al., 2009; Beck and Teboulle, 2009; Combettes and Pesquet, 2011; Parikh and Boyd, 2014). To derive the proximal operator of $h(\cdot)$, we use the fact that both $h(\cdot)$ and the squared Frobenius norm (and Euclidean distance) can both be decomposed into the orthogonal subspaces defined by $\mathbf{H}$ and $\frac{1}{n}\mathbf{J}$, as

$$\begin{aligned} h(\mathbf{X}) &= \tfrac{1}{n}\|\mathbf{X}\tfrac{1}{n}\mathbf{J}\|_*^2 + \tfrac{1}{n(n-1)}\|\mathbf{XH}\|_*^2 = \tfrac{1}{n}\|\bar{\mathbf{X}}\|_*^2 + \tfrac{1}{n(n-1)}\|\tilde{\mathbf{X}}\|_*^2 \\ &= \|\bar{\mathbf{x}}\|_2^2 + \tfrac{1}{n(n-1)}\|\tilde{\mathbf{X}}\|_*^2, \end{aligned} \tag{9}$$

and

$$\|\mathbf{X}\|_F^2 = \left\|\mathbf{X}(\tfrac{1}{n}\mathbf{J} + \mathbf{H})\right\|_F^2 = \left\|\bar{\mathbf{X}}\right\|_F^2 + \|\tilde{\mathbf{X}}\|_F^2 = n\|\bar{\mathbf{x}}\|_2^2 + \|\tilde{\mathbf{X}}\|_F^2,$$

where $\tilde{\mathbf{X}} = \mathbf{XH}$ is a centered version of $\mathbf{X}$, $\bar{\mathbf{X}} = \bar{\mathbf{x}}\mathbf{1}^\top$, $\bar{\mathbf{x}}$ is the average of the columns of $\mathbf{X}$, and $\|\bar{\mathbf{X}}\|_* = \|\bar{\mathbf{X}}\|_F = \sqrt{n}\|\bar{\mathbf{x}}\|_2$ since $\bar{\mathbf{X}}$ is rank-1. Thus, the proximal solution can be written as the sum of the terms $\bar{\mathbf{Y}}$ and $\tilde{\mathbf{Y}}$, each computed independently as

$$\mathbf{prox}_{\lambda h}(\mathbf{Z}) = \bar{\mathbf{Y}} + \tilde{\mathbf{Y}}, \tag{10}$$

with the two terms calculated as

$$\begin{aligned} \bar{\mathbf{Y}} &= \arg\min_{\mathbf{X}} \frac{1}{2}\left\|\bar{\mathbf{Z}} - \mathbf{X}\right\|_F^2 + \frac{\lambda}{n}\|\mathbf{X}\|_F^2 = \frac{n}{n+2\lambda}\bar{\mathbf{Z}}, \\ \tilde{\mathbf{Y}} &= \arg\min_{\mathbf{X}} \frac{1}{2}\left\|\tilde{\mathbf{Z}} - \mathbf{X}\right\|_F^2 + \frac{\lambda}{n(n-1)}\|\mathbf{X}\|_*^2 = \mathcal{Z}_\beta(\tilde{\mathbf{Z}}), \end{aligned}$$

where $\beta = \frac{2\lambda}{n(n-1)}$ and $\mathcal{Z}_\beta(\cdot)$ is the proximal operator of the squared trace norm $\frac{\beta}{2}\|\cdot\|_*^2$, which we define in Equation E.1 in Appendix E along with a proof of its optimality.

4.2.2 Correlation Matrix Denoising

Given the proximal operator for the Bures-based informativeness, we proceed to solve the maximally informative embedding problem of Equation 6. We employ the optimization problem

$$\min_{\mathbf{Z}\in\mathcal{OB}} \tfrac{1}{2}\|\mathbf{T} - \mathbf{Z}\|_F^2 + \lambda h(\mathbf{Z}), \tag{11}$$

where the ϵ-radius constraint is replaced with the penalty term $\frac{1}{2}\|\mathbf{T} - \mathbf{Z}\|_F^2$ and the informativeness is scaled by λ. The minimization's cost function is convex, but the oblique

manifold is not a convex set. We propose to relax the oblique manifold constraint, by replacing it with the constraint that each point in the embedding $\mathbf{Z}$ lies within the unit-sphere $\|\mathbf{z}_i\|_2 \leq 1, \forall i \in [n]$, rather than on it. The relaxed constraint corresponds to the convex hull of the oblique manifold, denoted by $\text{conv}(\mathcal{OB})$. We then add the penalty term $-\frac{\mu}{2}\|\mathbf{Z}\|_F^2$ to ensure the norm of each column is maximized, which pushes the solution closer to the oblique manifold. The relaxed optimization is written as

$$\min_{\mathbf{Z}\in\text{conv}(\mathcal{OB})} \tfrac{1}{2}\|\mathbf{T}-\mathbf{Z}\|_F^2 - \tfrac{\mu}{2}\|\mathbf{Z}\|_F^2 + \lambda h(\mathbf{Z}).$$

This is a convex optimization problem for $\mu \leq 1$. We choose the largest possible penalty $\mu = 1$ to satisfy convexity and approach feasibility, leading to the cost function $\langle \mathbf{Z}, -\mathbf{T}\rangle + \lambda h(\mathbf{Z})$.

To isolate the non-smooth term, we write an equivalent composite form of the optimization problem, using an auxiliary variable $\mathbf{X}$ as

$$\begin{aligned} \min_{\substack{\mathbf{X}\in\mathbb{R}^{b\times n}\\ \mathbf{Z}\in\text{conv}(\mathcal{OB})}} \quad & \langle \mathbf{Z}, -\mathbf{T}\rangle + \lambda h(\mathbf{X}) \\ \text{s.t. } & \mathbf{X} = \mathbf{Z}. \end{aligned} \tag{12}$$

An approximate solution to the composite optimization can be found by removing the equality constraint and adding the penalty term $\frac{\rho}{2}\|\mathbf{Z}-\mathbf{X}\|_F^2$ and using alternating optimization to find a solution to the relaxed problem. The optimization of $\mathbf{Z}$, for fixed $\mathbf{X}$, can be performed on each column independently, and amounts to a Euclidean projection to the unit-ball. Likewise, the optimization $\mathbf{X}$, for fixed $\mathbf{Z}$, is also convex with an analytic solution given by application of the proximal operator $\mathbf{prox}_{\frac{\lambda}{\rho}h}(\mathbf{Z})$.

Because the aforementioned alternating optimization procedure is limited by slow convergence and sensitivity to the penalty, we instead conduct an optimization through an augmented Lagrangian formulation. Specifically, we employ the alternating direction method of multipliers algorithms (ADMM) (Gabay and Mercier, 1976; Eckstein and Bertsekas, 1992; Combettes and Pesquet, 2011; Boyd et al., 2011) to find a fixed point of the augmented Lagrangian

$$\mathcal{L}_\rho(\mathbf{Z},\mathbf{X},\mathbf{M}) = I(\mathbf{Z}) + \langle \mathbf{Z}, -\mathbf{T}\rangle + \lambda h(\mathbf{X}) + \tfrac{\rho}{2}\|\mathbf{Z}-\mathbf{X}\|_F^2 + \langle \mathbf{M}, \mathbf{Z}-\mathbf{X}\rangle, \tag{13}$$

where I is the indicator function for $\text{conv}(\mathcal{OB})$: $f(\mathbf{Z}) = 0$ for $\mathbf{Z} \in \text{conv}(\mathcal{OB})$ and $f(\mathbf{Z}) = +\infty$ otherwise. This leads to a minimization that alternates between two minimization steps on each non-fixed variables $\mathbf{Z}$ and $\mathbf{X}$. Using scaled dual variables $\mathbf{U} = \frac{1}{\rho}\mathbf{M}$, the steps are

$$\begin{aligned} &\min_{\mathbf{Z}\in\text{conv}(\mathcal{OB})} \left\|\mathbf{Z}-\mathbf{X}+\mathbf{U}-\tfrac{1}{\rho}\mathbf{T}\right\|_F^2, \\ &\min_{\mathbf{X}\in\mathbb{R}^{b\times n}} \tfrac{\rho}{2}\|\mathbf{Z}-\mathbf{X}+\mathbf{U}\|_F^2 + \lambda h(\mathbf{X}), \end{aligned}$$

and are followed by an update of the dual variables $\mathbf{U}$ with the residuals $\mathbf{Z}-\mathbf{X}$. The proposed optimization procedure is summarized in Algorithm 1. The computational complexity of each iteration is $\mathcal{O}(b^2 n)$, which corresponds to the singular value decomposition of the embedding matrix required by the proximal operator of h.

Algorithm 1: Correlation Matrix Denoising

input : Initial embedding $\mathbf{T}$ of size $b \times n$, and parameters $\lambda > 0$, $\rho > 0$, and $\eta = \frac{\lambda}{\rho}$

$\mathbf{X} \leftarrow \mathbf{T}$, $\mathbf{U} \leftarrow \mathbf{0}$

while *not converged* **do**

- $\mathbf{W} \leftarrow \mathbf{X} - \mathbf{U} + \frac{1}{\rho}\mathbf{T}$
- $\mathbf{z}_i \leftarrow \frac{\mathbf{w}_i}{\max(1, \|\mathbf{w}_i\|_2)}$, to project each column to unit-ball
- $\mathbf{X} \leftarrow \mathbf{prox}_{\eta h}(\mathbf{Z}+\mathbf{U}) = \bar{\mathbf{X}} + \tilde{\mathbf{X}}$, as defined in Equation 10 with
  - $\bar{\mathbf{X}} = \frac{n}{n+2\eta}(\mathbf{Z}+\mathbf{U})\frac{1}{n}\mathbf{J}$,
  - $\tilde{\mathbf{X}} = \mathcal{Z}_\beta(\mathbf{ZH}+\mathbf{UH})$, where $\beta = \frac{2\eta}{n(n-1)}$ and $\mathcal{Z}_\beta$ defined in Equation E.1.
- $\mathbf{U} \leftarrow \mathbf{U} + \mathbf{Z} - \mathbf{X}$, for the dual update

end

output: A denoised embedding $\mathbf{Z}$

In practice, we use $\rho = 1$, and find that if λ is chosen appropriately, the solution at convergence lies on the oblique manifold, i.e., $\mathbf{Z} \in \mathcal{OB}$. This means that the solution to the relaxed problem (12) satisfies the constraint of the original problem (11). To understand why the convex relaxation yields solutions to the original problem, we note that for any λ, the relaxed problem can be written as a linear program over the convex set defined by the intersection of $\text{conv}(\mathcal{OB})$ and $\mathcal{I}_c = \{\mathbf{X} : \sqrt{h(\mathbf{X})} \leq c\}$, where small values of c correspond to large values of λ, and $\mathcal{I}_c$ corresponds to the set embeddings with informativeness above $1-c$. As any linear program is optimized at extremal points of a convex set, an optimizing point of the relaxed solution will either lie on the oblique manifold and within the interior of $\mathcal{I}_c$, or if λ is too large, on the boundary of $\mathcal{I}_c$ and in the interior of $\text{conv}(\mathcal{OB})$.

The convergence of the ADMM algorithm can be proven under mild conditions (Boyd et al., 2011), which are met for Algorithm 1. The conditions require the objective functions to be closed, proper, and convex; and the existence of a saddle point $(\mathbf{Z}^\star, \mathbf{X}^\star, \mathbf{M}^\star)$ of the unaugmented Lagrangian function $\mathcal{L}_0(\mathbf{Z},\mathbf{X},\mathbf{M})$, i.e., Equation 13 with $\rho = 0$, such that $\mathcal{L}_0(\mathbf{Z}^\star,\mathbf{X}^\star,\mathbf{M}) \leq \mathcal{L}_0(\mathbf{Z}^\star,\mathbf{X}^\star,\mathbf{M}^\star) \leq \mathcal{L}_0(\mathbf{Z},\mathbf{X},\mathbf{M}^\star)$ for all $\mathbf{Z},\mathbf{X},\mathbf{M}$. The saddle point $(\mathbf{Z}^\star,\mathbf{X}^\star,\mathbf{0})$ corresponds to the solution $(\mathbf{Z}^\star,\mathbf{X}^\star)$ of the relaxed optimization problem (12). This solution necessarily exists, since as mentioned, the relaxed problem is equivalent to a linear objective optimized over the convex set defined by the intersection of $\text{conv}(\mathcal{OB})$ and $\mathcal{I}_c$. Furthermore, the convergence rate of the ADMM algorithm for convex objectives is $\mathcal{O}(1/t)$, where t is the number of iterations (He and Yuan, 2015).

To obtain embeddings which are progressively more informative (low-rank and nearly centered), the algorithm can be ran with increasing values of λ and warm restarts. We start with $\lambda = 1$, increase it by 125% at each restart, and stop the sequence of warm restarts if the solution at convergence no longer lies on the boundary of the convex hull, that is, $\mathbf{Z} \notin \mathcal{OB}$. The main computational burden of the algorithm is the singular value decomposition required in computing the proximal operator of $h(\cdot)$. This burden can somewhat be alleviated by using lower dimensional embeddings (i.e., a smaller b). Rather than choosing a small initial b, which cannot preserve as much information about the target matrix, we use an adaptive truncation heuristic. Specifically, we note that $h(\cdot)$ is invariant to unitary

transformations of the embedding coordinates. Thus, we can use a singular value decomposition of the solution $\mathbf{Z} = \mathbf{\Psi}\mathbf{\Sigma}\mathbf{\Upsilon}^\top$ and transform $\mathbf{Z}$ and $\mathbf{T}$ with the left singular vectors, and then remove embedding dimensions (rows) from both $\mathbf{\Psi}^\top\mathbf{Z} = \mathbf{\Sigma}\mathbf{\Upsilon}^\top$ and $\mathbf{\Psi}^\top\mathbf{T}$, when the corresponding singular values of $\mathbf{Z}$ are less than $\frac{\sigma_1}{100}$, with $\sigma_1 = \|\mathbf{Z}\|_2$ being the maximum singular value. While truncation does distort the target matrix, this approach lowers the dimensions and decreases the processing requirements for the subsequent warm restarts.

5. Experiments

In this section we highlight example applications of the measures of informativeness[7] and compare the different measures listed in Table 2. Applications include selecting an appropriate kernel bandwidth, selecting the dimension of a low-rank kernel approximation, and detecting clustered graphs. We also demonstrate the correlation matrix denoising algorithm as a preprocessor for spectral clustering.

5.1 Kernel Bandwidth Selection

Gaussian kernel functions of the form $k_\sigma(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-1}{2\sigma^2}\|\mathbf{x} - \mathbf{y}\|_2^2\right)$ are widely used for kernel-based clustering and classification of vectors in Euclidean space, but an appropriate kernel bandwidth σ needs to be selected. For any choice of σ, evaluating the kernel function for a set of vectors yields a correlation matrix with entries $K_{i,j} = k_\sigma(\mathbf{x}_i, \mathbf{x}_j)$. If all of the vectors are distinct, the limit $\sigma \to 0$ yields the identity matrix $\mathbf{I}$, and the limit $\sigma \to \infty$ results in the constant matrix $\mathbf{J}$. Essentially, the kernel bandwidth parameterizes a curve within the elliptope that begins and finishes at non-informative correlation matrices. Since informativeness is a function of the distance to the set of non-informative correlation matrices, choosing the kernel bandwidth that maximizes the informativeness corresponds to finding an extremal point of this curve as illustrated in Figure 7. The example shows informativeness can be used to select a kernel bandwidth appropriate for the scale of the data. In comparison, measures such as the von Neumann entropy or distance from the identity matrix are maximized by the smallest and largest kernel bandwidths, respectively.

We compare the different informativeness measures on four synthetic examples in Figure 8. In these examples, the measures are maximized by kernel bandwidths dependent on the scale of the data. This holds for all measures except CKA, which is maximized by large kernel bandwidths that yield nearly constant matrices. We quantify the appropriateness of informativeness-based bandwidth selection for spectral clustering—specifically, the algorithm by Ng et al. (2002)—by using the normalized mutual information[8] between the labels corresponding to the true grouping and the clusters found when using different kernel bandwidths. For each example, 10 random instances are tested with the correct number of groups provided to the clustering algorithm. On the first example, each group is a convex shape and a large kernel bandwidth performs well. On the remaining examples, there is an intermediate kernel bandwidth that maximizes the normalized mutual information.

7. MATLAB code that implements the informativeness measures and reproduces the figures and tables is available at `http://pcwww.liv.ac.uk/~goulerma/software/brockmeier17a-code.zip`.
8. Normalized mutual information is the mutual information between two variables divided by the maximum entropy of the two (Kvålseth, 1987).

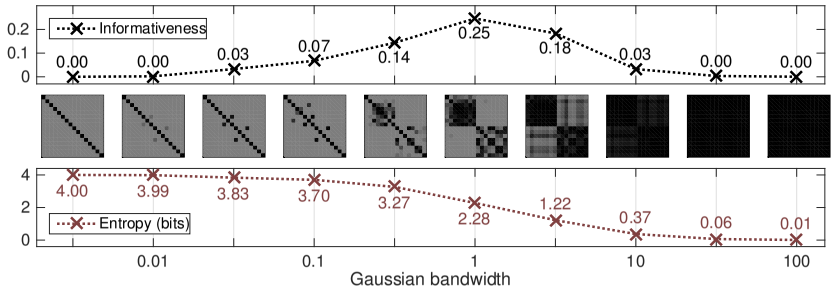


Figure 7: Informativeness versus von Neumann entropy for correlation matrices obtained from applying a Gaussian kernel with varying bandwidths to a sample drawn from a univariate mixture of two normal distributions separated by a distance of 4. (The Bures-based informativeness is used in this example.)

For each measure of informativeness and for Bartlett's and Lawley's test statistics, we collect the values of normalized mutual information corresponding to the bandwidths that maximize the measures. The results are reported as box and whisker plots in Figure 8. From the results, there appear to be four subsets of measures that select similar bandwidths: the cosine measure, which chooses the smallest kernel bandwidth, has the worst median performance on the first data set, but performs the best on the remaining; the Chernoff, QH, and Bures-based measures select a larger bandwidth, tie for the best performance on the first data set, achieve the second-best performance across the remaining; the Euclidean, HSIC, and sub-Bures measures select larger bandwidths and perform worse; and the CKA measure along with Bartlett's and Lawley's test statistics are maximized by the largest kernel bandwidth within the range, which yields a poor clustering on three of the four data sets. This example shows that the informativeness measures (except CKA) can adaptively select kernel bandwidths appropriate for the structure of the data.

5.2 Selecting an Embedding Dimension

Kernel principal component analysis (Schölkopf et al., 1998) approximates a kernel matrix using a lower-dimensional embedding obtained by truncating the eigenvalue decomposition of the centered kernel matrix. Choosing the embedding dimension (how many eigenvectors to retain) requires a criterion. We show that choosing the dimension that maximizes informativeness is an effective approach.

Specifically, kernel principal component analysis uses the top-k eigenvectors of the centered kernel matrix $\tilde{\mathbf{K}} = \mathbf{HKH}$. Let denote $\tilde{\mathbf{K}}^{(k)}$ the rank-k approximation of $\tilde{\mathbf{K}}$. In general $\tilde{\mathbf{K}}^{(k)}$ is positive definite but not a correlation matrix, and requires symmetric normalization as defined in Equation 1. However, if the original kernel matrix is far from being centered then applying normalization after centering and truncation can introduce severe distortion.

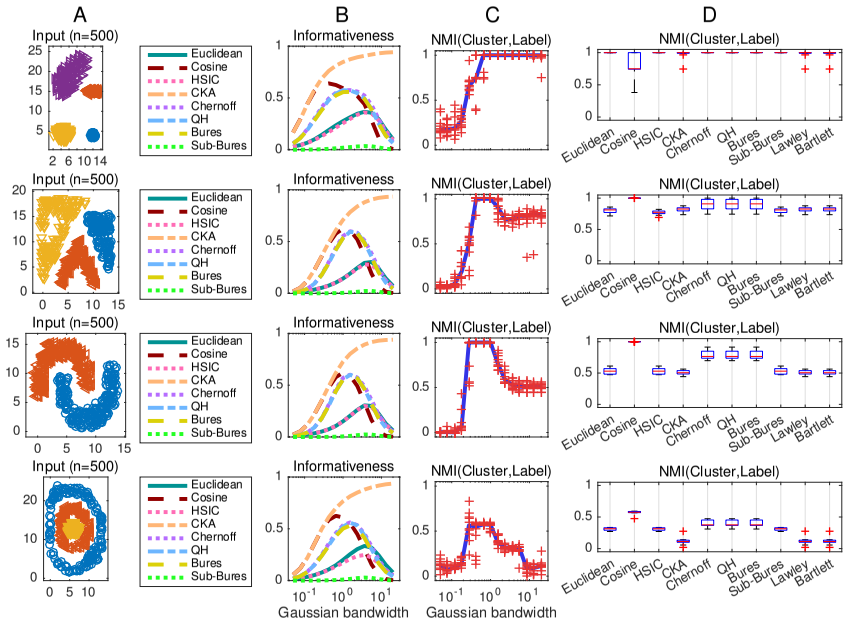


Figure 8: Selecting a Gaussian kernel bandwidth for spectral clustering by maximizing informativeness. (A) Samples are drawn from two-dimensional distributions with distinct groups. (B) Informativeness is plotted versus the kernel bandwidth. (C) For each kernel bandwidth, spectral clustering is applied with the correct number of clusters and the normalized mutual information (NMI) between the cluster label and the true label is recorded (crosses indicate values across 10 random samples; solid curve for the median). (D) Box and whisker plots of the NMI values for the clusterings obtained when the kernel bandwidth is selected by informativeness or by Lawley's and Bartlett's test statistics.

As an alternative, we truncate the original, uncentered correlation matrix and then perform renormalization.

We test the dimensionality selection method on a set of UCI data sets (Lichman, 2013). A Gaussian kernel function with a heuristic bandwidth is used for all cases. Specifically, the bandwidth is a linear combination of the minimum and maximum Euclidean distances $\sigma = 2d_{min} + \frac{2}{9}(d_{max} - 2d_{min})$, where d_{min} and d_{max} are the minimum and maximum pairwise distances (a similar heuristic was used by Shi and Malik, 2000).

Table 6: Nearest neighbor classification performance for kernel-based dimensionality reduction across 7 UCI data sets. Each entry lists the classification accuracy as a percentage, followed by the average dimensionality in parentheses. The last two rows list the average difference in performance versus using the original space and the average loss margin (absolute difference to the best method on the run) across 20 Monte Carlo runs and data sets (standard deviation across data sets).

	Original	CKA[1]	CKA[2]	Euclid.	Cosine	QH	Bures
breast cancer	95 (569)	90 (2.0)	87 (1.0)	91 (3)	95 (328)	95 (52)	95 (41)
sonar	83 (208)	83 (126.3)	82 (130.1)	62 (3)	83 (113)	83 (28)	83 (26)
ionosphere	85 (351)	74 (1.0)	88 (69.5)	84 (2)	78 (176)	94 (38)	94 (33)
parkinson	92 (195)	84 (21.4)	80 (39.5)	82 (2)	93 (100)	91 (19)	91 (19)
iris	94 (150)	85 (3.0)	84 (2.2)	80 (2)	94 (21)	93 (10)	94 (11)
glass	68 (214)	65 (181.0)	68 (211.6)	60 (3)	68 (112)	68 (20)	68 (19)
ecoli	80 (336)	78 (31.6)	77 (48.2)	80 (6)	80 (158)	80 (24)	80 (24)
vs. original	—	5.5±4	4.9±6	8.4±7	0.9±2	-0.9±4	-1.1±4
Loss margin	2.2±3	7.7±6	7.1±4	10.6±7	3.2±6	1.3±0.6	1.2±0.5

[1] Centered kernel alignment using the training portion of the uncentered kernel matrix
[2] Centered kernel alignment using the training portion of the centered kernel matrix

	HSIC	CKA	Chernoff	Sub-Bures	Bartlett	Lawley
breast cancer	90 (2)	70 (1)	95 (33)	91 (3)	70 (1)	70 (1)
sonar	62 (3)	53 (1)	84 (30)	62 (3)	57 (2)	57 (2)
ionosphere	84 (2)	74 (1)	94 (38)	84 (2)	84 (2)	84 (2)
parkinson	82 (2)	67 (1)	91 (15)	82 (2)	82 (2)	82 (2)
iris	80 (2)	66 (1)	93 (10)	80 (2)	80 (2)	80 (2)
glass	60 (3)	38 (1)	68 (18)	60 (3)	38 (1)	38 (1)
ecoli	74 (3)	40 (1)	80 (18)	80 (6)	66 (2)	66 (2)
vs. original	9.5±6	27.2±9	-1.0±4	8.4±7	17.3±10	17.3±10
Loss margin	11.8±6	29.4±7	1.3±0.3	10.6±7	19.5±9	19.5±9

For comparison, we use supervised selection of the dimensionality based on centered kernel alignment (Cortes et al., 2012). We test both the centered and uncentered truncations, and select the dimension that maximizes the centered kernel alignment to a target matrix obtained from the classification labels in the training set. In all cases, we use a first nearest neighbor classifier with half of the instances for training; the average classification accuracy is recorded in Table 6.

Across these data sets, the Bures, Chernoff, and QH-based informativeness measures perform the best. They select a lower dimensional embedding and improve the classification performance versus using the original distances. The cosine-based informativeness measure also outperforms using centered kernel alignment with training labels. In comparison, the Euclidean and HSIC-based informativeness measures select very low-dimensional embeddings that do not perform well, and the CKA-based informativeness and Bartlett's and

Lawley's test statistics are inappropriate for this task as they often select one-dimensional embeddings, which is an inadequate representation for a nearest neighbor classifier.

5.3 Informativeness as an Indicator of Structure in Graphs

We show that informativeness is indicative of structure in undirected graphs. We consider binary graphs whose edges are defined by a symmetric adjacency matrix $\mathbf{G} \in \{0,1\}^{n\times n}$, where $G_{i,j} = 1$ indicates an edge between vertices i and j. To enable informativeness to be applied directly, we represent a graph by its normalized graph Laplacian matrix $\mathbf{L}$, which is a correlation matrix under the following definition

$$L_{i,j} = \begin{cases} 1 & i = j, \\ \frac{-1}{\sqrt{d_i d_j}} & G_{i,j} = 1, \\ 0 & \text{otherwise}, \end{cases}$$

where d_i is the degree of the ith vertex. When the graph has unconnected vertices, the corresponding rows and columns of the normalized graph Laplacian are all zeros except at the diagonal entry. This is different than the standard convention (Chung, 1997), where the diagonal entry for an unconnected vertex would be zero.

Informativeness can be used to make relative comparisons between graphs with the same number of vertices. In particular, informativeness can be used to rank graphs and identify graphs that are more informative than a regular graph with the same number of edges. Firstly, we consider the informativeness for small graphs, $n = 5$, for which we evaluate all 34 non-isomorphic graphs. Figure 9 shows the Bures-based informativeness for each graph plotted against the number of edges; there is a clear separation of the values for the three most informative graphs, which correspond to clustered graphs. Of the remaining graphs, the path graph (a tree with maximum degree of 2) has the next highest informativeness. This ordering held for all informativeness measures. We then test the set of non-isomorphic trees for $n = 6$. The trees are shown in Figure 10 ordered by their informativeness. Once again, the path graph has the highest informativeness of all trees; the ordering is consistent for all informativeness measures.

For larger graphs, it becomes infeasible to enumerate all of the non-isomorphic graphs. As a baseline we consider regular graphs, which are defined as graphs whose vertices have the same degree. To further sample from the universe of possible graphs with $n = 100$ vertices, we consider the following random graph models:

- Erdős-Rényi model described by an edge-link probability (Erdős and Rényi, 1959),
- Barabási-Albert scale-free model with a power law degree distribution (Barabási and Albert, 1999),
- Watts and Strogatz small-world model (Watts and Strogatz, 1998), which randomly rewires each edge of a regular graphs with a certain probability (we use 5%),
- a clustered regular graph model, generated by removing all edges between vertices assigned to different clusters (the number of clusters is varied between 2 and 20).

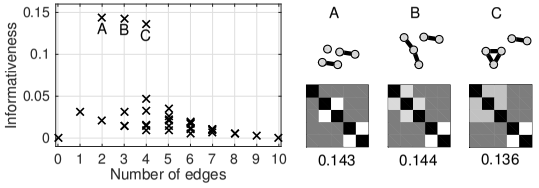


Figure 9: Bures-based informativeness of non-isomorphic graphs with $n = 5$ vertices. The three graphs (labeled A, B, C) with highest informativeness correspond to graphs with two or three clusters.

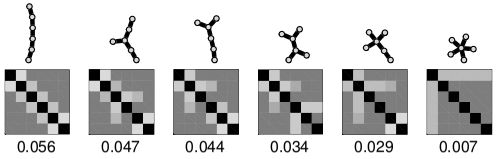


Figure 10: Non-isomorphic trees with 6 vertices sorted by Bures-based informativeness.

Erdős-Rényi graphs and Barabási-Albert scale-free graphs are both unstructured graph models that differ in their degree distributions. The degree distribution of the former is a Poisson distribution, whereas the scale-free graphs have a long-tailed degree distribution, meaning that a small number of vertices have a large number of edges.

We evaluate the informativeness of regular graphs and realizations of each random graph model, scanning any free parameters to vary the number of edges. Example graphs and the Bures-based informativeness as a function of the average degree[9] are shown in Figure 11. For any given number of edges, the informativeness is maximized by clustered graphs, and minimized by the Erdős-Rényi graphs and Barabási-Albert scale-free graphs. The random rewiring involved with the Watts and Strogatz small-world model does not drastically change the informativeness from that of regular graphs.

We compare informativeness to two existing test statistics for clustering in graphs: the modularity of two-way partitions identified using an eigenvector-based method (Newman, 2006) and the statistic defined by Bickel and Sarkar (2016) on the adjacency matrix. The latter statistic is used for testing the hypothesis that the graph is drawn from an Erdős-Rényi model; the test statistic is based on the largest eigenvalue of the difference between the adjacency matrix and the expectation over adjacency matrices under an Erdős-Rényi model, where the normalized graph Laplacian of this expectation is a non-informative correlation matrix. Bickel and Sarkar (2016) used the test statistic to identify cluster structure. Bickel and Sarkar's statistic is normalized, such that its distribution for Erdős-Rényi graphs is invariant to the number of edges. Figure 11 also shows the values obtained from these test

9. The average degree is given as $\bar{d} = \frac{2e}{n}$, where e is the number of edges.

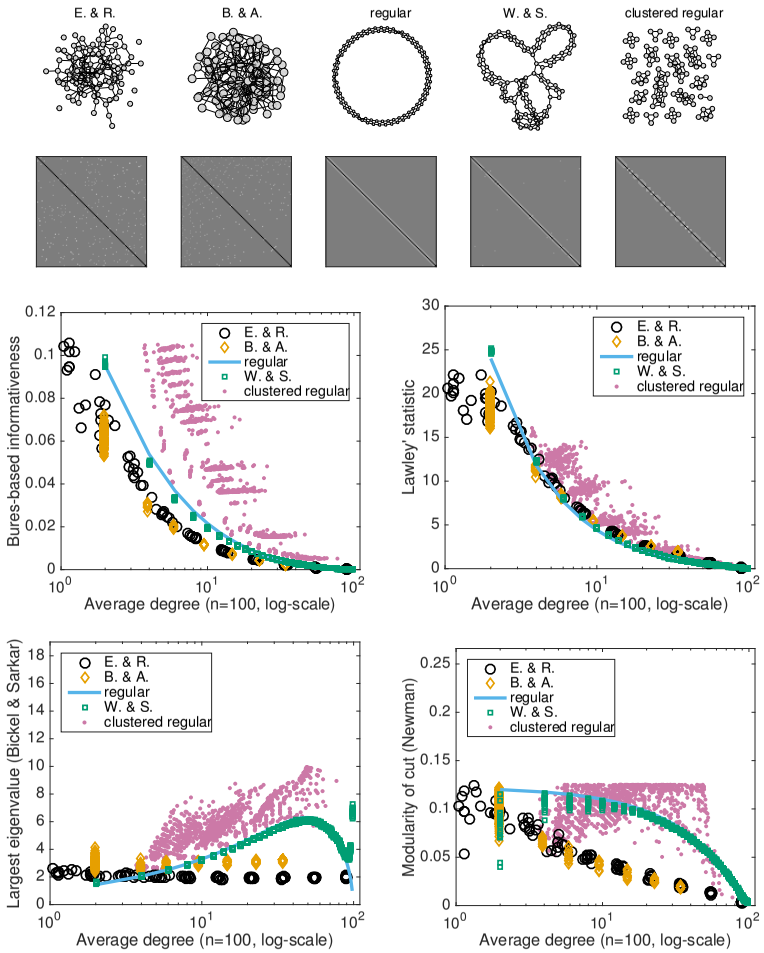


Figure 11: Comparison of informativeness and other statistics on different types of random graphs. Bures-based informativeness and Lawley's test statistic for homogeneous correlation coefficients are applied to the normalized graph Laplacians, and compared to the graph-clustering test statistics proposed by Bickel and Sarkar (2016) and Newman (2006). The random graphs were drawn from Erdős-Rényi (E. & R.), Barabási-Albert scale-free (B. & A.), Watts-Strogatz small-world (W. & S.), and clustered regular models.

Table 7: Proportion of graphs with informativeness, or other measure, greater than a threshold set as the value of a nearly-regular graph with the same number of edges. Graphs have 100 vertices. The random graphs were drawn from Erdős-Rényi (E. & R.), Barabási-Albert scale-free (B. & A.), Watts-Strogatz small-world (W. & S.), and clustered regular models.

Number of graphs:	E. & R. 200	B. & A. 200	W. & S. 990	Clustered regular 1200
Euclidean	0.241	0.195	0.986	0.968
Sub-Bures	0.241	0.195	0.986	0.968
Cosine	0.241	0.195	0.987	0.968
HSIC	0.134	**0**	0.969	0.957
CKA	0.134	**0**	0.974	0.957
Chernoff	0.053	**0**	0.550	**1**
QH	0.053	**0**	0.558	**1**
Bures	0.118	**0**	0.553	**1**
Bartlett	**0.016**	**0**	0.974	0.957
Lawley	0.364	0.340	1	0.996
Bickel and Sarkar	0.460	0.700	0.232	0.962
Newman	0.102	**0**	**0.099**	0.601

statistics on the various graphs. The test statistics have a different range as compared to the measures of informativeness; nonetheless, for a given statistic or measure of informativeness, the values can be used to make relative comparisons between the different graphs.

The value of informativeness is dependent on the number of edges, with graphs with fewer edges being more informative. As a threshold for identifying structure, we use the informativeness of a nearly-regular graph[10] with the same number of edges. The thresholds for the other statistics are formed in the same way. Ideally, the value of a clustered graph will be above the threshold, and the value will be below the threshold for an unstructured graph. Table 7 details the proportion of random graphs above the threshold for each measure.

The QH, Chernoff, and Bures-based measures appear to be the best measures to distinguish clustered graphs from unstructured graphs. The HSIC and CKA-based measures and Bartlett's statistic perform nearly as well. The Euclidean, sub-Bures, and cosine-based measures are not as specific—a larger proportion of unstructured graphs are evaluated above the threshold. Bickel and Sarkar's and Lawley's statistics are even less specific. Newman's modularity index (designed to detect binary clustering) evaluates fewer clustered graphs above the threshold; but, it identifies the lowest proportion of Watts-Strogatz small-world model, which are not clustered. To summarize, a subset of the informativeness measures and Bartlett's statistic are specific indicators of regular structure and clustering in graphs, and Newman's modularity is useful for identifying clustered graphs specifically.

10. A graph is regular if the number of edges is divisible by the number of vertices, otherwise a nearly-regular graph is constructed by adding edges to a regular graph, where the additional edges are evenly distributed, such that the degrees of any two vertices differ by at most 1.

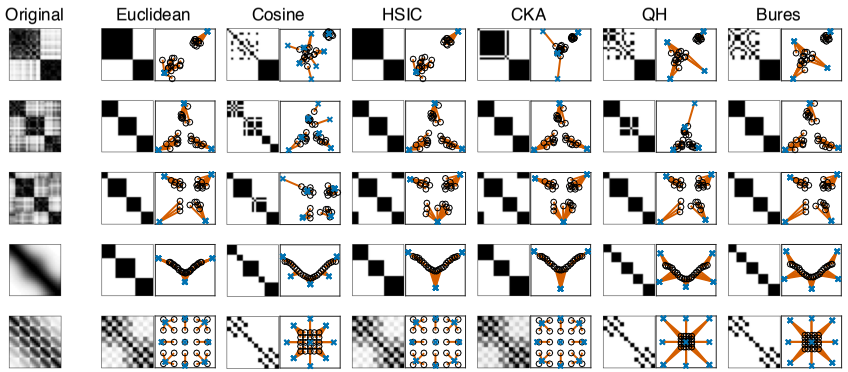


Figure 12: Sample denoising based on optimizing the informativeness of a Gaussian kernel matrix with respect to the sample's two-dimensional coordinates. Each row shows the original Gaussian kernel matrix ($n = 25$), the kernel matrices corresponding to the optimized coordinates, the location of the optimized coordinates, denoted as x's, and the original coordinates, denoted as o's.

5.4 Sample Denoising via Informativeness

To illustrate that informativeness can be used as an unsupervised objective function, we explore first-order optimization of informativeness with respect to a sample of points $\{\mathbf{x}_i\}_{i=1}^n$, defining a correlation matrix through a Gaussian function as $K_{i,j} = \kappa(\|\mathbf{x}_i - \mathbf{x}_j\|_2^2) = \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{x}_i - \mathbf{x}_j\|_2^2\right)$. We first find the bandwidth σ that maximizes the informativeness using a golden search over $\theta \in [-5, 5]$, where $2\sigma^2 = 10^{-\theta}$. Then, we optimize the sample coordinates using the conjugate gradient method implemented in minFunc (Schmidt, 2012) with the gradients given in Section 4.1. For all methods, the optimization is performed in terms of $\acute{\mathbf{K}} = (1-\eta)\mathbf{K} + \eta\mathbf{I}$, where $\eta = 10^{-6}$, to ensure the correlation matrix is positive definite. A log-barrier term of $-\log(1 - \frac{1}{n^2}\mathbf{1}^\top\mathbf{K}\mathbf{1})$ is added to the cost function for the CKA measure. For the Bures-based measure, a smoothing parameter of $\gamma = 10^{-9}$ is used.

We apply the first-order optimization on five synthetic data sets consisting of $n = 25$ two-dimensional points. As shown in Figure 12, the optimization exhibits mode-seeking—i.e., groups of nearby points cluster together, which enhances the cluster structure seen in the correlation matrices.

We then apply the denoising process to a set of grayscale images of handwritten digits in the USPS data set (Hull, 1994).[11] Taking a random set of 40 images for each digit yields a sample with $n = 400$. The resulting correlation matrices and the images corresponding to the optimized point locations are shown in Figure 13. There is a clear difference in the results for the different measures: With the Euclidean measure, different digits are

11. The USPS handwritten digits data set (Hull, 1994) is available from `http://web.stanford.edu/~hastie/ElemStatLearn/data.html`, where it is labeled 'ZIP code'.

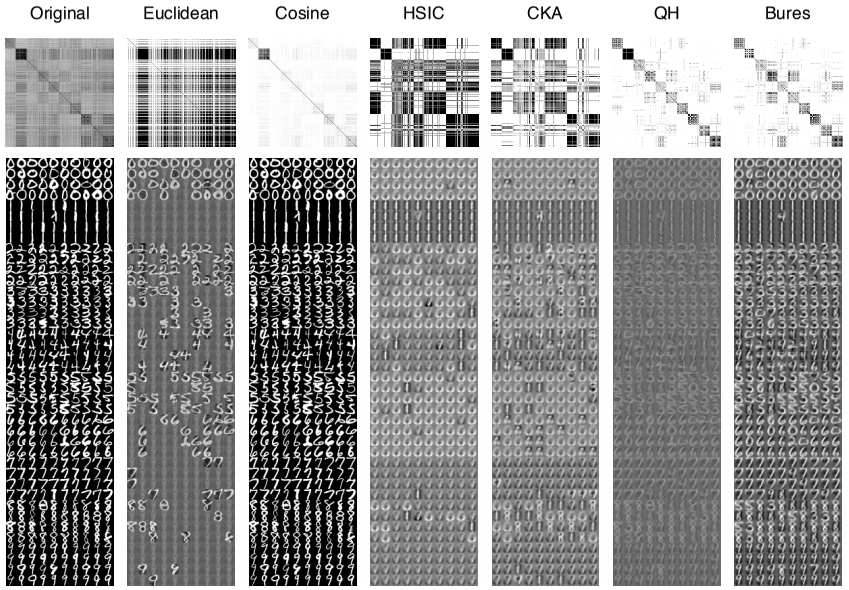


Figure 13: Sample denoising on 200 images from the USPS handwritten digits data set. The correlation matrix is formed from the Gaussian kernel applied to vectors of the pixel values. The first row shows the correlation matrices and the second row shows the corresponding images based on the optimized pixel values.

grouped together into indistinct images, while the remaining images remain unchanged. The cosine measure only slightly modifies the images. The HSIC and CKA measures seem to map most of the digits to a subset of stereotyped digits, but besides 0's, 1's, and 7's/9's the correspondence to the original digits is lost. On the other hand, the QH and Bures measures produce correlation matrices that correspond much more closely to the true class structure. Furthermore, the images appear to be denoised with the majority of the digits mapped to stereotyped versions of the original digits. To quantify the enhanced class structure, we compute the centered kernel alignment between the correlation matrix for the ground-truth labels and the optimized correlation matrix, and record the values in Table 8. As a baseline, we use the original coordinates (that is the original images) for the sample and select a kernel bandwidth that maximizes the centered kernel alignment. Only the Bures measure yields a correlation matrix with higher centered kernel alignment than this baseline.

The QH and Bures measures are also distinguished by their computation time as shown in Table 8, as the gradient calculations for these two measures have higher computational

Table 8: Performance results for sample denoising on the subset of 200 USPS digit images shown in Figure 13. The centered kernel alignment (CA) is calculated between the ground-truth label matrix (a binary correlation matrix) and the correlation matrix for the original sample and the denoised sample for each measure. Computation time logged in MATLAB R2015b on a 2.8 GHz Intel Core i7 with 16 GB RAM.

	Original	Euclidean	Cosine	HSIC	CKA	QH	Bures
CA to labels:	0.53	0.11	0.45	0.27	0.28	0.47	0.55
Time (s):	—	0.99	0.17	1.56	1.10	503.85	37.13

complexity $\mathcal{O}(n^3)$ versus $\mathcal{O}(n^2)$ for the other measures. The QH measure requires solving two Sylvester equations, and the Bures measure requires a single eigendecomposition. This makes the running time with the QH measure much longer in practice.

5.5 Correlation Matrix Denoising

We now demonstrate Algorithm 1 for correlation matrix denoising. The algorithm finds an embedding that maximizes the Bures-based informativeness within a neighborhood of a target embedding. The algorithm is completely unsupervised and can be used to obtain an arbitrary rank embedding. We use it to reduce the rank of an $n \times n$ kernel matrix to $\lfloor \sqrt{n} \rfloor$ as a preprocessing for clustering.

The initial correlation matrix is formed by using a Gaussian kernel between images for five thumbnail image data sets: ORL, MNIST, UMIST, USPS, and COIL-20.[12] The kernel bandwidth heuristic described in Section 5.2 is again used to select this parameter. The initial target embedding was given as $\mathbf{T} = \sqrt{\mathbf{\Sigma}}\mathbf{U}^\top$, where $\mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top = \mathbf{K}$ is the eigendecomposition of the Gaussian kernel matrix. Figure 14 shows a visualization of the various matrices and also plots the informativeness and centered kernel alignment to the label matrix as a function of rank. As designed, the denoised matrices are more informative than the original matrix and more informative than truncated and renormalized kernel matrices (centered or uncentered) of equal rank. Additionally, except on the ORL data set, the denoised matrices have higher centered kernel alignment.

To demonstrate that the denoising preserves and enhances task-relevant structure, we perform spectral clustering with and without denoising. Starting from an embedding of the original correlation matrix, we obtain a denoised embedding $\mathbf{Z}$ with a rank of $\lfloor \sqrt{n} \rfloor$. Since spectral clustering methods assume a non-negative affinity matrix as input, we apply non-negative thresholding on the entries of the denoised correlation matrix $\mathbf{K}^\star = \mathbf{Z}^\top\mathbf{Z}$ and treat the resulting non-negative, symmetric matrix $[\mathbf{K}^\star]_+$ as the input for Ng et al.'s normalized

12. The Olivetti Research Laboratory's (ORL) face image data set was previously hosted by AT&T Cambridge and is now hosted by Cambridge University Computer Laboratory: `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`. The MNIST handwritten digits test set is available from `http://yann.lecun.com/exdb/mnist/`. The UMIST data set (Graham and Allinson, 1998) is now the Sheffield Face Database `https://www.sheffield.ac.uk/eee/research/iel/research/face`. The COIL-20 data set consists of images of rotated objects (Nene et al., 1996), available at `http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php`.

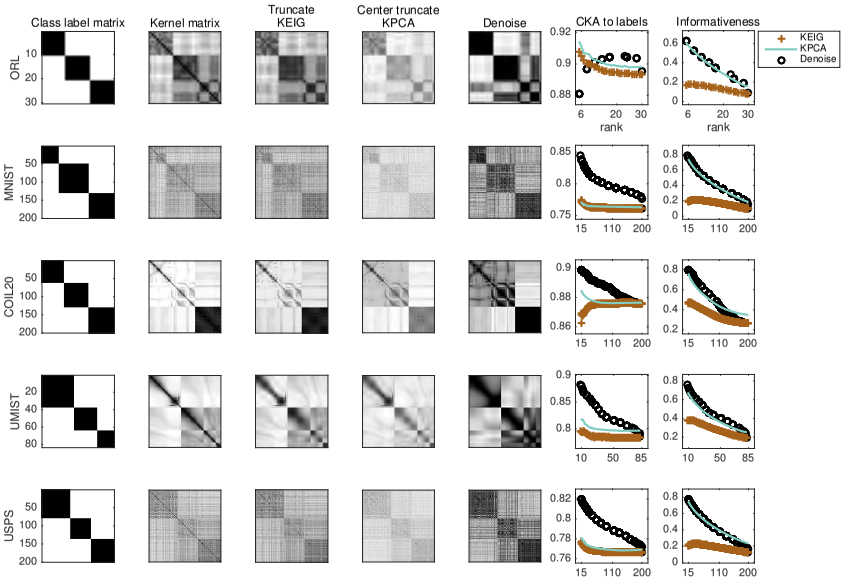


Figure 14: Examples of correlation matrix denoising on Gaussian kernel matrices obtained from samples of the ORL, MNIST, COIL-20, UMIST, and USPS data sets. Each sample contains three different classes. Each row shows the kernel matrix for the ground truth labels, the original Gaussian kernel matrix, the truncated and renormalized matrices corresponding to the uncentered kernel matrix (KEIG) and centered kernel matrix (KPCA), and the denoised correlation matrix. The latter three matrices have rank $\lfloor \sqrt{n} \rfloor$ and the symmetric normalization described in Equation 1 is applied to ensure they are correlation matrices. To the right of the matrices, the centered kernel alignment (CKA) to the ground-truth label matrix and the Bures-based informativeness is shown as a function of rank.

spectral clustering algorithm (Ng et al., 2002). For comparison we also use the spectral clustering method by Shi and Malik (2000) without denoising.

We compare clustering performance on the 20 Newsgroup text data set[13] and the five image data sets used in the previous example. Random subsets of different number of classes (2, 3, 4, and 5) are drawn with 200 samples in each Monte Carlo draw. For the text data set, each document is represented as a bag-of-words vector (a sparse vector of word counts for each document), and in each Monte Carlo division, words that are used in less than 4 documents are dropped. The remaining word counts are multiplied by the

13. A MATLAB/Octave version is provided by Jason Rennie `http://qwone.com/~jason/20Newsgroups/`.

Table 9: Clustering performance using normalized spectral clustering (NCut), the algorithm by Ng, Jordan, and Weiss (NJW), and the NJW algorithm after denoising the correlation matrix. In each case, k-means with correct number of classes is used to determine the cluster assignment from the spectral embedding. Values correspond to average accuracy (higher is better) or variation in information (lower is better) across 20 Monte Carlo runs; the average loss margin to the best performing method is shown parenthesized. Boldfaced entries indicate significantly better median performance (Wilcoxon signed-rank test with a significance threshold of 0.05).

	Accuracy (after mapping clusters to classes)				Variation in information		
	k	NCut	NJW	denoise+NJW	NCut	NJW	denoise+NJW
20 News	2	0.64 (0.29)	0.92 (0.01)	0.93 (0.01)	1.09 (0.46)	0.67 (0.04)	0.68 (0.06)
	3	0.60 (0.22)	0.73 (0.09)	**0.79 (0.03)**	1.70 (0.29)	1.64 (0.24)	1.51 (0.11)
	4	0.54 (0.21)	0.68 (0.07)	**0.75 (0.01)**	2.17 (0.42)	1.99 (0.25)	**1.82 (0.08)**
	5	0.53 (0.17)	0.62 (0.08)	**0.69 (0.01)**	2.34 (0.30)	2.30 (0.25)	**2.16 (0.11)**
COIL-20	2	0.70 (0.26)	0.91 (0.05)	**0.96 (0.00)**	1.20 (1.00)	0.50 (0.29)	**0.22 (0.01)**
	3	0.64 (0.26)	0.86 (0.04)	**0.88 (0.02)**	1.27 (0.77)	0.65 (0.15)	0.60 (0.11)
	4	0.60 (0.27)	0.77 (0.10)	**0.87 (0.00)**	1.50 (0.80)	0.94 (0.25)	**0.70 (0.01)**
	5	0.58 (0.20)	0.72 (0.06)	0.76 (0.02)	1.79 (0.66)	1.37 (0.24)	1.23 (0.10)
MNIST	2	0.78 (0.16)	0.92 (0.02)	**0.94 (0.00)**	1.17 (0.63)	0.70 (0.16)	**0.56 (0.01)**
	3	0.77 (0.09)	0.78 (0.08)	**0.85 (0.01)**	1.43 (0.35)	1.47 (0.40)	**1.14 (0.06)**
	4	0.70 (0.07)	0.71 (0.06)	**0.75 (0.01)**	1.89 (0.23)	1.86 (0.21)	**1.71 (0.05)**
	5	0.64 (0.07)	0.66 (0.05)	0.69 (0.02)	2.30 (0.20)	2.32 (0.22)	**2.14 (0.04)**
USPS	2	0.85 (0.09)	0.91 (0.03)	0.93 (0.01)	0.88 (0.37)	0.67 (0.16)	0.61 (0.10)
	3	0.75 (0.10)	0.76 (0.09)	0.80 (0.05)	1.33 (0.33)	1.30 (0.29)	**1.13 (0.13)**
	4	0.66 (0.13)	0.74 (0.05)	0.76 (0.03)	1.81 (0.57)	1.39 (0.15)	1.32 (0.08)
	5	0.66 (0.09)	0.71 (0.04)	0.73 (0.02)	1.93 (0.28)	1.77 (0.11)	1.79 (0.13)
UMIST	2	0.75 (0.15)	0.72 (0.18)	**0.87 (0.03)**	1.17 (0.62)	1.34 (0.80)	**0.66 (0.11)**
	3	0.64 (0.20)	0.67 (0.17)	**0.81 (0.03)**	1.40 (0.65)	1.31 (0.56)	**0.92 (0.17)**
	4	0.59 (0.15)	0.63 (0.11)	**0.69 (0.05)**	1.56 (0.42)	1.52 (0.38)	1.37 (0.23)
	5	0.54 (0.11)	0.59 (0.06)	0.61 (0.04)	1.81 (0.32)	1.82 (0.33)	1.69 (0.20)
ORL	2	0.78 (0.22)	0.98 (0.02)	1.00 (0.00)	1.00 (1.00)	0.11 (0.11)	0.00 (0.00)
	3	0.81 (0.17)	0.97 (0.01)	0.97 (0.01)	0.66 (0.54)	0.18 (0.06)	0.15 (0.04)
	4	0.77 (0.20)	0.92 (0.04)	0.95 (0.02)	0.84 (0.64)	0.41 (0.21)	0.29 (0.09)
	5	0.81 (0.16)	0.94 (0.03)	0.94 (0.03)	0.75 (0.56)	0.31 (0.12)	0.31 (0.12)

logarithm of the inverse of each word's document frequency (this is the standard TF-IDF weighting), and cosine similarity is used to compute the correlation matrix. For the image data sets, the Gaussian kernel (the kernel bandwidth heuristic described in Section 5.2 is used once again). We fix the number of clusters to the number of classes. The results in terms of accuracy and variation of information (Meilă, 2003) are recorded in Table 9. The clustering obtained when using the denoised and thresholded matrix better matches the ground truth (in terms of both performance metrics) than the clustering obtained using the original kernel matrix across almost every data set and number of classes.

6. Discussion

We have introduced a framework for identifying nontrivial structure in correlation matrices using informativeness, which is a distance-based framework we have proposed for measuring the equality of correlation coefficients (Bartlett, 1954; Anderson, 1963; Lawley, 1963; Gleser, 1968; Aitkin et al., 1968; Steiger, 1980; Brien et al., 1984). Correlation matrices appear in various contexts within machine learning and statistical analysis. Measures which yield correlation matrices include the normalized inner product; Pearson's, Spearman's, and Kendall's correlation coefficients between sets of vectors; kernel matrices formed from non-linear kernel functions on data of various types (since any positive semidefinite kernel function can be normalized); and the normalized graph Laplacian, which can represent any set of non-negative similarity measurements on an undirected graph.

We defined the informativeness of a correlation matrix to be proportional to the distance between it and the closest correlation matrix whose off-diagonal entries are all equal. Motivated by the fact that a scaled correlation matrix is a valid quantum density matrix, we explored quantum distance metrics, including the Bures distance. For specific distance metrics, we derived closed-form expressions of the minimal distance, and a lower-bound for the Bures distance. In particular, we proved the lower-bound, referred to as the Bures-based informativeness, is maximized by centered, rank-1 correlation matrices, which means a maximally informative matrix corresponds to a balanced cut.

The measures of informativeness are unsupervised and can be used as objective functions for machine learning applications. In particular, informativeness can be used for selecting kernel parameters or the embedding dimension for kernel-based dimensionality reduction, testing for structure in graphs, and first-order optimizations of correlation matrices. In the first case, we explored informativeness as a criterion for automatically selecting an appropriate kernel bandwidth for spectral clustering. For this task, the proposed informativeness measures (except the CKA-based measure) consistently outperform Bartlett's and Lawley's test statistics for equality of correlation coefficients, which like the CKA measure always select the largest possible kernel bandwidth. The results also indicate that there is a clear grouping of the different informativeness measures that behave similarly. The cosine measure performs best on three data sets, and the three measures applicable to quantum density matrices—Chernoff bound, QH, and Bures—have the second best performance on three data sets.

In Section 5.2, we investigated using informativeness to select the dimensionality of a kernel-based embedding in a semi-supervised case, where the kernel matrix is formed from both labeled and unlabeled samples, and the performance is assessed by the first nearest-neighbor classification error rates in the reduced dimension space. On this task, Bartlett's and Lawley's test statistics fail to select a meaningful dimensionality (selecting one or two dimensions), which yields an average classification error rate 17.3 percentage points (ppts) higher than the original, as detailed in Table 6. In comparison, the Chernoff, QH, and Bures-based measures have the best average performance with an error rate >0.9 ppts lower than using the distances in the original space. Furthermore, these measures of informativeness outperform centered kernel alignment (Cortes et al., 2012) on this task, which has an error rate of 4.9 ppts higher than the original, even though centered kernel alignment uses the training set labels.

Informativeness is also useful for quantifying the amount of structure in undirected graphs via their normalized graph Laplacian representation, as discussed in Section 5.3. Empirically, the measures of informativeness are all able to distinguish regular and clustered graphs from unstructured graphs. The quantum-based distances (Chernoff bound, QH, and Bures) perform well in distinguishing structured graphs from random graphs (identifying <15% of random graphs as structured and 100% of clustered regular graphs). These informativeness measures are better at identifying structured graphs than the test statistic for determining if the graph is a realization of an Erdős-Rényi graph (Bickel and Sarkar, 2016), which identifies 46% of random graphs and 96.2% of clustered regular graphs. For this task, Bartlett's test statistic also performs well (identifying >2% of random graphs as structured and 95.7% of clustered regular graphs), but Lawley's test statistics evaluates a larger proportion of random graphs higher than the regular graphs (36.4%).

Across these experiments, the three quantum-based distances (Chernoff bound, QH, and Bures) perform consistently well in identifying correlation matrices that are structured and more suitable for clustering or classification. One drawback of these measures is that their evaluations have a higher computation cost than the other measures of informativeness that are based on the Frobenius norm, e.g., the Euclidean distance, cosine similarity, HSIC, and CKA. Specifically, the Bures-based measure requires all of the eigenvalues of an $n \times n$ positive semidefinite matrix, the Chernoff-bound requires a full eigendecomposition, and the QH measures requires the matrix square root—a complexity of $\mathcal{O}(n^3)$ in each case. In comparison, the measures based on the Frobenius norm can be computed directly from the elements of the correlation matrix (or its centered version), with a complexity of $\mathcal{O}(n^2)$.

Closed-form measures of informativeness can also be used as optimization criteria for applications like sample denoising and metric learning through positive definite kernel functions. Using the gradients of smooth cost functions derived from the informativeness measures, we discussed first-order optimizations in Section 4.1. The experimental results for sample denoising in Section 5.4 show that when the coordinates of a sample of points are updated the points form local clusters to maximize informativeness. In this way, informativeness functions as an objective for finding local modes in a sample. This behavior is scale-invariant, since when a Gaussian or other applicable kernel is used, the scale can be automatically set by choosing the bandwidth that maximizes the informativeness. When the informativeness-based optimizations are applied to a thumbnail image data set, the optimization based on the Bures measures enhances the class structure, yielding a higher centered kernel alignment than the correlation matrix for the original sample.

Along similar lines, we investigated the optimization problem of finding maximally informative correlation matrices in terms of their lower-dimensional embeddings (Section 4.2). We proved that the Bures-based measure is especially suited for this task since it corresponds to a convex cost function on Euclidean embeddings. This cost function is a novel matrix norm, which can be used to simultaneously penalize high-rank and uncentered embeddings. Based on the proximal operator of this norm (Appendix E), we proposed a correlation matrix denoising algorithm using an ADMM scheme. In Section 5.5, we tested the proposed correlation matrix denoising algorithm on data sets with known cluster structure. On thumbnail image data sets, the denoising method is able to preserve information relevant to ground-truth classes. Furthermore, it consistently increases the clustering performance when used as a preprocessing for spectral clustering (Ng et al., 2002).

In conclusion, informativeness appears to be a well-suited criterion for a variety of unsupervised data analysis tasks that can be formulated in terms of correlation matrices, and offers a unique measure for assessing the organizational properties of a sample based on pairwise similarity measurements.

Acknowledgments

This work was supported by the Medical Research Council grant MR/L01078X/1 "Supporting Evidence-based Public Health Interventions using Text Mining."

Appendix A. Bures Distance

Bures (1969) defined a distance metric between two non-commutative, positive semidefinite Hermitian matrices that generalizes the Hellinger-Kakutani probability distance to non-commuting spaces. The Bures distance is a measure of the statistical distance between two quantum states, i.e., a symmetric measure of how likely it is for one state to transition to the other (Uhlmann, 1976). In particular, as shown later, when two matrices commute the Bures distance corresponds to the Hellinger-Kakutani distance between the eigenvalues.

Let $\mathbf{A}, \mathbf{B}$ denote two positive-semidefinite matrices, which are trace-normalized so that $\mathrm{tr}\mathbf{A} = \mathrm{tr}\mathbf{B} = 1$. The squared Bures distance between them can be computed (Uhlmann, 1976) as $d_B^2(\mathbf{A}, \mathbf{B}) = 2 - 2\mathrm{tr}\sqrt{\mathbf{A}^{\frac{1}{2}}\mathbf{B}\mathbf{A}^{\frac{1}{2}}}$, where all square roots denote matrix roots. The Bures distance is a chordal distance on the space of positive semidefinite matrices with trace one. The trace term in the previous expression, denoted here as $C(\mathbf{A}, \mathbf{B})$, is equal to the trace norm of the matrix $\mathbf{A}^{\frac{1}{2}}\mathbf{B}^{\frac{1}{2}}$, that is

$$C(\mathbf{A}, \mathbf{B}) = \mathrm{tr}\sqrt{\mathbf{A}^{\frac{1}{2}}\mathbf{B}\mathbf{A}^{\frac{1}{2}}} = \left\|\mathbf{A}^{\frac{1}{2}}\mathbf{B}^{\frac{1}{2}}\right\|_*.$$

This quantity is a symmetric similarity measure between positive definite matrices. From the properties of trace-normalized positive semidefinite matrices, we have $0 \leq C(\mathbf{A}, \mathbf{B}) \leq 1$; $C(\mathbf{A}, \mathbf{B}) = 0$ iff $\langle \mathbf{A}, \mathbf{B} \rangle = 0$ (since $\mathbf{AB}$ and $\mathbf{A}^{\frac{1}{2}}\mathbf{B}\mathbf{A}^{\frac{1}{2}}$ have the same eigenvalues), and $C(\mathbf{A}, \mathbf{B}) = 1$ iff $\mathbf{A} = \mathbf{B}$. In quantum information theory, the quantity $C(\mathbf{A}, \mathbf{B})$, or the concave quantity $C^2(\mathbf{A}, \mathbf{B})$ as originally defined, is known as fidelity (Jozsa, 1994). Fidelity is a measure of how well $\mathbf{A}$ preserves the information of $\mathbf{B}$. It can also be used to directly define a geodesic distance, which is known as the statistical distance or quantum angle, and can be computed using the standard chordal to geodesic conversion, as $\arccos\left(C(\mathbf{A}, \mathbf{B})\right) = \arccos\left(1 - \frac{1}{2}d_B^2(\mathbf{A}, \mathbf{B})\right)$.

If $\mathbf{A}$ and $\mathbf{B}$ commute, then they are simultaneously diagonalizable by an orthogonal matrix $\mathbf{M}$, that is $\mathbf{M}\mathbf{\Lambda}_A\mathbf{M}^\top = \mathbf{A}$ and $\mathbf{M}\mathbf{\Lambda}_B\mathbf{M}^\top = \mathbf{B}$, with $\mathbf{\Lambda}_A$ and $\mathbf{\Lambda}_B$ being the corresponding spectral matrices. Therefore, $\mathrm{tr}\sqrt{\mathbf{A}^{\frac{1}{2}}\mathbf{B}\mathbf{A}^{\frac{1}{2}}} = \mathrm{tr}\big(\mathbf{M}\sqrt{\mathbf{\Lambda}_A\mathbf{\Lambda}_B}\mathbf{M}^\top\big) = \mathrm{tr}(\sqrt{\mathbf{\Lambda}_A}\sqrt{\mathbf{\Lambda}_B})$ and $\mathrm{tr}\big(\mathbf{M}\sqrt{\mathbf{\Lambda}_A\mathbf{\Lambda}_B}\mathbf{M}^\top\big) = \langle\sqrt{\mathbf{A}}, \sqrt{\mathbf{B}}\rangle$, which makes the fidelity $C(\mathbf{A}, \mathbf{B})$ equal to the affinity measure in Table 1, or equivalently, the Bures and the quantum Hellinger distance to coincide because $d_B^2(\mathbf{A}, \mathbf{B}) = \|\sqrt{\mathbf{A}} - \sqrt{\mathbf{B}}\|_F^2$. In this case, by using $\mathbf{p}_A = \mathrm{diag}\left(\mathbf{\Lambda}_A\right)$ and $\mathbf{p}_B = \mathrm{diag}\left(\mathbf{\Lambda}_B\right)$ to denote vectors containing the eigenvalues, $C(\mathbf{A}, \mathbf{B})$ is also equal to $\langle\sqrt{\mathbf{p}_A}, \sqrt{\mathbf{p}_B}\rangle$. This last quantity is the Bhattacharyya coefficient between the eigenspectra

of $\mathbf{A}$ and $\mathbf{B}$, and hence, $d_B^2(\mathbf{A},\mathbf{B}) = \|\sqrt{\mathbf{p}_A} - \sqrt{\mathbf{p}_B}\|_2^2$ is the squared Euclidean distance between them.

A further point to note about the Bures distance is that it serves as a lower bound for the trace distance. Specifically, as shown by Fuchs and van de Graaf (1999), we have $\frac{1}{2}d_B^2(\mathbf{A},\mathbf{B}) = 1 - C(\mathbf{A},\mathbf{B}) \leq \frac{1}{2}\|\mathbf{A}-\mathbf{B}\|_*$. It is also a lower bound on quantum Hellinger distance (Luo and Zhang, 2004). This can also be directly seen from the fact that for a given matrix $\mathbf{Q}$, we have $\mathrm{tr}(\mathbf{Q}) \leq \max_{\|\mathbf{U}\|_2\leq 1}\mathrm{tr}(\mathbf{U}^\top\mathbf{Q}) = \|\mathbf{Q}\|_*$ from the duality of the trace and spectral norms. By setting $\mathbf{Q} = \sqrt{\mathbf{A}}\sqrt{\mathbf{B}}$, we have the fidelity measure $C(\mathbf{A},\mathbf{B})$ dominating the affinity measure, $\langle\sqrt{\mathbf{A}},\sqrt{\mathbf{B}}\rangle$.

For the context of this work, we will show that a useful property of the Bures distance is that in its computation, the matrix square roots $\sqrt{\mathbf{A}}$ and $\sqrt{\mathbf{B}}$ can be substituted with any arbitrary embeddings in a Hilbert space that represent the correlation structure. Assume we have two arbitrary embeddings $\mathbf{Z}_A \in \mathbb{R}^{b_A\times n}$ and $\mathbf{Z}_B \in \mathbb{R}^{b_B\times n}$ such that $\mathbf{Z}_A^\top\mathbf{Z}_A = \mathbf{A}$ and $\mathbf{Z}_B^\top\mathbf{Z}_B = \mathbf{B}$. Since $\|\mathbf{Q}\|_* = \left\|\mathbf{Q}^\top\right\|_* = \mathrm{tr}\sqrt{\mathbf{Q}^\top\mathbf{Q}}$, we have

$$\begin{aligned}\left\|\mathbf{Z}_A\mathbf{Z}_B^\top\right\|_* &= \mathrm{tr}\sqrt{\mathbf{Z}_A\mathbf{Z}_B^\top\mathbf{Z}_B\mathbf{Z}_A^\top} = \mathrm{tr}\sqrt{\mathbf{Z}_A\mathbf{B}\mathbf{Z}_A^\top} = \left\|\mathbf{Z}_A\mathbf{B}^{\frac{1}{2}}\right\|_* \\ &= \mathrm{tr}\sqrt{\mathbf{B}^{\frac{1}{2}}\mathbf{Z}_A^\top\mathbf{Z}_A\mathbf{B}^{\frac{1}{2}}} = \mathrm{tr}\sqrt{\mathbf{B}^{\frac{1}{2}}\mathbf{A}\mathbf{B}^{\frac{1}{2}}} = \left\|\mathbf{A}^{\frac{1}{2}}\mathbf{B}^{\frac{1}{2}}\right\|_*.\end{aligned} \tag{A.1}$$

A critical conclusion from these equations, is that the fidelity $\|\mathbf{A}^{\frac{1}{2}}\mathbf{B}^{\frac{1}{2}}\|_*$ and the Bures distance d_B can potentially use any of the forms in Equation A.1 for calculating the constituent trace norm, according to computational convenience and embeddings dimensionality, even though the embeddings $\mathbf{Z}_A$ and $\mathbf{Z}_B$ are not unique representations of $\sqrt{\mathbf{A}}$ and $\sqrt{\mathbf{B}}$. For instance, the constraint $\mathbf{Z}_A^\top\mathbf{Z}_A = \mathbf{A}$ cannot imply a unique $\mathbf{Z}_A$, as a row permutation, sign change, or any orthogonal transformation of the rows of $\mathbf{Z}_A$ would not alter the underlying similarity matrix. Furthermore, one can choose a transformation matrix $\mathbf{U}_A$, which is not necessarily square, with orthonormal columns (and, by definition, orthonormal rows if $\mathbf{U}_A$ is square) that introduces the embedding into a higher dimensional space. For example, if $\mathbf{U}_A \in \mathbb{R}^{q\times b_A}$, the projection $\mathbf{U}_A\mathbf{Z}_A$ is equivalent to applying an orthogonal transformation to $\binom{\mathbf{Z}_A}{\mathbf{0}} \in \mathbb{R}^{q\times n}$. These invariances lead to an alternative formulation of fidelity via the following theorem, which is known as Uhlmann's theorem in the more general case for quantum density matrices (Uhlmann, 1976; Jozsa, 1994; Nielsen and Chuang, 2000).

Theorem 1 *For positive semidefinite matrices $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times n}$,*

$$C(\mathbf{A},\mathbf{B}) = \left\|\mathbf{A}^{\frac{1}{2}}\mathbf{B}^{\frac{1}{2}}\right\|_* = \max_{\substack{\mathbf{Z}_A,\mathbf{Z}_B \\ \mathbf{Z}_A^\top\mathbf{Z}_A=\mathbf{A} \\ \mathbf{Z}_B^\top\mathbf{Z}_B=\mathbf{B}}} \langle\mathbf{Z}_A,\mathbf{Z}_B\rangle.$$

Proof Let $\bar{\mathbf{Z}}_A \in \mathbb{R}^{b_A\times n}$ and $\bar{\mathbf{Z}}_B \in \mathbb{R}^{b_B\times n}$ be arbitrary embeddings such that $\bar{\mathbf{Z}}_A^\top\bar{\mathbf{Z}}_A = \mathbf{A}$ and $\bar{\mathbf{Z}}_B^\top\bar{\mathbf{Z}}_B = \mathbf{B}$ (without loss of generality, we assume $b_A \leq b_B$). The set of valid embeddings for $\mathbf{A}$ and $\mathbf{B}$ can then be expressed as transformations of $\bar{\mathbf{Z}}_A$ and $\bar{\mathbf{Z}}_B$, i.e.,

$$\left\{(\mathbf{Z}_A,\mathbf{Z}_B) : \mathbf{Z}_A^\top\mathbf{Z}_A{=}\mathbf{A},\ \mathbf{Z}_B^\top\mathbf{Z}_B{=}\mathbf{B}\right\} \ = \ \left\{(\mathbf{U}_A\bar{\mathbf{Z}}_A,\mathbf{U}_B\bar{\mathbf{Z}}_B) : \mathbf{U}_A^\top\mathbf{U}_A{=}\mathbf{I},\ \mathbf{U}_B^\top\mathbf{U}_B = \mathbf{I}\right\},$$

where $\mathbf{U}_A \in \mathbb{R}^{q\times b_A}$ and $\mathbf{U}_B \in \mathbb{R}^{q\times b_A}$ are matrices with orthonormal columns and the same number of rows. From the equivalence of these two sets, we pose the equivalent maximization problem

$$\max_{\substack{\mathbf{U}_A,\mathbf{U}_B \\ \mathbf{U}_A^\top\mathbf{U}_A=\mathbf{I} \\ \mathbf{U}_B^\top\mathbf{U}_B=\mathbf{I}}} \left\{\langle\mathbf{U}_A\bar{\mathbf{Z}}_A,\mathbf{U}_B\bar{\mathbf{Z}}_B\rangle = \mathrm{tr}\left(\bar{\mathbf{Z}}_A^\top\mathbf{U}_A^\top\mathbf{U}_B\bar{\mathbf{Z}}_B\right) = \mathrm{tr}\left(\mathbf{U}_B^\top\mathbf{U}_A\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top\right)\right\}.$$

Since $\mathbf{U}_A$ and $\mathbf{U}_B$ are semi-orthogonal, they both have all their non-zero singular values equal to one, and from the sub-multiplicative property it follows that

$$\|\mathbf{U}_B^\top\mathbf{U}_A\|_2 \leq \|\mathbf{U}_A\|_2 \cdot \|\mathbf{U}_B\|_2 = 1,$$

and furthermore, the set of pairs of semi-orthogonal matrices is a strict subset of the set of pairs with product within the spectral norm unit ball:

$$\left\{(\mathbf{U}_A,\mathbf{U}_B) : \mathbf{U}_A^\top\mathbf{U}_A = \mathbf{I},\ \mathbf{U}_B^\top\mathbf{U}_B = \mathbf{I}\right\} \ \subset \ \left\{(\mathbf{U}_A,\mathbf{U}_B) : \|\mathbf{U}_B^\top\mathbf{U}_A\|_2 \leq 1\right\}.$$

Letting $\mathbf{U}^\top = \mathbf{U}_B^\top\mathbf{U}_A$, we have

$$\max_{\substack{\mathbf{U}_A,\mathbf{U}_B \\ \mathbf{U}_A^\top\mathbf{U}_A=\mathbf{I} \\ \mathbf{U}_B^\top\mathbf{U}_B=\mathbf{I}}} \mathrm{tr}\left(\mathbf{U}_B^\top\mathbf{U}_A\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top\right) \ \leq \ \max_{\substack{\mathbf{U} \\ \|\mathbf{U}\|_2\leq 1}} \mathrm{tr}\left(\mathbf{U}^\top\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top\right) = \|\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top\|_*,$$

where the equality follows from the duality of the trace and spectral norms. Furthermore, the optimization in terms of $\mathbf{U}_A, \mathbf{U}_B$ achieves the maximum when $\mathbf{U}_A = [\mathbf{R},\ \mathbf{0}_{b_A\times(b_B-b_A)}]^\top$ and $\mathbf{U}_B = \mathbf{V}$, where $\mathbf{RSV}^\top$ is a singular value decomposition of $\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top$, since in this case[14]

$$\begin{aligned}&\mathrm{tr}\left(\mathbf{U}_B^\top\mathbf{U}_A\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top\right) = \mathrm{tr}\left(\mathbf{U}_B^\top\mathbf{U}_A\mathbf{RSV}^\top\right) = \mathrm{tr}\left(\mathbf{V}^\top\mathbf{U}_B^\top\mathbf{U}_A\mathbf{RS}\right) \\ &= \mathrm{tr}\left(\mathbf{V}^\top\mathbf{V}\left[\mathbf{R}^\top\mathbf{R},\ \mathbf{0}_{b_A\times(b_B-b_A)}\right]^\top\mathbf{S}\right) = \mathrm{tr}\left(\left[\mathbf{I}_{b_A\times b_A},\ \mathbf{0}_{b_A\times(b_B-b_A)}\right]^\top\mathbf{S}\right) = \|\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top\|_*.\end{aligned}$$

By the properties of the trace norm (A.1), $\|\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top\|_* = \|\mathbf{A}^{\frac{1}{2}}\mathbf{B}^{\frac{1}{2}}\|_*$, which completes the proof. ■

It is a significant consequence of this theorem that $d_B(\mathbf{A},\mathbf{B})$ corresponds to the minimal distance $\|\mathbf{Z}_A - \mathbf{Z}_B\|_F$ between all possible embeddings represented by $\mathbf{A}$ and $\mathbf{B}$. This is very desirable, as it makes d_B independent to the choice of their embeddings used to measure it with any of the forms in Equation A.1.

We note that Theorem 1 can also be used to show that the symmetric normalization in Equation 1, discussed in Section 2.1, which transforms a positive semidefinite matrix to a correlation matrix, corresponds to finding the closest elliptope element in the Bures distance sense.

14. Letting $\mathbf{Q} = \mathbf{V}^\top\mathbf{U}_B^\top\mathbf{U}_A\mathbf{R} = (\mathbf{U}_B\mathbf{V})^\top\mathbf{U}_A\mathbf{R}$ we have $\mathrm{tr}(\mathbf{U}_B^\top\mathbf{U}_A\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top) = \mathrm{tr}(\mathbf{QS})$. We note that no element of $\mathbf{Q}$ can exceed unity by the Cauchy-Bunyakovsky-Schwarz inequality since $\mathbf{U}_A\mathbf{R}$ and $\mathbf{U}_B\mathbf{V}$ have orthonormal columns. From this, it follows that $\mathrm{tr}(\mathbf{U}_B^\top\mathbf{U}_A\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top) = \mathrm{tr}(\mathbf{QS}) \leq \mathrm{tr}(\mathbf{S}) = \|\bar{\mathbf{Z}}_A\bar{\mathbf{Z}}_B^\top\|_*$.

Lemma 2 *For a positive semidefinite matrix* $\tilde{\mathbf{K}} \in \mathbb{R}^{n\times n}$ *with* $\mathrm{tr}(\tilde{\mathbf{K}}) = n$,

$$\operatorname*{arg\,min}_{\mathbf{K}\in\mathcal{E}} \; d_B\left(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\tilde{\mathbf{K}}\right) \;=\; \mathbf{D}\tilde{\mathbf{K}}\mathbf{D},$$

where $\mathbf{D}$ *is a diagonal matrix with entries* $D_{i,i} = \frac{1}{\sqrt{\tilde{K}_{i,i}}}$. *Furthermore,*

$$\min_{\mathbf{K}\in\mathcal{E}} \; d_B\left(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\tilde{\mathbf{K}}\right) = \sqrt{\frac{2}{n}\sum_{i\in[n]}\left(1-(\tilde{K}_{i,i})^{\frac{1}{2}}\right)},$$

i.e., the distance depends on how far the roots of the diagonal entries of $\tilde{\mathbf{K}}$ *are from unity.*

Proof Let $\tilde{\mathbf{Z}}$ be an embedding of $\tilde{\mathbf{K}}$ such that $\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}} = \tilde{\mathbf{K}}$. Then for any choice of $\tilde{\mathbf{Z}}$,

$$\min_{\mathbf{K}\in\mathcal{E}} d_B\left(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\tilde{\mathbf{K}}\right) = \min_{\mathbf{Z}\in\mathcal{OB}} \left\|\tfrac{1}{\sqrt{n}}\mathbf{Z} - \tfrac{1}{\sqrt{n}}\tilde{\mathbf{Z}}\right\|_F = \sqrt{\frac{1}{n}\sum_{i\in[n]}\min_{\mathbf{x}:\|\mathbf{x}\|_2=1}\|\mathbf{x}-\tilde{\mathbf{z}}_i\|_2^2}\,,$$

where the second equality follows from the fact that both the oblique manifold constraint and the Frobenius norm are amenable to column-wise separation. The solution for the ith column is $\mathbf{z}_i = \frac{\tilde{\mathbf{z}}_i}{\|\tilde{\mathbf{z}}_i\|_2}$, which corresponds to finding the point on the unit sphere closest to the vector $\tilde{\mathbf{z}}_i$. This is equivalent to setting $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n] = \tilde{\mathbf{Z}}\mathbf{D}$, where $\mathbf{D}$ is a diagonal matrix with entries $D_{i,i} = \frac{1}{\|\tilde{\mathbf{z}}_i\|_2} = \frac{1}{\sqrt{\tilde{K}_{i,i}}}$, which yields the closest matrix in the elliptope as $\mathbf{D}\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}}\mathbf{D} = \mathbf{D}\tilde{\mathbf{K}}\mathbf{D} \in \mathcal{E}$.

The distance itself is given as

$$\begin{aligned} d_B\left(\tfrac{1}{n}\mathbf{D}\tilde{\mathbf{K}}\mathbf{D}, \tfrac{1}{n}\tilde{\mathbf{K}}\right) &= \sqrt{2 - 2\left\|\tfrac{1}{n}\tilde{\mathbf{Z}}\mathbf{D}\tilde{\mathbf{Z}}^\top\right\|_*} = \sqrt{2 - \frac{2}{n}\mathrm{tr}\left(\mathbf{D}\tilde{\mathbf{Z}}^\top\tilde{\mathbf{Z}}\right)} \\ &= \sqrt{2 - \frac{2}{n}\sum_{i\in[n]}\|\tilde{\mathbf{z}}_i\|_2} = \sqrt{\frac{2}{n}\sum_{i\in[n]}\left(1-(\tilde{K}_{i,i})^{\frac{1}{2}}\right)}\,. \end{aligned}$$

■

This simple normalization can be contrasted with the alternating projection algorithm required to find the closest correlation matrix in terms of the Euclidean distance (Higham, 2002). The latter could be considered more general, since it can be applied to any square matrix, whereas the Bures distance is only applicable to positive semidefinite matrices. Nonetheless, the symmetric normalization can be applied after first finding the nearest positive semidefinite matrix (Higham, 1988).

Another property of the Bures distance, is that its square is a convex function, which follows from the following theorem for real-valued positive-semidefinite matrices, which is a special case of the proof by Uhlmann (1976).

Theorem 3 *The squared fidelity measure* $C^2(\mathbf{A}, \mathbf{B}) = \|\mathbf{A}^{\frac{1}{2}}\mathbf{B}^{\frac{1}{2}}\|_*^2$ *is concave for positive-semidefinite matrices.*

Proof Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{R}^{n\times n}$ be three positive-semidefinite matrices. Let $\Psi_Z = \bigoplus_{i=1}^n \psi_{z_i} \in \mathcal{H}_X$ and $\Psi_X = \bigoplus_{i=1}^n \psi_{x_i} \in \mathcal{H}_X$ denote Hilbert-space embeddings defined via the direct sum $\mathcal{H}_X = \mathcal{H}_x \oplus \cdots \oplus \mathcal{H}_x = \bigoplus_{i=1}^n \mathcal{H}_x$ such that $Z_{i,j} = \langle\psi_{z_i}, \psi_{z_j}\rangle, X_{i,j} = \langle\psi_{x_i}, \psi_{x_j}\rangle$, and $\|\mathbf{Z}^{\frac{1}{2}}\mathbf{X}^{\frac{1}{2}}\|_* = \langle\Psi_Z, \Psi_X\rangle$. Likewise, let $\Phi_Z = \bigoplus_{i=1}^n \phi_{z_i} \in \mathcal{H}_Y$ and $\Psi_Y = \bigoplus_{i=1}^n \phi_{y_i} \in \mathcal{H}_Y$ denote Hilbert-space embeddings such that $Z_{i,j} = \langle\phi_{z_i}, \phi_{z_j}\rangle, Y_{i,j} = \langle\phi_{y_i}, \phi_{y_j}\rangle$, and $\|\mathbf{Z}^{\frac{1}{2}}\mathbf{Y}^{\frac{1}{2}}\|_* = \langle\Phi_Z, \Phi_Y\rangle$.

For any λ_X, λ_Y such that $\lambda_X^2 + \lambda_Y^2 = 1$, define

$$\Omega_Z = \bigoplus_{i=1}^n \omega_{z_i} = (\lambda_X\Psi_Z) \oplus (\lambda_Y\Phi_Z) \in \mathcal{H}_X \oplus \mathcal{H}_Y \equiv \mathcal{H}_W.$$

Then $Z_{i,j} = \langle\omega_{z_i}, \omega_{z_j}\rangle = \lambda_X^2\langle\psi_{z_i}, \psi_{z_j}\rangle + \lambda_Y^2\langle\phi_{z_i}, \phi_{z_j}\rangle$. Let $\mathbf{W} = \alpha\mathbf{X} + (1-\alpha)\mathbf{Y}$ denote a convex combination, where $0 \le \alpha \le 1$. Then $\Omega_W = (\sqrt{\alpha}\Psi_X) \oplus (\sqrt{1-\alpha}\Phi_Y) \in \mathcal{H}_W$ is a Hilbert-space embedding of $\mathbf{W}$. Consequently, $\langle\Omega_Z, \Omega_W\rangle = \lambda_X\sqrt{\alpha}\langle\Psi_Z, \Psi_X\rangle + \lambda_Y\sqrt{1-\alpha}\langle\Phi_Z, \Phi_Y\rangle$.

From Theorem 1 it follows that

$$\|\mathbf{Z}^{\frac{1}{2}}\mathbf{W}^{\frac{1}{2}}\|_*^2 \ge \left|\langle\Omega_Z, \Omega_W\rangle\right|^2 = \left|\lambda_X\sqrt{\alpha}\langle\Psi_Z, \Psi_X\rangle + \lambda_Y\sqrt{1-\alpha}\langle\Phi_Z, \Phi_Y\rangle\right|^2.$$

Maximizing over λ_X, λ_Y on the right hand side (which corresponds to an optimization along the positive quadrant of the circle parametrically defined by coordinates λ_X and λ_Y) we obtain

$$\|\mathbf{Z}^{\frac{1}{2}}\mathbf{W}^{\frac{1}{2}}\|_*^2 \ge \left|\sqrt{\alpha}\langle\Psi_Z, \Psi_X\rangle\right|^2 + \left|\sqrt{1-\alpha}\langle\Phi_Z, \Phi_Y\rangle\right|^2 = \alpha\|\mathbf{Z}^{\frac{1}{2}}\mathbf{X}^{\frac{1}{2}}\|_*^2 + (1-\alpha)\|\mathbf{Z}^{\frac{1}{2}}\mathbf{Y}^{\frac{1}{2}}\|_*^2.$$

■

Appendix B. Derivation of Informativeness Measures

In this appendix, we detail the derivation of the analytic expression of each measure of informativeness found in Table 2.

B.1 Euclidean Distance

We define the measure of informativeness for any correlation matrix $\mathbf{K} \in \mathcal{E}$ using the Euclidean distance as $d_{\mathcal{N}}(\mathbf{K}) = \frac{1}{n}\min_{\mathbf{N}_a\in\mathcal{N}}\|\mathbf{K} - \mathbf{N}_a\|_F$. Expanding the Frobenius norm term yields

$$\begin{aligned} &\|\mathbf{K} - \mathbf{N}_a\|_F^2 = \|\mathbf{K}\|_F^2 + \|\mathbf{N}_a\|_F^2 - 2\langle\mathbf{K}, \mathbf{N}_a\rangle, \\ &\|\mathbf{N}_a\|_F^2 = (an)^2 + (1-a)^2\frac{n^2}{n-1}, \\ &\langle\mathbf{K}, \mathbf{N}_a\rangle = a\mathbf{1}^\top\mathbf{K}\mathbf{1} + (1-a)\frac{n}{n-1}\mathrm{tr}(\mathbf{H}\mathbf{K}) = a\bar{k}n^2 + (1-a)(1-\bar{k})\frac{n^2}{n-1}, \end{aligned}$$

since $\mathrm{tr}(\mathbf{H}\mathbf{K}) = n - \frac{1}{n}\mathrm{tr}(\mathbf{J}\mathbf{K}) = n(1-\bar{k})$, where $\bar{k} = \frac{1}{n^2}\mathbf{1}^\top\mathbf{K}\mathbf{1}$. Taking the derivative with respect to a yields a linear function of a whose root is $\bar{k}$. That is, the value of the parameter

a that defines the closest non-informative matrix $\mathbf{N}_{a^\star}$ is

$$a^\star = \underset{0\le a\le 1}{\arg\min}\ \|\mathbf{K}-\mathbf{N}_a\|_F^2 = \bar{k}, \tag{B.1}$$

and

$$\|\mathbf{K}-\mathbf{N}_{\bar{k}}\|_F^2 = \|\mathbf{K}\|_F^2 - (n\bar{k})^2 - (1-\bar{k})^2\frac{n^2}{n-1} = \|\mathbf{K}\|_F^2 - \frac{n^2}{n-1}(n\bar{k}^2 - 2\bar{k} + 1).$$

The informativeness measure is

$$d_{\mathcal{N}}(\mathbf{K}) = \sqrt{\frac{1}{n^2}\|\mathbf{K}\|_F^2 - \frac{1}{n-1}(n\bar{k}^2 - 2\bar{k} + 1)}. \tag{B.2}$$

This expression is a specific case of the formula for the Euclidean distance of an arbitrary matrix to the closest correlation matrix with uniform off-diagonal elements (Borsdorf et al., 2010).

B.2 Cosine Distance

Using the cosine similarity, one can define a chordal distance, which amounts to Euclidean distance after normalizing the matrices by their Frobenius norm. By squaring the chordal distance and dividing by two, we obtain a measure of informativeness as $1-\max_{0\le a\le 1}\frac{\langle \mathbf{K},\mathbf{N}_a\rangle}{\|\mathbf{K}\|_F\|\mathbf{N}_a\|_F}$. For a specific value of a and $\mathbf{K}\in\mathcal{E}$ we have

$$\frac{\langle \mathbf{K},\mathbf{N}_a\rangle}{\|\mathbf{K}\|_F\|\mathbf{N}_a\|_F} = \frac{a\mathbf{1}^\top\mathbf{K}\mathbf{1} + (1-a)\frac{n}{n-1}\mathrm{tr}(\mathbf{H}\mathbf{K})}{\|\mathbf{K}\|_F\sqrt{(an)^2 + (1-a)^2\frac{n^2}{n-1}}} = \frac{a\bar{k}n + (1-a)(1-\bar{k})\frac{n}{n-1}}{\|\mathbf{K}\|_F\sqrt{(a^2n - 2a + 1)\frac{1}{n-1}}}.$$

Taking the derivative of this expression and solving for its root, yields

$$a^\star = \underset{0\le a\le 1}{\arg\max}\ \frac{\langle \mathbf{K},\mathbf{N}_a\rangle}{\|\mathbf{K}\|_F\|\mathbf{N}_a\|_F} = \bar{k}. \tag{B.3}$$

Substituting $a^\star$ back into the cosine similarity, leads to

$$\max_{0\le a\le 1}\frac{\langle \mathbf{K},\mathbf{N}_a\rangle}{\|\mathbf{K}\|_F\|\mathbf{N}_a\|_F} = \frac{\langle \mathbf{K},\mathbf{N}_{a^\star}\rangle}{\|\mathbf{K}\|_F\|\mathbf{N}_{a^\star}\|_F} = \frac{n\bar{k}^2 + (1-\bar{k})^2\frac{n}{n-1}}{\|\mathbf{K}\|_F\sqrt{(\bar{k}^2n - 2\bar{k} + 1)\frac{1}{n-1}}} = \frac{n}{\|\mathbf{K}\|_F}\sqrt{\frac{n\bar{k}^2 - 2\bar{k} + 1}{n-1}}.$$

The measure of informativeness based on the chordal distance is

$$\frac{1}{2}d_{\mathcal{N}}^2(\mathbf{K}) = 1 - \frac{n}{\|\mathbf{K}\|_F}\sqrt{\frac{n\bar{k}^2 - 2\bar{k} + 1}{n-1}}. \tag{B.4}$$

B.3 Hilbert-Schmidt Independence Criterion (HSIC)

The use of HSIC as a measure of similarity between a correlation matrix and the closest non-informative one leads to a trivial function. Specifically, for any a we have

$$\mathrm{HSIC}(\mathbf{K},\mathbf{N}_a) = \langle \mathbf{H}\mathbf{K}\mathbf{H},\mathbf{N}_a\rangle = \mathrm{tr}(\mathbf{H}\mathbf{K}\mathbf{H}\mathbf{N}_a) = \frac{(1-a)n}{n-1}\mathrm{tr}(\mathbf{H}\mathbf{K}).$$

This similarity is maximized for $a=0$ and the informativeness would be inversely proportional to $\mathrm{tr}(\mathbf{H}\mathbf{K}) = n(1-\bar{k})$ for $\mathbf{K}\in\mathcal{E}$, where $\bar{k}$ is as defined previously to be the average of the entries of $\mathbf{K}$.

A more meaningful measure of informativeness, however, is based on the Euclidean distance after centering (see Table 1). This is defined as

$$d_{\mathcal{N}}(\mathbf{K}) = \frac{1}{n}\min_{\mathbf{N}_a\in\mathcal{N}}\|\mathbf{H}\mathbf{K}\mathbf{H} - \mathbf{H}\mathbf{N}_a\mathbf{H}\|_F = \frac{1}{n}\min_{\mathbf{N}_a\in\mathcal{N}}\left\|\mathbf{H}\mathbf{K}\mathbf{H} - \frac{(1-a)n}{n-1}\mathbf{H}\right\|_F.$$

Expanding the norm we have

$$\left\|\mathbf{H}\mathbf{K}\mathbf{H} - \frac{(1-a)n}{n-1}\mathbf{H}\right\|_F^2 = \|\mathbf{H}\mathbf{K}\mathbf{H}\|_F^2 + \frac{(1-a)^2n^2}{n-1} - 2\frac{(1-a)n}{n-1}\mathrm{tr}(\mathbf{H}\mathbf{K}).$$

Taking the derivative of this expression and solving for its root yields

$$a^\star = \underset{0\le a\le 1}{\arg\min}\ \|\mathbf{H}\mathbf{K}\mathbf{H} - \mathbf{H}\mathbf{N}_a\mathbf{H}\|_F^2 = \bar{k}, \tag{B.5}$$

and the measure of informativeness is

$$d_{\mathcal{N}}(\mathbf{K}) = \sqrt{\frac{1}{n^2}\|\mathbf{H}\mathbf{K}\mathbf{H}\|_F^2 - \frac{1}{n-1}(1-\bar{k})^2}. \tag{B.6}$$

B.4 Centered Kernel Alignment (CKA)

The CKA similarity measurement between kernel matrices can be used to define a chordal distance between centered kernel matrices. By squaring the chordal distance and dividing by two, we define the measure of informativeness as

$$\begin{aligned}\frac{1}{2}d_{\mathcal{N}}^2(\mathbf{K}) &= 1 - \max_{0\le a\le 1}\frac{\langle \mathbf{H}\mathbf{K}\mathbf{H},\mathbf{N}_a\rangle}{\|\mathbf{H}\mathbf{K}\mathbf{H}\|_F\|\mathbf{H}\mathbf{N}_a\mathbf{H}\|_F} = 1 - \max_{0\le a\le 1}\frac{\frac{(1-a)n}{n-1}\mathrm{tr}(\mathbf{H}\mathbf{K})}{\|\mathbf{H}\mathbf{K}\mathbf{H}\|_F\frac{(1-a)n}{n-1}\|\mathbf{H}\|_F}\\ &= 1 - \frac{\mathrm{tr}(\mathbf{H}\mathbf{K})}{\|\mathbf{H}\mathbf{K}\mathbf{H}\|_F\sqrt{n-1}} = 1 - \frac{n(1-\bar{k})}{\|\mathbf{H}\mathbf{K}\mathbf{H}\|_F\sqrt{n-1}},\end{aligned} \tag{B.7}$$

where $\bar{k}$ is as defined before, and $\mathrm{tr}(\mathbf{H}\mathbf{K}) = n(1-\bar{k})$ for $\mathbf{K}\in\mathcal{E}$. The measure is undefined for the constant matrix $\mathbf{K}=\mathbf{N}_1=\mathbf{J}$, because $\|\mathbf{H}\mathbf{K}\mathbf{H}\|_F = 0$, but since $\mathbf{J}$ is non-informative, the informativeness measure in this case should be defined to be zero. Furthermore, for a rank-1 matrix $\mathbf{K}=\mathbf{v}\mathbf{v}^\top$, this measure is constant with value of $1-\frac{1}{\sqrt{n-1}}$, since $\mathrm{tr}(\mathbf{H}\mathbf{K}) = \mathbf{v}^\top\mathbf{H}\mathbf{v} = \sqrt{\mathrm{tr}(\mathbf{v}^\top\mathbf{H}\mathbf{v}\mathbf{v}^\top\mathbf{H}\mathbf{v})} = \|\mathbf{H}\mathbf{v}\mathbf{v}^\top\mathbf{H}\|_F = \|\mathbf{H}\mathbf{K}\mathbf{H}\|_F$.

B.5 Chernoff Bound

The quantum Chernoff bound is a lower bound on both the affinity and fidelity, and it can thus be used to define an upper bound on a function of the quantum Hellinger and Bures distances (Audenaert et al., 2007). We define a measure of informativeness based on the Chernoff bound $Q(\frac{1}{n}\mathbf{K},\frac{1}{n}\mathbf{N}_a) = \min_{0\le s\le 1}\frac{1}{n}\langle \mathbf{K}^s,\mathbf{N}_a^{1-s}\rangle$, for $\mathbf{K}\in\mathcal{E}$ as

$$d_{\mathcal{N}}(\mathbf{K}) = 1 - \max_{0\le a\le 1}Q(\tfrac{1}{n}\mathbf{K},\tfrac{1}{n}\mathbf{N}_a) = 1 - \max_{0\le a\le 1}\min_{0\le s\le 1}f(\mathbf{K},a,s),$$

where

$$f(\mathbf{K}, a, s) \equiv \frac{(an)^{1-s}}{n}\text{tr}(\mathbf{K}^s\mathbf{J}) + \left(\frac{(1-a)n}{n-1}\right)^{1-s}\text{tr}(\mathbf{K}^s\mathbf{H}). \tag{B.8}$$

For a fixed s, the value of a that maximizes $f(\mathbf{K}, a, s)$ can be found analytically. When $\text{tr}(\mathbf{KJ}) = 0$, letting $a = 0$ maximizes the quantity $f(\mathbf{K}, a, s)$. Otherwise, taking the derivative with respect to a, yields

$$\frac{\partial f(\mathbf{K}, s, a)}{\partial a} = \frac{1-s}{n^{s+1}}a^{-s}\text{tr}(\mathbf{K}^s\mathbf{J}) - \frac{1-s}{n-1}\left(\frac{(1-a)n}{n-1}\right)^{-s}\text{tr}(\mathbf{K}^s\mathbf{H}).$$

Equating this quantity to zero, rearranging, and taking the sth root of both sides, gives the maximizing value defined as

$$a_s^* \equiv \left(1 + \sqrt[s]{\frac{\text{tr}(\mathbf{K}^s\mathbf{H})}{\text{tr}(\mathbf{K}^s\mathbf{J})}\frac{n}{(n-1)^{1-s}}}\right)^{-1}, \tag{B.9}$$

for $\text{tr}(\mathbf{K}^s\mathbf{J}) > 0$. Now defining

$$q(\mathbf{K}) \equiv \min_{0\le s\le 1} f(\mathbf{K}, s, a_s^*) = \min_{0\le s\le 1}\max_{0\le a\le 1}\langle \mathbf{K}^s, \mathbf{N}_a^{1-s}\rangle, \tag{B.10}$$

the measure of informativeness using the Chernoff bound is

$$d_{\mathcal{N}}(\mathbf{K}) = 1 - q(\mathbf{K}). \tag{B.11}$$

By precomputing the eigenvalue decomposition of $\mathbf{K} = \mathbf{U}\Lambda\mathbf{U}^\top$, each evaluation of $f(\mathbf{K}, s, a_s^*)$ requires computations on the order of $\mathcal{O}(n)$ (Mendonça et al., 2008). Specifically, $\text{tr}(\mathbf{K}^s\mathbf{J}) = \text{tr}(\Lambda^s\mathbf{U}^\top\mathbf{1}\mathbf{1}^\top\mathbf{U}) = \sum_i \lambda_i^s\langle\mathbf{u}_i, \mathbf{1}\rangle^2$, and $\text{tr}(\mathbf{K}^s\mathbf{H}) = \sum_i \lambda_i^s(1 - \frac{1}{n}\langle\mathbf{u}_i, \mathbf{1}\rangle^2)$. By consequence, the complexity of computing $q(\mathbf{K})$ via a line search on s, is the same as computing the Chernoff bound.

B.6 Quantum Hellinger (QH)

The quantum version of Hellinger distance shares some of the same properties as the Bures distance (Luo and Zhang, 2004). Squaring the quantum Hellinger distance (a chordal distance on the space of trace-normalized positive semidefinite matrices) and dividing by two yields a measure of informativeness for correlation matrices $\mathbf{K} \in \mathcal{E}$ as

$$\frac{1}{2}d_{\mathcal{N}}^2(\mathbf{K}) = \min_{0\le a\le 1} 1 - \frac{1}{n}\langle\sqrt{\mathbf{K}}, \sqrt{\mathbf{N}_a}\rangle.$$

Firstly, we expand the affinity term as

$$\frac{1}{\sqrt{n}}\langle\sqrt{\mathbf{K}}, \sqrt{\mathbf{N}_a}\rangle = \langle\sqrt{\mathbf{K}}, \tfrac{\sqrt{a}}{n}\mathbf{J}\rangle + \langle\sqrt{\mathbf{K}}, \sqrt{\tfrac{1-a}{n-1}}\mathbf{H}\rangle = \frac{\sqrt{a}}{n}\mathbf{1}^\top\sqrt{\mathbf{K}}\mathbf{1} + \frac{\sqrt{1-a}}{\sqrt{n-1}}\text{tr}(\mathbf{H}\sqrt{\mathbf{K}}).$$

Maximizing the latter, corresponds to an optimization along the positive quadrant of the circle parametrically defined by coordinates $\sqrt{a}$ and $\sqrt{1-a}$. The maximizing value is

$$a^\star = \arg\max_{0\le a\le 1}\langle\sqrt{\mathbf{K}}, \sqrt{\mathbf{N}_a}\rangle = \frac{\left(\frac{1}{n}\mathbf{1}^\top\sqrt{\mathbf{K}}\mathbf{1}\right)^2}{\left(\frac{1}{n}\mathbf{1}^\top\sqrt{\mathbf{K}}\mathbf{1}\right)^2 + \frac{1}{n-1}\text{tr}^2(\mathbf{H}\sqrt{\mathbf{K}})}. \tag{B.12}$$

Substituting $a^\star$ back into the expression of affinity leads to

$$\max_{0\le a\le 1}\langle\sqrt{\mathbf{K}}, \sqrt{\mathbf{N}_a}\rangle = \langle\sqrt{\mathbf{K}}, \sqrt{\mathbf{N}_{a^\star}}\rangle = \sqrt{\left(\tfrac{1}{n}\mathbf{1}^\top\sqrt{\mathbf{K}}\mathbf{1}\right)^2 + \tfrac{1}{n-1}\text{tr}^2(\mathbf{H}\sqrt{\mathbf{K}})},$$

and the measure of informativeness is

$$\frac{1}{2}d_{\mathcal{N}}^2(\mathbf{K}) = 1 - \sqrt{\tfrac{1}{n^3}(\mathbf{1}^\top\sqrt{\mathbf{K}}\mathbf{1})^2 + \tfrac{1}{n(n-1)}\text{tr}^2(\mathbf{H}\sqrt{\mathbf{K}})}. \tag{B.13}$$

B.7 Bures

By employing the Bures distance d_B for trace-normalized matrices in Equation 3, we can define a measure of informativeness of an arbitrary correlation matrix $\mathbf{K} \in \mathcal{E}$ as

$$\frac{1}{2}d_{\mathcal{N}}^2(\mathbf{K}) = \frac{1}{2}\min_{\mathbf{N}\in\mathcal{N}} d_B^2\left(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\mathbf{N}\right) = 1 - 1\max_{0\le a\le 1}\tfrac{1}{n}\left\|\sqrt{\mathbf{K}}\sqrt{\mathbf{N}_a}\right\|_*. \tag{B.14}$$

Since $0 \le \frac{1}{n}\left\|\sqrt{\mathbf{K}}\sqrt{\mathbf{N}_a}\right\|_* \le 1$ (discussed in Appendix A), this distance is bounded as $0 \le d_{\mathcal{N}}^2(\mathbf{K}) \le 2$, with $d_{\mathcal{N}}(\mathbf{K}) = 0$ when $\mathbf{K} \in \mathcal{N}$.

We now define for convenience the function

$$f(\mathbf{K}, a) \equiv \frac{1}{\sqrt{n}}\left\|\sqrt{\mathbf{K}}\sqrt{\mathbf{N}_a}\right\|_* = \left\|\tfrac{\sqrt{a}}{n}\sqrt{\mathbf{K}}\mathbf{J} + \sqrt{\tfrac{1-a}{n-1}}\sqrt{\mathbf{K}}\mathbf{H}\right\|_*,$$

where the right hand side is obtained by using the equation $\sqrt{\mathbf{N}_a} = \sqrt{\frac{a}{n}}\mathbf{J} + \sqrt{\frac{(1-a)n}{n-1}}\mathbf{H}$ introduced in Section 3.1. We note that $\frac{1}{n}f^2(\mathbf{K}, a) = C^2(\frac{1}{n}\mathbf{K}, \frac{1}{n}\mathbf{N}_a)$, where $C^2(\cdot,\cdot)$ is the concave fidelity measure (Jozsa, 1994) discussed in Appendix A. The function simplifies for $a = 0$ and $a = 1$, yielding the following two quantities that will be useful for subsequent calculations

$$f^2(\mathbf{K}, 1) = \frac{1}{n^2}\left\|\sqrt{\mathbf{K}}\mathbf{J}\right\|_*^2 = \frac{1}{n^2}\text{tr}^2\sqrt{\mathbf{JKJ}} = \frac{1}{n}\mathbf{1}^\top\mathbf{K}\mathbf{1}\,,$$
$$f^2(\mathbf{K}, 0) = \frac{1}{n-1}\left\|\sqrt{\mathbf{K}}\mathbf{H}\right\|_*^2 = \frac{1}{n-1}\text{tr}^2\sqrt{\mathbf{HKH}}\,.$$

For general correlation matrices $\mathbf{K}$, exact computation of the informativeness $d_{\mathcal{N}}(\mathbf{K})$ requires finding the value of a that maximizes $f(\mathbf{K}, a)$, or equivalently $f^2(\mathbf{K}, a)$. Since the latter expression is concave, this concave maximization is tractable and can be done using line search. However, each evaluation of $f(\mathbf{K}, a)$ requires the calculation of a trace norm, and therefore the singular values of an $n \times n$ matrix; for large matrices, this would be a costly search. Consequently, as a surrogate, we seek a lower bound on $d_{\mathcal{N}}(\mathbf{K})$, or equivalently an upper bound on $\max_a\|\sqrt{\mathbf{K}}\sqrt{\mathbf{N}_a}\|_*$. One such useful bound can be found by simply employing the sub-additive property $\|\mathbf{A} + \mathbf{B}\|_* \le \|\mathbf{A}\|_* + \|\mathbf{B}\|_*$ for matrices $\mathbf{A}$ and $\mathbf{B}$ of equal size. Thus, for any $a \in [0, 1]$, we have

$$\begin{aligned} f(\mathbf{K}, a) &\le \frac{\sqrt{a}}{n}\left\|\sqrt{\mathbf{K}}\mathbf{J}\right\|_* + \sqrt{\frac{1-a}{n-1}}\left\|\sqrt{\mathbf{K}}\mathbf{H}\right\|_* = \sqrt{\frac{a}{n}\mathbf{1}^\top\mathbf{K}\mathbf{1}} + \sqrt{\frac{1-a}{n-1}}\left\|\sqrt{\mathbf{K}}\mathbf{H}\right\|_* \\ &= \sqrt{a}\, f(\mathbf{K}, 1) + \sqrt{1-a}\, f(\mathbf{K}, 0) \;\equiv\; \overline{f}(\mathbf{K}, a). \end{aligned} \tag{B.15}$$

Additionally, we have $\max_a f(\mathbf{K},a) \le \max_a \overline{f}(\mathbf{K},a)$. As the maximization of $\overline{f}(\mathbf{K},a)$ corresponds to a linear program constrained to a circle parameterized by coordinates $\sqrt{a}$ and $\sqrt{1-a}$, the unique optimizing value of a is given as

$$a^* = \operatorname*{arg\,max}_{0\le a\le 1} \overline{f}(\mathbf{K},a) = \frac{f^2(\mathbf{K},1)}{f^2(\mathbf{K},1)+f^2(\mathbf{K},0)} = \frac{\mathbf{1}^\top\mathbf{K}\mathbf{1}}{\mathbf{1}^\top\mathbf{K}\mathbf{1}+\frac{n}{n-1}\mathrm{tr}^2\sqrt{\mathbf{H}\mathbf{K}\mathbf{H}}}. \tag{B.16}$$

Substituting a^* into Equation B.15 yields $\overline{f}(\mathbf{K},a^*) = \sqrt{f^2(\mathbf{K},1)+f^2(\mathbf{K},0)}$. This finally provides a lower bound for $d^2_{\mathcal{N}}(\mathbf{K})$ as

$$\begin{aligned}\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) &\ge 1-\frac{1}{\sqrt{n}}\sqrt{f^2(\mathbf{K},1)+f^2(\mathbf{K},0)}\\ &= 1-\sqrt{\tfrac{1}{n^2}\mathbf{1}^\top\mathbf{K}\mathbf{1}+\tfrac{1}{n(n-1)}\mathrm{tr}^2\sqrt{\mathbf{H}\mathbf{K}\mathbf{H}}} \;\equiv\; i(\mathbf{K}).\end{aligned} \tag{B.17}$$

We refer to $i(\mathbf{K})$ as the *Bures-based informativeness*, since $\sqrt{2i(\mathbf{K})}$ is a lower bound on the Bures distance between $\mathbf{K}$ and the closest member in the set of non-informative correlation matrices $\mathcal{N}$.

Theorem 4 *The Bures-based informativeness $i(\mathbf{K})$ is convex. For $\mathbf{K}_1,\mathbf{K}_0 \in \mathcal{E}$ and $0 \le \alpha \le 1$, $i(\alpha\mathbf{K}_1+(1-\alpha)\mathbf{K}_0) \le \alpha i(\mathbf{K}_1)+(1-\alpha)i(\mathbf{K}_0)$.*

Proof We write the Bures-based informativeness as the composition $i(\mathbf{K}) = g(f(\mathbf{K}))$ where $f(\mathbf{K}) = \frac{1}{n^2}\mathbf{1}^\top\mathbf{K}\mathbf{1}+\frac{1}{n(n-1)}\mathrm{tr}^2\sqrt{\mathbf{H}\mathbf{K}\mathbf{H}}$ and $g(u) = 1-\sqrt{u}, \quad 0 \le u$. The function $f(\mathbf{K}) = \langle \mathbf{K}, \frac{1}{n^2}\mathbf{J}\rangle + \frac{1}{n(n-1)}\|\sqrt{\mathbf{K}}\sqrt{\mathbf{H}}\|^2_*$ is concave as it is the sum of a linear function and the concave function $\frac{1}{n(n-1)}\|\sqrt{\mathbf{K}}\sqrt{\mathbf{H}}\|^2_*$, which is itself proportional to the concave fidelity function (Uhlmann, 1976; Jozsa, 1994) (see Theorem 3 in Appendix A). Furthermore, the function g is convex and nonincreasing. From these properties, it follows that

$$\begin{aligned} i(\alpha\mathbf{K}_1+(1-\alpha)\mathbf{K}_0) = g(f(\alpha\mathbf{K}_1+(1-\alpha)\mathbf{K}_0)) &\le g(\alpha f(\mathbf{K}_1)+(1-\alpha)f(\mathbf{K}_0))\\ &\le \alpha g(f(\mathbf{K}_1))+(1-\alpha)g(f(\mathbf{K}_0))\\ &= \alpha i(\mathbf{K}_1)+(1-\alpha)i(\mathbf{K}_0),\end{aligned}$$

where the first inequality follows from the concavity of f and the monotonicity of g, and the second inequality follows from the convexity of g. ■

Special case 1: If $\mathbf{K}$ is centered, that is $\mathbf{K}\mathbf{1} = \mathbf{0}$, we have $\sqrt{\mathbf{K}}\mathbf{1} = \mathbf{0}$ and $f(\mathbf{K},a) = \sqrt{\frac{1-a}{n-1}}\mathrm{tr}\sqrt{\mathbf{K}}$. Therefore, in this case we obtain a closed-form for Equation B.14, given as

$$\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = 1-\frac{1}{\sqrt{n(n-1)}}\mathrm{tr}\sqrt{\mathbf{K}}. \tag{B.18}$$

Thus, Equation B.17 yields an equality with $\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = i(\mathbf{K})$. Centered correlation matrices have zero-mean embeddings, that is, $\mathbf{Z}\mathbf{1} = \mathbf{0}$, for any $\mathbf{Z}$ such that $\mathbf{Z}^\top\mathbf{Z} = \mathbf{K}$.

Special case 2: If $\mathbf{K}$ is a rank-1, then $\left\|\sqrt{\mathbf{K}}\sqrt{\mathbf{N}_a}\right\|_* = \left\|\sqrt{\mathbf{K}}\sqrt{\mathbf{N}_a}\right\|_F = \sqrt{\mathrm{tr}(\mathbf{N}_a\mathbf{K})}$. In this case, we have

$$f^2(\mathbf{K},a) = \frac{a}{n}\mathbf{1}^\top\mathbf{K}\mathbf{1}+\frac{1-a}{n-1}\mathrm{tr}(\mathbf{H}\mathbf{K}) = \frac{an^2(\bar{k}-\frac{1}{n})}{n-1}+\frac{n(1-\bar{k})}{n-1},$$

where $\bar{k} = \frac{1}{n^2}\mathbf{1}^\top\mathbf{K}\mathbf{1}$ is the average of all the entries of $\mathbf{K}$, and $\mathrm{tr}(\mathbf{H}\mathbf{K}) = n(1-\bar{k})$. This shows that $f^2(\mathbf{K},a)$ is linear with respect to a, and will be maximized by the extreme values 1 or 0 of a (depending on whether or not, $\bar{k}$ is greater than $\frac{1}{n}$). Substituting the expression of $f(\mathbf{K},a)$ into Equation B.14 and simplifying yields

$$\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = 1-\sqrt{\max\left(\bar{k},\frac{1-\bar{k}}{n-1}\right)}. \tag{B.19}$$

Similarly, when $\mathbf{K}$ is rank-1, then $\mathrm{tr}^2\sqrt{\mathbf{H}\mathbf{K}\mathbf{H}} = \mathrm{tr}(\mathbf{H}\mathbf{K}\mathbf{H}) = n(1-\bar{k})$, and Equation B.17 becomes

$$i(\mathbf{K}) = 1-\sqrt{\bar{k}+\frac{1-\bar{k}}{n-1}} = 1-\sqrt{\frac{\bar{k}(n-2)+1}{n-1}}. \tag{B.20}$$

Rank-1 correlation matrices $\mathbf{K} = \mathbf{v}\mathbf{v}^\top$ with $\mathbf{v} \in \{\pm 1\}^n$ correspond to vertices of the elliptope (discussed in Section 2.1). Using the cut vector $\mathbf{v}$, it is easy to see that $\mathbf{1}^\top\mathbf{K}\mathbf{1} = (\mathbf{v}^\top\mathbf{1})^2 = (2c-n)^2 = 4(c-\frac{n}{2})^2$, where c is the number of positive entries of $\mathbf{v}$. Thus, Equation B.19 can be directly expressed in terms of c as

$$\frac{1}{2}d^2_{\mathcal{N}}(\mathbf{K}) = \begin{cases} 1-\frac{2}{n}\sqrt{\frac{c(n-c)}{n-1}}, & \text{if } c \in \left[\frac{n-\sqrt{n}}{2},\frac{n+\sqrt{n}}{2}\right],\\ 1-\frac{2}{n}\left|c-\frac{n}{2}\right|, & \text{otherwise,}\end{cases}$$

and likewise Equation B.20 can be expressed as

$$i(\mathbf{K}) = 1-\frac{2}{n}\sqrt{\left(c-\frac{n}{2}\right)^2\frac{n-2}{n-1}+\frac{n^2}{4(n-1)}}.$$

Figure 15 compares $\sqrt{2i(\mathbf{K})}$ and $d_{\mathcal{N}}(\mathbf{K})$ for rank-1 matrices. For this case, $i(\mathbf{K})$ provides a tight lower bound except near the balanced cut.

Finally, we note the following result for the Bures-based informativeness.

Theorem 5 *For $n > 2$ and even, the Bures-based informativeness is maximized by rank-1 correlation matrices that are centered. That is, if $\mathbf{K}^\star \in \mathbb{R}^{n\times n}$ is a rank-1 correlation matrix with $\bar{k}^\star = \frac{1}{n^2}\mathbf{1}^\top\mathbf{K}^\star\mathbf{1} = 0$, then $i(\mathbf{K}^\star) = \max_{\mathbf{K}\in\mathcal{E}} i(\mathbf{K})$.*

Proof From Equation B.20, it follows that $i(\mathbf{K}^\star) = 1-\sqrt{\frac{1}{n-1}}$. Maximizing $i(\mathbf{K})$ over $\mathbf{K} \in \mathcal{E}$ corresponds to minimizing the expression $\bar{k}+\frac{1}{n(n-1)}\|\mathbf{H}\sqrt{\mathbf{K}}\|^2_*$. Using the matrix norm inequality $\|\mathbf{A}\|_* \ge \|\mathbf{A}\|_F$, it follows that

$$\frac{1}{n(n-1)}\left\|\mathbf{H}\sqrt{\mathbf{K}}\right\|^2_* \ge \frac{1}{n(n-1)}\left\|\mathbf{H}\sqrt{\mathbf{K}}\right\|^2_F = \frac{1}{n(n-1)}\mathrm{tr}\left(\mathbf{H}\mathbf{K}\right) = \frac{1-\bar{k}}{n-1}.$$

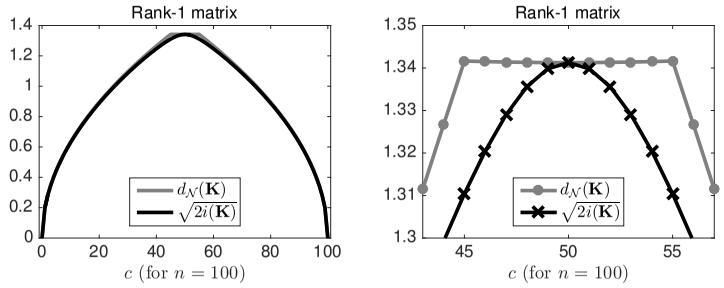


Figure 15: Comparison of the distance to the non-informative matrix family $d_{\mathcal{N}}(\mathbf{K})$ using the exact Bures distance d_B versus the lower bound $\sqrt{2i(\mathbf{K})}$ for rank-1 matrices $\mathbf{K} = \mathbf{v}\mathbf{v}^\top$, where c is the number of positive elements in the cut vector $\mathbf{v}$. The plot on the right magnifies the area around a balanced cut.

Substituting this expression back into the informativeness measure and maximizing over the possible values of $\bar{k}$, it follows that

$$i(\mathbf{K}) \leq 1 - \sqrt{\bar{k} + \frac{1-\bar{k}}{n-1}} \leq 1 - \sqrt{\frac{1}{n-1}} = i(\mathbf{K}^\star).$$

■

B.8 Sub-Bures Dissimilarity via Super-Fidelity

The modified fidelity or super-fidelity provides an upper bound of fidelity used in the Bures distance, and its computation has a lower complexity (Mendonça et al., 2008; Miszczak et al., 2009). Using super-fidelity, a lower bound on the Bures distance can be formed, which we refer to as the sub-Bures dissimilarity. (We use the term dissimilarity since it is not a metric, although a distance metric version can be obtained by removing the square root in the corresponding expression.) Using the sub-Bures dissimilarity, the informativeness is

$$\frac{1}{2} d^2_{\mathcal{N}}(\mathbf{K}) = 1 - \max_{0 \leq a \leq 1} \sqrt{G(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\mathbf{N}_a)}.$$

The super-fidelity between $\mathbf{K}$ and a non-informative matrix $\mathbf{N}_a$ is given by

$$\begin{aligned} G(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\mathbf{N}_a) &= \frac{1}{n^2}\langle \mathbf{K}, \mathbf{N}_a \rangle + \sqrt{\left(1 - \left\|\tfrac{1}{n}\mathbf{K}\right\|_F^2\right)\left(1 - \left\|\tfrac{1}{n}\mathbf{N}_a\right\|_F^2\right)} \\ &= \frac{a(n\bar{k}-1) + 1 - \bar{k}}{n-1} + \sqrt{\tfrac{n(1-a^2)+2(1-a)}{n-1}\left(1 - \left\|\tfrac{1}{n}\mathbf{K}\right\|_F^2\right)}, \end{aligned} \tag{B.21}$$

where $\bar{k}$ is as defined before, and $\left\|\frac{1}{n}\mathbf{N}_a\right\|_F^2 = a^2 + \frac{(1-a)^2}{n-1}$ was used to simplify the right hand side.

If $\mathbf{K}$ is rank-1, then $\|\frac{1}{n}\mathbf{K}\|_F = 1$ and the super-fidelity is proportional to $\langle \mathbf{K}, \mathbf{N}_a \rangle$. The inner product is maximized for $a = 1$ if $\bar{k} > \frac{1}{n}$, and $a = 0$ otherwise. In these two cases, super-fidelity obtains the values of $\bar{k}$ and $\frac{1-\bar{k}}{n-1}$, respectively. For more general cases, the maximizing value of a can be found using the derivative of super-fidelity given by

$$\frac{\partial G(\frac{1}{n}\mathbf{K}, \frac{1}{n}\mathbf{N}_a)}{\partial a} = \frac{\bar{k}n - 1}{n-1} + \frac{1 - an}{\sqrt{n(1-a^2) + 2(1-a)}} \sqrt{\frac{1 - \|\frac{1}{n}\mathbf{K}\|_F^2}{n-1}},$$

which vanishes when

$$\frac{\bar{k}n - 1}{n-1} = \frac{an - 1}{\sqrt{n(1-a^2) + 2(1-a)}} \sqrt{\frac{1 - \|\frac{1}{n}\mathbf{K}\|_F^2}{n-1}}.$$

Squaring both sides yields a quadratic expression whose roots are given by

$$a^{\pm} = \frac{b + cn \pm \sqrt{(cn+b)^2 - (cn^2 + bn)(c + b(2-n))}}{cn^2 + bn},$$

where $b = \frac{(\bar{k}n-1)^2}{(n-1)^2}$ and $c = \frac{1 - \|\frac{1}{n}\mathbf{K}\|_F^2}{n-1}$.

The super-fidelity is maximized when a is at one of its limits or by the root that satisfies $\operatorname{sign}(\bar{k} - \frac{1}{n}) = \operatorname{sign}(a - \frac{1}{n})$. Now, if we define $a_1 \equiv \min(1, a^+)$ and $a_0 \equiv \max(0, a^-)$, we have

$$a^\star = \operatorname*{arg\,max}_{0 \leq a \leq 1} G(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\mathbf{N}_a) = \begin{cases} a_1 & \text{if } G(\frac{1}{n}\mathbf{K}, \frac{1}{n}\mathbf{N}_{a_1}) \geq G(\frac{1}{n}\mathbf{K}, \frac{1}{n}\mathbf{N}_{a_0}) \\ a_0 & \text{otherwise.} \end{cases} \tag{B.22}$$

The maximal super-fidelity is

$$\max_{0 \leq a \leq 1} G(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\mathbf{N}_a) = \max\left(G(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\mathbf{N}_{a_1}),\ G(\tfrac{1}{n}\mathbf{K}, \tfrac{1}{n}\mathbf{N}_{a_0}) \right) \equiv g(\mathbf{K}), \tag{B.23}$$

and, hence, the sub-Bures measure of informativeness is

$$\frac{1}{2} d^2_{\mathcal{N}}(\mathbf{K}) = 1 - \sqrt{g(\mathbf{K})}. \tag{B.24}$$

Although this closed-form expression seems algebraically complex, it only depends on $\bar{k}$ and the norm of $\mathbf{K}$.

Appendix C. Statistics for Equality of Correlation Coefficients

In this appendix, we present Bartlett's and Lawley's test statistics for equality of correlation coefficients (Lawley, 1963) using notation consistent with the proposed measures of informativeness.

C.1 Lawley's Test Statistic

Lawley proposed a test statistic for equality of correlation coefficients (Lawley, 1963), which is a function of the Frobenius norm, element-wise mean, and variance of the row sums of a correlation matrix $\mathbf{K}$. The test assumes the correlation matrix is estimated from m observations of n-dimensional vectors. Under the null hypothesis the true correlation matrix is $\mathbf{N}_{a^*}$ for some value $0 \leq a^* \leq 1$. Letting $\lambda = \frac{n(1-a^*)}{n-1}$ and $\bar{k} = \frac{1}{n^2}\mathbf{1}^\top\mathbf{K}\mathbf{1}$, the test statistic can be written as

$$\begin{aligned} T_1(\mathbf{K}) &= \frac{m}{\lambda^2}\left(\frac{1}{2}\left\|\mathbf{K}-\mathbf{N}_{\bar{k}}\right\|_F^2 - \frac{1-\lambda^2}{n-\lambda^2(n-2)}\left\|\mathbf{K}\mathbf{1}-\mathbf{N}_{\bar{k}}\mathbf{1}\right\|_2^2\right) \\ &= \frac{m}{\lambda^2}\left(\frac{1}{2}\left(\|\mathbf{K}\|_F^2 - \frac{n^2}{n-1}(\bar{k}^2 n - 2\bar{k}+1)\right) - \frac{1-\lambda^2}{n-\lambda^2(n-2)}\left\|\mathbf{K}\mathbf{1}-n\bar{k}\mathbf{1}\right\|_2^2\right), \end{aligned} \tag{C.1}$$

where the simplification comes from $\|\mathbf{K}-\mathbf{N}_{\bar{k}}\|_F^2 = \|\mathbf{K}\|_F^2 - (n\bar{k})^2 - (1-\bar{k})^2\frac{n^2}{n-1} = \|\mathbf{K}\|_F^2 - \frac{n^2}{n-1}(n\bar{k}^2 - 2\bar{k}+1)$.

Asymptotically (as $m \to \infty$), the test statistic under the null hypothesis has a χ^2 distribution with $\frac{1}{2}(n+1)(n-2)$ degrees of freedom. In practice, a^* is unknown, but $\bar{k}$ can be used with $\lambda = \frac{n(1-\bar{k})}{n-1}$. This substitution does not affect the χ^2 limiting distribution (Lawley, 1963). Given only $\mathbf{K}$, m is unknown, and while it must be equal or greater than the rank of $\mathbf{K}$, we use the quantity $\frac{1}{m}T_1(\mathbf{K})$ for comparisons.

C.2 Bartlett's Asymptotic Test Statistic

Similarly, Bartlett's asymptotic test statistic (Lawley, 1963) is also a function of the Frobenius norm, element-wise mean, and variance of the rows sum of the given correlation matrix. Letting $\bar{k}$ and λ be defined as before, this test statistic is

$$\begin{aligned} T_2(\mathbf{K}) &= \frac{m}{\lambda^2}\left(\frac{1}{2}\left\|\mathbf{K}-\mathbf{N}_{\bar{k}}\right\|_F^2 - \frac{1}{n}\left\|\mathbf{K}\mathbf{1}-\mathbf{N}_{\bar{k}}\mathbf{1}\right\|_2^2\right) \\ &= \frac{m}{\lambda^2}\left(\frac{1}{2}\left(\|\mathbf{K}\|_F^2 - \frac{n^2}{n-1}(\bar{k}^2 n - 2\bar{k}+1)\right) - \frac{1}{n}\left\|\mathbf{K}\mathbf{1}-\bar{k}n\mathbf{1}\right\|_2^2\right). \end{aligned} \tag{C.2}$$

For $n = 3$ the asymptotic distribution of the test statistic under the null hypothesis is a scaled χ^2 distribution with two degrees of freedom (Anderson, 1963). For $n > 3$ the test statistic under the null hypothesis has a distribution described by the conic combination of two χ^2 distributions; however, this combination depends on the unknown value of a^* in terms of λ (Lawley, 1963). Again, we use the quantity $\frac{1}{m}T_2(\mathbf{K})$ since we assume m is unknown.

Appendix D. Matrix Norm Properties of $\sqrt{h(\cdot)}$

Theorem 6 *The cost function* $\sqrt{h(\mathbf{X})} = \sqrt{\frac{1}{n}\|\mathbf{X}\frac{1}{n}\mathbf{J}\|_*^2 + \frac{1}{n(n-1)}\|\mathbf{X}\mathbf{H}\|_*^2}$*, derived from the Bures-based informativeness measure, is a matrix norm for* $\mathbf{X} \in \mathbb{R}^{b\times n}$ *that satisfies the*

following properties:

$$\sqrt{h(\mathbf{X})} \geq 0, \tag{D.1}$$

$$\sqrt{h(c\mathbf{X})} = |c|\sqrt{h(\mathbf{X})} \quad \forall c, \tag{D.2}$$

$$\sqrt{h(\mathbf{X}+\mathbf{Y})} \leq \sqrt{h(\mathbf{X})} + \sqrt{h(\mathbf{Y})}, \tag{D.3}$$

$$\sqrt{h(\mathbf{X})} = 0 \text{ iff } \mathbf{X} = \mathbf{0}. \tag{D.4}$$

Furthermore, $\sqrt{h(\cdot)}$ *is convex.*

Proof The first two properties of non-negativity and scalability are readily apparent. We proceed to show that $\sqrt{h(\cdot)}$ satisfies the triangle inequality (D.3). For compactness, we set $\alpha_1 = \frac{1}{n}$, $\alpha_2 = \frac{1}{n(n-1)}$, and $\acute{\mathbf{J}} = \frac{1}{n}\mathbf{J}$. For all conformable matrices $\mathbf{X}$ and $\mathbf{Y}$, we have

$$\begin{aligned} h(\mathbf{X}+\mathbf{Y}) &= \alpha_1\|(\mathbf{X}+\mathbf{Y})\acute{\mathbf{J}}\|_*^2 + \alpha_2\|(\mathbf{X}+\mathbf{Y})\mathbf{H}\|_*^2 \\ &\leq \alpha_1(\|\mathbf{X}\acute{\mathbf{J}}\|_* + \|\mathbf{Y}\acute{\mathbf{J}}\|_*)^2 + \alpha_2(\|\mathbf{X}\mathbf{H}\|_* + \|\mathbf{Y}\mathbf{H}\|_*)^2 \\ &= h(\mathbf{X}) + h(\mathbf{Y}) + 2\sqrt{\left(\alpha_1\|\mathbf{X}\acute{\mathbf{J}}\|_*\|\mathbf{Y}\acute{\mathbf{J}}\|_* + \alpha_2\|\mathbf{X}\mathbf{H}\|_*\|\mathbf{Y}\mathbf{H}\|_*\right)^2} \\ &= h(\mathbf{X}) + h(\mathbf{Y}) \\ &\quad + 2\sqrt{\alpha_1^2\|\mathbf{X}\acute{\mathbf{J}}\|_*^2\|\mathbf{Y}\acute{\mathbf{J}}\|_*^2 + \alpha_2^2\|\mathbf{X}\mathbf{H}\|_*^2\|\mathbf{Y}\mathbf{H}\|_*^2 + 2\alpha_1\alpha_2\|\mathbf{X}\acute{\mathbf{J}}\|_*\|\mathbf{Y}\acute{\mathbf{J}}\|_*\|\mathbf{X}\mathbf{H}\|_*\|\mathbf{Y}\mathbf{H}\|_*} \\ &\leq h(\mathbf{X}) + h(\mathbf{Y}) \\ &\quad + 2\sqrt{\alpha_1^2\|\mathbf{X}\acute{\mathbf{J}}\|_*^2\|\mathbf{Y}\acute{\mathbf{J}}\|_*^2 + \alpha_2^2\|\mathbf{X}\mathbf{H}\|_*^2\|\mathbf{Y}\mathbf{H}\|_*^2 + \alpha_1\alpha_2\left(\|\mathbf{X}\acute{\mathbf{J}}\|_*^2\|\mathbf{Y}\mathbf{H}\|_*^2 + \|\mathbf{Y}\acute{\mathbf{J}}\|_*^2\|\mathbf{X}\mathbf{H}\|_*^2\right)} \\ &= h(\mathbf{X}) + h(\mathbf{Y}) + 2\sqrt{h(\mathbf{X})h(\mathbf{Y})} = \left(\sqrt{h(\mathbf{X})} + \sqrt{h(\mathbf{Y})}\right)^2, \end{aligned}$$

where the first inequality follows from the sub-additivity and non-negativity of the trace norm, and the second one from the arithmetic and geometric means inequality. Therefore, taking the square root yields $\sqrt{h(\mathbf{X}+\mathbf{Y})} \leq \sqrt{h(\mathbf{X})} + \sqrt{h(\mathbf{Y})}$.

To ensure that $\sqrt{h(\mathbf{X})}$ is a norm, rather than a seminorm, we now show that property (D.4) holds. Firstly, if $\mathbf{X} = \mathbf{0}$, then $h(\mathbf{X}) = \alpha_1\|\mathbf{0}\acute{\mathbf{J}}\|_*^2 + \alpha_2\|\mathbf{0}\mathbf{H}\|_*^2 = 0$. We prove the converse by contradiction for any $\alpha_1, \alpha_2 > 0$. Suppose that $h(\mathbf{X}) = 0$ and $\mathbf{X} \neq \mathbf{0}$. For the trace norm, $\|\mathbf{X}\acute{\mathbf{J}}\|_*^2 = 0$ iff $\mathbf{X}\acute{\mathbf{J}} = \mathbf{0}$, and $\|\mathbf{X}\mathbf{H}\|_*^2 = 0$ iff $\mathbf{X}\mathbf{H} = \mathbf{0}$. Since the trace-norm is non-negative and $\alpha_1, \alpha_2 > 0$, $h(\mathbf{X}) = 0$ implies that $\mathbf{X}\acute{\mathbf{J}} = \mathbf{0}$ and $\mathbf{X}\mathbf{H} = \mathbf{0}$. Summing these yields $\mathbf{X}\acute{\mathbf{J}} + \mathbf{X}\mathbf{H} = \mathbf{0}$, which, since $\acute{\mathbf{J}} + \mathbf{H} = \mathbf{I}$, yields $\mathbf{X} = \mathbf{0}$, but this contradicts the assumption that $\mathbf{X} \neq \mathbf{0}$.

The convexity of $\sqrt{h(\cdot)}$ follows from the norm properties. For $\mathbf{X}_1, \mathbf{X}_0$ and $0 \leq \alpha \leq 1$, $\sqrt{h(\alpha\mathbf{X}_1 + (1-\alpha)\mathbf{X}_0)} \leq \sqrt{h(\alpha\mathbf{X}_1)} + \sqrt{h((1-\alpha)\mathbf{X}_0)} = \alpha\sqrt{h(\mathbf{X}_1)} + (1-\alpha)\sqrt{h(\mathbf{X}_0)}$. ■

Appendix E. Second-Order Spectral Soft-Thresholding Operator

The trace norm is well known as the tightest convex surrogate for the rank of a matrix (Fazel et al., 2001; Recht et al., 2010) and its proximal operator, the spectral soft-thresholding

operator, has been used in the context of matrix completion (Cai et al., 2010; Mazumder et al., 2010; Ma et al., 2011). The generalization of proximal operators on vectors to matrices holds for any vector functional that is absolutely symmetric (invariant to permutation and change of sign), or equivalently any matrix functional that is unitarily invariant, since in these cases Lewis (1995) proved that the subdifferential of the matrix functional can be written as a composition of the subdifferential of the vector functional and the singular value functional (Lewis's result generalizes the classical theorem by von Neumann (1937) relating symmetric gauge functions to unitary invariant norms). For example, the trace norm can be written as the ℓ_1-norm of the singular values of a matrix, and thus, the spectral soft-thresholding operator is the matrix analog of the soft-thresholding operator Donoho et al. (1995), which is the proximal operator of the ℓ_1-norm (Wright et al., 2009).

By the same reasoning, the squared trace-norm is the matrix analog to the squared ℓ_1-norm. The squared ℓ_1-norm has been used in the context of multiple kernel learning (Lanckriet et al., 2004; Bach, 2008; Kowalski and Torrésani, 2009; Martins et al., 2011), and its proximal operator uses the soft-thresholding operator (Donoho et al., 1995) with a data-dependent penalty (Bach, 2008; Kowalski and Torrésani, 2009; Martins et al., 2011). Since the squared ℓ_1-norm is absolutely symmetric, the subdifferential of the squared trace norm is found by composition of the subdifferential of the squared ℓ_1-norm and the singular value functional (Lewis, 1995). The proximal operator for the squared trace norm applies the proximal operator of the squared ℓ_1-norm to the singular values of the target matrix.

Let $\mathbf{X} \in \mathbb{R}^{m\times n}$ denote an arbitrary matrix (without loss of generality we assume $m \leq n$), and let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ denote the singular value decomposition of $\mathbf{X}$, such that $\mathbf{U}, \mathbf{V}$ are unitary and $\mathbf{\Sigma}$ is a diagonal matrix with its first m entries $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$ and any remaining entries being 0. The spectral soft-thresholding operator is defined as

$$\mathcal{S}_\tau(\mathbf{X}) = \arg\min_{\mathbf{Y}} \frac{1}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 + \tau\|\mathbf{Y}\|_* = \mathbf{U}[\mathbf{\Sigma}-\tau\mathbf{I}]_+\mathbf{V}^\top,$$

where $[\cdot]_+$ denotes the element-wise non-negative thresholding operation that sets any negative values to zero. Using the proximal operator for the squared ℓ_1-norm (Kowalski and Torrésani, 2009; Martins et al., 2011), we define the second-order spectral soft-thresholding operator $\mathcal{Z}_\beta(\cdot)$ as

$$\begin{aligned}\mathcal{Z}_\beta(\mathbf{X}) &= \mathcal{S}_{\tau^\star}(\mathbf{X}), \qquad\qquad (E.1)\\ \tau^\star &= \tfrac{\beta}{1+k^\star\beta}\|\mathbf{X}\|_{k^\star},\\ k^\star &= \max\{k\in[m] \;:\; \sigma_k - \tfrac{\beta}{1+k\beta}\|\mathbf{X}\|_k > 0\},\\ \|\mathbf{X}\|_k &= \sigma_1+\cdots+\sigma_k \quad \text{(Ky-Fan } k\text{-norm)}.\end{aligned}$$

Theorem 7 *$\mathcal{Z}_\beta(\cdot)$ is the proximal operator of the function $\frac{\beta}{2}\|\cdot\|_*^2$ such that*

$$\mathcal{Z}_\beta(\mathbf{X}) = \arg\min_{\mathbf{Y}} \tfrac{1}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 + \tfrac{\beta}{2}\|\mathbf{Y}\|_*^2.$$

Proof

Since $\frac{1}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 + \frac{\beta}{2}\|\mathbf{Y}\|_*^2$ is convex, to prove the optimality of the second-order spectral thresholding operator it is sufficient to show that

$$\mathbf{0} \in \partial\tfrac{1}{2}\|\mathbf{X}-\mathcal{Z}_\beta(\mathbf{X})\|_F^2 + \partial\tfrac{\beta}{2}\|\mathcal{Z}_\beta(\mathbf{X})\|_*^2,$$

where $\partial f(\mathbf{X})$ denotes the subdifferential of the function f at $\mathbf{X}$, defined as

$$\partial f(\mathbf{X}) = \{\mathbf{G} : \forall \mathbf{Y}\in\mathbb{R}^{m\times n}, f(\mathbf{Y}) \geq f(\mathbf{X}) + \langle \mathbf{Y}-\mathbf{X}, \mathbf{G}\rangle\}.$$

To do this we use the optimality of the spectral thresholding operator (Cai et al., 2010; Mazumder et al., 2010; Ma et al., 2011), which implies that for any $\tau \geq 0$

$$\mathbf{0} \in \partial\tfrac{1}{2}\|\mathbf{X}-\mathcal{S}_\tau(\mathbf{X})\|_F^2 + \partial\tau\|\mathcal{S}_\tau(\mathbf{X})\|_*,$$

where the subdifferential of the trace norm (Watson, 1992) is

$$\partial\|\mathbf{X}\|_* = \{\mathbf{U}_1\mathbf{V}_1^\top + \mathbf{M} : \mathbf{U}_1^\top\mathbf{M} = \mathbf{0}, \mathbf{M}\mathbf{V}_1 = \mathbf{0}, \|\mathbf{M}\|_2 \leq 1\},$$

where the columns of $\mathbf{U}_1$ and $\mathbf{V}_1$ are the left and right singular vectors associated to non-zero singular values.

We note that any subgradient of squared trace-norm is a scaled version of the subgradient of the trace-norm with location-dependent scaling, i.e., $(2\|\mathbf{X}\|_*)\mathbf{G} \in \partial\|\mathbf{X}\|_*^2$ iff $\mathbf{G} \in \partial\|\mathbf{X}\|_*$.

Given $\mathcal{Z}_\beta(\mathbf{X}) = \mathcal{S}_{\tau^\star}(\mathbf{X})$, $\tau^\star = \frac{\beta}{1+k^\star\beta}\|\mathbf{X}\|_{k^\star}$, and $\|\mathcal{Z}_\beta(\mathbf{X})\|_* = \|\mathbf{X}\|_{k^\star} - k^\star\tau^\star = \frac{\tau^\star}{\beta}$, it follows that

$$\partial\tfrac{\beta}{2}\|\mathcal{Z}_\beta(\mathbf{X})\|_*^2 = \tfrac{\beta}{2}(2\|\mathcal{Z}_\beta(\mathbf{X})\|_*)\partial\|\mathcal{Z}_\beta(\mathbf{X})\|_* = \tau^\star\partial\|\mathcal{S}_{\tau^\star}(\mathbf{X})\|_*.$$

Thus, $\partial\frac{1}{2}\|\mathbf{X}-\mathcal{Z}_\beta(\mathbf{X})\|_F^2 + \partial\frac{\beta}{2}\|\mathcal{Z}_\beta(\mathbf{X})\|_*^2 = \partial\frac{1}{2}\|\mathbf{X}-\mathcal{S}_{\tau^\star}(\mathbf{X})\|_F^2 + \partial\tau^\star\|\mathcal{S}_{\tau^\star}(\mathbf{X})\|_*$. ■

References

P.-A. Absil and K. A. Gallivan. Joint diagonalization on the oblique manifold for independent component analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages V–945–V–948, 2006.

M. A. Aitkin, W. C. Nelson, and Karen H. Reinfurt. Tests for correlation matrices. *Biometrika*, 55(2):327–334, 1968.

T. W. Anderson. Asymptotic theory for principal component analysis. *The Annals of Mathematical Statistics*, 34(1):122–148, 1963.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

K. M. R. Audenaert, J. Calsamiglia, R. Munoz-Tapia, E. Bagan, Ll. Masanes, A. Acin, and F. Verstraete. Discriminating states: the quantum Chernoff bound. *Physical Review Letters*, 98(16):160501, 2007.

Francis R. Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

Richard H. Bartels and G. W. Stewart. Solution of the matrix equation $AX + XB = C$. *Communications of the ACM*, 15(9):820–826, 1972.

M. S. Bartlett. A note on the multiplying factors for various χ^2 approximations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 16(2):296–298, 1954.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Peter J. Bickel and Purnamrita Sarkar. Hypothesis testing for automated community detection in networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):253–273, 2016.

Rüdiger Borsdorf, Nicholas J. Higham, and Marcos Raydan. Computing a nearest correlation matrix with factor structure. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2603–2622, 2010.

Jonathan M. Borwein and Adrian S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. CMS Books in Mathematics. Springer New York, 2010.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Samuel L. Braunstein and Carlton M. Caves. Statistical distance and the geometry of quantum states. *Physical Review Letters*, 72(22):3439–3443, 1994.

C. J. Brien, W. N. Venables, A. T. James, and O. Mayo. An analysis of correlation matrices: equal correlations. *Biometrika*, 71(3):545–554, 1984.

Thomas R. Bromley, Marco Cianciaruso, Rosario Lo Franco, and Gerardo Adesso. Unifying approach to the quantification of bipartite correlations by Bures distance. *Journal of Physics A: Mathematical and Theoretical*, 47(40):405302, 2014.

Arne Brøndsted. *An Introduction to Convex Polytopes*, volume 90 of *Graduate Texts in Mathematics*. Springer New York, 2012.

Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

Donald Bures. An extension of Kakutani's theorem on infinite product measures to the tensor product of semifinite w*-algebras. *Transactions of the American Mathematical Society*, 135:199–212, 1969.

Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Zeineb Chebbi and Maher Moakher. Means of Hermitian positive-definite matrices based on the log-determinant α-divergence function. *Linear Algebra and its Applications*, 436 (7):1872–1889, 2012.

Anoop Cherian, Suvrit Sra, Arindam Banerjee, and Nikolaos Papanikolopoulos. Efficient similarity search for covariance matrices via the Jensen-Bregman logdet divergence. In *2011 International Conference on Computer Vision*, pages 2399–2406, 2011.

Fan R. K. Chung. *Spectral Graph Theory*. Number 92 in CBMS regional conference series in mathematics. American Mathematical Society, 1997.

Andrzej Cichocki, Sergio Cruces, and Shun-ichi Amari. Log-determinant divergences revisited: Alpha-beta and gamma log-det divergences. *Entropy*, 17(5):2988–3034, 2015.

Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In Heinz H. Bauschke, Regina S. Burachik, Patrick L. Combettes, Veit Elser, D. Russell Luke, and Henry Wolkowicz, editors, *Fixed-point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, New York, NY, 2011.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13:795–828, 2012.

Nello Cristianini, John Shawe-Taylor, André Elisseeff, and Jaz Kandola. On kernel-target alignment. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.

David L. Donoho, Iain M. Johnstone, Gérard Kerkyacharian, and Dominique Picard. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(2):301–369, 1995.

Jonathan Eckstein and Dimitri P. Bertsekas. On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992.

P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6: 290–297, 1959.

Maryam Fazel, Haitham Hindi, and Stephen P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American Control Conference*, pages 4734–4739, 2001.

Christopher Fuchs and Jeroen van de Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999.

Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

Leon Jay Gleser. On testing a set of correlation coefficients for equality: Some asymptotic results. *Biometrika*, 55(3):513–517, 1968.

Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

Daniel B. Graham and Nigel M. Allinson. Characterising virtual eigensignatures for general purpose face recognition. In Harry Wechsler, P. Jonathon Phillips, Vicki Bruce, Françoise Fogelman Soulié, and Thomas S. Huang, editors, *Face Recognition: From Theory to Applications*, pages 446–456. Springer, Berlin, Heidelberg, 1998.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In Sanjay Jain, Hans Ulrich Simon, and Etsuji Tomita, editors, *Algorithmic Learning Theory*, volume 3734 of *Lecture Notes in Computer Science*, pages 63–77. Springer, Berlin, Heidelberg, 2005.

Igor Grubišić and Raoul Pietersz. Efficient rank reduction of correlation matrices. *Linear Algebra and its Applications*, 422(2):629–653, 2007.

Masahito Hayashi. *Quantum Information*. Springer, 2006.

Bingsheng He and Xiaoming Yuan. On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers. *Numerische Mathematik*, 130(3):567–577, 2015.

Nicholas J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, 1988.

Nicholas J. Higham. Computing the nearest correlation matrix—a problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329–343, 2002.

Michael Hintermüller and Tao Wu. A smoothing descent method for nonconvex TV^q-models. In Andrés Bruhn, Thomas Pock, and Xue-Cheng Tai, editors, *Efficient Algorithms for Global Optimization Methods in Computer Vision*, volume 8293 of *Lecture Notes in Computer Science*, pages 119–133. Springer, Berlin, Heidelberg, 2014.

Peter J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 20(5):2327–2351, 2010.

Richard Jozsa. Fidelity for mixed quantum states. *Journal of Modern Optics*, 41(12): 2315–2323, 1994.

Vladimir Koltchinskii and Dong Xia. Optimal estimation of low rank density matrices. *Journal of Machine Learning Research*, 16:1757–1792, 2015.

Risi Kondor and Tony Jebara. A kernel between sets of vectors. In *Proceedings of the 20th International Conference on Machine Learning*, pages 361–368, 2003.

Matthieu Kowalski and Bruno Torrésani. Structured sparsity: from mixed norms to structured shrinkage. In *SPARS'09 - Signal Processing with Adaptive Sparse Structured Representations*, 2009.

Tarald O. Kvålseth. Entropy and correlation: some comments. *IEEE Transactions on Systems, Man, and Cybernetics*, 3(17):517–519, 1987.

Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

Monique Laurent and Svatopluk Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223–224:439–461, 1995.

Monique Laurent and Svatopluk Poljak. On the facial structure of the set of correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 17(3):530–547, 1996.

D. N. Lawley. On testing a set of correlation coefficients for equality. *The Annals of Mathematical Statistics*, 34(1):149–151, 1963.

Hosoo Lee and Yongdo Lim. Invariant metrics, contractions and nonlinear matrix equations. *Nonlinearity*, 21(4):857–878, 2008.

A. S. Lewis. The convex analysis of unitarily invariant matrix functions. *Journal of Convex Analysis*, 2(1/2):173–183, 1995.

A. S. Lewis. Derivatives of spectral functions. *Mathematics of Operations Research*, 21(3): 576–588, 1996.

Chi-Kwong Li and Bit-Shun Tam. A note on extreme correlation matrices. *SIAM Journal on Matrix Analysis and Applications*, 15(3):903–908, 1994.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

David G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 7(1):72–85, 1995.

Shunlong Luo and Qiang Zhang. Informational distance on quantum-state space. *Physical Review A*, 69(3):032106, 2004.

Shiqian Ma, Donald Goldfarb, and Lifeng Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2):321–353, 2011.

A. P. Majtey, P. W. Lamberti, and D. P. Prato. Jensen-Shannon divergence as a measure of distinguishability between mixed quantum states. *Physical Review A*, 72(5):052310, 2005.

Jérôme Malick. A dual approach to semidefinite least-squares problems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):272–284, 2004.

André F. T. Martins, Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. Online learning of structured predictors with multiple kernels. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 507–515, 2011.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

Marina Meilă. Comparing clusterings by the variation of information. In B. Schölkopf and M. K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 173–187. Springer, Berlin, Heidelberg, 2003.

Paulo E. M. F. Mendonça, Reginaldo d. J. Napolitano, Marcelo A. Marchiolli, Christopher J. Foster, and Yeong-Cherng Liang. Alternative fidelity measure between quantum states. *Physical Review A*, 78(5):052330, 2008.

Jarosław Adam Miszczak, Zbigniew Puchała, Paweł Horodecki, Armin Uhlmann, and Karol Życzkowski. Sub-and super-fidelity as bounds for quantum fidelity. *Quantum Information & Computation*, 9(1&2):103–130, 2009.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société mathématique de France*, 93:273–299, 1965.

Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (COIL-20). Technical report, CUCS-005-96, 1996.

Yu. Nesterov. Gradient methods for minimizing composite objective function. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, 2007.

M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856, Cambridge, MA, 2002. MIT Press.

Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

Peder A. Olsen, Steven J. Rennie, and Vaibhava Goel. Efficient automatic differentiation of matrix functions. In Shaun Forth, Paul Hovland, Eric Phipps, Jean Utke, and Andrea Walther, editors, *Recent Advances in Algorithmic Differentiation*, volume 87 of *Lecture Notes in Computational Science and Engineering*, pages 71–81. Springer, Berlin, Heidelberg, 2012.

Neal Parikh and Stephen P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

Il Memming Park, Sohan Seth, Antonio R. C. Paiva, Lin Li, and Jose C. Principe. Kernel methods on spike train space for neuroscience: a tutorial. *IEEE Signal Processing Magazine*, 30(4):149–160, 2013.

Houduo Qi and Defeng Sun. A quadratically convergent Newton method for computing the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, 28(2): 360–385, 2006.

Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

Luis Gonzalo Sanchez Giraldo, Murali Rao, and Jose C. Principe. Measures of entropy from data using infinitely divisible kernels. *IEEE Transactions on Information Theory*, 61(1): 535–548, 2015.

J. Saunderson, V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Diagonal and low-rank matrix decompositions, correlation matrices, and ellipsoid fitting. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1395–1416, 2012.

Mark Schmidt. minFunc, 2012. Software available at `http://www.di.ens.fr/~mschmidt/Software/minFunc.html`.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5):2263–2291, 2013.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

James H. Steiger. Tests for comparing elements of a correlation matrix. *Psychological Bulletin*, 87(2):245–251, 1980.

Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

Nickolay T. Trendafilov and Ross A. Lippert. The multimode procrustes problem. *Linear Algebra and its Applications*, 349(1):245–264, 2002.

A. Uhlmann. The transition probability in the state space of a*-algebra. *Reports on Mathematical Physics*, 9(2):273–279, 1976.

John von Neumann. Some matrix inequalities and metrization of matric-space. *Tomsk University Review*, 1:286–300, 1937.

John von Neumann. *Mathematical Foundations of Quantum Mechanics*. Princeton University Press, 1955.

G. A. Watson. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45, 1992.

Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

Stephen J. Wright, Robert D. Nowak, and Mário A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 505–512, Cambridge, MA, 2003. MIT Press.

Journal of Machine Learning Research 18 (2017) 1-36 Submitted 6/16; Revised 5/17; Published 8/17

Time for a Change: a Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis

Alessio Benavoli[†] alessio@idsia.ch
Giorgio Corani[†] giorgio@idsia.ch
Janez Demšar[♯] janez.demsar@fri.uni-lj.si
Marco Zaffalon[†] zaffalon@idsia.ch
[♯]*Faculty of Computer and Information Science, University of Ljubljana, Vecna pot 113, SI-1000 Ljubljana, Slovenia*
[†]*Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA) Galleria 2, 6928 Manno, Switzerland*

Editor: David Barber

Abstract

The machine learning community adopted the use of null hypothesis significance testing (NHST) in order to ensure the statistical validity of results. Many scientific fields however realized the shortcomings of frequentist reasoning and in the most radical cases even banned its use in publications. We should do the same: just as we have embraced the Bayesian paradigm in the development of new machine learning methods, so we should also use it in the analysis of our own results. We argue for abandonment of NHST by exposing its fallacies and, more importantly, offer better—more sound and useful—alternatives for it.

Keywords: comparing classifiers, null hypothesis significance testing, pitfalls of p-values, Bayesian hypothesis tests, Bayesian correlated t-test, Bayesian hierarchical correlated t-test, Bayesian signed-rank test

1. Introduction

Progression of Science and of the scientific method go hand in hand. Development of new theories requires—and at the same time facilitates—development of new methods for their validation.

Pioneers of machine learning were playing with ideas: new approaches, such as induction of classification trees, were worthy of publication for the sake of their interestingness. As the field progressed and found more practical uses, variations of similar ideas began emerging, and with that the interest in determining which of them work better in practice. A typical example are the different measures for assessing the quality of attributes; deciding which work better than others required tests on actual, real-world data. Papers thus kept introducing new methods and measured, for instance, classification accuracies to prove their advantages over the existing methods. To ensure the validity of such claims, we adopted—starting with the work of Dietterich (1998) and Salzberg (1997), and later followed by Demšar (2006)—the common statistical methodology used in all scientific areas relying on empirical observations: the null hypothesis significance testing (NHST).

©2017 Alessio Benavoli, Giorgio Corani, Janez Demsar, Marco Zaffalon.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-305.html.

This spread the understanding that the observed results require statistical validation. On the other hand, NHST soon proved inadequate for many reasons (Demšar, 2008). Noteworthy, the American Statistical Association has recently made a statement against p-values (Wasserstein and Lazar, 2016). NHST nowadays it is also falling out of favour in other fields of science (Trafimow and Marks, 2015). We believe that the field of machine learning is ripe for a change as well.

We will spend a whole section demonstrating the many problems of NHST. In a nutshell: it does not answer the question we ask. In a typical scenario, a researcher proposes a new method and desires to prove that it is more accurate than another method on a *single* data set or on a *collection* of data sets. She thus runs the competing methods and records their results (classification accuracy or another appropriate score) on one or more data sets, which is followed by NHST. The difference between what the researcher has in mind and what the NHST provides for is evident from the following quote from a recently published paper: "*Therefore, at the 90% confidence level, we can conclude that (...) method is able to significantly outperform the other approaches.*" This is wrong. The stated *90% confidence level* is not the probability of one classifier outperforming another. The NHST computes the probability of getting the observed (or a larger) difference between classifiers if the null hypothesis of equivalence was true, which is not the probability of one classifier being more accurate than another, given the observed empirical results. Another common problem is that the claimed statistical significance might have no practical impact. Indeed, the common usage of NHST relies on the wrong assumptions that the p-value is a reasonable proxy for the probability of the null hypothesis and that statistical significance implies practical significance.

As we wrote at the beginning, development of Science not only requires but also facilitates the improvement of scientific methods. Advancement of computational techniques and power reinvigorated the interest for Bayesian statistics. Bayesian modelling is now widely adopted for designing principled algorithms for learning from data (Bishop, 2007; Murphy, 2012). It is time to also switch to Bayesian statistics when it comes to analysis of our own results.

The questions we are actually interested in—e.g., is method A better than B? Based on the experiments, how probably is A better? How high is the probability that A is better by more than 1%?—are questions about posterior probabilities. These are naturally provided by the Bayesian methods (Edwards et al., 1963; Dickey, 1973; Berger and Sellke, 1987). The core of this paper is thus a section that establishes the Bayesian alternatives to frequentist NHST and discusses their inference and results. We eventually describe also the software libraries with the necessary algorithms and give short instructions for their use.

2. Frequentist analysis of experimental results

Why do we need to go beyond the frequentist analysis of experimental results? To answer this question, we will focus on a practical case: the comparison of the accuracy of classifiers on different datasets. We initially consider two classifiers: naive Bayes (nbc) and averaged one-dependence estimator (aode). A description of these algorithms with exhaustive references is given in the book by Witten et al. (2011). Assume that our aim is to compare *nbc* versus *aode*. These are the steps we must follow:

1. choose a comparison metric;

2. select a group of datasets to evaluate the algorithms;

3. perform m runs of k-fold cross-validation for each classifier on each dataset.

We have performed these steps in WEKA, choosing *accuracy* as metric, on a collection of 54 data sets downloaded from the the WEKA website[1] and with 10 runs of 10-fold cross-validation. Table 1 reports the accuracies obtained on each dataset by each classifier.

First of all, we aim at knowing which is the best classifier for each dataset. The answer to this question is probabilistic, since on each data set we have only estimates of the performance of each classifier.

Datasets	10 runs of 10-fold cross-validation						
anneal	94.44	98.89	94.44	98.89	...	94.38	97.75
anneal	96.67	100.0	96.67	100.0	...	96.63	97.75
audiology	73.91	69.56	73.91	60.87	...	72.73	59.09
audiology	73.91	69.56	78.26	60.87	...	72.73	59.09
breast-cancer	90.32	90.32	87.1	86.67	...	86.67	90.0
breast-cancer	87.1	87.1	87.1	86.67	...	83.33	86.67
cmc	51.35	50.68	54.73	59.18	...	50.34	48.3
cmc	52.7	50.68	52.7	55.1	...	52.38	48.98
...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...
wine	100.0	95.71	97.14	94.29	...	97.14	97.1
wine	100.0	95.71	97.14	92.86	...	97.14	97.1
yeast	57.72	55.03	59.06	58.39	...	55.4	55.4
yeast	57.05	55.03	59.06	58.39	...	54.05	55.4
zoo	81.82	100.0	100.0	90.0	...	90.0	100.0
zoo	90.91	100.0	100.0	90.0	...	90.0	100.0

Table 1: Accuracies for 10 runs of 10-fold cross-validation on 54 UCI datasets for *nbc* (blue row) versus *aode* (white rows).

.

Since during cross-validation we have provided both classifiers with the same training and test sets, we can compare the classifiers by considering the difference in accuracies on each test set. This yields the vector of *differences of accuracies* $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$, where $n = 100$ (10 runs of 10-fold cross-validation). We can compute the mean of the vector of differences $\boldsymbol{x}$, i.e., the mean difference of accuracy between the two classifiers, and statistically evaluate whether the mean difference is significantly different from zero. In frequentist analysis, we would perform a NHST using the t-test. The problem is that the t-test assumes the observations to be independent. However, the differences of accuracies

1. See `http://www.cs.waikato.ac.nz/ml/weka/`.

are not independent of each other because of the overlapping training sets used in cross-validation. Thus the usual t-test is *not* calibrated when applied to the analysis of cross-validation results: when sampling the data under the null hypothesis its rate of Type I errors is much larger than α (Dietterich, 1998). Moreover, the correlation cannot be estimated from data; Nadeau and Bengio (2003) have proven that there is no unbiased estimator of the correlation of the results obtained on the different folds. Introducing some approximations, they have proposed a heuristic to choose the correlation parameter: $\rho = \frac{n_{te}}{n_{tot}}$, where n_{te}, n_{tr} and $n_{tot} = n_{te} + n_{tr}$ respectively denote the size of the training set, of the test set and of the whole available data set.[2]

Frequentist correlated t-test

The correlated t-test is based on the modified Student's t-statistic:

$$t(\boldsymbol{x}, \mu) = \frac{\overline{x} - \mu}{\sqrt{\hat{\sigma}^2(\frac{1}{n} + \frac{\rho}{1-\rho})}} = \frac{\overline{x} - \mu}{\sqrt{\hat{\sigma}^2(\frac{1}{n} + \frac{n_{te}}{n_{tr}})}}, \tag{1}$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\hat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$ are the sample mean and sample standard deviation of the data $\boldsymbol{x}$, ρ is the correlation between the observations and μ is the value of the mean we aim at testing. The statistic follows a Student distribution with $n-1$ degrees of freedom:

$$St\left(\bar{x}; n-1, \mu, \left(\frac{1}{n} + \frac{\rho}{1-\rho}\right)\hat{\sigma}^2\right). \tag{2}$$

For $\rho = 0$, we obtain the traditional t-test. For $\rho = \frac{n_{te}}{n_{tot}}$, we obtain the correlated t-test proposed by Nadeau and Bengio (2003) to account for the correlation due to the overlapping training sets. Usually the test is run in a two-sided fashion. Its hypotheses are: $H_0 : \mu = 0$; $H_1 : \mu \neq 0$. The p-value of the statistic under the null hypotheses is:

$$p = 2 \cdot (1 - \mathcal{T}_{n-1}(|t(\boldsymbol{x}, 0)|)), \tag{3}$$

where $\mathcal{T}_{n-1}(|t(\boldsymbol{x}, 0)|)$ denotes the cumulative distribution of the standardized Student distribution with $n-1$ degrees of freedom in $|t(\boldsymbol{x}, \mu)|$ for $\mu = 0$. For instance, for the first data set in Table 1 we have that $\bar{x} = -0.0194$, $\hat{\sigma} = 0.01583$, $\rho = 1/10$, $n = 100$ and so $t(\bar{x}, 0) = -3.52$. Hence, the two-sided p-value is $p = 2 \cdot (1 - \mathcal{T}_{n-1}(|t(\boldsymbol{x}, 0)|)) = 0.00065 \approx 0.001$. Sometimes the directional one-sided test is performed. If the alternative hypothesis is the positive one, the hypotheses of the one-sided test are: $H_0 : \mu \leq 0$; $H_1 : \mu > 0$. The p-value is $p = 1 - \mathcal{T}_{n-1}(t(\boldsymbol{x}, 0))$.

Table 2 reports the two-sided p-values for each comparison on each dataset obtained via the correlated t-test. The common practice in NHST is to declare all the comparisons such that $p \leq 0.05$ as significant, i.e., the accuracy of the two classifiers is significantly different on that dataset. Conversely, all the comparisons with $p > 0.05$ are declared not significant.

2. Nadeau and Bengio (2003) considered the case in which random training and test sets are drawn from the original data set. This is slightly different from k-fold cross-validation, in which the folds are designed not to overlap. However the correlation heuristic by Nadeau and Bengio (2003) has since become commonly used to analyse the cross-validation results (Bouckaert, 2003).

Note that these significance tests can, under the NHST paradigm, only be considered in isolation, while combined they require either an omnibus test like ANOVA or corrections for multiple comparisons.

Dataset	p-value	Dataset	p-value	Dataset	p-value
anneal	0.001	audiology	0.622	breast-cancer	0.598
cmc	0.338	contact-lenses	0.643	credit	0.479
german-credit	0.171	pima-diabetes	0.781	ecoli	0.001
eucalyptus	0.258	glass	0.162	grub-damage	0.090
haberman	0.671	hayes-roth	1.000	cleeland-14	0.525
hungarian-14	0.878	hepatitis	0.048	hypothyroid	0.287
ionosphere	0.684	iris	0.000	kr-s-kp	0.646
labor	1.000	lier-disorders	0.270	lymphography	0.018
monks1	0.000	monks3	0.220	monks	0.000
mushroom	0.000	nursery	0.000	optdigits	0.000
page	0.687	pasture	0.000	pendigits	0.452
postoperatie	0.582	primary-tumor	0.492	segment	0.000
solar-flare-C	0.035	solar-flare-m	0.596	solar-flare-X	0.004
sonar	0.777	soybean	0.049	spambase	0.000
spect-reordered	0.198	splice	0.004	squash-stored	0.940
squash-unstored	0.304	tae	0.684	credit	0.000
owel	0.000	waveform	0.417	white-clover	0.463
wine	0.671	yeast	0.576	zoo	0.435

Table 2: Two sided p-values for each dataset. The difference is significant ($p < 0.05$) in 19 out of 54 comparisons.

2.1 NHST: the pitfalls of black and white thinking

Despite being criticized from its inception, NHST is still considered necessary for publication, as $p \leq 0.05$ is trusted as an objective proof of the method's quality. One of the key problems of decisions based on $p \leq 0.05$ is that it leads to "black and white thinking", which ignores the fact that (i) a *statistically significant difference* is completely different from a *practically significant difference* (Berger and Sellke, 1987); (ii) two methods that *are not statistically significantly different are not necessarily equivalent*. The NHST and this p-value-related "black and white thinking" do not allow for making informed decisions. Hereafter, we list the limits of NHST in order of severity using, as a working example, the assessment of the performance of classifiers.

NHST does not estimate probabilities of hypotheses. *What is the probability that the performance of two classifiers is different (or equal)?* This is *the question* we are asking when we compare two classifiers; and NHST cannot answer it.

In fact, the p-value represents the probability of getting the observed (or larger) differences assuming that the performance of the classifiers is equal (H_0). Formally, $p = p(t(\boldsymbol{x}) > \tau|H_0)$, where $t(\boldsymbol{x})$ is the statistic computed from the data $\boldsymbol{x}$, and τ is the critical value corresponding to the test and the selected α. This is not the probability of the hypothesis, $p(H_0|\boldsymbol{x})$, in which we are interested.

Yet, researchers want to know the probability of the null and the alternative hypotheses on the basis of the observed data, rather than the probability of the data assuming the null hypothesis to be true. Sentences like "*at the 95% confidence level, we can conclude that* (...)", are formally correct, but they seem to imply that $1 - p$ is the probability of the alternative hypothesis, while in fact $1 - p = 1 - p(t(\boldsymbol{x}) > \tau|H_0) = p(t(\boldsymbol{x}) < \tau|H_0)$, which is not the same as $p(H_1|\boldsymbol{x})$. This is summed up in Table 3.

what we compute	**what we would like to know**
$p(t(\boldsymbol{x}) > \tau\|H_0)$	$p(H_0\|\boldsymbol{x})$
$1 - p(t(\boldsymbol{x}) > \tau\|H_0) = p(t(\boldsymbol{x}) < \tau\|H_0)$	$1 - p(H_0\|\boldsymbol{x}) = p(H_1\|\boldsymbol{x})$

Table 3: Difference between the probabilities of interest for the analyst and the probabilities computed by the frequentist test.

Point-wise null hypotheses are practically always false. *The difference between two classifiers can be very small; however there are no two classifiers whose accuracies are perfectly equivalent.*

By using a NHST, the null hypothesis is that the classifiers are equal. However, the null hypothesis is practically always false! By rejecting the null hypothesis NHST indicates that the null hypothesis is unlikely; but this is known even before running the experiment. This problem of the NHST has been pointed out in many different scientific domains (Lecoutre and Poitevineau, 2014, Sec 4.1.2.2). A consequence is that, since the null hypothesis is always false, **by adding enough data points it is possible to claim significance even when the effect size is trivial**. This is because the p-value is affected both by the sample size and the effect size, as discussed in the next section. Quoting Kruschke and Liddell (2015): *"null hypotheses are straw men that can virtually always be rejected with enough data."*

The p-value does not separate between the effect size and the sample size. The usual explanation for this phenomenon is that if the effect size H_0 is small, more data is needed to demonstrate the difference. Enough data can confirm arbitrarily small effects. Since the sample size is manipulated by the researcher and the null hypothesis is always wrong, *the researcher can reject it by testing the classifiers on enough data.* On the contrary, conceivable differences may fail to yield small p-values if there are not enough suitable data for testing the method (e.g., not enough datasets). Even if we pay attention not to confuse the p-value with the probability of the null hypothesis, the p-value is intuitively understood as the indicator of the effect size. In practice, it is the function of effect size and sample size: same p-values do not imply same effect sizes.

Figure 1 reports the *density plots* of the differences of accuracy between nbc and aode on the dataset *hepatitis* in two cases: (1) considering only 15 of the 100 accuracies in Table 1 (left); (2) considering all the 100 accuracies (right). The *two orange vertical lines*

define the region in which the differences of accuracy is less than 1%—the meaning of these lines will be clarified in the next sections.

The p-value is 0.077 in the first case and so the null hypothesis cannot be rejected. The p-value becomes 0.048 in the second case and so the null hypothesis can be rejected. This demonstrates how adding data leads to rejection of the null hypothesis although the difference between the two classifiers is very small in this dataset (all the mass is inside the two orange vertical lines). Practical significance can be equated with the effect size, which is what the researcher is interested in. Statistical significance—the p-value—is not a measure of practical significance, as shown in the example.

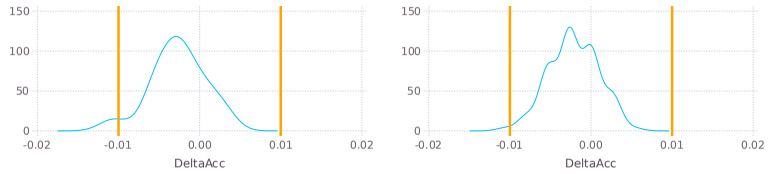


Figure 1: Density plot for the differences of accuracy between *nbc* and *aode* for the dataset *hepatitis* considering only 15 of the 100 data (left) or all the data (right). Left: the null hypothesis cannot be rejected ($p = 0.077 > 0.05$) using half the data. Right: the null hypothesis is rejected when all the data are considered ($p = 0.048 < 0.05$), despite the very small effect size.

NHST ignores magnitude and uncertainty. *A very important problem with NHST is that the result of the test does not provide information about the magnitude of the effect or the uncertainty of its estimate, which are the key information we should aim at knowing.*

A consequence of this limit is that: (i) a null hypothesis can be rejected despite a very small effect; (ii) a null hypothesis can be rejected even though there is a large uncertainty in the effect's magnitude and the region of uncertainty includes (or is extremely close to) zero, that is, no effect. Figure 1 (right) shows a case for which $p = 0.048 < 0.05$ and the result is therefore declared to be statistically significant. However, from the density plot, it is clear that the magnitude of the effect is very small (all inside the orange vertical lines bounding the less than 1% difference of accuracy region). Thus rejecting a null hypothesis does not provide any information about the magnitude of the effect and whether or not the effect is trivial. Figure 2 shows two cases for which $p \approx 0.001 < 0.05$ and the result is therefore declared to be statistically significant. Such p-values are similarly low, but the two cases are extremely different. For the dataset *ecoli* (left), the differences of accuracy are spread from 0.1 to -0.25 (the magnitude of the uncertainty is very large, about 35%), while in the second case the data are spread from 0 to -0.07, a much smaller uncertainty. Thus, rejecting a null hypothesis does not provide us with any information about the uncertainty of the estimate.

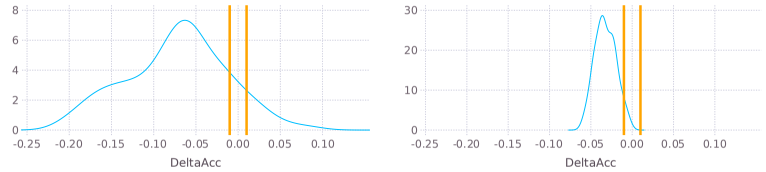


Figure 2: Density plot for the differences of accuracy (DeltaAcc) between *nbc* and *aode* for the datasets *ecoli* (left) and *iris* (right). The null hypothesis is rejected ($p < 0.05$) with similar p-values, even though the two cases have very different uncertainty. For *ecoli*, the uncertainty is very large and includes zero.

NHST yields no information about the null hypothesis. *What can we say when NHST does not reject the null hypothesis?*

The scientific literature contains examples of non-significant tests interpreted as evidence of no difference between the two methods/groups being compared. This is wrong since NHST cannot provide evidence in favour of the null hypothesis (see also Lecoutre and Poitevineau (2014, Sec 4.1.2.2) for further examples on this point). When NHST does not reject the null hypothesis, no conclusion can be made.

Researchers may be interested in the probability that two classifiers are equivalent. For instance, a recent paper contains the following conclusion: "*there is no significant difference between (...) under the significance level of* 0.05. *This is quite a remarkable conclusion.*" This is not correct, we cannot conclude anything in this case! Consider for example Figure 3 (left), which shows the density plot of the differences of accuracy between nbc and aode for the dataset *audiology*. The correlated t-test gives a p-value $p = 0.622$ and so it fails to reject the null hypothesis. However, NHST does not allow us to reach any conclusion about the possibility that the two classifiers may actually be equivalent in this dataset, although the majority of data (density plot) seems to support this hypothesis. We tend to interpret these cases as acceptance of the null hypothesis or to even (mis)understand the p value as a "62.2% probability that the performance of the classifiers is the same". This is also evident from Figure 3 (right), where a very similar p-value, $p = 0.598$, corresponds to a different density plot—with more uncertainty and so less evidence in favour of the null.

There is no principled way to decide the α level. In the above examples, we rejected the null hypothesis at $\alpha = 0.05$. We understand this difference as significant, but not as significant as if we could reject it at $\alpha = 0.01$. What is the actual meaning of this? Can we reject it at $\alpha = 0.03$? The α level represents the crucial threshold to declare the experiment successful. With its importance, it needs to be set with care. However, since it is used to compare the meaningless p-values, α is equally meaningless. By using the NHST, the researcher is forced to select an important threshold with no practical meaning. Using the customary thresholds of 0.05 and 0.01 merely allows her/him to shift the responsibility to the unsubstantiated traditional habit.

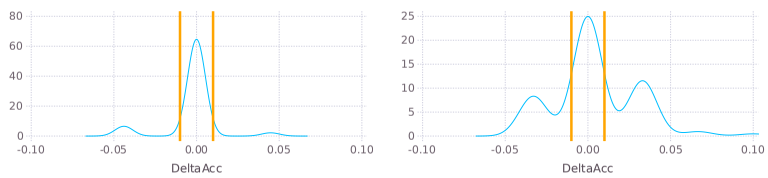


Figure 3: Density plot for the differences of accuracy between *nbc* and *aode* for the datasets *audiology* (left) and *breast-cancer* (right). In *audiology*, we can only say that the null hypothesis cannot be rejected $p = 0.622$ (> 0.05). NHST does not allow the conclusion that the null hypothesis is true although practically all the data lies in the interval $[-0.01, 0.01]$ (the differences of accuracy are less than 1%). A similar p-value corresponds to a very different situation for *breast-cancer*.

We pretend that α is the proportion of cases in which H_0 would be falsely rejected if the same experiment was repeated. This would be true if the p-value represented the probability of the null hypothesis. In reality, α is the proportion of cases in which experiments yield the data that is more extreme than expected under H_0; the actual probability of falsely rejecting H_0 is also related to the probability of H_0 and the data.

For α to be meaningful, it would need to set the required effect size or at least the probability of the hypothesis, not the likelihood of the data.

The inference depends on the sampling intention. Consider analysing a data set of n observations with a NHST test. The sampling distribution used to determine the critical value of the test assumes that our intention was to collect exactly n observations. If the intention was different—for instance in machine learning you typically compare two algorithms on all the datasets that are available—, the sampling distribution changes to reflect the actual sampling intentions (Kruschke, 2010). This is never done, given the difficulty of formalizing one's intention and of devising an appropriate sampling distribution. This problem is thus important but generally ignored. Thus for the data set the hypothesis test (and thus the p-value) should be computed differently, depending on the intention of the person who collected the data (Kruschke, 2010).

3. Bayesian analysis of experimental results

There are two main Bayesian approaches for the analysis of experimental results. The first Bayesian approach, as NHST, is also based on a null value. The analyst has to set up two competing models of what values are possible. One model assumes that only the null value is possible. The alternative model assumes a broad range of other values is also possible. Bayesian inference is used to compute which model is more credible, given the data. This method is called *Bayesian model comparison* and uses so-called "Bayes factors" (Berger, 1985; Aitkin, 1991; Kass and Raftery, 1995; Berger and Pericchi, 1996).

The second Bayesian approach does not set any null value. The analyst simply has to set up a range of candidate values (prior model), including the zero effect, and use Bayesian inference to compute the relative credibilities of all the candidate values (the posterior distribution). This method is called *Bayesian estimation* (Gelman et al., 2014; Kruschke, 2015).

The choice of the method depends on the situation – is a point-wise null hypothesis plausible ? (Bayarri and Berger, 2013, Sec.18.2.3.1)

In machine learning the estimation approach is preferable because there is not plausible reason to believe that two *different* classifiers have exactly the same accuracy. For this reason, we will focus on the Bayesian estimation approach that hereafter we will simply call *Bayesian analysis*.

The first step in Bayesian analysis is establishing a descriptive mathematical model of the data. In a parametric model, this mathematical model is the the *likelihood function* that provides the probability of the observed data for each candidate value of the parameter(s) $p(Data|\theta)$. The second step is to establish the credibility for each value of the parameter(s) before observing data, the *prior distribution* $p(\theta)$. The third step is to use Bayes' rule to combine likelihood and prior to obtain the *posterior distribution* of the parameter(s) given the data $p(\theta|Data)$. The questions we pose in statistical analysis can be answered by querying this posterior distribution in different ways.

As a concrete example of Bayesian analysis we will compare the accuracies of two competing classifiers via cross-validation on multiple data sets (Table 1). For this purpose, we will adopt the *correlated Bayesian t-test* proposed by Corani and Benavoli (2015).

Bayesian correlated t-test

The Bayesian correlated t-test is used for the analysis of cross-validation results on a single dataset and it accounts for the correlation due to the overlapping training sets. The test is based on the following (generative) model of the data:

$$\mathbf{x}_{n\times 1} = \mathbf{1}_{n\times 1}\mu + \mathbf{v}_{n\times 1}, \tag{4}$$

where $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ is the vector of differences of accuracy, $\mathbf{1}_{n\times 1}$ is a vector of ones, μ is the parameter of interest (the mean difference of accuracy) and $\mathbf{v} \sim \text{MVN}(0, \boldsymbol{\Sigma}_{n\times n})$ is a multivariate Normal noise with zero mean and covariance matrix $\boldsymbol{\Sigma}_{n\times n}$. The covariance matrix $\boldsymbol{\Sigma}$ is characterized as follows: $\boldsymbol{\Sigma}_{ii} = \sigma^2$ and $\boldsymbol{\Sigma}_{ij} = \sigma^2\rho$ for all $i \neq j \in 1, \ldots, n$, where ρ is the correlation and σ^2 is the variance and, therefore, the covariance matrix takes into account the correlation due to cross-validation. Hence, the likelihood model of data is

$$p(\mathbf{x}|\mu, \boldsymbol{\Sigma}) = \frac{\exp(-\frac{1}{2}(\mathbf{x} - \mathbf{1}\mu)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \mathbf{1}\mu))}{(2\pi)^{n/2}\sqrt{|\boldsymbol{\Sigma}|}}. \tag{5}$$

The likelihood (5) does not allow to estimate ρ from data, since the maximum likelihood estimate of ρ is $\hat{\rho} = 0$ regardless the observations (Corani and Benavoli, 2015). This confirms that ρ is not identifiable: thus the Bayesian correlated t-test adopts the same heuristic $\rho = \frac{n_{te}}{n_{tot}}$ suggested by Nadeau and Bengio (2003).

In Bayesian estimation, we aim at estimating the unknown parameters $\mu, \nu = 1/\sigma^2$ and in particular μ, which is the parameter of interest in the Bayesian correlated t-test. To this

end, we consider the following prior:

$$p(\mu, \nu|\mu_0, k_0, a, b) = N\left(\mu; \mu_0, \frac{k_0}{\nu}\right) G(\nu; a, b) = NG(\mu, \nu; \mu_0, k_0, a, b),$$

which is a Normal-Gamma distribution (Bernardo and Smith, 2009, Chap. 5) with parameters (μ_0, k_0, a, b). The Normal-Gamma prior is conjugate to the likelihood (5). If we choose the prior parameters $\{\mu_0 = 0,\ k_0 \to \infty,\ a = -1/2,\ b = 0\}$ (matching prior), the resulting posterior distribution of μ is the following Student distribution:

$$p(\mu|\boldsymbol{x}, \mu_0, k_0, a, b) = St\left(\mu; n-1, \bar{x}, \left(\frac{1}{n} + \frac{\rho}{1-\rho}\right)\hat{\sigma}^2\right), \tag{6}$$

where $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$ and $\hat{\sigma}^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}$. For these values of the prior parameters, the posterior distribution of μ (6) coincides with the Student distribution used in the frequentist correlated t-test in (2). For instance, consider the first data set in Table 1, we have that $\bar{x} = -0.0194$, $\hat{\sigma} = 0.01583$, $\rho = 1/10$, $n = 100$ and so $p(\mu|\boldsymbol{x}, \mu_0, k_0, a, b) = St(\mu; 99, -0.0194, 0.000030)$. The output of the Bayesian analysis is the posterior of μ, $p(\mu|\boldsymbol{x}, \mu_0, k_0, a, b)$, which we can plot and query.

In (6), we have reported the posterior distribution obtained under the matching prior—for which the probability of the Bayesian correlated t-test and the p-value of the frequentist correlated t-test are numerically equivalent. Our aim is to show that although they are numerically equivalent the inferences drawn by the two approaches are very different. In particular we will show that a different interpretation of the same numerical value can completely change our prospective and allow us to make informative decisions. In other words, in this case **the cassock does make the priest**!

3.1 Comparing nbc and aode through Bayesian analysis: a colour thinking

Consider the dataset *squash-unsorted*, with the posterior computed by the Bayesian correlated t-test for the difference between *nbc* and *aode*, as shown in Figure 4. The vertical orange lines mark again the region corresponding to a difference of accuracy of less than 1% (we will clarify the meaning of this region later in the section). In Bayesian analysis, the experiment is summarized by the posterior distribution (in this case a Student distribution). The posterior describes the distribution of the mean difference of accuracies between the two classifiers.

By querying the posterior distribution, we can evaluate the probability of the hypothesis. We can for instance infer the probability that *nbc* is better/worse than *aode*. Formally $P(nbc > aode) = 0.165$ is the integral of the posterior distribution from zero to infinity or, equivalently, the posterior probability that the mean of the differences of accuracy between *nbc* and *aode* is greater than zero. $P(aode > nbc) = 1 - P(nbc > aode) = 0.835$ is the integral of the posterior between minus infinity and zero or, equivalently, the posterior probability that the mean of the differences of accuracy is less than zero.

Can we say anything about the probability that *nbc* is practically equivalent to *aode*? Bayesian analysis can answer this question. First, we need to define the meaning of "practically equivalent". In classification, it is sensible to define that two classifiers whose mean difference of accuracies is less that 1% are practically equivalent. The interval $[-0.01, 0.01]$ thus defines a *region of practical equivalence* (rope) (Kruschke and Liddell, 2015) for classifiers.[3] Once we have defined a rope, from the posterior we can compute the probabilities:

- $P(nbc \ll aode)$: the posterior probability of the mean difference of accuracies being *practically* negative, namely the integral of the posterior on the interval $(-\infty, -0.01)$.
- $P(nbc = aode)$: the posterior probability of the two classifiers being practically equivalent, namely the integral of the posterior over the rope interval.
- $P(nbc \gg aode)$: the posterior probability of the mean difference of accuracies being *practically* positive, namely the integral of the posterior on the interval $(0.01, \infty)$.

$P(nbc = aode) = 0.086$ is the integral of the posterior distribution between the vertical lines (the rope) shown in Figure 4 and it represents the probability that the two classifiers are practically equivalent. Similarly, we can compute the probabilities that the two classifiers are practically different, which are $P(nbc \ll aode) = 0.788$ and $P(nbc \gg aode) = 0.126$.

The posterior also shows the uncertainty in the estimate, because the distribution shows the relative credibility of values across the continuum. One way to summarize the uncertainty is by marking the span of values that are the most credible and cover $q\%$ of the distribution (e.g., $q = 90\%$). These are called the *High Density Intervals* (HDIs) and they are shown in Figure 5 (center) for $q = 50, 60, 70, 80, 90, 95, 99$.

Thus the posterior distribution equipped with rope:

1. estimates the posterior probability of a sensible null hypothesis (the area within the rope);
2. claims significant differences that also have a practical meaning (the area outside the rope);

3. In classification 1% seems to be a reasonable choice. However, in other domains a different value could be more suitable.

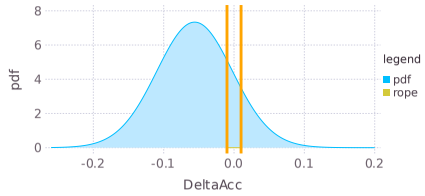


Figure 4: Posterior of the Bayesian correlated t-test for the difference between *nbc* and *aode* in the dataset *squash-unsorted*.

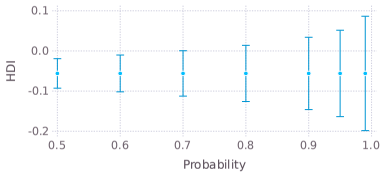


Figure 5: Posterior HDIs of the Bayesian correlated t-test for the difference between *nbc* and *aode* in the dataset *squash-unsorted*.

3. represents magnitude (effect size) and uncertainty (HDIs);

4. does not depend on the sampling intentions.

To see that, we apply Bayesian analysis to the critical cases for NHST presented in the previous section. Figure 6 shows the posterior for *hepatitis* (top), *ecoli* (left) and *iris* (right). For *hepatitis*, all the probability mass is inside the *rope* and so we can conclude that *nbc* and *aode* are practically equivalent: $P(nbc = aode) = 1$. For *ecoli* and *iris*, the probability mass is all in $(-\infty, 0]$ and so $nbc < aode$. However, the posterior gives us more information: there is much more uncertainty in the *iris* dataset. The posteriors thus provide us the same information as the density plots shown in Figures 1–3.

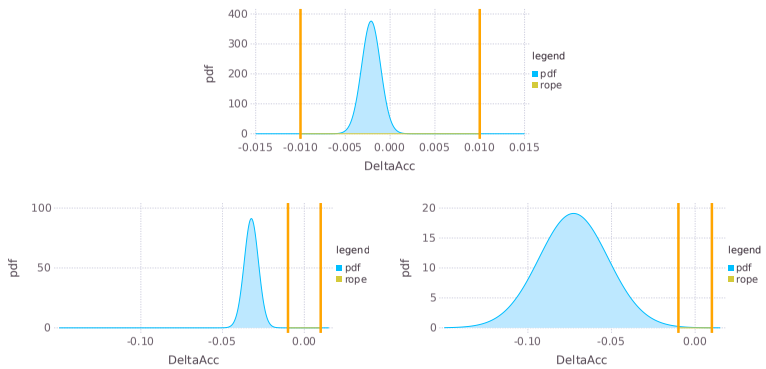


Figure 6: Posterior for *nbc* vs. *aode* in *hepatitis* (top), *ecoli* (left) and *iris* (right).

Consider now Figure 7; those are two examples for which we cannot clearly say whether *nbc* and *aode* are practically equivalent or practically different. We cannot decide in an

obvious way and this is evident from the posterior. Compare these figures with Figure 4, which is similar.

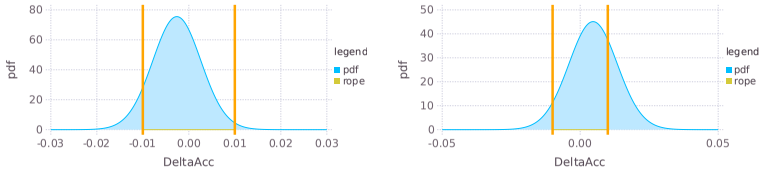


Figure 7: Posterior for *nbc* vs. *aode* in *audiology* (left) and *breast-cancer* (right).

Let us repeat the previous analysis for all 54 datasets: Figure 8 reports the posteriors for *nbc* versus *aode* in all those cases. Looking at the posteriors, we see that there are 12 cases where *aode* is practically better than *nbc*, since all the posterior is outside (to the left of) the rope (in the datasets *ecoli, iris, monks1, monks, mushroom, nursey, optdigits, pasture, segment, spambase, credit, owel*). There are 6 datasets where *nbc* and *aode* are practically equivalent (*hayes-roth*, *hungarian14*, *hepatitis*, *labor*, *wine*, *yeast*), since the entire posterior is inside the rope. We see also that there are no cases where *nbc* is practically better than *aode* (posterior to the right of the rope). The posteriors give us information about the magnitude of effect size, practical difference and equivalence, as well as the related uncertainty.

3.2 Sensible automatic decisions

In machine learning, we often need to perform many analyses. So it may be convenient to devise a tool for automatic decision from the posterior. This means that we have to summarize the posterior in some way, with a few numbers. However, we must be aware that every time we do that we go back to that sort of black and white analysis whose limitations have been discussed before. In fact, by summarizing the posterior, we lose information, but we can do so in a conscious way. We have already explained the advantages of introducing a *rope*, so we can make automatic decisions based on the three probabilities $P(nbc \ll aode)$, $P(nbc = aode)$ and $P(nbc \gg aode)$. In this way, we lose information but we introduce shades in the black and white thinking. $P(nbc \ll aode)$, $P(nbc = aode)$ and $P(nbc \gg aode)$ are probabilities and their interpretation is clear. $P(nbc = aode)$ is the area of the posterior within the rope and represents the probability that *nbc* and *aode* are practically equivalent. $P(nbc \ll aode)$ is the area of the posterior to the left of the rope and corresponds to the probability that *nbc* is practically better than *aode*. Finally, $P(nbc \gg aode)$ is the area to the right of the rope and represents the probability that *aode* is practically better than *nbc*. Since these are the actual probabilities of the decisions we are interested in, in classification, we need not think in terms of Type-I errors to make decisions. We can simply make decisions using these probabilities, which have a direct interpretation—contrarily to *p*-values. For instance we can decide

1. $nbc \ll aode$ if $P(nbc \ll aode) > 0.95$;

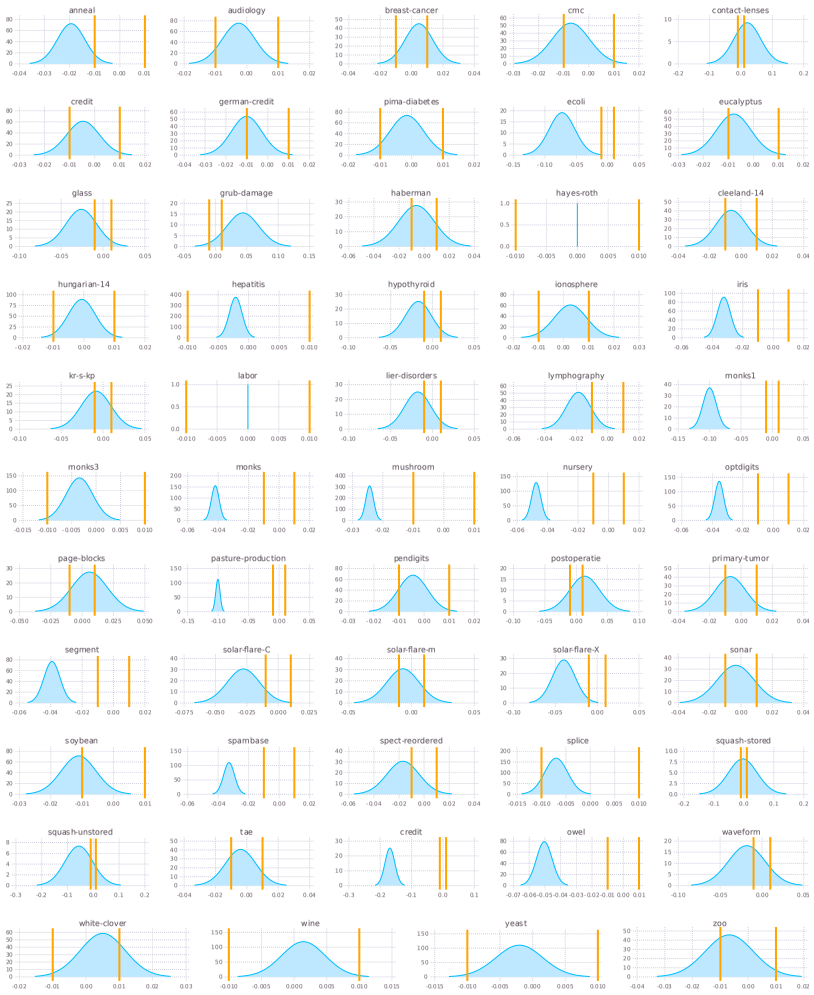


Figure 8: Posteriors of *nbc* versus *aode* for all 54 datasets.

2. $nbc \gg aode$ if $P(nbc \gg aode) > 0.95$;

3. $nbc = aode$ if $P(nbc = aode) > 0.95$.

We can also decide with a probability of, for instance, 0.90 or 0.80 (if this is appropriate in the given context). Table 4 compares the Bayesian decisions based on the above rule with the NHST decisions. The NHST fails to reject the null in 35/54 datasets; Bayesian analysis declares the two classifiers equivalent in 6 of such datasets. Conversely, when NHST rejects the null (in 19/54 datasets), the Bayesian analysis declares nbc$\ll$aode or nbc$\gg$aode in 14 cases, nbc=aode in 1 case and no decision in 4 cases. Overall, Bayesian analysis allows us to make a decision in $6 + 1 + 14 = 21$ datasets, while NHST makes a decision only in 19 cases.

When NHST does not reject the null

pair	Data sets (out of 54)	Bayesian decision	
		nbc=aode	No decision
nbc-aode	35	6	29

When NHST rejects the null

pair	Data sets (out of 54)	Bayesian decision		
pair		nbc=aode	nbc$\ll$aode or nbc$\gg$aode	No decision
nbc-aode	19	1	14	4

Table 4: Results referring to comparisons in which the NHST *rejects* the null.

Sometimes also in Bayesian analysis it may be convenient to think in terms of errors. We can easily do that by defining a loss function. A loss function defines the loss we incur in making the wrong decision. The decisions we can take are nbc$\ll$aode (denoted as a_l), nbc$\gg$aode (a_r), nbc=aode (a_c) or none of them (a_n). Consider for instance the following loss matrix.

$$\begin{array}{c} \\ a_l \\ a_c \\ a_r \\ a_n \end{array} \begin{array}{c} \begin{array}{ccc} a_l & a_c & a_r \end{array} \\ \begin{bmatrix} 0 & 20 & 20 \\ 20 & 0 & 20 \\ 20 & 20 & 0 \\ 1 & 1 & 1 \end{bmatrix} \end{array} \tag{7}$$

The first row gives us the loss we incur in deciding a_l when a_l is true (zero loss), a_c is true (loss is 20) and a_r is true (loss is 20). Similarly for the second and third rows. The last row is the loss we incur in making no decision. The expected loss can be simply obtained by multiplying the above matrix L for the vector of posterior probabilities of nbc$\ll$aode, nbc=aode and nbc$\gg$aode ($p = [p_l, p_c, p_r]^T$), i.e., Lp. The row of Lp corresponding to the lowest loss determines the final decision. Since $0.05 \cdot 20 = 1$, this leads to the same decision rule discussed previously ($P(\cdot) > 0.95$).

3.3 Comparing NHST and Bayesian analysis for other classifiers

In this section we extend the previous analysis to other classifiers besides nbc and aode: hidden naive Bayes (hnb), j48 decision tree (j48) and j48 grafted (j48-gr).

The aim of this section is to show that the pattern described above is general and it also holds for other classifiers. The results are presented in two tables. First we report on the cases in which NHST does not reject the null (Tab. 5). Then we report on the comparisons in which the NHST rejects the null (Tab. 6).

The NHST test does not reject the null hypothesis (Tab. 5) in 341/540 comparisons. In these cases the NHST does not make any conclusion: it cannot tell whether the null hypothesis is true[4] or whether is false but the evidence is too weak to reject it.

By applying the Bayesian correlated t-test with rope and taking decisions as discussed in the previous section, we can draw more informative conclusions. In 74/341=22% of the rejections failed by NHST, the posterior probability of the rope is larger than 0.95, allowing to declare that the two analyzed classifiers are practically equivalent. In the remaining cases, no conclusion can be drawn with probability 0.95.

The rope thus provides a sensible null hypothesis which can be accepted on the basis of the data. When this happens we conclude that the two classifiers are practically equivalent. This is impossible with the NHST.

When NHST does not reject the null

pair	Data sets (out of 54)	Bayesian decision P(rope) >.95	Bayesian decision No decision
nbc–aode	35	6	29
nbc–hnb	30	0	30
nbc–j48	27	2	25
nbc–j48gr	27	2	25
aode–hnb	40	6	34
aode–j48	33	6	27
aode–j48gr	35	6	29
hnb–j48	32	3	29
hnb–j48gr	32	3	29
j48–j48gr	50	40	10
total	341	74	267
rates		74/341= 0.22	267/341=0.78

Table 5: Results referring to comparisons in which the NHST correlated t-test does not reject the null.

Let us consider now the comparisons in which NHST claims significance (Tab. 6). There are 199 such cases. The Bayesian estimation procedure confirms the significance of 142/199 (71%) of them: in these cases it estimates either $P(left) > 0.95$ or $P(right) > 0.95$.[5] In these cases the accuracy of the two compared classifiers are *practically* different. In 51/199 cases (26%) the Bayesian test does not make any conclusion. This means that a sizeable amount of probability lies within the rope, despite the statistical significance claimed by the

4. A point null is however always false, as already discussed.
5. In the comparison nbc-aode, left means $nbc \ll aode$ and right $nbc \gg aode$.

When NHST rejects the null

pair	Data sets (out of 54)	Bayesian decision rope	Bayesian decision difference	Bayesian decision no decision
nbc–aode	19	1	14	4
nbc–hnb	24	0	19	5
nbc–j48	27	0	20	7
nbc–j48gr	27	0	21	6
aode–hnb	14	1	6	7
aode–j48	21	1	14	6
aode–j48gr	19	1	13	5
hnb–j48	22	0	17	5
hnb–j48gr	22	0	17	5
j48–j48gr	4	2	1	1
total	199	6	142	51
rates		6/199= 0.03	142/199=0.71	51/199=0.26

Table 6: Results referring to comparisons in which the NHST *rejects* the null.

frequentist test. In the remaining cases (6/199=3%) the Bayesian test concludes that the two classifiers are practically equivalent ($P(rope) > 0.95$) despite the significance claimed by the NHST. In this case it draws the opposite conclusion from the NHST.

Summing up, the Bayesian test with rope is more conservative, reducing the claimed significances by 30% as compared with the NHST. The Bayesian test is thus more conservative due to the rope, which constitutes a sensible null hypothesis, while the null hypothesis of the NHST is surely wrong. However counting the detection of practically equivalent classifiers as decisions, the Bayesian test with rope takes more decisions than the NHST (222 vs. 199).

4. Comparing two classifiers on multiple data sets

So far we have discussed how to compare two classifiers on the same data set. In machine learning, another important problem is how to compare two classifiers on a collection of q different data sets, after having performed cross-validation on each data set.

4.1 The frequentist approach

There is no direct NHST able to perform such statistical comparison, i.e., one that takes as inputs the m runs of the k-fold cross-validation differences of accuracy for each dataset and returns as output a statistical decision about which classifier is better in all the datasets. The usual NHST procedure that is employed for performing such an analysis has two steps:

1. compute the mean *difference* of accuracy for each dataset (averaging the differences of accuracies obtained in the m runs of the k-fold cross-validation);

2. perform a NHST to establish if the two classifiers have different performance or not based on these mean differences of accuracy.

For our case study, *nbc* vs. *aode*, the mean *differences* of accuracy in each dataset computed from Table 1 are shown in Table 7. We denote these measures generically with $\boldsymbol{z} = \{z_1, \ldots, z_q\}$ (in our case $q = 54$). The recommended NHST for this task is the signed-rank test (Demšar, 2006).

Dataset	Mean Dif.	Dataset	Mean Dif.	Dataset	Mean Dif.
anneal	-1.939	audiology	-0.261	breast-cancer	0.467
cmc	-0.719	contact-lenses	2.000	credit	-0.464
german-credit	-1.014	pima-diabetes	-0.151	ecoli	-7.269
eucalyptus	-0.790	glass	-2.600	grub-damage	4.362
haberman	-0.614	hayes-roth	0.000	cleeland-14	-0.625
hungarian-14	-0.069	hepatitis	-0.212	hypothyroid	-1.683
ionosphere	0.267	iris	-3.242	kr-s-kp	-0.833
labor	0.000	lier-disorders	-1.762	lymphography	-1.863
monks1	-10.002	monks3	-0.343	monks	-4.190
mushroom	-2.434	nursery	-4.747	optdigits	-3.548
page-blocks	0.583	pasture	-10.043	pendigits	-0.443
postoperatie	1.333	primary-tumor	-0.674	segment	-3.922
solar-flare-C	-2.776	solar-flare-m	-0.688	solar-flare-X	-3.996
sonar	-0.338	soybean	-1.112	spambase	-3.284
spect-reordered	-1.684	splice	-0.699	squash-stored	-0.367
squash-unstored	-5.600	tae	-0.400	credit	-16.909
owel	-5.040	waveform	-1.809	white-clover	0.500
wine	0.143	yeast	-0.202	zoo	-0.682

Table 7: Mean difference of accuracy (0–100) for each dataset for nbc minus aode

Frequentist signed-rank test
The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test used when comparing two paired samples. The signed-rank test assumes the observations $z_1, \ldots, z_q$ to be i.i.d. and generated from a symmetric distribution. The test is miscalibrated if the distribution is not symmetric. A strict usage of the test should thus include first a test for symmetry. One should run the signed-rank only if "the symmetry test fails to reject the null" (note that, as we discussed before, this does not actually prove that the distribution is symmetric!). However this would make the whole testing procedure cumbersome, requiring also corrections for the test of multiple hypotheses. In the common practice thus the test for symmetry is not performed, and we follow this practice in this paper.

The test is typically used as follows. The null hypothesis is that the median of the distribution from which the z_i's are sampled is 0; when the test rejects the null hypothesis, it claims that it is significantly different from 0. The test ranks the z_i's according to their absolute value and then compares the ranks of the positive differences and negative

differences. The test statistic is:

$$t = \sum_{\{i:\ z_i \geq 0\}} r_i(|z_i|) \quad = \sum_{1 \leq i \leq j \leq q} t^+_{ij},$$

where $r_i(|z_i|)$ is the rank of $|z_i|$ and

$$t^+_{ij} = \begin{cases} 1 & \textit{if } z_i \geq -z_j, \\ 0 & \textit{otherwise.} \end{cases}$$

For instance, let us consider the following two cases $\boldsymbol{z} = \{-2, -1, 4, 5\}$ or $\boldsymbol{z} = \{-1, 4, 5\}$, then the statistic is $t = 7$ and, respectively, $t = 5$. For a large enough number of samples (e.g., $q > 10$), the statistic under the null hypothesis is approximately normally distributed and in this case the two-sided test is performed as follows:

$$w = \frac{t - \frac{q(q+1)}{4}}{\sqrt{\frac{q(q+1)(2q+1) - \text{tie}}{24}}}, \qquad p = 2(1 - \Phi(|w|)), \tag{8}$$

where p denotes the p-value computed w.r.t. Φ, which is the cumulative distribution function of the standard Normal distribution; *tie* is an adjustment for ties in the data $|\boldsymbol{z}|$, i.e., $z_i = -z_j$ for some i, j, required by the nonparametric test (Sidak et al., 1999; Hollander et al., 2013), while it is zero in case of no ties.

Being non-parametric, the signed-rank is robust to outliers. It assumes commensurability of differences, but only qualitatively: greater differences count more as they top the rank; yet their absolute magnitudes are ignored (Demšar, 2006).

4.1.1 Experimental results

If we apply this method to compare *nbc* vs. *aode*, we obtain p-value=10^{-6} (the rank $t = 162$ with no ties and w is -4.8). Since the p-value is less than 0.05, the NHST concludes that the null hypothesis can be rejected and that *nbc* and *aode* are significantly different. Table 8 reports the p-values of all comparisons of the five classifiers. The pairs *nbc-aode*, *nbc-hnb*, *j48-j48gr* are statistically significantly different. Again by applying this black and white mechanism, we encounter the same problems as before, i.e., we do not have any idea of the magnitude of the effect size, the uncertainty, the probability of the null hypothesis, et cetera. The density plot of the data (the mean differences of accuracy in each dataset) for *nbc* versus *aode* shows for instance that there are many datasets where the mean difference is small (close to zero), see Figure 9. Instead for *j48* versus *j48gr*, it is clear that the difference of accuracy is very small.

4.2 The Bayesian analysis approach

We will present two ways of approaching the comparison between two classifiers in multiple datasets. The first will be based on a nonparametric approach that directly extends the Wilcoxon signed-rank test. The second is a hierarchical model.



Figure 9: Density plot for *nbc* versus *aode* (left) and *j48* versus *j48gr* (right)

Classif. 1	Classif. 2	p-value
nbc	aode	0.000
nbc	hnb	0.001
nbc	j48	0.463
nbc	j48gr	0.394
aode	hnb	0.654
aode	j48	0.077
aode	j48gr	0.106
hnb	j48	0.067
hnb	j48gr	0.084
j48	j48r	0.000

Table 8: p-values for the comparison of the five classifiers.

4.2.1 Nonparametric test

Benavoli et al. (2014) have proposed a Bayesian counterpart of the frequentist sign and signed-rank test, which is based on the Dirichlet process.

Bayesian sign and signed-tank test
Let z denote the scalar variable of interest and $\boldsymbol{z} = \{z_1, \dots, z_q\}$ denotes a vector of i.i.d. observations of z. To derive the Bayesian sign and signed-rank test, we assume a Dirichlet Process (DP) prior on the probability distribution of z. A DP is a distribution over probability distributions such that marginals on finite partitions are Dirichlet distributed. Like the Dirichlet distribution, the DP is therefore completely defined by two parameters: the prior strength $s > 0$ and the prior mean that, for the DP, is a probability measure G_0 on z. If we choose $G_0 = \delta_{z_0}$, i.e., a Dirac's delta centred on the pseudo-observation z_0, the posterior probability density function of Z has this simple expression:

$$p(z) = w_0\delta_{z_0}(z) + \sum_{j=1}^{n} w_j\delta_{z_j}(z), \quad (w_0, w_1, \dots, w_n) \sim Dir(s, 1, \dots, 1), \tag{9}$$

i.e., it is a mixture of Dirac's deltas centred on the observations z_j for $j = 1, \dots, q$ and on the prior pseudo-observation z_0, whose weights are Dirichlet distributed with parameters $(s, 1, \dots, 1)$. We can think about (9) as a hierarchical model: $p(z)$ depends on the weights

w that are Dirichlet distributed. The model (9) is therefore a posterior distribution of the probability distribution of z and encloses all the information we need for the experimental analysis. We can summarize it in different way. If we compute:

$$\theta_l = P(z < -r) = \sum_{i=0}^{q} w_i I_{(-\infty,-r)}(z_i),$$
$$\theta_e = P(|z| \le r) = \sum_{i=0}^{q} w_i I_{[-r,r]}(z_i),$$
$$\theta_r = P(z > r) = \sum_{i=0}^{q} w_i I_{(r,\infty)}(z_i),$$

where the indicator $I_A(z) = 1$ if $z_0 \in A$ and zero otherwise, then we obtain a Bayesian version of the **sign test** that also accounts for the rope $[-r, r]$. In fact, $\theta_l, \theta_e, \theta_r$ are respectively the probabilities that the mean difference of accuracy is in the interval $(-\infty, -r)$, $[-r, r]$, or (r, ∞). Since $(w_0, w_1, \dots, w_n) \sim Dir(s, 1, \dots, 1)$, it can easily be shown that

$$\theta_l, \theta_e, \theta_r \sim Dirichlet(n_l + sI_{(-\infty,-r]}(z_0), n_e + sI_{[-r,r]}(z_0), n_r + sI_{[r,\infty)}(z_0)), \tag{10}$$

where n_l is the number of observations z_i that fall in $(-\infty, -r]$, n_e is the number of observations z_i that fall in $[-r, r]$ and n_r is the number of observations z_i that fall in $[r, \infty)$, obviously $n_l + n_e + n_r = q$. If we neglect $sI_{(-\infty,-r]}(z_0), sI_{[-r,r]}(z_0), sI_{[r,\infty)}(z_0)$, then (10) says that the posterior probability of $\theta_l, \theta_e, \theta_r$ is Dirichlet distributed with parameters (n_l, n_e, n_r). The terms $sI_{(-\infty,-r]}(z_0), sI_{[-r,r]}(z_0), sI_{[r,\infty)}(z_0)$ are due to the prior. Therefore, to fully specify the Bayesian sign test, we must choose the value of the prior strength s and where to place the pseudo-observation z_0 in $(-\infty, -r]$ or $[-r, r]$ or $[r, \infty)$. We will return to this choice in Section 4.3. Instead, if we compute

$$\theta_l = \sum_{i=0}^{q}\sum_{j=0}^{q} w_i w_j I_{(-\infty,-2r)}(z_j + z_i),$$
$$\theta_e = \sum_{i=0}^{q}\sum_{j=0}^{q} w_i w_j I_{[-2r,2r]}(z_j + z_i),$$
$$\theta_r = \sum_{i=0}^{q}\sum_{j=0}^{q} w_i w_j I_{(2r,\infty)}(z_j + z_i), \tag{11}$$

then we derive a Bayesian version of the **signed rank test** (Benavoli et al., 2014) that also accounts for the rope $[-r, r]$. This time the distribution of $\theta_l, \theta_e, \theta_r$ has not a simple closed form but we can easily compute it by Monte Carlo sampling the weights $(w_0, w_1, \dots, w_n) \sim Dir(s, 1, \dots, 1)$. Also in this case we must choose s, z_0, see Section 4.3.

It should be observed that the Bayesian signed-rank test does not require the symmetry assumption about the distribution of the observations z_i. This test works also in case the distribution is asymmetric thanks to the Bayesian estimation approach (i.e., it estimates

the distribution from data). This is another advantage of the Bayesian estimation approach w.r.t. the frequentist null hypothesis tests (Benavoli et al., 2014).

4.2.2 Experiments

Let us start by comparing *nbc* vs. *aode* by means of the Bayesian sign-rank tests *without* rope ($r = 0$). Hereafter we will choose the prior parameter of the Dirichlet as $s = 0.5$ and $z_0 = 0$; we will return to this choice in Section 4.3. Since without rope $\theta_r = 1 - \theta_l$, we have only reported the posterior of θ_l (denoted as "Pleft") that represents the probability that *aode* is better than *nbc*. The samples of the posteriors are shown in Figure 10: this is simply the histogram of 150′000 samples of θ_l generated according to (11). For all samples, it results in θ_l greater than 0.5 and so $\theta_r = 1 - \theta_l$. So we can conclude with probability ≈ 1 that *aode* is better than *nbc*. We can in fact think about the comparison of two classifiers as the inference on the bias (θ_l) of a coin. In this case, all the 150′000 sampled coins from the posterior have a bias that is greater than 0.5 and, therefore, all the coins are always biased towards *aode* (which is then preferable to *nbc*).

This conclusion is in agreement with that derived by the frequentist sign-rank test (very small p-value, see Table 8).

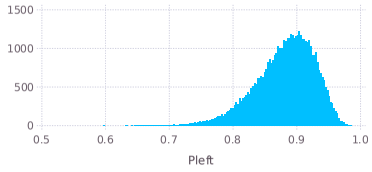


Figure 10: Posterior for *nbc* vs. *aode* for the Bayesian sign-rank test.

The introduction of the *rope* partially changes the previous conclusion. A way to visualize the posterior of $\theta_l, \theta_e, \theta_r$ in this case, is by plotting the 150′000 Monte Carlo samples of these probabilities in barycentric coordinates: each trinomial vector of probabilities is a point in the simplex having vertices $\{(1,0,0),(0,1,0),(0,0,1)\}$. The three vertices are respectively denoted as "aode", "rope" and "nbc" and represent decisions with certainty in favour of "aode", "rope" and, respectively, "nbc". Figure 11 reports the simplex as well as the two-dimensional projections of the posterior for the Bayesian sign-rank test. In particular Figure 11 reports the marginal of the posterior distribution of "aode" vs. "rope" (left); the marginal of the posterior distribution "nbc" vs. "rope" (right). From these two figures we can deduce the other marginal since $\theta_l + \theta_e + \theta_r = 1$. Finally, Figure 11 (bottom) reports the joint of the three variables in barycentric coordinates (we are again exploiting the fact that $\theta_l + \theta_e + \theta_r = 1$). In particular, Figure 11 (bottom) reports the samples from the posteriors (cloud of points), the simplex (the large orange triangle) and three regions (in orange) that are limited by the level curves: $\theta_i \geq \max(\theta_j, \theta_k)$ with $i \neq j \neq k$ (hypothesis i is more probable than both hypotheses j, k together). For instance, the region at the bottom-right of the triangle is relative to the case where "aode" is more probable than "rope" and "nbc" together; the region at the top of the triangle represents the case where "rope" is more

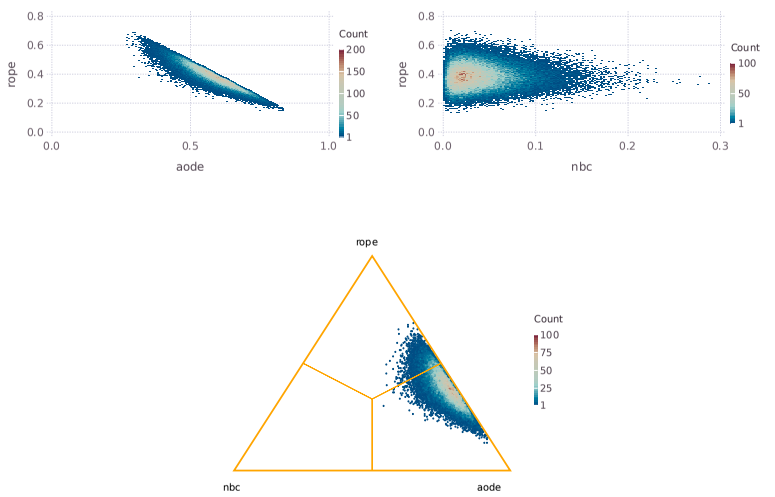


Figure 11: Posterior for *nbc* vs. *aode* for the Bayesian sign-rank test.

probable than "aode" and "nbc" together; the region at the left of the triangle corresponds to the case where "nbc" is more probable than "aode" and "rope" together. Hence, if all the points fall inside one of these three regions, we conclude that such hypothesis is true with probability ≈ 1. Looking at Figure 11 (bottom), it is evident that the majority of cases support *aode* more than *rope* and definitively more than *nbc*. We can quantify this numerically by counting the number of points that fall in the three regions, see first row in Table 9. *aode* is better in 90% of cases, while *rope* is selected in the remaining 10%. We can therefore conclude with probability 90% that *aode* is practically better than *nbc*. Table 9 reports also these probabilities for the other comparisons of classifiers computed using 150′000 Monte Carlo samples. We conclude that *hnb* is practically better than *nbc* with probability 0.999; *aode* and *hnb* are equivalent with probability 0.95; *aode* is better than *j48* and *j48gr* with probability 0.9; *hnb* is better than *j48* and *j48gr* with probability greater than 0.95 and finally *j48* and *j48gr* are practically equivalent. These conclusions are in agreement with the data.

The computational complexity of the Bayesian sign-rank test is low. The comparison of two classifiers (based on 150′000 samples) takes less than one second on a standard computer.

4.3 Choice of the prior

In the previous section, we have selected the prior parameters of the Dirichlet process as $s = 0.5$ and $z_0 = 0$. In terms of rank, this basically means that the prior strength is

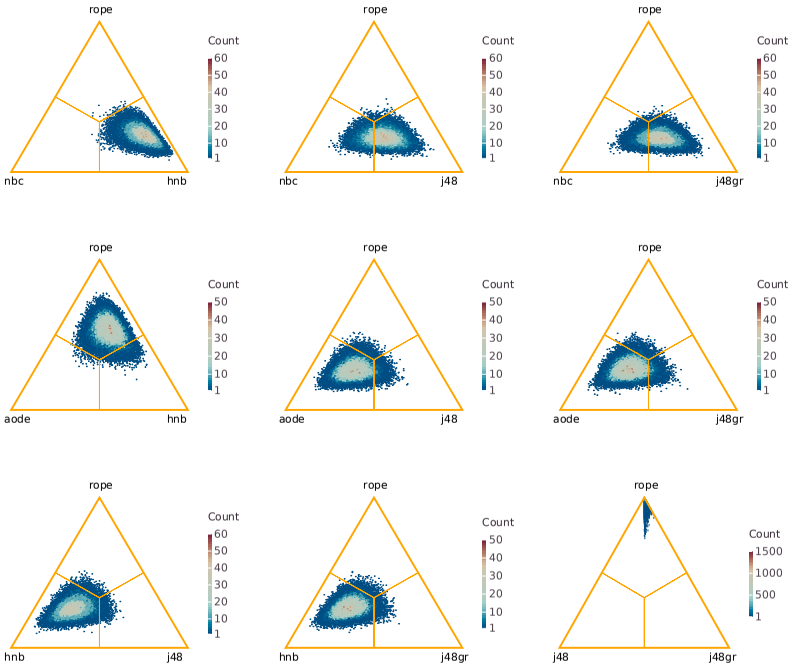


Figure 12: Posterior for *nbc* vs. *aode* for Bayesian sign-rank test.

equivalent to that of one pseudo-observation that is located inside the rope (Benavoli et al., 2014). How are the inferences sensitive to this choice? For instance, we can see how the probabilities on Table 9 would change based on z_0. We have considered two extreme cases $z_0 = -\infty$ and $z_0 = \infty$ and reported these probabilities in Table 10 and 11 (this is an example of robust Bayesian analysis (Berger et al., 1994)). It is evident that the position of z_0 has only a minor effect on the probabilities. We could have performed this analysis jointly by considering all the possible Dirichlet process priors obtained by varying $z_0 \in \mathbb{R}$. This set of Dirichlet priors is called "Imprecise Dirichlet Process" (IDP). IDP allows us to start the inference with very weak prior assumptions, much in the direction of letting data speak for themselves. More details about the properties of IDP and the choice of the prior can be found in Benavoli et al. (2014, 2015b); Walley (1996).

Classif. 1	Classif. 2	left	rope	right
nbc	aode	0.000	0.103	0.897
nbc	hnb	0.000	0.001	0.999
nbc	j48	0.228	0.004	0.768
nbc	j48gr	0.182	0.002	0.815
aode	hnb	0.001	0.956	0.042
aode	j48	0.911	0.026	0.063
aode	j48gr	0.892	0.035	0.073
hnb	j48	0.966	0.015	0.019
hnb	j48gr	0.955	0.020	0.025
j48	j48gr	0.000	1.000	0.000

Table 9: Probabilities for the ten comparisons of classifiers. Left and right refer to the columns Classif. 1 (left) and Classif. 2 (right).

Classif. 1	Classif. 2	left	rope	right
nbc	aode	0.000	0.112	0.888
nbc	hnb	0.000	0.001	0.999
nbc	j48	0.262	0.004	0.734
nbc	j48gr	0.213	0.003	0.784
aode	hnb	0.002	0.961	0.037
aode	j48	0.922	0.024	0.053
aode	j48gr	0.906	0.033	0.061
hnb	j48	0.971	0.014	0.016
hnb	j48gr	0.961	0.018	0.021
j48	j48gr	0.000	1.000	0.000

Table 10: Probabilities for the ten comparisons of classifiers with $z_0 = \infty$. Left and right refer to the columns Classif. 1 (left) and Classif. 2 (right).

4.3.1 Hierarchical models

In Section 3 we have presented the Bayesian correlated t-test that is used for the analysis of cross-validation results on a single dataset. In particular, it makes inference about the mean difference of accuracy between two classifiers in the i-th dataset (μ_i) by exploiting three pieces of information: the sample mean ($\bar{x}_i$), the variability of the data (sample standard deviation $\hat{\sigma}_i$) and the correlation due to the overlapping training set (ρ). This test can only be applied to a single dataset. We have already discussed the fact that there is no direct NHST able to extend the above statistical comparison to multiple datasets, i.e., that takes as inputs the m runs of the k-fold cross-validation results for each dataset and returns as output a statistical decision about which classifier is better in all the datasets. The usual NHST procedure that is employed for performing such analysis has two steps: (1) compute

Classif. 1	Classif. 2	left	rope	right
nbc	aode	0.000	0.096	0.904
nbc	hnb	0.000	0.001	0.999
nbc	j48	0.201	0.004	0.795
nbc	j48gr	0.159	0.002	0.839
aode	hnb	0.001	0.950	0.049
aode	j48	0.892	0.028	0.080
aode	j48gr	0.872	0.037	0.091
hnb	j48	0.957	0.017	0.027
hnb	j48gr	0.944	0.022	0.034
j48	j48gr	0.000	1.000	0.000

Table 11: Probabilities for the ten comparisons of classifiers with $z_0 = -\infty$. Left and right refer to the columns Classif. 1 (left) and Classif. 2 (right).

the mean *difference* of accuracy for each dataset $\bar{x}_i$; (2) perform a NHST to establish if the two classifiers have different performance or not based on these mean differences of accuracy. This discards two pieces of information: the correlation ρ and sample standard deviation $\hat{\sigma}_i$ in each dataset. The standard deviation is informative about the accuracy of $\bar{x}_i$ as an estimator of μ_i. The standard deviation can largely vary across data sets, as a result of each data set having its own size and complexity. The aim of this section is to present an extension of the Bayesian correlated t-test that is able to make inference on multiple datasets and at the same time to account for all the available information (mean, standard deviation and correlation). In Bayesian estimation, this can be obtained by defining a hierarchical model (Corani et al., 2017). Hierarchical models are among the most powerful and flexible tools in Bayesian analysis.

Bayesian hierarchical correlated t-test

The hierarchical correlated t-test is based on following hierarchical probabilistic model:

$$\mathbf{x}_i \sim MVN(\mathbf{1}\mu_i, \mathbf{\Sigma_i}), \tag{12}$$

$$\mu_1 ... \mu_q \sim t(\mu_0, \sigma_0, \nu), \tag{13}$$

$$\sigma_1 ... \sigma_q \sim \text{unif}(0, \bar{\sigma}). \tag{14}$$

Equation (12) models the fact that the cross-validation measures $\boldsymbol{x_i} = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ of the i-th data set are jointly multivariate-normal distributed with the same mean (μ_i), same variance (σ_i) and correlation (ρ). In fact, it states that, for each dataset i, $\mathbf{x}_i$ is multivariate normal with mean $\mathbf{1}\mu_i$ (where $\mathbf{1}$ is a vector of ones) and covariance matrix $\mathbf{\Sigma}_i$ defined as follows: the diagonal elements are σ_i^2 and the out-of-diagonal elements are $\rho\sigma_i^2$, where $\rho = \frac{n_{te}}{n_{tr}}$. This model is the same we discussed in Section 3. Equation (13) models the fact that the mean difference of accuracies in the single datasets, μ_i, depends on μ_0 that is the average difference of accuracy between the two classifiers on the population of data sets. This is the quantity we aim at estimating. Equation (13) assumes the μ_i's to

be drawn from a high-level Student distribution with mean μ_0, variance σ_0^2 and degrees of freedom ν. The choice of a Student distribution at this level of the hierarchical model enables the model to robustly deal with data sets whose μ_i's are far away from the others (Gelman et al., 2014; Kruschke, 2013). Moreover the heavy tails of the Student make more cautious the conclusions drawn by the model.

The hierarchical model assigns to the i-th data set its own standard deviation σ_i, assuming the σ_i's to be drawn from a common distribution, see Equation (14). In this way it realistically represents the fact the estimates referring to different data sets data sets have different uncertainty. The high-level distribution of the σ_i's is unif$(0, \bar{\sigma})$, as recommended by Gelman (2006), as it yields inferences which are insensitive to $\bar{\sigma}$, if $\bar{\sigma}$ is large enough. To this end we set $\bar{\sigma} = 1000 \cdot \bar{s}$ (Kruschke, 2013), where $\bar{s} = \sum_i^q \hat{\sigma}_i / q$.

We complete the model with the prior on the parameters δ_0, σ_0 and ν of the high-level distribution. We assume δ_0 to be uniformly distributed within 1 and -1. This choice works for all the measures bounded within ± 1, such as accuracy, AUC, precision and recall. Other type of indicators might require different bounds.

For the standard deviation σ_0 we adopt the prior $unif(0, \bar{s_0})$, with $\bar{s_0} = 1000 s_{\bar{x}}$, where $s_{\bar{x}}$ is the standard deviation of the $\bar{x}_i$'s.

As for the prior $p(\nu)$ on the degrees of freedom, there are two proposals in the literature. Kruschke (2013) proposes an exponentially shaped distribution which balances the prior probability of nearly normal distributions (ν >30) and heavy tailed distributions (ν <30). We re-parameterize this distribution as a Gamma(α,β) with α=1, β= 0.0345. Juárez and Steel (2010) proposes instead $p(\nu) = \text{Gamma}(2, 0.1)$, assigning larger prior probability to normal distributions.

We have no reason for preferring a prior over another, but the hierarchical model shows some sensitivity on the choice of $p(\nu)$. We model this uncertainty by representing the coefficients α and β of the Gamma distribution as two random variables (hierarchical prior). In particular we assume $p(\nu) = \text{Gamma}(\alpha, \beta)$, with $\alpha \sim \text{unif}(\underline{\alpha}, \bar{\alpha})$ and $\beta \sim \text{unif}(\underline{\beta}, \bar{\beta})$, setting $\underline{\alpha}$=0.5, $\bar{\alpha}$=5, $\underline{\beta}$=0.05, $\bar{\beta}$=0.15. The simulations in (Corani et al., 2017) show that the inferences of the model are stable with respect to perturbations of $\underline{\alpha}$, $\bar{\alpha}$, $\underline{\beta}$, and $\bar{\beta}$, and that the resulting hierarchical generally fits well the experimental data.

These considerations are reflected by the following probabilistic model:

$$\nu \sim Ga(\alpha, \beta), \tag{15}$$

$$\alpha \sim \text{unif}(\underline{\alpha}, \overline{\alpha}), \tag{16}$$

$$\beta \sim \text{unif}(\underline{\beta}, \overline{\beta}), \tag{17}$$

$$\mu_0 \sim \text{unif}(-1, 1), \tag{18}$$

$$\sigma_0 \sim \text{unif}(0, \bar{\sigma_0}). \tag{19}$$

We want to make inference about the μ_i's and μ_0. Such inferences are computed by marginalizing out the σ_i's, and thus accounting for the different uncertainty which characterizes each data set. This characteristic is unique among the methods discussed so far. Computations in hierarchical models are obtained by Markov-Chain Monte Carlo sampling.

A further merit of the hierarchical model is that it jointly estimates the μ_i's while the existing methods estimate independently the difference of accuracy on each data set using the $\bar{x}_i$'s. The consequence of the joint estimation performed by the hierarchical model is that *shrinkage* is applied to the $\bar{x}_i$'s. The hierarchical model thus estimates the μ_i's more accurately than the $\bar{x}_i$'s adopted by the other tests. This result is valid under general assumptions, such as a severe misspecification between the high-level distributions of the true generative model and of the fitted model (Corani et al., 2017). By applying the rope on the posterior distribution of the μ_i's and the μ_0 in a similar way to what discussed for the Bayesian correlated t-test, the model is able to detect equivalent classifiers and to claim significances that have a practical impact.

4.3.2 Experiments

In the experiments, we have computed the posterior of μ_0, σ_0, ν for the ten pairwise comparisons between the classifiers *nbc*, *aode*, *hnb*, *j48* and *j48gr*. As inference we have computed the prediction on the next (unseen) dataset, which is formally equivalent to the inference computed by the Bayesian signed-rank test. For instance, for *nbc* vs. *aode*, we have computed the probabilities that in the next dataset *nbc* is better than *aode* (θ_r), *nbc* is equivalent to *aode* (θ_e), *aode* is better than *nbc* (θ_l). This is the procedure we have followed:

1. we have sampled μ_0, σ_0, ν from the posteriors of these parameters;

2. for each sample of μ_0, σ_0, ν we have defined the posterior of the mean difference of accuracy on the next dataset, i.e., $t(\mu_{next}; \mu_0, \sigma_0, \nu)$;

3. from $t(\mu_{next}; \mu_0, \sigma_0, \nu)$ we have computed the probabilities θ_l (integral on $(-\infty, r]$), θ_e (integral on $[-r, r]$) and θ_r (integral on $[r, \infty)$).

We have repeated this procedure $4'000$ times, obtaining $4'000$ samples of $(\theta_l, \theta_e, \theta_r)$ and the results are shown in Figure 13. The results are quite in agreement with those of the Bayesian signed-rank test. For instance, it is evident that *aode* is clearly better than *nbc*. We can quantify this numerically by counting the number of points that fall in the three regions (see the first row in Table 12). *aode* is better in almost 100% of cases. Table 12 reports also these probabilities for the other comparisons of classifiers computed using $4'000$ Monte Carlo samples. By comparing Tables 12 and 9, we can see that the two tests are substantially in agreement apart from differences in *aode* vs. *j48* and *j48gr*. The hierarchical test is taking into account of all available infromation: the sample mean ($\bar{x}_i$), the variability of the data (sample standard deviation $\hat{\sigma}_i$) and the correlation due to the overlapping training set (ρ), while the Bayesian signed rank only considers $\bar{x}_i$. Therefore, when the two tests differ substantially, it means that there is substantial variability of the cross-validation estimate.

We have implemented the hierarchical model in Stan (`http://mc-stan.org`) (Carpenter et al., 2016), a language for Bayesian inference. The analysis of the results of 10 runs of 10-fold cross-validation on 54 data sets (that means a total of 5400 observations) takes about three minutes on a standard computer.

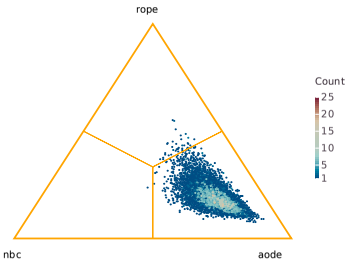


Figure 13: Posterior of *nbc* versus *aode* for all 54 datasets.

Classif. 1	Classif. 2	left	rope	right
nbc	aode	0	0.28	0.72
nbc	hnb	0	0	1
nbc	j48	0.2	0.01	0.79
nbc	j48gr	0.15	0.01	0.84
aode	hnb	0	1	0
aode	j48	0.46	0.51	0.03
aode	j48gr	0.41	0.56	0.03
hnb	j48	0.91	0.07	0.02
hnb	j48gr	0.92	0.05	0.03
j48	j48gr	0	1	0

Table 12: Probabilities for the ten comparisons of classifiers. Left and right refer to the columns Classif. 1 (left) and Classif. 2 (right).

4.4 Choice of the hyper-priors parameters

The choice of of the hyper-priors parameters can be critical in Bayesian hierarchical models. We have conducted a sensitivity analysis by using different constants in the top-level gamma and uniform distributions, to check whether they have any notable influence on the resulting posterior distribution. Whether all gamma/uniform distributions are assumed, the results are essentially identical. More details about this sensitivity analysis are reported in Corani et al. (2017). This means that for the hierarchical model inferences and decisions are stable w.r.t. the choice of the hyper-parameters.

4.5 Bayesian signed rank or hierarchical model?

So far we have presented two methods for comparing two classifiers on multiple datasets: Bayesian signed-rank and hierarchical model. Which one should we use for comparing

classifiers? In our opinion, the hierarchical model is preferable because it takes as inputs the m runs of the k-fold cross-validation results for each dataset and so it makes inference about the mean difference of accuracy between two classifiers in the i-th dataset (μ_i) by exploiting all available information: the sample mean ($\bar{x}_i$), the variability of the data (sample standard deviation $\hat{\sigma}_i$) and the correlation due to the overlapping training set (ρ). Conversely, the Bayesian signed rank only considers $\bar{x}_i$. On the other hand, the hierarchical model is slower than the Bayesian signed rank. In machine learning, we often need to run statistical tests hundreds of times for instance for features selection or algorithms racing and, in this case, it is more convenient to use a light test as the Bayesian signed rank.

5. Comparisons of multiple classifiers

Another important problem with NHST is the issue of multiple hypothesis testing. Considering the results in Table 8, which reports the p-values for the comparison of the five classifiers obtained by the Wilcoxon signed-rank test. From the p-values, we concluded that "*nbc* was found significantly better than *aode* and *hnb*, and algorithms *j48* and *j48gr* were significantly different, while there were no significant differences between other pairs". When many tests are made, the probability of making at least one Type 1 error in any of the comparisons increases. One of the most popular fixes to this problem is the Bonferroni correction. The Bonferroni correction adjusts the p-value at which a test is evaluated for significance according to the number of tests being performed. More specifically, the adjusted p-value is calculated as the original p-value divided by the number of tests being performed. Implicitly, Bonferroni's correction assumes that these test statistics are independent. So in our current example an overall desired significance level of 0.05 would translate into individual tests each using a p-value threshold of $0.05/10 = 0.005$ (we are performing 10 comparisons). In this case, this would not change our previous sections, since all significant p-values were less than 0.005. The Bonferroni correction reduces false rejections but it also increases the number of instances in which the null is not rejected when actually it should have been. Thus, the Bonferroni adjustment can reduce the power to detect an important effect. Motivated by this issue of the Bonferroni correction, researchers have proposed alternative procedures. The goal of these methods typically is to reduce the family-wise error rate (that is, the probability of having at least one false positive) without sacrificing power too much. A natural way to achieve this is by considering the dependence across tests (Westfall et al., 1993).

We have already discussed in Section 2 the pitfalls of NHST Type I error thinking. Type I error underlies these corrections and, therefore, corrections inherit all its problems. The most critical one is that the correction factor depends on the way the analyst intends to conduct the comparison. For instance, the analyst may want to compare *nbc* with the other four classifiers (in this case the Bonferroni correction would be $0.05/4 = 0.0125$—he/she is conducting only four comparisons) or to perform all the ten comparisons (0.05/10) and so on. This creates a problem because two analysts can c draw different conclusions from the same data because of the variety of comparisons that they made. Another important issue with the multiple-comparison procedure based on mean-ranks test is described in Benavoli et al. (2016).

How do we manage the problem of multiple hypothesis testing in Bayesian analysis? Paraphrasing Gelman et al. (2012): "in Bayesian analysis we usually do not have to worry about multiple comparisons. The reason is that we do not worry about Type I error, because the null hypothesis is hardly believable to be true." How does Bayesian Analysis mitigate false alarms? Gelman et al. (2012) suggest using multilevel analysis (in our case a hierarchical Bayesian model on multiple classifiers). Multilevel models perform partial pooling; they shift estimates toward each other. This means that the comparisons of the classifiers are more conservative, in the sense that intervals for comparisons are more likely to include zero. This may be a direction to pursue in future research. In this paper, we instead mitigate false alarms through the rope. The rope mitigates false alarms because it decreases the asymptotic false alarm rate (Kruschke, 2013).

6. Software and available Bayesian tests

All the tests that we have presented in this paper are available in *R* and *Python* code at

`https://github.com/BayesianTestsML/tutorial/`.

Moreover, the code that is necessary to replicate all the analyses we performed in this paper is also available at the above URL in form of *Ipython* notebooks (implemented in *Python* and *Julia*). The software is open source and that can be freely used, changed, and shared (in modified or unmodified form).

Machine learning researchers may be interested in using other Bayesian tests besides the ones we have discussed in this paper. General Bayesian parametric tests can be found in Kruschke (2015) (together with R code) and also in Gelman et al. (2013). We have specialized some of these tests to the case of correlated data, such as the Bayesian correlated t-test (Corani and Benavoli, 2015) discussed in Section 3. We have also implemented several Bayesian nonparametric tests for comparing algorithms: Bayesian rank test (Benavoli et al., 2015b), Friedman test (Benavoli et al., 2015a) and tests that account for censored data (Mangili et al., 2015). Finally, we have developed an extension of the Bayesian sign test to compare algorithms taking into account multiple measures at the same time (accuracy and computational time for instance) (Benavoli and Campos, 2015). For the analysis of multiple data sets, another approach has been proposed by Lacoste et al. (2012) that models each data set as an independent Bernoulli trial. The two possible outcomes of the Bernoulli trial are the first classifier being more accurate than the second or vice versa. This approach yields the posterior probability of the first classifier being more accurate than the second classifier on more than half of the q data sets. A shortcoming is that its conclusions apply only to the q available data sets without generalizing to the whole population of data sets

7. Conclusions

We discourage the use of frequentist null hypothesis significance tests (NHST) in machine learning and, in particular, for comparison of the performance of classifiers. In this, we follow the current trends in other scientific areas. For instance, the journal of Basic and Applied Social Psychology, has banned the use of NHSTs and related statistical procedures (Trafimow and Marks, 2015). We believe that also in machine learning is time to move on from NHST and p-values.

In this paper, we have discussed how Bayesian analysis can be employed instead of NHST. In particular, we have presented three Bayesian tests: Bayesian correlated t-test, Bayesian signed rank test and a Bayesian hierarchical model that can be used for comparing the performance of classifiers and that solve the drawbacks of the frequentist tests. All the code of these tests is freely available and so researchers can already use these tests for their analysis. In this paper, we have mainly discussed the use of Bayesian tests for comparing the performance of algorithms. However, in machine learning, NHST statistical tests are also employed inside the algorithms. For instance, nonparametric tests are used in racing algorithms, independence tests are used to learn the structure of Bayesian networks, etcetera. Bayesian tests can be used to replace all NHST tests because of their advantages (for instance, Bayesian tests can assess whether two algorithms are similar through the use of the rope (Benavoli et al., 2015a)).

Acknowledgements

Research partially supported by the Swiss NSF grant no. IZKSZ2_162188.

References

Murray Aitkin. Posterior Bayes factors. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 111–142, 1991.

M.J. Bayarri and James O Berger. Hypothesis Testing and Model Uncertainty. In *Bayesian Theory and Applications*, pages 365–394. OUP Oxford, 2013.

Alessio Benavoli and Cassio P. Campos. *Advanced Methodologies for Bayesian Networks: Second International Workshop, AMBN 2015, Yokohama, Japan, November 16-18, 2015. Proceedings*, chapter Statistical tests for joint analysis of performance measures. Springer International Publishing, Cham, 2015.

Alessio Benavoli, Francesca Mangili, Giorgio Corani, Marco Zaffalon, and Fabrizio Ruggeri. A Bayesian Wilcoxon signed-rank test based on the Dirichlet process. In *Proceedings of the 30th International Conference on Machine Learning (ICML 2014)*, pages 1–9, 2014.

Alessio Benavoli, Giorgio Corani, Francesca Mangili, and Marco Zaffalon. A Bayesian nonparametric procedure for comparing algorithms. In *Proceedings of the 31th International Conference on Machine Learning (ICML 2015)*, pages 1–9, 2015a.

Alessio Benavoli, Francesca Mangili, Fabrizio Ruggeri, and Marco Zaffalon. Imprecise Dirichlet process with application to the hypothesis test on the probability that X≤Y. *Journal of Statistical Theory and Practice*, 9(3):658–684, 2015b.

Alessio Benavoli, Giorgio Corani, and Francesca Mangili. Should we really use post-hoc tests based on mean-ranks? *Journal of Machine Learning Research*, 17(5):1–10, 2016.

James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer Series in Statistics, New York, 1985.

James O. Berger and Luis R Pericchi. The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, 91(433):109–122, 1996.

James O. Berger and Thomas Sellke. Testing a point null hypothesis: The irreconcilability of p-values and evidence. *Journal of the American statistical Association*, 82(397):112–122, 1987.

James O. Berger, E. Moreno, L. R. Pericchi, M. J. Bayarri, Bernardo, et al. An overview of robust Bayesian analysis. *Test*, 3(1):5–124, 1994.

José M Bernardo and Adrian FM Smith. *Bayesian theory*, volume 405. Wiley Chichester, 2009.

Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007.

Remco R Bouckaert. Choosing between two learning algorithms based on calibrated tests. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 51–58, 2003.

Bob Carpenter, Daniel Lee, Marcus A Brubaker, Allen Riddell, Andrew Gelman, Ben Goodrich, Jiqiang Guo, Matt Hoffman, Michael Betancourt, and Peter Li. Stan: A probabilistic programming language. *Journal of Statistical Software*, in press, 2016.

Giorgio Corani and Alessio Benavoli. A Bayesian approach for comparing cross-validated algorithms on multiple data sets. *Machine Learning*, 100(2):285–304, 2015. doi: 10.1080/s10994-015-5486-z.

Giorgio Corani, Alessio Benavoli, Janez Demsar, Francesca Mangili, and Marco Zaffalon. Statistical comparison of classifiers through Bayesian hierarchical modelling. *Machine Learning in press*, 2017. doi: 10.1007/s10994-017-5641-9.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Janez Demšar. On the appropriateness of statistical tests in machine learning. In *Workshop on Evaluation Methods for Machine Learning in conjunction with ICML*, 2008.

James Dickey. Scientific reporting and personal probabilities: Student's hypothesis. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 285–305, 1973.

Thomas G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1924, 1998.

Ward Edwards, Harold Lindman, and Leonard J Savage. Bayesian statistical inference for psychological research. *Psychological review*, 70(3):193, 1963.

Andrew Gelman. Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian analysis*, 1(3):515–534, 2006.

Andrew Gelman, Jennifer Hill, and Masanao Yajima. Why we (usually) don't have to worry about multiple comparisons. *Journal of Research on Educational Effectiveness*, 5 (2):189–211, 2012.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. CRC press, 2013.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian Data Analysis*, volume 2. Taylor & Francis, 2014.

Myles Hollander, Douglas A Wolfe, and Eric Chicken. *Nonparametric Statistical Methods*, volume 751. John Wiley & Sons, 2013.

Miguel A Juárez and Mark FJ Steel. Model-based clustering of non-gaussian panel data based on skew-t distributions. *Journal of Business & Economic Statistics*, 28(1):52–66, 2010.

Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795, 1995.

John K. Kruschke. Bayesian data analysis. *Wiley Interdisciplinary Reviews: Cognitive Science*, 1(5):658–676, 2010.

John K. Kruschke. Bayesian estimation supersedes the t-test. *Journal of Experimental Psychology: General*, 142(2):573, 2013.

John K. Kruschke. *Doing Bayesian Data Analysis: A Tutorial with R, Jags and Stan*. Academic Press, 2015.

John K. Kruschke and Torrin M Liddell. The Bayesian New Statistics: Two Historical Trends Converge. *Available at SSRN 2606016*, 2015.

Alexandre Lacoste, François Laviolette, and Mario Marchand. Bayesian comparison of machine learning algorithms on single and multiple datasets. In *Proc.of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-12)*, pages 665–675, 2012.

Bruno Lecoutre and Jacques Poitevineau. *The Significance Test Controversy Revisited*. Springer, 2014.

Francesca Mangili, Alessio Benavoli, Cassio P. de Campos, and Marco Zaffalon. Reliable survival analysis based on the Dirichlet Process. *Biometrical Journal*, 57:10021019, 2015.

Kevin P Murphy. *Machine Learning: a Probabilistic Perspective*. MIT press, 2012.

Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003.

Steven L. Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1:317–328, 1997.

Zbynek Sidak, Pranab Sen, and Jaroslav Hajek. *Theory of Rank Tests*. Probability and Mathematical Statistics. Elsevier Science, 1999.

David Trafimow and Michael Marks. Editorial. *Basic and applied social psychology*, 37(1): 1–2, 2015.

Peter Walley. Inferences from multinomial data: learning about a bag of marbles. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):3–57, 1996.

Ronald L Wasserstein and Nicole A Lazar. The ASA's statement on p-values: context, process, and purpose. *The American Statistician*, (just-accepted):00–00, 2016.

Peter H Westfall, S Stanley Young, and S Paul Wright. On adjusting p-values for multiplicity. *Biometrics*, 49(3):941–945, 1993.

Ian H Witten, Eibe Frank, and Mark Hall. *Data Mining: Practical Machine Learning Tools and Techniques (third edition)*. Morgan Kaufmann, 2011.

Journal of Machine Learning Research 18 (2017) 1-44 Submitted 6/16; Revised 6/17; Published 8/17

Relational Reinforcement Learning for Planning with Exogenous Effects

David Martínez [1] dmartinez@iri.upc.edu
Guillem Alenyà [1] galenya@iri.upc.edu
Tony Ribeiro [2] tony.ribeiro@ls2n.fr
Katsumi Inoue [3] inoue@nii.ac.jp
Carme Torras [1] torras@iri.upc.edu
[1] *Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain*
[2] *Laboratoire des sciences du numérique de Nantes (LS2N), Nantes, France*
[3] *National Institute of Informatics, Tokyo, Japan*

Editor: George Konidaris

Abstract

Probabilistic planners have improved recently to the point that they can solve difficult tasks with complex and expressive models. In contrast, learners cannot tackle yet the expressive models that planners do, which forces complex models to be mostly handcrafted. We propose a new learning approach that can learn relational probabilistic models with both action effects and exogenous effects. The proposed learning approach combines a multi-valued variant of inductive logic programming for the generation of candidate models, with an optimization method to select the best set of planning operators to model a problem. We also show how to combine this learner with reinforcement learning algorithms to solve complete problems. Finally, experimental validation is provided that shows improvements over previous work in both simulation and a robotic task. The robotic task involves a dynamic scenario with several agents where a manipulator robot has to clear the tableware on a table. We show that the exogenous effects learned by our approach allowed the robot to clear the table in a more efficient way.

Keywords: Learning Models for Planning, Model-Based RL, Probabilistic Planning, Active Learning, Robot Learning

1. Introduction

Task planning has been a useful tool to solve complex problems where an agent has to reach a goal by executing actions, including robotic tasks (Kulick et al., 2013; Martínez et al., 2015a) and scheduling (Zhu et al., 2014). Task planners can find solutions that optimize a reward function for any state, even if this state was unexpected. The reward function encodes the priorities when solving the task, and new actions can be added easily to extend tasks.

Traditionally, applications have mostly used deterministic planners (Hoffmann and Nebel, 2001; Helmert, 2006) as they are easier to set-up and can solve complex problems with many variables. Recently the planning community has improved successfully the existing planners to solve more expressive and complex tasks. In the latest International Probabilistic Planning Competition (IPPC 2014) (Vallati et al., 2015), the best planners were able to solve probabilistically interesting tasks (Little and Thiebaux, 2007) that included both action effects and exogenous effects (Kolobov et al., 2012; Keller and Eyerich, 2012) and many relational variables. We consider that action effects are those that occur when the agent executes an action (e.g. a robot moves to a new position, the robot grasps an object), while exogenous effects are those that do not depend on an action (e.g. people moving in the street, a traffic light turns red, a plant grows).

However, these planners rely on a model, that has to be either handcrafted or learned. In contrast to planners, learners cannot tackle yet the expressive models as planners do, which forces complex models to be mostly handcrafted. Relational action models with uncertain effects from a log of completely observable state-action-state transitions can be learned using the approach by Pasula et al. (2007), but in this paper we propose a new method that, in addition to relational probabilistic action models, it can also learn exogenous effects.

Relational model learners can be integrated in Reinforcement Learning (RL) approaches such as REX (Lang et al., 2012), allowing an agent to learn a task autonomously. However, heuristic model learners may have trouble learning certain domains if exploration is scarce when using REX. Thus we show that our approach can be integrated in V-MIN (Martínez, Alenyà, and Torras, 2015b), an extension of REX that can successfully learn such domains by actively requesting a few teacher demonstrations.

Finally, we will present how this learner can be integrated in a robot to improve its performance in tasks where exogenous effects are important. Specifically, we show a robot that has to help clearing the tableware on a table. Exogenous effects allow us to model the frequency at which different types of tableware arrive, and to coordinate with the waiter robots that take piles of tableware back to the kitchen. When a model with such exogenous effects can be learned, the performance of the task is increased significantly.

To summarize, we propose a novel method that takes as input a set of state-action-state transitions, and learns a relational model with probabilistic and exogenous effects to be used for planning. We also show that the learner can be integrated in a RL framework to learn and solve new tasks. This RL approach can be integrated in a robot decision-maker to improve the performance in the task of clearing the tableware of a table.

This work is an extension of Martínez et al. (2016), where the model learner was presented. The most significant additions are the integration with RL, the experiments with a robot, and showing the way input transitions must be normalized.

2. Background

In this section we present the formulation that we will use throughout this paper, as well as the background on reinforcement learning required to understand Section 4.

2.1 Formulation

Two types of representations are combined in this work, a relational one for the planning operators, and a propositional one that will be used internally by some parts of the learner. We assume complete observability and uncertain effects.

©2017 David Martínez, Guillem Alenyà, Tony Ribeiro, Katsumi Inoue and Carme Torras.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-326.html.

2.1.1 Relational Formulation

Literals l are expressions of the form $(\neg)var(t_1, ..., t_m)$ where var is a predicate symbol that represents a variable, $(\neg)$ means that the atom may be optionally negated, and t_i are the terms. Terms can be variables, which have a preceding "?" symbol (e.g. ?X), and can also be objects, which are represented without an "?" symbol (e.g. box1). We use a relational representation where expressions take objects as arguments to define their grounded counterparts. A state s is defined as a conjunction of grounded literals l^g that follow the closed world assumption $s = l^g_1, ..., l^g_N$.

A planning operator $o \in \mathcal{O}$ defines the probability of a literal taking a value based on a set of preconditions. Operators take the form

$$o(t_1, ..., t_n) = l_h : p_o \leftarrow l_1 \wedge \cdots \wedge l_m, (a) \tag{1}$$

where l_h is the head of the operator, p_o is the probability of l_h being true in the next state given that the body and the action are satisfied, $l_1 \wedge \cdots \wedge l_m$ are the literals in the body, (a) is an optional action, and t_i are the terms that may appear in the head, body and action. The action is optional so that operators can capture both action effects when there is an action, and exogenous effects when there is no action. Operators are not Horn clauses as negation can appear in both the body and the head. Also note that this representation does not model correlated effects, as each operator can only have one effect. There can be two operators with the same body and different heads, but they would be independent.

Example 1 *An example operator for an action is:*
$o1(?X, ?Y, ?Z) = \textit{robot-at}(?X, ?Y) : 0.82 \leftarrow \textit{robot-at}(?X, ?Z) \wedge \textit{adj}(?Z, ?Y) \wedge \textit{move}(?X, ?Y)$.
If the "move" action is applied, the robot would move to $(?X, ?Y)$ *with a probability of* 0.82 *if it is on an adjacent position.*

A grounded operator only has objects as terms. If an operator o has n variables, its groundings $Gr(o)$ are a set of operators, each taking one of the possible combinations of n objects.

Example 2 *Having the objects* $\{x1, y1, y2\}$ *and the operator* $o1(?X, ?Y)$*, the possible groundings can be obtained by substituting* $?X$ *and* $?Y$ *for every permutation of 2 objects. One possible grounding is:*
$o1(x1, y2, y1) = \textit{robot-at}(x1, y2) : 0.82 \leftarrow \textit{robot-at}(x1, y1) \wedge \textit{adj}(y1, y2) \wedge \textit{move}(x1, y2)$.

The transition dynamics are defined by a set of planning operators $\mathcal{O}$. A grounded operator o_g is said to cover a state-action pair (s, a) when the literals of the body are in s, and the optional action of the operator is either a, or the operator has no action:

$$cov(o_g, s, a) = (body(o_g) \subset s) \wedge ((action(o_g) = a) \vee (action(o_g) = \varnothing)).$$

Likewise, a non-grounded operator o covers (s, a) if one of its groundings does:

$$cov(o, s, a) = \exists\, o_g \in Gr(o) \mid cov(o_g, s, a).$$

Transitions T are defined as triples $t = (s, a, s')$ where s' is the successor state of s after executing a. A successor state s' is obtained by applying all groundings of all operators to (s, a). When a grounded operator o_g is applied to (s, a), its head is added to the state s' with a probability p_{o_g} if $cov(o_g, s, a)$.

Example 3 *An example transition is:*
s*:* $\textit{adj}(y1, y2) \wedge \textit{adj}(x1, x2) \wedge \textit{robot-at}(x1, y1) \wedge \textit{moving-obstacle}(x1, y2)$
a*:* $\textit{move}(x2, y1)$
s'*:* $\textit{adj}(y1, y2) \wedge \textit{adj}(x1, x2) \wedge \textit{robot-at}(x2, y1) \wedge \textit{moving-obstacle}(x2, y2)$
Where the action "move" modifies the literal "robot-at", and an exogenous effect modifies "moving-obstacle".

We require operators to be mutually exclusive, there cannot be two operators with the same head atom that cover the same (s, a) as their heads may conflict. One example of such conflict would be $[o_{g,1}(r1) = at(r1) : 0.8 \leftarrow ...]$ and $[o_{g,2}(r1) = \neg at(r1) : 0.6 \leftarrow ...]$ where both heads cannot hold at the same time as one contradicts the other. The objective of this work is to learn models that can be used by planners, and planners require conflict-free operators. A planner has to know precisely the expected effects of applying a planning operator. If two different operators were to make conflicting changes, the effects would be undefined. This constraint is similar to the one in Pasula et al. (2007)'s learner, but in this case a different operator can be applied to each head.

If s' is a successor state of s, we define $changes(s, s')$ as the set of literals $\{c \in s', c \notin s\}$. Given a transition $t = (s, a, s')$ and a grounded operator o_g, a transition change $c \in changes(s, s')$ has a likelihood

$$P(c \mid o_g) = \begin{cases} p_{o_g}, & cov(o_g, s, a) \wedge (c = head(o_g)) \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

And a set of non-grounded operators $\mathcal{O}$ gives the following likelihood to a change c:

$$P(c|\mathcal{O}) = \begin{cases} P(c|o_g), & \exists!\, o_g \in Gr(\mathcal{O}) \mid P(c|o_g) > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $\exists!$ is the operator for uniqueness quantification. If more than one operator covers the same change given the same state-action pair, there is a conflict and the behavior is undefined, so a likelihood of 0 is given.

2.1.2 Propositional Formulation

On the propositional level, atoms are multi-valued variables that take the form p^x, where p is a predicate symbol, x is the value, and atoms have no parameters. A *state* is a conjunction of propositional atoms $s = var_1^{val_1} \wedge \cdots \wedge var_n^{val_n}$ such that $var^{val'}, var^{val''} \in s$ implies $val' = val''$.

We consider a *probabilistic multi-valued logic program* as a set of *propositional rules* r of the form

$$r = var_0^{val_0} : p_r \leftarrow var_1^{val_1} \wedge \cdots \wedge var_n^{val_n} \tag{4}$$

where $var_i^{val_i}$ for all $i \geq 0$ are atoms and p_r is a probability. A propositional rule r is interpreted as follows: with probability p_r the variable var_0 takes the value val_0 in the next state if all variables var_i have the value val_i in the current state. For any rule r, $var_0^{val_0}$ is called the *head* and $var_1^{val_1} \wedge \cdots \wedge var_n^{val_n}$ are the literals in the *body*.

Example 4 *Let's consider the following rules,*
$r_1 = a^1 : 0.7 \leftarrow b^1$,
$r_2 = b^2 : 1.0 \leftarrow a^1 \wedge b^0$,
$r_3 = a^0 : 0.3 \leftarrow b^2$,
then the logic program $P = \{r_1, r_2, r_3\}$ *is a probabilistic multi-valued logic program.*

Finally, propositional transitions E are defined as pairs of states $e = (s, s')$ where s' is the successor state of s.

2.2 Reinforcement Learning

Reinforcement learning (RL) is a framework for sequential decision-making problems where the dynamics of the environment are not known in advance. At each time step, the agent takes an observation, decides which action to execute, and is given a reward. The agent's goal is to maximize the total sum of discounted rewards over time. In model-based RL the agent maintains a model that is updated with every transition, and a planner uses this model to select the action to be executed.

Formally, fully-observable problems with uncertainty are represented with Markov Decision Processes (MDP). A finite MDP is a five-tuple $\langle S, A, M, R, \gamma \rangle$ where S is a set of states, A is the set of actions that an agent can perform, $M : S \times A \times S \rightarrow [0, 1]$ is the transition model, $R : S \times A \rightarrow \mathbb{R}$ is the reward function and $\gamma \in [0, 1)$ is the discount factor.

The goal of RL algorithms is to find a policy $\pi : S \rightarrow A$ that chooses the actions to maximize the value function. The sum of expected rewards is the value function $V^\pi(s) = E[\sum_t \gamma^t R(s_t, a_t) | s_0 = s, \pi]$ which is to be maximized.

As the model is not known in advance, RL approaches have to balance exploration (visiting state-action pairs to learn the dynamics related to unknown parts of the model, thereby allowing better policies to be obtained in the future) and exploitation (executing actions to maximize rewards based on the current model).

3. Model Learner

In this section we show how to learn a relational probabilistic model with exogenous effects from a log of input state-action transitions. The learned model will consist of a set of planning operators that define both action effects and exogenous effects.

There exist previous works that tackle the problem of learning models. Some of them estimate the parameters of a model (Moldovan et al., 2012; Thon et al., 2011), but the structure of the model has to be provided and only its parameters are learned. The work by Jiménez et al. (2008) can capture the preconditions and uncertainty in the models given that the possible effects are known in advance. A more complete approach by Sykes et al. (2013) learns probabilistic logic programs, but the restrictions for the initial set of candidate rules need to be manually coded. In contrast, we learn the complete model and no restrictions are required.

Most approaches that learn complete models handle deterministic tasks, and although they can tackle partial observability (Molineaux and Aha, 2014; Mourão et al., 2012; Zhuo and Kambhampati, 2013) or apply transfer learning (Zhuo and Yang, 2014), they do not consider uncertain effects. In this work we focus on models with uncertain effects.

The most similar approaches to ours are those that learn relational action models with uncertain effects (Pasula et al., 2007; Deshpande et al., 2007; Mourão, 2014). They learn the effects that each action may have for each set of preconditions, which are then represented with probabilistic STRIPS-like models. However, they do not learn exogenous effects that are not related to any action.

These methods cannot be easily extended to learn exogenous effects. Pasula et al. (2007) and Deshpande et al. (2007)'s approach use a local search algorithm that works well when transitions are explained by one rule, but faces many local minima when tackling domains with exogenous effects, as two or more new rules may have to be added to properly explain a transition. Mourão (2014)'s approach exhibits a similar problem since it learns one rule per transition. To favor a better understanding of this problem, let's assume that we have a transition that is optimally modeled by 2 rules, one of which relates to an exogenous effect. These 2 rules are likely to have a very low score individually as they only cover part of the transition. The local search in Pasula's learner would choose a rule that covered the complete transition to maximize the score, even if the rule had a lot of noise and low probability to guess the effect. As a local search is used, the learner can only make one change at a time, and selecting only one of the 2 optimal rules would decrease the score, so they would never be chosen.

Note that although those methods cannot learn exogenous effects, they have other advantages. Pasula et al. (2007)'s approach can learn correlated effects while Mourão (2014)'s learner can also tackle partial observability. Therefore the best method will depend on the type of domain being learned.

The problem of learning minimal effects from a log of input data transitions is known to be NP-Hard (Walsh, 2010). The approaches shown before, as well as our approach, apply heuristics to find solutions with any number of input experiences. Optimal approaches that learn complete probabilistic models have also been proposed (Walsh et al., 2009), but they require too many input experiences or assumptions not made in the current work.

The learned model will consist of a set of planning operators that define the different effects. The proposed method can be divided into two parts:

- *Candidate planning operator generation.* Candidates are generated with the LFIT (Learning From Interpretation Transitions) framework (Inoue et al., 2014). LFIT induces a set of propositional rules that realize the given input transitions. Specifically, an algorithm that guarantees to learn the set of minimal rules is used (Ribeiro and Inoue, 2014).

- *Planning operator selection.* To select the best subset of candidates, we define a score function that is maximized by candidates that explain input transitions while being general enough. Based on this score function, a search optimization method guided by an heuristic function is proposed. Moreover, suboptimal solutions to make complex tasks tractable are provided.

Our approach combines (a) LFIT on the propositional level to ensure that candidates are minimal, (b) an optimization method that works on the relational level to apply relational generalizations when selecting subsets, and (c) grounded input data. Since, as mentioned, the approach requires three different types of data (grounded, relational and propositional), data transformation methods are needed.

3.1 LFIT

The LFIT framework (Inoue et al., 2014) is used to obtain the set of probabilistic candidate rules that model the dynamics. Given a batch of propositional transitions (s, s'), LFIT induces a *normal logic program* that realizes the given transitions. This framework has been extended (Ribeiro and Inoue, 2014) with a new algorithm that guarantees that the learned rules are minimal: the body of each rule constitutes a prime implicant to infer the head. It is based on a top-down method that generates hypotheses by *specialization* from the most general rules. Moreover, the framework has been adapted to capture also probabilistic dynamics (Martínez et al., 2015c).

In this section we explain briefly how to obtain probabilistic logic programs with minimal rules. For more details please refer to the work by Ribeiro and Inoue (2014) and Martínez et al. (2015c).

3.1.1 Rule Specialization

To learn multi-valued logic programs with minimal rules we need to define the ground resolution and the least specialization for multi-valued propositional variables.

Definition 1 (Subsumption) *Let r_1 and r_2 be two rules. If $head(r_1) = head(r_2)$ and $body(r_1) \subseteq body(r_2)$ then r_1* subsumes *r_2. Let P be a logic program and r be a rule. P* subsumes *r if there exists a rule $r' \in P$ that subsumes r.*

We say that a rule r_1 is *more general* than another rule r_2 if r_1 subsumes r_2 and $body(r_1) \subset body(r_2)$. In particular, a rule r is *most general* if there is no rule $r'(\neq r)$ that subsumes r ($body(r) = \emptyset$).

Example 5 *Let r_1 and r_2 be the two following rules: $r_1 = (a^1 \leftarrow b^1)$, $r_2 = (a^1 \leftarrow a^0 \wedge b^1)$, r_1 subsumes r_2 because $(body(r_1) = \{b^1\}) \subset (body(r_2) = \{a^0, b^1\})$. When r_1 appears in a logic program P, r_2 is useless for P, because whenever r_2 can be applied, r_1 can be applied.*

Definition 2 (Complement) *Let r_1 and r_2 be two rules, r_2 is a* complement *of r_1 on var^{val} if $var^{val} \in body(r_1)$, $var^{val'} \in body(r_2), val \neq val'$ and $(body(r_2) \setminus \{var^{val'}\}) \subseteq (body(r_1) \setminus \{var^{val}\})$.*

Definition 3 (Ground resolution) *Let r be a rule, P be a logic program and $\mathcal{B}$ be a set of atoms, r can be* generalized *on var^{val} if $\forall var^{val'} \in \mathcal{B}, val \neq val'$, $\exists r' \in P$ such that r' is a complement of r on var^{val}:*

$$generalise(r, P) = head(r) \leftarrow body(r) \setminus \{var^{val}\}$$

Definition 4 (Least specialization) *Let r_1 and r_2 be two rules such that $head(r_1) = head(r_2)$ and r_1 subsumes r_2. Let $\mathcal{B}$ be a set of atoms. The least specialization $ls(r_1, r_2, \mathcal{B})$ of r_1 over r_2 w.r.t $\mathcal{B}$ is*

$$ls(r_1, r_2, \mathcal{B}) = \{head(r_1) \leftarrow body(r_1) \wedge var^{val'}\} \text{ such that}$$
$$var^{val} \in body(r_2) \setminus body(r_1),\ var^{val'} \in \mathcal{B},\ val' \neq val$$

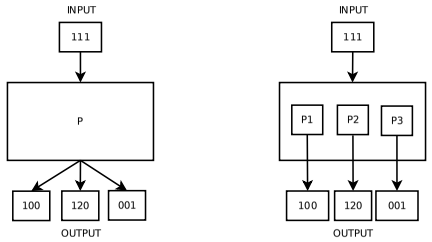


Figure 1: **Left:** A black-box probabilistic system where one program has multiple possible next states. **Right:** The LFIT algorithm uses a set of deterministic programs that form an equivalent probabilistic system. Each deterministic program outputs one of the possible next states.

Least specialization can be used on a rule r to avoid the subsumption of another rule with a minimal reduction of the generality of r. By extension, least specialization can be used on the rules of a logic program P to avoid the subsumption of a rule with a minimal reduction of the generality of P. Let P be a logic program, $\mathcal{B}$ be a set of atoms, r be a rule and S be the set of all rules of P that subsume r. The least specialization $ls(P, r, \mathcal{B})$ of P by r w.r.t $\mathcal{B}$ is as follows:

$$ls(P, r, \mathcal{B}) = (P \setminus S) \cup (\bigcup_{r_P \in S} ls(r_P, r, \mathcal{B}))$$

3.1.2 Probabilistic LFIT

To learn probabilistic systems, LFIT represent them as a set of deterministic programs. Given a state s, we compute the set of possible next states by querying each program. Figure 1 (left) shows a probabilistic system P where the possible successors of the state 111 can be 100, 120 or 001. This behavior can be captured by a set of three deterministic programs $\{P1, P2, P3\}$ (right), such that $P1$ obtains a transition $e_1 = (\{a^1, b^1, c^1\}, \{a^1, b^1, c^1\})$, $P2$ obtains $e_2 = (\{a^1, b^1, c^1\}, \{a^1, b^2, c^0\})$, and $P3$ obtains $e_3 = (\{a^1, b^1, c^1\}, \{a^0, b^0, c^0\})$. Each program will also encode the behavior of the system for all possible other states. This representation allows us to capture any probabilistic system dynamics. It provides a model than can reproduce the possible behavior but no information about which transition is more likely to occur. To output such information, probabilities have to be extracted from observed transitions and reflected in the model. There are many ways of encoding likelihood and probabilities in logic programs, hence here we choose to simply consider rules likelihood independently for each rule. For each rule, we simply check how many times it could be applied in the input transitions and how many times the next state could effectively be the result of the use of this rule. This provides us with a simple likelihood for each rule, a ratio matches/realizes that can be used to give the likelihood of the possible next states. Other methods and normalization of the probabilities could be performed with the same sets of programs.

In this work we use an extension of *LFIT* (Martínez et al., 2015c) to learn a set of deterministic logic programs that can model probabilistic domains. The main idea is that when two transitions are not consistent, we need two different programs to realize them. The first program will realize the first transition and the second one will realize the second transition. The algorithm will output a set of logic programs such that every transition given as input is realized by at least one of those programs.

Probabilistic LFIT

- **Input**: a set of propositional transitions E and a set of atoms $\mathcal{B}$.
- Step 1: Initialize a set of logic programs P with one program P_1 with fact rules for each atom of $\mathcal{B}$.
- Step 2: Pick $e = (s, s')$ in E, check consistency of e with all programs of P:
- If there is no logic program in P that realizes e then
  - Copy one of the logic programs P_i into a P'_i and add rules in P'_i to realize e.
  - Use ground resolution to generalize P'_i.
- Step 3: Revise all logic programs that realize e by using least specialization.
- Step 4: If there is a remaining transition in E, go to step 2.
- Step 5: Compute the probability of each rule of all programs P_i according to E.
- **Output**: P a set of multi-valued logic programs that realize E.

The detailed pseudo-code of the *Probabilistic LFIT* is given in Algorithm 1. The algorithm starts with one logic program that contains all fact rules (lines 1-7). Each input transition e is analyzed one by one. If no program can realize the observed transition (lines 10-20), one is copied and rules are added into this copy so that it realizes the transition (lines 21-28). The programs that realize e are then revised using least specialization like in LFIT (Ribeiro and Inoue, 2014) (lines 29-36). The programs that do not realize the transition realize another one previously observed that is not consistent with the new one because of the non-determinism of the system. Those programs cannot be specialized by this transition because we would lose the information of the previous transition they realize. Finally, the likelihood of each rule is computed by just counting the number of observations covered/realized (lines 39-53). The algorithm outputs a set of logic programs, each one of them realizes some transitions of E, and all transitions of E are realized by at least one program. The complexity belongs to $O(d \cdot |E| \cdot nv^n)$ for run time and $O(d \cdot 2^n)$ for memory, with d the maximal out-degree of an observed state (degree of non-determinism), E the set of input transitions, n the number of variables and v the maximal domain of a variable (the maximal number of values a variable may have).

Finally, all the learned programs P are combined in one set that will become the set of planning operator candidates.

Algorithm 1 Probabilistic LFIT($E, \mathcal{B}$)

Input: a set of pair of states E and a set of atoms $\mathcal{B}$
Output: P a set of logic programs

```
1: E' := ∅
2: P := ∅
   // Initialize P: one program with the most general rules
3: P_1 := ∅
4: for each var^val ∈ B do
5:     P_i := P_i ∪ {var^val ←}
6: end for
7: P := P ∪ P_1
   // 2) Revise P to realize every transition
8: while E ≠ ∅ do
9:     Pick e = (s, s'); E := E \ {e}
       // 2.1) Check if e = (s, s') is realizable
10:    for each logic program P_i of P do
11:        realize_e := true
12:        for each var^val ∈ s' do
13:            if ∄r ∈ P_i | body(r) ⊆ s, head(r) ∈ s' then
14:                realize_e := false
15:            end if
16:        end for
17:        if realize_e = true then
18:            break
19:        end if
20:    end for
       // 2.2) Construct a logic program that realizes e
21:    if realize_e = false then
22:        for each var^val ∈ s' do
23:            r := var^val : 1.0 ← ⋀_{B_i ∈ s} B_i
24:            choose a P_i ∈ P
25:            P'_i := P_i
26:            P := AddRule(P'_i, r, B)
27:        end for
28:    end if
       // 3) revise logic programs that realize e
29:    for each logic program P_i of P do
30:        for each var^val ∈ s' do
31:            for each var^val' ∈ B, val' ≠ val do
32:                r^s_{var^val'} := var^val : 1.0 ← ⋀_{m_i ∈ s} m_i
33:                P := specialize(P, r^s_{var^val'})
34:            end for
35:        end for
36:    end for
       // 4) Remember e and continue
37:    E' := E' ∪ e
38: end while
   // 5) Compute the likelihood of each rule
39: for each logic program P_i of P do
40:    for each rule r ∈ P_i do
41:        // r = var^val : p_r ← body(r)
42:        i := 0, j := 0
43:        for each e = (s, s') ∈ E' do
44:            if body(r) ⊆ s then
45:                j := j + 1
46:                if head(r) ∈ s' then
47:                    i := i + 1
48:                end if
49:            end if
50:        end for
           // Update the probability p_r of the rule r
51:        p_r = i/j
52:    end for
53: end for
54: return P
```

3.2 Candidate Planning Operator Generation

The input of the method is a set T of training transitions which are triples $t = (s, a, s')$ where s' is a successor state of s when the action a was executed. The state s' would have the effects of the action and all the applicable exogenous effects. The output is a set of planning operators $\mathcal{O}$ that define the model. Figure 2 shows the data transformations and processing done in the algorithm, which are described below:

- Transform grounded transitions to relational (non-grounded) ones. The objective is to learn relational operators that can take objects as arguments to generalize. Note that grounded literals are different from propositional ones because the objects and the variables in their terms can be identified. The transformation to relational transitions requires to substitute objects with variables.
- Obtain candidate operators. LFIT is used to obtain all possible candidate operators for a given set of transitions. The main advantage of using LFIT is that it obtains the set of minimal rules that model both action effects and exogenous effects. To use LFIT, first the relational transitions have to be transformed to propositional ones, and later, the output propositional rules to planning operators.
- Select the subset of candidate planning operators that best models the training transitions. This is detailed in Section 3.3.

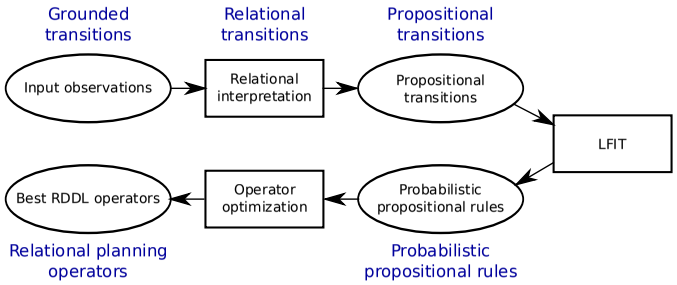


Figure 2: Data representation used for each module. The input and output data are shown in ellipses, and the processing modules are shown in boxes. The data representation used is indicated at each step.

The aim of these transformations is to generalize better, a relational representation is more compact and general as an infinite number of objects can be represented by each variable. By replacing objects with variables, LFIT learns rules that model relational data. The trade-off of learning relational generalizations is that the number of generated relational transitions is larger than the number of input grounded transitions, which increases the learning time.

3.2.1 Input Normalization

The first step is to normalize the input so that all state-action pairs (s, a) in the training data get the same number of transitions. Given a transition $t = (s_t, a_t, s'_t)$, the coverage of a pair (s, a) by t is defined as $cov(t, s, a) = (s_t = s) \wedge (a_t = a)$.

This step is specially important if the learner is integrated in a RL method, as state-action pairs with high expected values will be visited much often than those with low expected values. Without normalizing the input, when using heuristic learners, the selection of the best operators (Section 3.3) would be biased to perfectly model repeated state-action pairs and ignore rare ones.

The normalization is done by repeating transitions so that every (s, a) gets a similar number of covering transitions:

- *Input*: grounded transitions T.
- Get maximum coverage $n_{max} = \max_{(s,a)} |\{t \mid cov(t, s, a), t \in T\}|$.
- For each (s, a) covered by $T' \subset T$ (with $|T'| > 1$):
  - Number of times each transitions should be repeated: $n_{rep} = n_{max}/|T'|$.
  - Add n_{rep} times each $t' \in T'$ to T_{out}.
- *Output*: normalized transitions T_{out}.

Note that this input normalization is used exclusively to calculate the likelihood of the sets of operators (Sec. 3.3), and it is not used for the RL exploration nor the confidence in the operators. Moreover, it does not have an impact on performance if implemented correctly. Every transition should have a counter representing the number of times it is repeated to avoid extra computation.

3.2.2 Grounded to Relational Transitions

The goal is to obtain a relational representation that can generalize to different objects. If the dynamics of an object are learned, the same dynamics can be applied to other objects, not requiring examples of every possible grounding.

Since generating all possible relational combinations of every transition would be highly inefficient, we limit the number of relational variables to a fixed number ω, which imposes a limit on the maximum number of variables that learned operators will have. This type of constraint has been frequent in previous works to tackle complex domains (Mourão et al., 2012; Walsh et al., 2009; Amir and Chang, 2008).

Selecting the right value of ω is important. To learn effects that involve n objects, a value $\omega \geq n$ is required. However, the number of relational transitions scales exponentially with ω, thus a large value of ω is intractable.

This module generates all possible relational transitions with at most ω variables. For every transition $t \in T$, the following method is applied:

- *Input*: grounded transition $t = (s, a, s')$, max variables ω. We define the objects in s as $b_{s,i}$ and the objects in a as $b_{a,i}$. The action a has m objects.
- Initialize an empty set V_{obj} of object combinations.
- Obtain combinations of ω objects. For each combination of $\omega - m$ objects $(b_{s,1}, \ldots b_{s,\omega-m})$ that are not in the action $(b_{s,i} \neq b_{a,j}, \forall i, j)$ do:
  - Create $v_{obj} = (b_{a,1}, \ldots b_{a,m}, b_{s,1}, \ldots b_{s,\omega-m})$ where $(b_{a,1}, \ldots b_{a,m})$ are the objects in the action a.
  - Add v_{obj} to V_{obj}.
- For each $v_{obj} \in V_{obj}$:
  - A new transition $t' = t$ is created.
  - Replace in t' all objects in v_{obj} for variables.
  - Remove from t' any remaining literal with objects.
  - Add the new transition t' to T'.
- *Output*: a set of relational transitions T'.

Example 6 *Given a grounded transition* $(s, a \rightarrow s')$ *:*

$$at(r1) \wedge road(r2, r3) \wedge road(r1, r3), move(r3) \rightarrow$$
$$road(r1, r3) \wedge road(r2, r3) \wedge at(r3),$$

the following relational transitions are generated (ω=2):

$$v_{obj} = (r3, r1): \; at(?Y) \wedge road(?Y, ?X), move(?X) \rightarrow road(?Y, ?X) \wedge at(?X);$$
$$v_{obj} = (r3, r2): \; road(?Y, ?X), move(?X) \rightarrow road(?Y, ?X) \wedge at(?X).$$

3.2.3 Relational to Propositional Transitions

To create the input that LFIT requires, which are pairs (s, s') of propositional states, a library L_{conv} that converts between relational and propositional atoms is created. For each relational atom d_r, a new propositional atom d_p is created and the pair (d_r, d_p) is added to L_{conv}. Using L_{conv}, everything is substituted by its propositional counterpart:

- Relational literals are represented with propositional atoms that take the values 1 (true) or 0 (false).
- Relational transitions are triples (s, a, s'), while propositional transitions are pairs (s, s'). Therefore, an additional multi-valued atom is added to propositional transitions to represent the action (if the agent executed an action during that transition). This atom takes as value the corresponding action name in L_{conv}, or "*noaction*" if there is no action. In the input, the "*noaction*" value will only be used in those transitions where the robot did not execute any action. In the output, it identifies exogenous effects. Note that literals are binary, so the multi-valued representation is only required to model the action, as there may be more than 2 different actions, and only one action can be executed per transition.

Example 7 *Using the relational transitions in example 6, the following library is created* $L_{conv} = \{(at(?Y), b1), (at(?X), b2), (road(?Y, ?X), b3), (move(?X), b4)\}$. *The propositional transitions obtained by using* L_{conv} *are:*

$$(b1{=}1) \wedge (b3{=}1) \wedge (action{=}b4) \rightarrow (b3{=}1) \wedge (b2{=}1);$$
$$(b3{=}1) \wedge (action{=}b4) \rightarrow (b3{=}1) \wedge (b2{=}1).$$

3.2.4 Generation of Rules

The LFIT framework is used to obtain the set of probabilistic candidate rules from a set of proposition transitions (s, s'). LFIT induces a set of *normal logic programs* that realize the given transitions. Our approach uses the specialization and probabilistic algorithm of LFIT, so it learns the sets of minimal probabilistic rules that model all effects appearing in the input transitions. The learned sets are combined into a single set before converting them to planning operators. LFIT learns both action effects and exogenous effects because the action is just another atom that may or may not appear in the body of a rule.

Example 8 *Using a larger set of propositional transitions such as the ones obtained previously, LFIT could obtain a set of rules such as:*

$$(b2{=}1) : 0.8 \leftarrow (b1{=}1) \wedge (b3{=}1) \wedge (action{=}b4);$$
$$(b2{=}1) : 0.1 \leftarrow (b1{=}1) \wedge (b3{=}0) \wedge (action{=}b4);$$
$$(b1{=}1) : 0.6 \leftarrow (b2{=}1);$$
$$\ldots$$

3.2.5 Propositional Rules to Planning Operators

Planning operators (eq. 1) can be reconstructed from probabilistic rules (eq. 4) by using the library L_{conv} created before. For each propositional rule:

- The atoms in the body and head of the rule are transformed to relational ones (using L_{conv}), and added to the body and head of a planning operator.
- The action is extracted from the multi-valued action atom in the rule body. If the atom is present, the corresponding action in L_{conv} is added to the operator.

Example 9 *Given that LFIT had learned the following rule*

$$(b2{=}1) : 0.8 \leftarrow (b1{=}1) \wedge (b3{=}1) \wedge (action{=}b4),$$

using the library L_{conv} *generated in example 7, the resulting operator* $o(?X, ?Y)$ *is*

$$at(?X) : 0.8 \leftarrow at(?Y) \wedge road(?Y, ?X), move(?X).$$

Traditionally, PPDDL (Younes and Littman, 2004) has been the standard language to model probabilistic domains, but it cannot model exogenous effects directly (PPDDL1.0 rules model actions, not exogenous effects). Therefore, our approach uses the RDDL language (Sanner, 2010), which has been the standard for the latest probabilistic planning competitions (IPPC 2011 and 2014). Writing our planning operators with RDDL is straightforward, and this language can be used directly by state-of-the art planners. RDDL objects and variables have types, and a variable can only be substituted by an object of the same type. However, for clarity and simplicity, we assume through the paper that there are no types, as adding them is trivial.

3.3 Planning Operator Selection

LFIT provides the set of minimal rules (that have been transformed to planning operators) that describe all the transitions. Note that LFIT learns the set of minimal rules, and not the minimal set of rules, so many operators may model the same changes and underfit or overfit. The subset of operators to model the transition dynamics is selected as follows:

- A score function is defined to evaluate the planning operators.
- A heuristic search algorithm selects the set of operators that maximizes the score. Note that this set may differ from the actual model, as it depends on the coverage of the input transitions and the quality of the score function.
- The subsumption tree is used to improve efficiency by partitioning the set of candidates into smaller subsets.

3.3.1 Score Function

The score function values the quality of a set of operators. The following functions are used by the score function:

- The likelihood is the probability that a transition $t = (s, a, s')$ is covered by a set of planning operators $\mathcal{O}$:

$$P(t \mid \mathcal{O}) = \prod_{c \in \text{changes}(t)} P(c \mid \mathcal{O}, s, a). \tag{5}$$

- The penalty term $Pen(o) = |body(o)|$ is the number of atoms in the operator bodies.
- The confidence $Conf(T, \widehat{o}, \epsilon)$ is obtained from (Hoeffding, 1963)'s inequality. The probability that an estimate $\widehat{o_{prob}}$ is accurate enough, i.e. $|\widehat{o_{prob}} - o_{prob}| \leq \epsilon$, is bounded by $Conf(T, \widehat{o}, \epsilon) \leq 1 - e^{-2\epsilon^2 |T_{\widehat{o}}|}$, where $|T_{\widehat{o}}|$ is the number transitions covered by $\widehat{o}$.

Finally, the proposed score function is defined as

$$s(\mathcal{O}, T) = \underset{t \in T}{E}[\log(P(t|\mathcal{O}))] - \alpha \sum_{o \in \mathcal{O}} \frac{Pen(o)}{Conf(T, o, \epsilon)}, \tag{6}$$

where $\alpha > 0$ is a scaling parameter for the penalty term. This score function is based on Pasula et al. (2007)'s one, where the likelihood is maximized to obtain operators that explain the transitions well, and the penalty term is minimized to prefer general operators when specific ones have very limited contributions. In contrast to Pasula et al.'s approach, the confidence term is added so that the penalty is increased when few transitions are available, as the estimates are less reliable.

3.3.2 Heuristic Search

Given a set of operators $\mathcal{O}$ with the same head, a heuristic search method is used to find the best subset of operators that maximizes the score function. To that end, we define the heuristic version of the change likelihood (eq. 3) as:

$$P_h(c|\mathcal{O}) = \begin{cases} P(c|o_g), & \exists! o_g \in Gr(\mathcal{O}) \,|P(c|o_g) > 0 \\ 1 - \delta, & \nexists\, o_g \in Gr(\mathcal{O}) \,|cov(o_g, s, a) \\ 0, & \text{otherwise } (|Gr(\mathcal{O})| > 1), \end{cases} \tag{7}$$

where δ is a parameter that can trade quality for efficiency. This heuristic modifies the change likelihood (eq. 3) when no operator covers the change, giving a likelihood of $1 - \delta$ instead of 0. The heuristic score function $s_h(\mathcal{O}, T)$ is defined as the score function (eq. 6) but replacing the standard change likelihood (eq. 3) with this heuristic likelihood.

This heuristic gets the expected likelihood that can be obtained by adding new operators to $\mathcal{O}$. When $\delta = 0$, it works as an admissible heuristic (prop. 1) as it gives the maximum likelihood = 1 to uncovered changes. When $\delta > 0$ but close to 0, then the heuristic penalizes very specific operators when more general operators with a high likelihood are also

available. The practical result is that the algorithm usually runs faster, but the heuristic is not admissible anymore.

Algorithm 2 selects the best subset of operators to explain the input transitions. In line 1, the candidate list Γ is initialized by creating one separate subset for each operator in the input set of candidates $\mathcal{O}_{input}$. Note that Γ is a set of sets of planning operators, which is initialized to $\Gamma = \{\{o_1\}, \ldots, \{o_n\}\}$ assuming that $\mathcal{O}_{input} = \{o_1, \ldots, o_n\}$. Afterwards, lines 2-3 find the best subset in Γ (which is the best set with only one operator). From that point, the candidate sets in Γ are iteratively joined together to find the best set with any number of operators, until none has $s_h(\mathcal{O}, T) > max_{score}$ (lines 4-17). In lines 5-6, the candidate $\mathcal{O}$ with the largest heuristic score is selected and removed from Γ. Then, in lines 7-9, new candidates are generated by combining the selected subset $\mathcal{O}$ with every subset in Γ. The *IsNew* method checks that the new candidate has not been already analyzed. If any of the new candidates has a new best score, it is saved as the best candidate (lines 10-13). Finally, the new candidates are added to Γ.

This method works as a search algorithm guided by an heuristic. The nodes to be analyzed are the subsets of operators stored in Γ, where they are ordered by the heuristic score value. The search tree is expanded by joining one subset with every other subset. Finally, the algorithm continues until no subset has a heuristic score larger than the best score so that the solution has been found.

The search can be used as an *anytime* algorithm, it can be stopped at any point to get the best solution found so far. Moreover, there are two options to limit in advance the processing time of the algorithm in complex problems: set a time limit, or set a limit in the size of Γ (only maintain the κ sets with the highest score in Γ). Experimental tests showed that in some domains the heuristic leads quickly to the best set, and subsequent processing is done only to confirm that no other set is better.

Property 1 *If $\delta = 0$, then the heuristic s_h is admissible: $\forall \mathcal{O}, s_h(\mathcal{O}) \geq s(\mathcal{O}^*) \mid \mathcal{O}^* = \underset{\mathcal{O}' \supset \mathcal{O}}{\text{argmax}}\, s(\mathcal{O}')$. Therefore the optimal set will be found.*

Proof The two parts of the score function in eq. 6 (likelihood and regularization) can be analyzed separately. Note that subsets of operators may only increase in size, as they start with one operator and can only be combined with other subsets.

- When adding operators to a set, the likelihood only increases when transition changes that were not covered before are covered by the added operators. In the heuristic score (eq. 7) all non-covered transition changes are already set to the maximum value of 1, so adding new operators cannot improve the result over the heuristic.
- The regularization part of the score function ($Reg(\mathcal{O}) = -\alpha \sum_{o \in \mathcal{O}} \frac{Pen(o)}{Conf(T,o,\epsilon)}$) is monotonically decreasing with respect to adding operators. If new operators are added, the new penalty $\frac{Pen(o)}{Conf(T,o,\epsilon)}$ will be positive as $Conf \in (0, 1]$, $Pen \geq 0$ and $\alpha > 0$.

■

Algorithm 2 OperatorSelection($\mathcal{O}_{input}$, T)

Current candidates $\Gamma \leftarrow \{o\}, \forall o \in \mathcal{O}_{input}$
$max_{score} = \max_{\mathcal{O}\in\Gamma} s(\mathcal{O}, T)$
$\mathcal{O}_{best} = \operatorname{argmax}_{\mathcal{O}\in\Gamma} s(\mathcal{O}, T)$
while $\max_{\mathcal{O}\in\Gamma} s_h(\mathcal{O}, T) > max_{score}$ **do**
  $\mathcal{O} = \operatorname{argmax}_{\mathcal{O}\in\Gamma} s_h(\mathcal{O}, T)$
  Remove $\mathcal{O}$ from Γ
  for $\mathcal{O}' \in \Gamma$ **do**
    if IsNew($\mathcal{O} \cup \mathcal{O}'$) **then**
      $\mathcal{O}_{new} = \mathcal{O} \cup \mathcal{O}'$
      if $s(\mathcal{O}_{new}, T) > max_{score}$ **then**
        $max_{score} = s(\mathcal{O}_{new}, T)$
        $\mathcal{O}_{best} = \mathcal{O}_{new}$
      end if
      Add $\mathcal{O}_{new}$ to Γ
    end if
  end for
end while
Output $\mathcal{O}_{best}$

Property 2 *When relaxing the admissibility criterion with $\delta > 0$, the solution found by Algorithm 2 is bounded to be no worse than $\mathcal{C} \cdot \log(1-\delta)$ plus the optimal score, where $\mathcal{C}$ is the average number of literals with the same predicate that change in a transition.*

Proof Operators with different head predicates are analyzed separately as their effects are independent, so $\mathcal{C}$ only depends on the average changes to one predicate literals. Also note that $\delta \in [0, 1)$, and thus, $\log(1-\delta) \leq 0$. Let $\mathcal{O}_n$ be the optimal solution with n operators and $s(\mathcal{O}_n) = opt$, then $\forall i < n$, at least one predecessor of i operators $\mathcal{O}_i$ and $s_h(\mathcal{O}_i) \geq opt + \mathcal{C} \cdot \log(1-\delta)$ will exist.

- The regularization term is monotonically decreasing (see explanation of property 1), so $Reg(\mathcal{O}_n) \leq Reg(\mathcal{O}_i)$.
- The maximum difference between $P(\mathcal{O}_n)$ and $P_h(\mathcal{O}_i)$ is $P(t|\mathcal{O}_n) = 1$ and $P_h(t|\mathcal{O}_i) = (1-\delta)^x$, which is the case where $\mathcal{O}_n$ has perfect coverage, $\mathcal{O}_i$ has no coverage (all the coverage is obtained from $\mathcal{O}_n \setminus \mathcal{O}_i$) and the transition has x changes. Then, if we take the worst case for all transitions, and an average of $\mathcal{C}$ changes per transition, $P(\mathcal{O}_n) - P_h(\mathcal{O}_i) = E[\log(P(T|\mathcal{O}_n))] - E[\log(P_h(T|\mathcal{O}_i))] = \log(1) - \log(1-\delta)^{\mathcal{C}} = \mathcal{C} \cdot \log(1-\delta)$.

Therefore, the predecessors of the optimal solution are checked (and thus the optimal solution found) unless a solution with $s(\mathcal{O}) \geq opt + \mathcal{C} \cdot log(1-\delta)$ is found before. ■

Algorithm 3 OperatorSelectionSubsumption($Tree_{\mathcal{O}}, T$)

1: **do**
2: $\quad\mathcal{O}_L = leaves(Tree_{\mathcal{O}})$
3: $\quad\mathcal{O}_{L,best} =$ OperatorSelection$(\mathcal{O}_L, T)$
4: $\quad$Remove $(\mathcal{O}_L \setminus \mathcal{O}_{L,best})$ from $Tree_{\mathcal{O}}$
5: $\quad\mathcal{O}_P = \mathcal{O}_{L,best} \cup parents(Tree_{\mathcal{O}}, \mathcal{O}_{L,best})$
6: $\quad\mathcal{O}_{P,best} =$ OperatorSelection$(\mathcal{O}_P, T)$
7: $\quad$Remove $(\mathcal{O}_L \setminus \mathcal{O}_{P,best})$ from $Tree_{\mathcal{O}}$
8: **while** $Tree_{\mathcal{O}}$ changed
9: **Output** $= leaves(Tree_{\mathcal{O}})$

A value of $\delta > 0$ can speed up the algorithm considerably at the expense of optimality. When two operators have similar likelihoods, $\delta > 0$ penalizes the most specific one. This results in general operator sets being analyzed first, and thus models with better likelihoods are obtained earlier.

3.3.3 Subsumption Tree

In this section we present a method to speed up the approach by partitioning the set of candidates into smaller groups. The idea is to organize the operators in a tree using the OI-subsumption (Esposito et al., 2004), then start with the specialized operators, and iteratively check if more general operators yield better scores.

Definition 5 (OI-subsumption relation) *First, we define the extended body of an operator as $body_{ext}(o) = body(o)(\wedge a)$ where $(\wedge a)$ only appears if o has an action. Let o_1 and o_2 be two planning operators with $head(o_1) = head(o_2)$, o_1 is OI-subsumed by o_2 if $(body_{ext}(o_1) \supseteq body_{ext}(o_2))$, given that different terms t_i cannot take the same object.*

Definition 6 (OI-subsumption tree) *The OI-subsumption tree $Tree_{\mathcal{O}}$ of a set of planning operators $\mathcal{O} = \{o_1, ..., o_n\}$ is a directed graph with arcs (o_i, o_j) when o_i OI-subsumes o_j and $|body_{ext}(o_j)| - |body_{ext}(o_i)| = 1$. We call the set of leaves $L(Tree_{\mathcal{O}})$. Figure 3 shows an example of a OI-subsumption tree.*

The OI-subsumption tree orders the operators in levels that represent the generality of the operators: the less literals the more general the operator. Based on this tree, Algorithm 3 selects the operators. The idea is to start by identifying the best specific operators, and then check if their generalizations improve the results. In lines 2-4, the best subset of leaves $\mathcal{O}_{L,best}$ is identified, and all leaves not in $\mathcal{O}_{L,best}$ are removed from the tree. Then, in lines 5-7, a new set of operators $\mathcal{O}_P$ is created that includes $\mathcal{O}_{L,best}$ and the operators that OI-subsume them (their parents in the tree). The best subset $\mathcal{O}_{P,best}$ in $\mathcal{O}_P$ is selected, and $\mathcal{O}_L \setminus \mathcal{O}_{P,best}$ are removed. This is repeated until nothing is changed in the tree.

The performance is improved by using the OI-subsumption tree: it divides the candidates into subsets, and the planning operator selection is much faster with smaller sets of candidates. Although it sacrifices optimality, experimental tests showed that the results obtained were in many cases optimal or near optimal. This happens due to the fact that

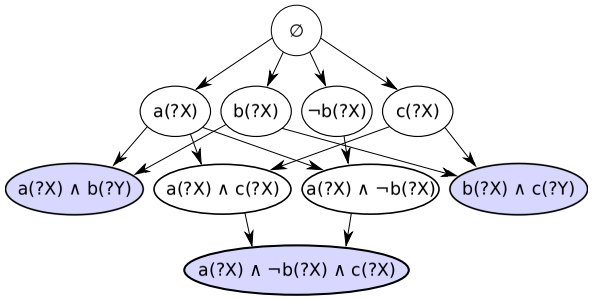


Figure 3: Example of a OI-subsumption tree. Each letter (a, b, c) represents a literal in the extended body of a planning operator. The leaves are the nodes painted in blue.

in most cases $P(\mathcal{O}) >> Reg(\mathcal{O})$. The operators in the leaves maximize $P(\mathcal{O})$ as they are more specialized, while the operators near the root maximize $Reg(\mathcal{O})$ as they are more general. Thus, the subset of leaves selected in the first iteration usually is near optimal, and afterwards the method only has to find the right level of generalization.

Note that if the OI-subsumption tree is used, the best operators may be close to the root of the tree (and thus they will be analyzed at the end), so the learner shouldn't be used as an anytime algorithm. However, the processing time can still be bounded with satisfactory results by limiting the size of Γ to κ sets in Algorithm 2.

4. Integration with RL

In this section we present how to integrate the model learner with a RL approach. The objective is to allow an agent to learn and solve a task from scratch by using a task planner and the learner presented before.

Model-based RL has proven to be an excellent tool to learn and solve a task that is unknown initially (Littman, 2015). There are several RL approaches that can learn complete relational models for planning (Džeroski et al., 2001; Diuk et al., 2008; Walsh et al., 2009), but they require a large amount of samples to make sure that they can solve the problem (Walsh, 2010). Other approaches use heuristics to reduce the number of samples needed to make the problem tractable, such as TEXPLORE (Hester and Stone, 2013) and REX (Lang et al., 2012). Specifically, REX uses Pasula et al. (2007)'s model learner internally, and proposes an exploration method based on R-MAX (Brafman and Tennenholtz, 2003) and E^3 (Kearns and Singh, 2002) that takes advantage of the relational representation to generalize and reduce the exploration required to learn relational tasks. However, REX can obtain models that are a local minimum as they cut exploration.

Our objective is to learn models without requiring too many input transitions. Therefore we choose to use the V-MIN algorithm (Martínez, Alenyà, and Torras, 2015b) as the RL framework. V-MIN extends the REX (Lang et al., 2012) with active learning: it explores

to learn the basic dynamics of the actions, but when it cannot find a plan with value larger than a certain threshold V_{min}, it requests help from a teacher. In contrast to REX, V-MIN can ensure that it will solve the task with a value larger than V_{min} (given that the internal learner can obtain the model for the task). Moreover, it also performs well with scarce exploration, because even if it misses the exploration of an important state-action pair, V-MIN will visit it later (if it is needed to get V_{min}) by requesting demonstrations.

Many RL approaches focus on learning specific tasks, but have trouble when parts of the problem change. Model-based approaches have been proposed to learn models that generalize between different states and number of objects (Diuk et al., 2008), while others have applied transfer learning in combination with RL to reuse knowledge in similar tasks (Konidaris et al., 2012). The REX and V-MIN algorithms learn relational operators that can be used by standard planners, which provides a great flexibility in the changes that can be made to the task without having to relearn or adapt the model. Using Taylor and Stone (2009)'s transfer learning categorization, the following changes can be done:

- The initial state, the goal state, and the number of objects can be changed without requiring further learning.
- The reward function can change without having to modify the rest of the model.

V-MIN in addition can incorporate new actions easily to the model through demonstrations if they are needed to get a value $> V_{min}$. This results in an approach with good adaptation capabilities when compared to other works (Taylor and Stone, 2009), and it also has the advantage of learning models with less exploratory actions. The drawback is that a teacher is required.

The inclusion of a teacher in the learning loop has been also tackled by other approaches. In some methods the teacher issues corrective demonstrations when the robot doesn't perform as desired (Meriçli et al., 2012; Walsh et al., 2010), which implies that the teacher has to actively monitor the agent. The teacher has also been used to provide approval or disapproval of the agent performance (Knox and Stone, 2012). When using V-MIN, the agent is the one that actively requests help whenever it is needed, and thus the teacher is free from monitoring the agent. Other approaches also require help directly from the agent (Grollman and Jenkins, 2007; Chernova and Veloso, 2009), but they only learn from demonstrations, so they have to request a larger number of demonstrations to learn. Agostini et al. (2017)'s approach requests demonstrations from the teacher when the planner cannot find a solution with its current set of planning operators. However, this approach only works in goal-driven deterministic problems, and the interaction with the teacher lacks flexibility as it does not consider rewards. Finally, Walsh et al. (2011) proposed a similar approach to V-MIN where the agent communicates the expected value to the teacher, and if it is lower than the optimal value then the teacher demonstrates an action. This approach can learn STRIPS rules efficiently, but it would require many input experiences to learn exogenous effects in the same way because the number of possible rules would be much larger.

4.1 The V-MIN Algorithm

The REX algorithm (Lang et al., 2012) is devised to apply relational generalizations to R-MAX, reducing the exploration needed. State-action pairs are grouped in contexts that have the same dynamics (i.e. that are covered by the same relational planning operator). Based on these contexts, exploration is reduced by defining a context-based density function that considers all possible state-action pairs in a context to be equivalent when deciding if a state is known. Although exploration generalizations improve the learning performance, they also have drawbacks. Using a context-based count function implies that not all states are explored before considering them as known, as we assume that all states within a context behave likewise. Thus, state-action pairs needed to attain the best policy may not be visited and their contexts (i.e. planning operators) not learned.

The V-MIN algorithm requests teacher demonstrations to learn new actions and contexts, and improve performance. It uses the concept of V_{min}, which is the minimum expected value. If the planner cannot obtain a value $V^{\pi}(s) \geq V_{min}$, it actively requests help from a teacher, who will demonstrate the best action $a = \pi^*(s_i)$ for the current state s_i. Therefore, when exploration and exploitation can no longer yield the desired results, demonstrations are requested to learn yet unknown actions, or to obtain demonstrations of actions whose effects were unexpected (i.e. new unknown contexts). The result is that good models can be learned even if exploration is scarce. The V_{min} parameter provides flexibility to the system:

- A high V_{min} forces the system to learn good policies at the cost of a higher number of demonstrations, whereas a lower V_{min} leads to a faster and easier learning process, but worse policies are learned.
- The teacher can change V_{min} online until the system performs as desired, forcing it to find either better or easier policies.
- Demonstrations can be used to learn new actions that weren't required before.

4.1.1 Algorithm Details

The V-MIN algorithm uses a model M that consists of a set of planning operators $\mathcal{O}$, which define the contexts. The algorithm creates an extension M_{vmin} of the model M to implicitly plan exploration and demonstration requests. As in the R-MAX version of the REX algorithm, every unknown context (those that have been explored less than a threshold ζ) gets the maximum reward R_{max}, implicitly following the principle of "optimism under uncertainty". Moreover, a special action "TeacherRequest()" is added which leads to an absorbing state with a V_{min} value. "TeacherRequest()" is also limited to be the only action in a plan when selected, as otherwise a value of $V(s')+V_{min}$ could be obtained by combining it in a plan.

Algorithm 4 shows a pseudocode for V-MIN. This algorithm maintains a model M that represents the robot actions with a set of relational planning operators. The model is updated in every iteration (line 3) to adapt to newer experiences using the learner presented in Section 3. The M_{vmin} model is then created to plan the action to execute, implicitly selecting exploratory actions until all the relevant contexts become known, and requesting teacher demonstrations whenever a value $V^{\pi}(s) > V_{min}$ cannot be obtained. Finally a new state is observed, and a transition is added to the log of experiences.

Algorithm 4 V-MIN

Input: Reward function R, confidence threshold ζ
Updates: Set of input transitions T

```
1:  Observe state s_0
2:  loop
3:      Update transition model M according to T
4:      Create M_vmin(M, R, ζ)
5:      Plan an action a_t using M_vmin
6:      if a_t == "TeacherRequest()" then
7:          Request demonstration
8:          a_t = demonstrated action
9:      else
10:         Execute a_t
11:     end if
12:     Observe new state s_{t+1}
13:     Add {(s_t, a_t, s_{t+1})} to E
14: end loop
```

The V-MIN algorithm provides the framework that selects which actions to explore and when a demonstration should be requested, but the most important piece is the learner used to obtain the model. Using the learner presented in Section 3 implies that a probabilistic relational model with exogenous effects will be learned.

Another important piece is the planner used, as it has to support planning with the learned model. PROST (Keller and Eyerich, 2012) is the planner used as it can obtain good results with probabilistic models containing exogenous effects.

Note that in this work we consider the reward function to be known. A KWIK algorithm can learn a reward function with a polynomial number of samples (Li et al., 2011), but it remains as future work to show how to effectively combine that reward learner with the V-MIN algorithm.

5. Experiments

This section describes the experimental evaluation of our approach. Three domains of the 2014 International Probabilistic Planning Competition (IPPC) (Vallati et al., 2015) were used in the experiments. The standalone learner introduced in Section 3 was used to learn from a batch of given input transitions, while the RL approach presented in Section 4 was used to solve tasks.

5.1 Domains

Three IPPC 2014 domains were used in the experiments. Note that they were slightly modified to remove redundancy (e.g. a *north(?X,?Y)* literal is equivalent to *south(?Y,?X)*, so one can be replaced by the other).

- *Triangle Tireworld.* This domain is the easiest one, it has uncertain effects, but no exogenous effects. It is modeled with 5 different predicates, 3 actions, 7 operators, and operators require at most 2 terms ($\omega = 2$). This domain serves as a good baseline to compare with the state of the art as there are not exogenous effects.

- *Crossing Traffic.* This domain has an intermediate difficulty. It has uncertain effects and exogenous effects, which makes it more challenging, but the complexity of the model is still moderate: 8 predicates, 4 actions, 6 operators, and operators take at most 3 terms.

- *Elevators.* This is the most challenging domain. It has uncertain effects and exogenous effects. It is modeled with 10 predicates, 4 actions, 17 operators, and operators take at most 3 terms.

5.2 Evaluation of the Model Learner

To evaluate the learner presented in Section 3 the scheme used by Pasula et al. (2007) was followed. The learners had to obtain models from sets of input transitions $(s, a, s') \in T$ that were generated randomly. To create a transition, first the state s is constructed by randomly assigning a value (positive or negative) to every literal, but ensuring that the resulting state is valid (e.g. in the elevators domain, an elevator cannot be in two different floors at the same time). Then, the action a arguments are picked randomly, and the state s' is obtained by applying all operators to (s, a). The distribution used to construct s is biased to guarantee that, in at least half of the examples, the operators that contain a have a chance of changing the state.

The evaluation of the learned models is carried out by calculating the *average variational distance* between the true model $\mathcal{O}$ and the estimate $\hat{\mathcal{O}}$. This evaluation uses a new set of similarly generated random transitions T':

$$D(\mathcal{O}, \hat{\mathcal{O}}) = \frac{1}{|T'|} \sum_{t \in T'} \left| P(t|\mathcal{O}) - P(t|\hat{\mathcal{O}}) \right|. \tag{8}$$

As the *average variational distance* may be difficult to interpret, here we give an intuition about the utility of the learned models. A planner usually ($p > 0.9$) yields a plan that can solve the task (optimally or suboptimally) when the *average variational distance* is below:

- 0.09 in the *Triangle Tireworld* domain.

- 0.15 in the *Crossing Traffic* domain.

- 0.1 in the *Elevators* domain.

As the *average variational distance* becomes lower, the planner will obtain better solutions.

We analyze experimentally the proposed algorithm, its parameters, and how it compares with the state of the art. The learner uses the following parameters: α, ϵ, and δ, which are the score function parameters; κ, which is the size limit of Γ (Algorithm 2); and "*tree*" to denote that the subsumption tree is being used.

The difficulty to learn a domain is mostly given by:

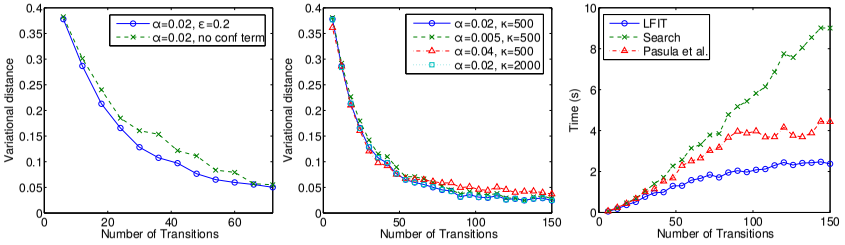


Figure 4: Evaluation of different configurations in the Triangle Tireworld domain, MDP-2. Unless stated otherwise, the following parameters were used: ($\alpha = 0.02$, $\epsilon = 0.2$, $\omega = 2$, $\delta = 0$, $\kappa = 500$, no-tree). The results shown are the means obtained from 50 runs. The evaluation was done with 3000 transitions. **Left:** Influence of the confidence term. **Center:** Comparing different values of α and κ. **Right:** Execution time of the search (optimization) and the LFIT modules. The total learning time would be the sum of LFIT plus one of the search configurations.

- The maximum number of terms ω that operators may have. The number of terms increases exponentially the number of relational transitions generated from the input grounded transitions (Section 3.2.2), and therefore the number of candidate rules. If the value of ω is larger than the number of terms that operators actually require, the learning time increases while the quality of the models remains the same.

- The number of predicates, both constant and variable, used to represent the states. The candidates that LFIT generates consider all combinations of predicates that are consistent with the transitions.

- The number of uncertain and exogenous effects. LFIT generates all candidates that may explain an effect, including operators that overfit and underfit, and all combinations of action effects and exogenous effects.

5.2.1 Configuration Parameters

Here we discuss the impact of the different configuration parameters on the quality of the models learned.

Triangle Tireworld domain:

- As seen in Fig. 4-left, the confidence term in the score function improves the quality of the models learned when few input transitions are available. This confidence term penalizes very specialized operators in cases where there is a large uncertainty in the predictions. Once many transitions are available, the impact of this term disappears. Note that only probabilistic operators are improved as deterministic ones are completely specialized anyway. The impact of this term is relatively small in the experiments because it only affects 1 probabilistic operator.

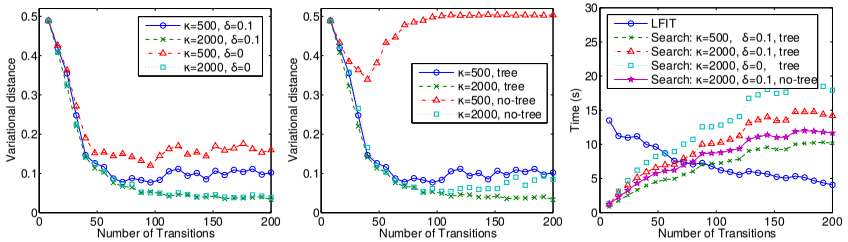


Figure 5: Evaluation of different configurations in the Crossing Traffic domain, MDP-1. Unless stated otherwise, the following parameters were used: ($\alpha = 0.02$, $\epsilon = 0.2$, $\omega = 3$, $\delta = 0.1$, $\kappa = 500$, tree). The results shown are the means obtained from 50 runs. The evaluation was done with 4000 transitions. **Left:** Influence of the κ and the δ terms. **Center:** Influence of κ and the subsumption tree. **Right:** Execution time of the search (optimization) and the LFIT modules. The total learning time would be the sum of LFIT plus one of the search configurations.

- Figure 4-middle compares different values of α. Small changes of the α term produce almost the same results, and any value ~ 0.02 yielded good results in all the domains analyzed in this paper. Moreover, as the Triangle Tireworld domain is simple, a value of $\kappa = 500$ already produces good models, and increasing it does not improve the results.
- The execution time is shown in Fig. 4-right. Even with 150 input transitions only a few seconds are required to learn a model. Most of the execution time is spent in obtaining the likelihood of the operators as there are a lot of possible candidates in this domain.

Crossing Traffic domain:

- As can be seen in all the experiments, the number of operator candidates that the algorithm analyzes (κ) has a great influence on both the quality of the learned domain and the learning time. The best value of κ will depend on the quality required for the domains and the time available for learning.
- Figure 5-left shows the advantages of a non-admissible heuristic ($\delta > 0$). A small value of δ prioritizes operators that explain many transitions with a high likelihood, and very specific operators that may not be interesting are given a smaller heuristic score. Although the same results could be obtained with a larger value of κ, the non-admissible heuristic speeds up the search (Fig. 5-right).
- Figure 5-center shows the advantages of the subsumption tree. This tree divides the search problem in smaller ones, and thus a much lower κ is enough to learn good models, specially when the number of input transitions is large. Since to get the same model quality a much larger κ would be required, the learning time is reduced.

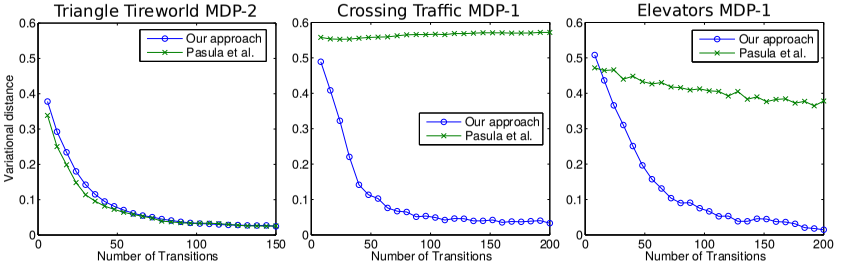


Figure 6: Comparison with Pasula et al. The results shown are the means obtained from 50 runs. The evaluation was done with 5000 random transitions. The parameters used were $\alpha = 0.02$, $\epsilon = 0.2$, $\delta = 0.1$, tree. **Left:** Triangle Tireworld ($\kappa = 500$). **Center:** Crossing Traffic ($\kappa = 2000$). **Right:** Elevators ($\kappa = 2000$).

5.2.2 Comparison with Pasula et al.

Comparison is performed only with Pasula et al. (2007)'s learner, as Deshpande et al. (2007)'s learner is an extension to include transfer learning, and Mourão (2014)'s learner yields similar results to Pasula et al.'s approach in completely observable problems. The experiments were carried out with the implementation by Lang and Toussaint (2010) and the fix by Mourão (2014).

Figure 6-left shows the results of learning the *Triangle Tireworld* domain. It can be easily learned by both, ours, and Pasula et al.'s approach, as there are no exogenous effects. With this experiment we want to show that our method can learn domains without exogenous effects as well as state-of-the-art learners. Pasula et al.'s learner gets slightly better results with few transitions as one of the actions is easier to learn with correlated effects.

Figures 6-center and 6-right refer to two domains with exogenous effects. As Pasula et al.'s learner cannot learn exogenous effects, it tries to build overcomplicated operators. These operators try to explain every possible combination of action effects and exogenous effects at the same time instead of learning each effect separately, and thus it is not able to yield a good general model. In contrast, our approach is able to distinguish action effects from exogenous effects once enough transitions are given as input.

5.2.3 Limitations of the Learner

The main limitation of the algorithm is the scalability. If the number of generated propositional predicates is large (either because the domain is represented with a large number of predicates, or because ω takes a high value), the problem may become intractable. The *Elevators* domain has the highest complexity that our approach can learn within a reasonable time. With 100 input transitions, the planning operator selection took an average of 2 minutes, and LFIT took up to 2 hours.

Moreover, in comparison with previous approaches, our learner cannot model correlated effects so it is not able to learn domains such as slippery gripper (Pasula et al., 2007). Our learner considers all effects as candidates, and thus, it would have to consider all possible

combinations of effects to learn correlated effects, which would be intractable. In contrast, approaches that do not learn exogenous effects can afford to find correlated effects because they search for a single effect per transition.

Finally, many IPPC 2014 domains cannot be directly learned. The following features would be needed to support all domains:

- The universal quantifier (*forall*) and the negative existential quantifier ($\sim$ *exists*) are not supported. When doing the translation from propositional rules to relational planning operators, new candidates could be generated that considered a positive atom in the preconditions as a (*forall*), or a negative atom as a ($\sim$ *exists*). It remains as future work to analyze the effectiveness of such candidates, and to check if the operator selection process would still be efficient with the addition of new candidates.

- Some IPPC domains have operators whose probabilities are encoded as a function. Unfortunately, our approach can only consider real numbers as probabilities.

5.3 Evaluation of the RL Integration

Here we evaluate how the learner can be integrated in a RL approach to learn models from scratch as explained in Section 4. The learner proposed by (Pasula et al., 2007) was also integrated in the V-MIN approach to compare against our approach.

5.3.1 Triangle Tireworld

In this domain, a car has to move to its destination, but it has a probability of getting a flat tire while it moves. The car starts with no spare tires but it can pick them up in some locations. The actions available in this domain are: a "Move" action to go to an adjacent position, a "Change Tire" action to replace a flat tire with a spare tire, and a "Load Tire" action to load a spare tire into the car if there are any in the current location. The main difficulty in the *Triangle Tireworld* domain is the dead end when the agent gets a flat tire and no spare tires are available. Safe and long paths exist with spare tires, but the shortest paths do not have any spare tires.

It should be noted that in the IPPC 2014 representation of the *Triangle Tireworld* domain there is one exogenous effect: when the goal reward is received, the "goal-reward-received" literal becomes true. As this is modeled as an exogenous effect and Pasula et al.'s approach cannot learn it, that planning operator is given to it at the beginning. Our approach can learn this effect easily because it appears isolated after the goal is reached, so it doesn't have a significant impact on the performance.

Figure 7 shows the results of learning the *Triangle Tireworld* domain. Initially the robot chooses the shortest path and thus 50% of the times gets an irrecoverable flat tire. The learners have to learn that the robot cannot move with a flat tire, and that a spare tire is required to recover from a flat tire before the planner opts to go through the safe but longer path. MDP-4 is learned in less episodes because more actions are executed per episode. As expected, the performance of our learner is similar to Pasula et al.'s one, both learners can obtain the model of the task with few transitions so they also work well when integrated in V-MIN. Pasula et al.'s learner performs slightly better in the MDP-2 and our learner got slightly better results in the MDP-4, but the differences are not significant.

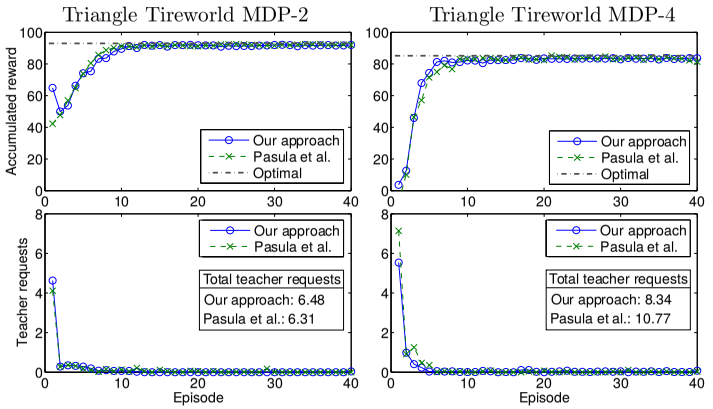


Figure 7: Learning the Triangle Tireworld domain with V-MIN. Comparison of using V-MIN with our learner and Pasula et al.'s learner. The difference between MDP-2 and MDP-4 is that in the latter the state space is larger, and more actions are required to reach the goal. The results shown are the means obtained from 100 runs. The exploration threshold is $\zeta = 3$ and V_{min} is 85 in MDP-2, and 65 in MDP-4.

5.3.2 Crossing Traffic

The Crossing Traffic domain is a grid where a robot must get to a goal and avoid cars arriving randomly and moving left. The goal is located on the top right of the grid, and the robot starts on the bottom right position. If a car overlaps with the robot, the robot disappears and can no longer move around. The robot can "duck" underneath a car by deliberately moving right when a car is to the right of it. The robot receives -1 for every time step it has not reached the goal. The best strategy is to move first to the left to be able to see if a car is coming, and crossing whenever there are no cars in the way.

The difficulty in this domain is learning that if the robot ends in the same position as a car, it will disappear independently of the action executed. Our model learner has to see one transition of the disappear effect for every action until it can learn the disappear effect completely. Figure 8 shows the results of V-MIN and REX with our model learner. Pasula et al.'s learner was not compared here because, as mentioned previously, it cannot learn the exogenous effects in this task.

The main problem for REX in this domain is that some action effects can only be learned if the robot has moved previously and has not collided with a car. For example, the "move-right" action can only be learned if the robot moved first to the left (because it starts in the bottom-right corner) and if it did not collide before with any car. Thus, if the exploration threshold is low, REX may start to exploit with an incomplete model that provides suboptimal results.

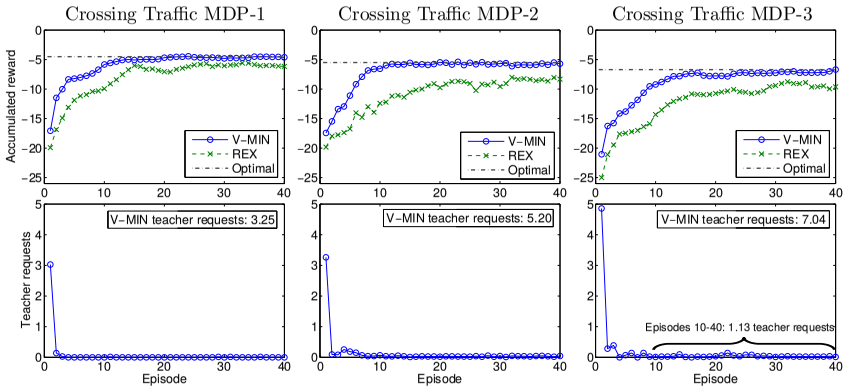


Figure 8: Learning the Crossing Traffic domain with V-MIN and REX. The results shown are the means obtained from 300 runs. MDP-1 is the easiest set up: its state space is small and the probability that a car arrives to each middle lane is 0.3. MDP-2 has also a small state space but cars arrive with a probability of 0.6. MDP-3 has a larger state space and a 0.3 probability of car arrivals. The exploration threshold is $\zeta = 3$ and V_{min} is -8 in MDP-1, -12 in MDP-2, and -15 in MDP-3.

In contrast, V-MIN requests demonstrations when the models do not provide good enough policies, and the demonstrations quickly teach the missing action effects. Below we analyze each MDP separately:

- In MDP-1, the naive strategy of going directly to the goal has a high success ratio (70%), and thus it may take some episodes until the robot collides with a car. The robot always takes the shortest route until it learns that it disappears after a collision with a car independently of the action taken. This in an easy problem so V-MIN only requires a few teacher demonstrations during the first episodes to get good results. REX takes longer to get a high reward because it has to explore more to obtain a good model.

- In MDP-2, the success ratio of reaching the goal with the shortest route is only 40%. In this problem V-MIN requires a few extra demonstrations in episodes 2-6 if it could not learn the "move-right" action before. In contrast, REX failed to obtain good models in a few iterations because it had already collided or was in an edge every time it tried to explore an important action.

- MDP-3 works similarly to MDP-1, but as it has a larger state space, the probability of reaching the goal without dodging cars is around 50%. The larger state space also implies that new problematic situations may appear. For example, the robot may start to go up, and if too many cars appear so that the robot cannot dodge them, it has to go back down by executing a "move-south" action. The "move-south" action is not needed in MDPs 1 and 2, but it is needed in some cases in MDP-3. After the 10th episode, most of the teacher demonstrations requested by V-MIN are to learn how to solve these unexpected problems that didn't appear in the simpler problems. As REX cannot request teacher demonstrations, it has a harder time dealing with these unexpected situations, and this is reflected in a lower accumulated reward mean when compared to MDP-1.

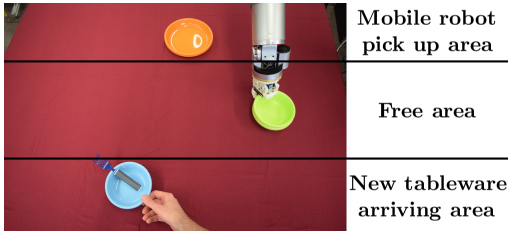


Figure 9: Table clearing task. On the bottom area, people leave the used tableware. On the top area, the robot has to prepare piles for the mobile robots that take tableware to the kitchen. The middle space is free for the robot to store objects while piling them up and waiting for mobile robots.

6. Robot Table Clearing

This section describes a task where a robot has to clear the tableware laying on a table. To that end, V-MIN with the learner presented previously was used as the decision-maker of the robot to solve the task.

This task represents a robotized restaurant. The manipulator robot that we control has to clear the tableware on a table. It has to cooperate with mobile robots that can take a pile of tableware from the table to the kitchen to be cleaned. The number of mobile robots is limited, so our robot should pile tableware together to minimize the number of piles to be taken. The difficulty of the task comes from the fact that people will continuously bring new used tableware to the table. The robot has to organize continuously the tableware on the table so that there is always enough space for people to leave new tableware, and also to ensure that the mobile robots always have a prepared pile of tableware when they come to the table.

This task is illustrated in Fig. 9. People leave used tableware on the bottom area, the robot has the central area to organize tableware and create piles, and the top area is where the mobile robots pick a pile up when they come to the table. The tableware being used includes plates, cups and cutlery.

The robot system has three modules, the decision-maker module that includes the planner and the learner, the perception module that obtains a representation of the scene that can be used by the decision-maker, and the manipulation module that executes actions.

The perception module uses a camera that is located on top of the table (hanging from the ceiling) to update continuously a symbolic state that represents the table. This state consists of a set of literals that describe the different locations on the table. These literals are:

- "ArrivingLocation(?loc)" indicates that ?loc is a location where people will leave new tableware. This literal is constant as people always leave objects on the bottom side of the table.
- "PickUpLocation(?loc)" indicates that ?loc is a location where mobile robots will pick up piles to take them to the kitchen. This literal is constant as mobile robots always pick up piles from the top side of the table.
- "mobileRobotPickingUp(?loc)" indicates that a mobile robot will pick up a pile during the next iteration.
- "plate(?loc)" indicates that there is at least one plate at ?loc.
- "cup(?loc)" indicates that there is at least one cup at ?loc.
- "cutlery(?loc)" indicates that there is at least one fork, knife or spoon at ?loc.
- "stable(?loc)" indicates that the pile at ?loc is stable. An unstable pile cannot be picked up by a mobile robot.

The objects in this domain are the locations. The perception module generates the location objects dynamically by assigning one location to each pile of tableware, and then generates extra locations for (large enough) empty areas.

The manipulator module executes the actions planned. It takes the 3D perceptions obtained by the camera and generates the arm trajectories and gripper movements required to move the objects as desired. The symbolic actions that can be planned are:

- The "put(?loc1, ?loc2)" action to put the top object from a pile in location ?loc1 into location ?loc2.
- The "movePile(?loc1, ?loc2)" action that drags a whole pile from ?loc1 to ?loc2 if ?loc2 is empty. This action has high failure rates with unstable piles.

Piles may become unstable if objects are not piled properly. For example, if a plate is placed on top of a pile containing a cup and a fork, there is a high probability that it will become unstable. To obtain stable piles, in general, plates should be placed at the bottom, cups on top of them, and cutlery at the top. Unstable piles are harder to move, and mobile robots cannot pick them successfully.

Finally the decision-maker module uses the V-MIN algorithm (Section 4) in combination with the learner presented in Section 3 to learn, and the PROST planner (Keller and Eyerich, 2012) to plan.

Figures 10 and 11 show examples of the Table Clearing task. The robot plans the optimal action to execute based on the state of the table.

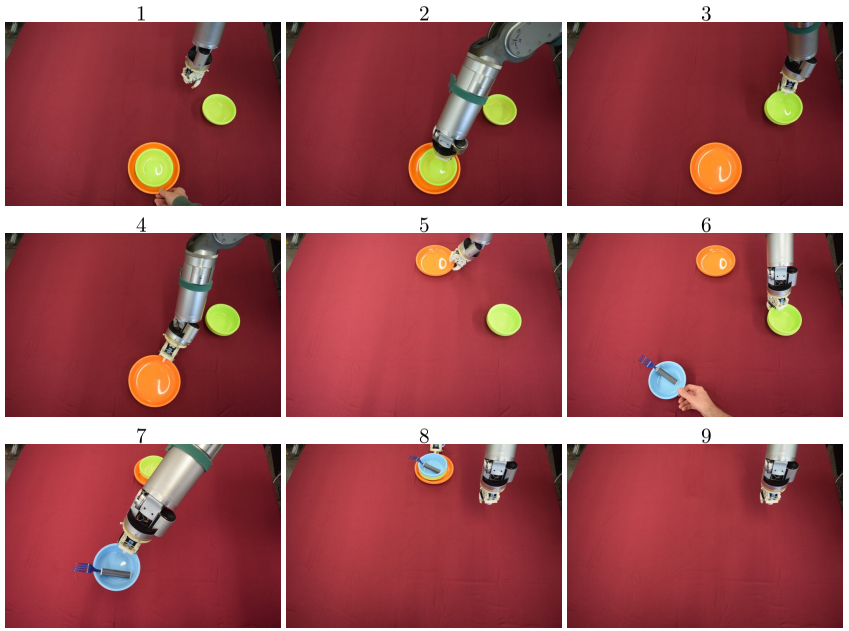


Figure 10: Table clearing with a learned model. As the robot does not know if new tableware will arrive shortly, it decides first to place the two cups together (images $1-3$), and then moves the plate to the *pick up* area (images $4-5$). Afterwards, a new pile arrives that contains no plates (image 6), so the robot decides to move the cups to the picking up area (image 7), and finally, as a mobile robot is arriving, the robot places the last pieces of tableware on the *pick up* area (image 8). Image 9 shows the final state after a mobile robot picked up the pile.

6.1 Learning a Model with RL

Using the setup described above, we executed the learner to obtain a model of the task from scratch while solving it. To make the experimentation easier, instead of having mobile robots picking up piles from the *pick up* area, a person did it. The robot had to solve experiments of increasing difficulty, and it transferred the learned knowledge from one experiment to the next. All experiments lasted 30 iterations during which piles were picked up 5 times from the *pick up* area.

1. The easiest experiment was executed once. People left 10 objects on the table during the experiment, including only plates and cups.

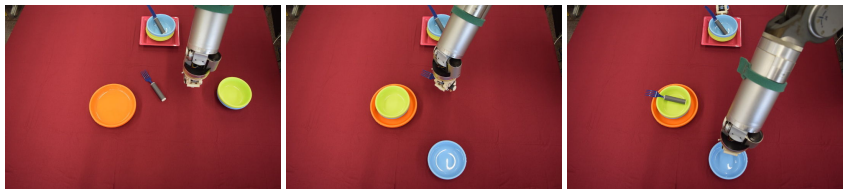


Figure 11: Table clearing with a learned model. **Initially**, the robot has already one pile prepared for the mobile robots. The planner opts to prepare a new pile with the tableware on the table as it expects that at some point a mobile robot will pick up the pile on top. **In the second image**, the new pile is halfway done, and a new cup arrives. The planner knows that in the next iteration a mobile robot is coming, so it decides to finish the new pile. **In the third image**, a mobile robot picks the top pile, and the manipulator robot plans that the best action is to move the new cup to the center free area to make more space in the arriving area. Once the mobile robot takes the top pile, the manipulator robot will plan the action of taking the center pile to the *pick up* area.

2. The medium experiment was executed twice. People left 13 objects on the table that included plates, cups and cutlery.

3. The difficult experiment was executed twice. People left 18 objects on the table that included plates, cups and cutlery.

The goals of the robot were to minimize the number of objects on the table, and to have empty space in the *arriving* area so that people always had space to place new tableware. Therefore big enough piles had to be given to the mobile robot at the same time that the arriving area had to be cleared quickly.

The learner parameters used were $\alpha = 0.01$, $\epsilon = 0.1$, $\omega = 2$, $\delta = 0.05$, $\kappa = 1000$, and the subsumption was enabled. The V-MIN exploration threshold was $\zeta = 3$ and V_{min} was selected and updated by the teacher depending on the robot performance.

6.1.1 Learning Results

Table 1 summarizes the performance of the manipulator robot. During the easy experiment, 7 teacher demonstration requests were required to complete the task. The robot had to learn the "put" action to pick and place objects, and also the operators defining how mobile robots took piles from the *pick up* area.

In the second experiment, as cutlery appeared for the first time, and piles that have cutlery in the middle are usually unstable, the robot needed 6 extra demonstrations to learn how to manipulate cutlery and how unstable piles could be unpiled to make them stable again. The third experiment needed only 3 more demonstrations as the robot already knew most of the dynamics.

	Easy 1	Medium 1	Medium 2	Difficult 1	Difficult 2
Number of objects	10*	13	13	18	18
Sum of reward	−48	−90	−82	−148	−130
Optimal rewards	−41	−64	−65	−97	−101
Teacher requests	7	6	3	4	1

*In the easy setup no cutlery was used (only plates and cups).

Table 1: Learning results in the robot table clearing task. The columns show the different learning experiments, which were executed in order of increasing difficulty. The rows show the number of objects that arrived in each experiment, the sum of rewards, and the number of teacher demonstration requests.

Finally, in the last difficult experiments, 5 teacher demonstrations were needed once many objects accumulated on the table. The robot had to learn that it could assemble a pile in the middle area, and then move it directly to the *pick up* area by using the action "movePile(?loc1, ?loc2)". In previous experiments this action was not needed as less tableware arrived and the robot could just make piles directly on the *pick up* area with the "put" action.

Appendix A explains in more detail the execution of the learning process as well as the behavior of the manipulator robot during the easy experiment and the first medium experiment.

It should be noted that during the learning experiments we observed that the effectiveness of the robot actions was very important. If actions failed often, many more action executions were needed to obtain a proper model. Therefore, before running the experiments presented here, the robot actions were improved so that they succeeded in most cases to make the task simpler.

6.2 Evaluation of the Learned Model

Finally, an experiment was carried out to evaluate the usefulness of learning exogenous effects. Here we compare the performance when using the model learned in the previous section, with another model learned with no exogenous effects (Pasula et al., 2007). The exogenous effects allow the planner to know the probabilities with which each type of tableware arrives, and the probability with which mobile robots come to pick piles up. There was a reward of −1 for every object on the table at each step, and a reward of −2 if new tableware was arriving but the arriving area was full. The results are shown in Table 2.

- An optimal action sequence obtained a reward of −99. Note that this optimal action sequence was not obtained in a fair way: it was created knowing in advance when and which type of tableware was arriving at each step, and when a mobile robot approached. In contrast the manipulator robot did not have this information, it could only know the probabilities of these effects.

- The model containing exogenous effects obtained a reward of −119. In this experiment the manipulator robot performed well in general, but it created some inefficient piles

	Optimal	Model with exogenous	Model with no exogenous
Sum of rewards	−99	−119	−136
Operators (actions)	-	8	20
Operators (exogenous)	-	11	0

Table 2: Execution results with the learned models. The columns show the results with an optimal sequence of actions, a learned model considering exogenous effects, and a learned model not considering exogenous effects. The rows show the sum of rewards and the number of planning operators that represent actions and exogenous effects.

as a mobile robot would take shorter or longer than expected to arrive. Moreover, the manipulator robot took a conservative strategy and it didn't complete fast enough some piles, as it did not know if new tableware would be arriving shortly. The model consisted of 8 planning operators for action effects, and 11 planning operators for exogenous effects.

- The model without exogenous effects obtained a reward of −136. This model had more trouble to complete the task. The manipulator robot did not know that new tableware would arrive because it did not have the exogenous effect for it. Thus, it tried to make the best piles with the given objects, which resulted in the robot redoing the piles whenever new tableware arrived (e.g. if a new cup arrived, the robot would put it on top of the other cups in an existing pile, and below the cutlery). The model had 20 operators that represented action effects. As exogenous effects were not learned separately, each operator actually represented combinations of an action effect with exogenous effects. However, these operators were less effective to represent the model than separate action and exogenous operators, so the planner selected worse policies.

Analyzing the results we can see that the proposed learner allows a robot to learn tasks including exogenous effects. Given that only 5 episodes of 30 action executions were used for training, the results were good. The experiment also shows that a model considering exogenous effects improves the performance when external agents interact. Here the planner was able to find much better plans when it anticipated that new tableware or a mobile robot were arriving. However, extending this task with a richer representation and more external agents would be costly. For every literal and exogenous effect that gets added, the number of input transitions and computational time required to learn a model increases significantly.

7. Conclusions

We have introduced a new method that, given a set of input transitions, learns a general model explaining them. In contrast to previous approaches, it can learn exogenous effects (effects not related to any action), while still being similarly good at obtaining a relational representation of the problem and at learning uncertain effects. Moreover optimal and suboptimal search methods are provided, so the best approach can be chosen depending on

the quality requirements, the difficulty of the problem, and the learning time available. The main limitation of the algorithm is scalability when the number of generated propositional predicates becomes large.

This learner can be combined with a RL algorithm to learn models from scratch. We validated experimentally that the integration in the V-MIN algorithm allowed an agent to learn models with a relatively small number of actions and teacher demonstration requests.

The learner was also integrated in a robot to perform the task of clearing the tableware on a table. In this task external agents interacted, people brought new tableware continuously and the manipulator robot had to cooperate with mobile robots to take the tableware to the kitchen. The learner was able to learn a usable model in just 5 episodes of 30 action executions. Finally, the model was used to complete the task, and learning models with such exogenous effects proved to increase the obtained reward significantly.

Many robotic problems cannot be learned yet. As future work, the learner could be extended to tackle correlated effects and partial observations.

- The challenge with correlated effects is to identify the candidates. A possible path to explore is to replace LFIT by Ribeiro et al. (2015)'s approach, in which preconditions can be literals in both the previous state (s_t) and the resulting state (s_{t+1}). In this case, an effect (a literal in s_{t+1}) can be a precondition for another effect, so they would be correlated effects.

- Partial observations would require to change the heuristic search and the cost function to consider that the given state transitions may be wrong or incomplete.

Acknowledgments

This work has been supported by the MINECO project RobInstruct TIN2014-58178-R and the European Union's Horizon 2020 research and innovation programme under grant agreement H2020-ICT-2016-1-731761 IMAGINE. D. Martínez is also supported by the Spanish Ministry of Education, Culture and Sport via a FPU doctoral grant (FPU12-04173).

Appendix A. Learning the Table Clearing Task

In Sec. 6.1 we explained how the manipulator robot learned the table clearing task. This appendix shows the detailed execution of the easy experiment and the first medium experiment to provide a deeper understanding of the learning process.

A.1 Easy Experiment

The states at each iteration of the easy experiment are shown in Fig. 12. The robot starts with no prior knowledge and has to learn the complete model. Below is an explanation of the actions taking place at each iteration:

1. Initially the robot does not have prior knowledge and its model is empty, so no action is taken. A person brings a plate (exogenous effect).

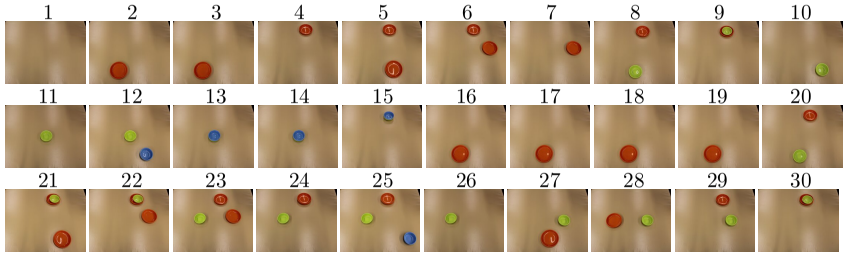


Figure 12: The state at the beginning of each iteration during the easy experiment.

2. No action is taken.
3. The manipulator robot learns that plates remain in the same location if no action is taken. A demonstration is requested to learn how to deal with plates, and the teacher demonstrates the "put" action to move the plate to the *pick up* area.
4. The manipulator robot executes the "put" action with an empty location as parameter, so it does not move anything. A new plate arrives.
5. The new plate is moved to the middle of the table.
6. The manipulator robot executes "put" with an empty location. Meanwhile, a mobile robot comes and picks the plate on the *pick up* area (another type of exogenous effect).
7. After learning that mobile robots pick up plates on the *pick up* area, the manipulator robot moves the plate on the middle to the *pick up* area. A cup arrives to the table.
8. A demonstration is requested, and the teacher moves it to the *pick up* area with a "put" action.
9. It is not known yet that mobile robots also pick cups up, so a demonstration is requested. The teacher does not execute any action because a mobile robot is already arriving to take the pile. A new cup appears.
10. The learned model is not good enough yet, so a demonstration is requested. The teacher puts the cup in the center of the table.
11. An exploratory "put" action with an empty location is taken. A new cup arrives.
12. A demonstration is requested. The teacher shows that both cups can be piled together.
13. No action is taken.
14. As a mobile robot is arriving, the cups are moved to the *pick up* area.
15. The mobile robot takes the cups and a new plate appears.

16. The robot executes a "put" action that fails.
17. An exploratory "put" action is executed with an empty location (a new planning operator with the action "put" and the precondition "cup(?X)" was generated and it is explored).
18. An exploratory "put" action is executed with an empty location.
19. The plate is moved to the *pick up* area. A cup appears.
20. The new cup is moved to the *pick up* area. A new plate appears.
21. A demonstration is requested (the robot does not know yet that it can pile plates together). The teacher decides to put the new plate on an empty location because piling it on the *pick up* area could be unstable.
22. As an exploratory action, the cup is moved to an empty location.
23. A demonstration is requested, and the teacher piles the two plates together.
24. No action was taken (exploration is not needed, so it decides to wait until new objects arrive or a mobile robot approaches). A new cup arrives.
25. The new cup is moved to the pile in the *pick up* area because a mobile robot is arriving.
26. As an exploration action, the cup is moved to a different location on the center of the table. A plate arrives.
27. The plate is moved to the center.
28. The plate is moved to the *pick up* area.
29. The cup is put on top of the plate.
30. A mobile robot comes and takes the last pile.

A.2 First Medium Experiment

Here the robot reuses the knowledge that it obtained during the easy experiment. The states at each iteration of the first medium experiment are shown in Fig. 13. Below is an explanation of the actions taking place at each iteration:

1. Initially there is no tableware, so no action is taken. A plate arrives.
2. The plate is "put" on the *pick up* area. A cup arrives.
3. An exploratory "put" action that does nothing is executed (the first parameter is an empty location and the second one is the cup, so it does not move anything).
4. The operators learned are not complete enough, so a demonstration is requested. The teacher places the cup in the middle of the table. A plate arrives

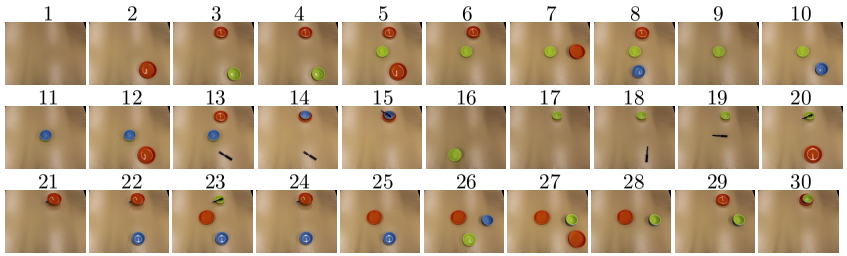


Figure 13: The state at the beginning of each iteration during the first medium experiment.

5. The new plate is moved to the *pick up* area.
6. As an exploratory action, the plates are moved to the center of the table.
7. The plates are moved back to the *pick up* area. A cup arrives.
8. As a mobile robot is arriving, it is piled up directly in the *pick up* area. The pile on the *pick up* area is taken.
9. No action is taken (exploration is not needed, so it decides to wait until new objects arrive or a mobile robot approaches). A new cup arrives
10. The new cup is piled with the other one.
11. No action is taken and a plate arrives.
12. The new plate is moved to the *pick up* area.
13. Cutlery appears for the first time. A demonstration is requested and the teacher decides to move the cups to the *pick up* area. Note that the teacher shows the best action, and not the action that would make the robot learn the most. In this case it is better to first put the cup on the plate, and later put the cutlery on the cup.
14. As the actions to manipulate cutlery have not been learned yet, another demonstration is requested. This time the teacher executes a "put" action to pile up the fork.
15. The manipulator robot has still to learn that mobile robots also take cutlery, so another demonstration is requested. As a mobile robot is arriving to take the pile, the teacher does not execute any action. A cup arrives.
16. The new cup is moved to the *pick up* area.
17. No action is taken. A fork arrives.
18. The new fork is moved to the center of the table.
19. The fork is put on top of the cup at the *pick up* area. A plate arrives.

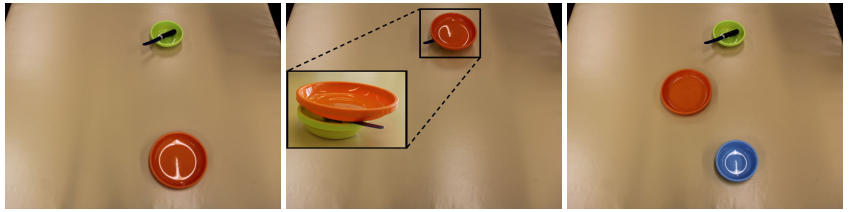


Figure 14: Unstable pile during the first medium experiment. **Left:** the state before the manipulator robot placed the plate on the top pile. **Center:** the pile became unstable, and thus it was not taken by mobile robots. **Right:** to make the pile stable again, the plate was positioned back in the table.

20. The new plate is moved to the pile at the *pick up* area. However, the pile becomes unstable as a plate is on top of a cup with a knife (see Fig. 14).
21. A mobile robot comes, but it does not take the pile because it is unstable. A cup arrives.
22. The manipulator robot has learned that mobile robots do not take unstable piles, so a demonstration is requested. The teacher shows that putting the plate back to the table makes the pile stable again.
23. As an exploratory action, the plate is put again on the pile, making it unstable.
24. The manipulator robot makes the pile stable again by removing the plate. A mobile robot comes and takes the pile on the *pick up* area as it is stable.
25. The cup on the *arriving* area is moved to the center of the table. A new cup arrives.
26. The model has updated its probabilities with which new objects and mobile robots arrive, and the resulting plan has a value $V(s) < V_{min}$ even though it is a good plan. The problem is that V_{min} was too high, and the teacher updates it after piling the two cups together. A plate arrives.
27. The plate is piled on top of the other plate.
28. The plates are moved to the *pick up* area.
29. The cups are piled on top of the plates.
30. A mobile robot comes and takes the last pile.

References

Alejandro Agostini, Carme Torras, and Florentin Wörgötter. Efficient interactive decision-making framework for robotic applications. *Artificial Intelligence*, 247:187–212, 2017.

Eyal Amir and Allen Chang. Learning partially observable deterministic action models. *Journal of Artificial Intelligence Research*, 33:349–402, 2008.

Ronen I Brafman and Moshe Tennenholtz. R-max - A general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2003.

Sonia Chernova and Manuela Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34(1):1–25, 2009.

Ashwin Deshpande, Brian Milch, Luke S Zettlemoyer, and Leslie Pack Kaelbling. Learning probabilistic relational dynamics for multiple tasks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 83–92, 2007.

Carlos Diuk, Andre Cohen, and Michael L Littman. An object-oriented representation for efficient reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, pages 240–247, 2008.

Sašo Džeroski, Luc De Raedt, and Kurt Driessens. Relational reinforcement learning. *Machine Learning*, 43(1-2):7–52, 2001.

Floriana Esposito, Stefano Ferilli, Nicola Fanizzi, Teresa Maria Altomare Basile, and Nicola Di Mauro. Incremental learning and concept drift in INTHELEX. *Intelligent Data Analysis*, 8(3):213–237, 2004.

Daniel H Grollman and Odest Chadwicke Jenkins. Dogged learning for robots. In *Proceedings of the International Conference on Robotics and Automation*, pages 2483–2488, 2007.

Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

Todd Hester and Peter Stone. TEXPLORE: real-time sample-efficient reinforcement learning for robots. *Machine Learning*, 90(3):385–429, 2013.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, pages 253–302, 2001.

Katsumi Inoue, Tony Ribeiro, and Chiaki Sakama. Learning from interpretation transition. *Machine Learning*, 94(1):51–79, 2014.

Sergio Jiménez, Fernando Fernández, and Daniel Borrajo. The PELA architecture: integrating planning and learning to improve execution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1294–1299, 2008.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002.

Thomas Keller and Patrick Eyerich. PROST: Probabilistic Planning Based on UCT. In *Proceedings of the International Conference on Automated Planning and Scheduling*, pages 119–127, June 2012.

W Bradley Knox and Peter Stone. Reinforcement learning from simultaneous human and MDP reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 475–482, 2012.

Andrey Kolobov, Peng Dai, Mausam, and Daniel S Weld. Reverse iterative deepening for finite-horizon MDPs with large branching factors. In *Proceedings of the International Conference on Automated Planning and Scheduling*, pages 146–154, 2012.

George Konidaris, Ilya Scheidwasser, and Andrew G Barto. Transfer in reinforcement learning via shared features. *The Journal of Machine Learning Research*, 13(1):1333–1371, 2012.

Johannes Kulick, Marc Toussaint, Tobias Lang, and Manuel Lopes. Active learning for teaching a robot grounded relational symbols. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1451–1457, 2013.

Tobias Lang and Marc Toussaint. Planning with noisy probabilistic relational rules. *Journal of Artificial Intelligence Research*, 39:1–49, 2010.

Tobias Lang, Marc Toussaint, and Kristian Kersting. Exploration in relational domains for model-based reinforcement learning. *Journal of Machine Learning Research*, 13:3691–3734, 2012.

Lihong Li, Michael L Littman, Thomas J Walsh, and Alexander L Strehl. Knows what it knows: a framework for self-aware learning. *Machine Learning*, 82(3):399–443, 2011.

Iain Little and Sylvie Thiebaux. Probabilistic planning vs. replanning. In *Proceedings of the ICAPS Workshop on IPC: Past, Present and Future*, 2007.

Michael L Littman. Reinforcement learning improves behaviour from evaluative feedback. *Nature*, 521(7553):445–451, 2015.

David Martínez, Guillem Alenyà, and Carme Torras. Planning robot manipulation to clean planar surfaces. *Engineering Applications of Artificial Intelligence*, 39:23–32, 2015a.

David Martínez, Guillem Alenyà, and Carme Torras. V-MIN: Efficient reinforcement learning through demonstrations and relaxed reward demands. In *Proceedings of The AAAI Conference on Artificial Intelligence*, pages 2857–2863, 2015b.

David Martínez, Tony Ribeiro, Katsumi Inoue, Guillem Alenyà, and Carme Torras. Learning probabilistic action models from interpretation transitions. In *Technical Communication of the International Conference on Logic Programming, CEUR Workshop Proceedings*, volume 1433(30), 2015c.

David Martínez, Guillem Alenyà, Carme Torras, Tony Ribeiro, and Katsumi Inoue. Learning relational dynamics of stochastic domains for planning. In *International Conference on Automated Planning and Scheduling*, pages 235–243, 2016.

Çetin Meriçli, Manuela Veloso, and H Levent Akın. Multi-resolution corrective demonstration for efficient task execution and refinement. *International Journal of Social Robotics*, 4(4):423–435, 2012.

Bogdan Moldovan, Plinio Moreno, Martijn van Otterlo, José Santos-Victor, and Luc De Raedt. Learning relational affordance models for robots in multi-object manipulation tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4373–4378, 2012.

Matthew Molineaux and David W Aha. Learning unknown event models. In *Proc. of the AAAI Conference on Artificial Intelligence*, pages 395–401, 2014.

Kira Mourão. Learning probabilistic planning operators from noisy observations. In *Proceedings of the Workshop of the UK Planning and Scheduling Special Interest Group*, 2014.

Kira Mourão, Luke S Zettlemoyer, Ronald Petrick, and Mark Steedman. Learning STRIPS operators from noisy and incomplete observations. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 614–623, 2012.

Hanna M Pasula, Luke S Zettlemoyer, and Leslie Pack Kaelbling. Learning symbolic models of stochastic domains. *Journal of Artificial Intelligence Research*, 29(1):309–352, 2007.

Tony Ribeiro and Katsumi Inoue. Learning prime implicant conditions from interpretation transition. In *Proceedings of the International Conference on Inductive Logic Programming, LNAI*, volume 9046, pages 108–125, 2014.

Tony Ribeiro, Morgan Magnin, Katsumi Inoue, and Chiaki Sakama. Learning multi-valued biological models with delayed influence from time-series observations. In *Proc. of the International Conference on Machine Learning and Applications*, pages 25–31, 2015.

Scott Sanner. Relational dynamic influence diagram language (RDDL): Language description. *Unpublished ms. Australian National University*, 2010.

Daniel Sykes, Domenico Corapi, Jeff Magee, Jeff Kramer, Alessandra Russo, and Katsumi Inoue. Learning revised models for planning in adaptive systems. In *Proceedings of the International Conference on Software Engineering*, pages 63–71, 2013.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.

Ingo Thon, Niels Landwehr, and Luc De Raedt. Stochastic relational processes: Efficient inference and applications. *Machine Learning*, 82(2):239–272, 2011.

Mauro Vallati, Lukáš Chrpa, Marek Grześ, Thomas L McCluskey, Mark Roberts, and Scott Sanner. The 2014 international planning competition: Progress and trends. *AI Magazine*, 36(3):90–98, 2015.

Thomas J Walsh. *Efficient learning of relational models for sequential decision making.* PhD thesis, Rutgers, The State University of New Jersey, 2010.

Thomas J Walsh, István Szita, Carlos Diuk, and Michael L Littman. Exploring compact reinforcement-learning representations with linear regression. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 591–598, 2009.

Thomas J Walsh, Kaushik Subramanian, Michael L Littman, and Carlos Diuk. Generalizing apprenticeship learning across hypothesis classes. In *Proceedings of the International Conference on Machine Learning*, pages 1119–1126, 2010.

Thomas J Walsh, Daniel K Hewlett, and Clayton T Morrison. Blending autonomous exploration and apprenticeship learning. In *Advances in Neural Information Processing Systems*, pages 2258–2266, 2011.

Håkan LS Younes and Michael L Littman. PPDDL1. 0: An extension to PDDL for expressing planning domains with probabilistic effects. *Technical Report CMU-CS-04-162*, 2004.

George Zhu, Dan Lizotte, and Jesse Hoey. Scalable approximate policies for Markov decision process models of hospital elective admissions. *Artificial Intelligence in Medicine*, 61(1): 21–34, 2014.

Hankz Hankui Zhuo and Subbarao Kambhampati. Action-model acquisition from noisy plan traces. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2444–2450, 2013.

Hankz Hankui Zhuo and Qiang Yang. Action-model acquisition for planning via transfer learning. *Artificial Intelligence*, 212:80–103, 2014.

Journal of Machine Learning Research 18 (2017) 1-31 Submitted 7/16; Published 8/17

Bayesian Tensor Regression

Rajarshi Guhaniyogi* rguhaniy@ucsc.edu
Department of Applied Mathematics & Statistics
University of California
Santa Cruz, CA 95064, USA

Shaan Qamar* siqamar@gmail.com
Google Inc.
Mountain View, CA 94043, USA

David B. Dunson dunson@duke.edu
Department of Statistical Science
Duke University
Durham, NC 27708-0251, USA

Editor: Robert McCulloch

Abstract

We propose a Bayesian approach to regression with a scalar response on vector and tensor covariates. Vectorization of the tensor prior to analysis fails to exploit the structure, often leading to poor estimation and predictive performance. We introduce a novel class of multiway shrinkage priors for tensor coefficients in the regression setting and present posterior consistency results under mild conditions. A computationally efficient Markov chain Monte Carlo algorithm is developed for posterior computation. Simulation studies illustrate substantial gains over existing tensor regression methods in terms of estimation and parameter inference. Our approach is further illustrated in a neuroimaging application.

Keywords: Multiway Shrinkage Prior, Magnetic Resonance Imaging (MRI), Parafac Decomposition, Posterior Consistency, Tensor Regression

1. Introduction

In many application areas, it is common to collect predictors that are structured as a multiway array or tensor. For example, the elements of this tensor may correspond to voxels in a brain image (Lindquist, 2008; Lazar, 2008; Hinrichs et al., 2009; Ryali et al., 2010). Existing approaches for quantifying associations between an outcome and such tensor predictors mostly fall within two groups. The first approach assesses the association between each cell (for brain images referred to as voxel) and the response independently, providing a p-value 'map' (Lazar, 2008). The p-values can be adjusted for multiple comparisons to identify 'significant' sub-regions of the tensor. Although this approach is widely used and appealing in its simplicity, clearly such independent screening approaches have key disadvantages relative to methods that take into account the joint impact of the overall tensor simultaneously. Unfortunately, the literature on simultaneous analysis approaches is sparse.

One naive approach is to simply vectorize the tensor and then use existing methods for high-dimensional regression. Such vectorization fails to preserve spatial structure, making it more difficult to learn low-dimensional relationships with the response. Efficient learning is of critical importance, as the sample size is typically massively smaller than the total number of cells. Alternative approaches within the regression framework include functional regression and two stage approaches. The former views the tensor as a discretization of a continuous functional predictor. Most of the literature on functional predictors focuses on 1D functions; Reiss and Ogden (2010) consider the 2D case, but substantial challenges arise in extensions to 3D due to dimensionality and collinearity among cells. Recently Wang et al. (2014) considered 3D regularized functional regression with Haar wavelet basis. The article is essentially frequentist in nature with simulation studies showing only the mean squared error and the percentage of correctly identified zero and nonzero elements. Additionally, the article reveals that functional regression is largely affected by the choice of proper basis functions. The second set of approaches, i.e. Two stage approaches first conduct a dimension reduction step, commonly using PCA, and then fit a model using lower dimensional predictors (Caffo et al., 2010). A clear disadvantage of such approaches is that the main principal components driving variability in the random tensor may have relatively limited impact on the response variable. Potentially, supervised PCA could be used, but it is not clear how to implement such an approach in 3D or higher dimensions.

Zhou et al. (2013) propose extending generalized linear regression to include a tensor structured parameter corresponding to the measured tensor predictor. To circumvent difficulties with extensions to higher order tensor predictors, they impose additional structure on the tensor parameter, supposing it decomposes as a rank-R parafac sum (see Section 2.1). This massively reduces the effective number of parameters to be estimated. They develop a penalized likelihood approach where adaptive lasso penalties are be imposed on individual margins of the parafac decomposition, focusing on good point estimation for the tensor parameter. However, their method relies heavily on cross-validation for selecting tuning parameters which are sensitive to the tensor dimension, the signal-to-noise ratio (degree of sparsity) and the parafac rank. Given that there is no automatic selection procedure for the tuning parameters provided in Zhou et al. (2013), they have to be fed manually by the end user which is problematic for an unknown tensor regression problem.

Of practical interest is a "self calibrating" procedure which adapts the complexity of the model to the data. We propose a principled method to effectively shrink unimportant cell coefficients to zero while maintaining accuracy in estimating important cell coefficients. Our framework gives rise to the automatic selection of tuning parameters, with carefully constructed shrinkage priors that naturally induce sparsity within and across components in the tensor factorization of the tensor coefficient for optimal region selection. In addition, the need for valid measures of uncertainty on parameter (predictive) estimates is crucial, especially in settings with low or moderate sample sizes, which naturally motivates our Bayesian approach. Recently Suzuki (2015) proposed a Bayesian tensor regression approach with naive Gaussian prior on the components of the tensor factorization of the tensor coefficients. In contrast, our proposed prior on the tensor coefficient is more sophisticated in the sense that it imparts shrinkage in three ways: at a global level, at a local level of

. ***These authors contributed equally**

©2017 Rajarshi Guhaniyogi and Shaan Qamar and David B. Dunson.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-362.html.

individual parameters, and also provides shrinkage towards low rank decomposition of the tensor coefficient. Similarly, Bayesian tensor regression framework proposed in Goldsmith et al. (2014) uses binary indicators to determine whether a cell in the tensor predictor is predictive of the response. For a tensor predictor with $30 \times 30 \times 30$ cells, such an approach requires to update 27000 binary indicators in each MCMC iteration and is deemed unsatisfactory due to mixing issues and poor inference.

Our approach differs from image reconstruction literature as we do not model the distribution of the tensor $\boldsymbol{X}$ (Qiu, 2007). There is a considerable recent literature on frequentist tensor modeling in which one typically encounters time series (generally to study social networks or images evolving over time) with response at every time point is an array/tensor (Gerard and Hoff, 2015; Hoff et al., 2015). There is also a Bayesian literature that facilitates joint modeling of a large number of unordered categorical variables (Zhou et al., 2015). Our framework is fundamentally different from these approaches in the sense that *these are all unsupervised tensor modeling approach while we propose a framework for supervised tensor regression.* To the best of our knowledge, we are the first to propose a novel multiway shrinkage prior in Bayesian tensor regression framework (with scalar response on a tensor predictor) that accommodates shrinkage of the tensor coefficient for the appropriate identification of important cells in the tensor predictor. Besides, we offer strong posterior consistency results on Bayesian tensor regression framework with multiway shrinkage prior.

Remainder of the manuscript evolves as follows. In Section 2, we propose the basic framework of the tensor regression model with a scalar response, vector predictors and a tensor predictor. Section 3 characterizes desirable criteria for a multiway shrinkage prior and proposes a novel multiway shrinkage prior on the tensor coefficient. Sections 4 and 5 provide theoretical results on the convergence of posterior distribution under the mutiway shrinkage prior and details on posterior computation respectively. Various simulation studies with 2D and 3D tensor predictors are presented in Sections 6 and 7 respectively to study effectiveness of the Bayesian tensor regression under various degrees of sparsity and signal strength. Section 8 is devoted to a real brain connectome data analysis using the proposed Bayesian tensor regression model along with its competitors. The manuscripts ends with a discussion.

2. Tensor Regression

This section provides details on the tensor regression model.

2.1 Basic Notation

Let $\boldsymbol{\beta}_1 = (\beta_{11}, \ldots, \beta_{1p_1})'$ and $\boldsymbol{\beta}_2 = (\beta_{21}, \ldots, \beta_{2p_2})'$ be vectors of length p_1 and p_2, respectively. The vector outer product $\boldsymbol{\beta}_1 \circ \boldsymbol{\beta}_2$ is a $p_1 \times p_2$ matrix with (i,j)-th entry $\beta_{1i}\beta_{2j}$. A D-way outer product between vectors $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jp_j})$, $1 \leq j \leq D$, is a $p_1 \times \cdots \times p_D$ multi-dimensional array denoted $\boldsymbol{B} = \boldsymbol{\beta}_1 \circ \boldsymbol{\beta}_2 \circ \cdots \circ \boldsymbol{\beta}_D$ with entries $(\boldsymbol{B})_{i_1,\ldots,i_D} = \prod_{j=1}^{D} \beta_{ji_j}$. Define a vec$(\boldsymbol{B})$ operator as stacking elements of this D-way tensor into a column vector of length $\prod_{j=1}^{D} p_j$. From the definition of outer products, it is easy to see that $\text{vec}(\boldsymbol{\beta}_1 \circ \boldsymbol{\beta}_2 \circ \cdots \circ \boldsymbol{\beta}_D) = \boldsymbol{\beta}_D \otimes \cdots \otimes \boldsymbol{\beta}_1$. As a higher order generalization of matrix singular value decomposition, Tucker decomposition of a D-way tensor $\boldsymbol{B} \in \otimes_{j=1}^{D} \Re^{p_j}$ is

often considered. The Tucker decomposition (Kolda and Bader, 2009) can be expressed as

$$\boldsymbol{B} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_D=1}^{R_D} \lambda_{r_1,\ldots,r_D} \boldsymbol{\beta}_1^{(r_1)} \circ \boldsymbol{\beta}_2^{(r_2)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r_D)} \tag{1}$$

where $\boldsymbol{\beta}_j^{(r_j)}$ is a p_j dimensional vector, $1 \leq j \leq D$, and $\boldsymbol{\Lambda} = (\lambda_{r_1,\ldots,r_D})_{r_1,\ldots,r_D=1}^{R_1,\ldots,R_D}$ is referred to as the *core tensor*. If one considers $\{\boldsymbol{\beta}_j^{(r_j)}; 1 \leq r_j \leq R_j, 1 \leq j \leq D\}$ as "factor loadings" and $\lambda_{r_1,\ldots,r_D}$ to be the corresponding coefficients, then the Tucker decomposition may be thought of as a multiway analogue to factor modeling.

A rank-R parafac decomposition emerges as a special case of Tucker decomposition 1 when $R_1 = R_2 = \cdots = R_D = R$ and $\lambda_{r_1,\ldots,r_D} = I(r_1 = r_2 = \cdots = r_D)$. In particular, $\boldsymbol{B} \in \otimes_{j=1}^{D} \Re^{p_j}$ assumes a rank-R parafac decomposition if

$$\boldsymbol{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)} \tag{2}$$

where $\boldsymbol{\beta}_j^{(r)}$, $1 \leq j \leq D$ and $1 \leq r \leq R$ are the p_j dimensional 'margins'. The parafac decomposition is more widely used due to its relative simplicity.

2.2 Model Framework

Let $y \in \mathcal{Y}$ denotes a response variable, with $\boldsymbol{z} \in \mathcal{X} \subset \Re^p$ and $\boldsymbol{X} \in \otimes_{j=1}^{D} \Re^{p_j}$ scalar and tensor predictors, respectively. We consider a tensor regression model having a general form

$$y \sim f\left(\alpha + \boldsymbol{z}'\boldsymbol{\gamma} + \langle \boldsymbol{X}, \boldsymbol{B} \rangle, \sigma\right), \quad \langle \boldsymbol{X}, \boldsymbol{B} \rangle = \text{vec}(\boldsymbol{X})'\text{vec}(\boldsymbol{B}), \tag{3}$$

where $f(\mu, \sigma)$ is a family of distributions having location μ and scale σ, $\boldsymbol{\gamma}$ is a $p \times 1$ coefficient for scalar preditors and $\boldsymbol{B} \in \otimes_{j=1}^{D} \Re^{p_j}$ is the tensor parameter corresponding to measured tensor predictor $\boldsymbol{X}$. We focus more specifically on the Gaussian linear model case with

$$y = \alpha + \boldsymbol{z}'\boldsymbol{\gamma} + \langle \boldsymbol{X}, \boldsymbol{B} \rangle + \epsilon, \; \epsilon \sim N(0, \sigma^2). \tag{4}$$

The coefficient tensor $\boldsymbol{B}$ has $\prod_{j=1}^{D} p_j$ elements, necessitating substantial dimensionality reduction. A rank-1 parafac decomposition assumes $\boldsymbol{B} = \boldsymbol{\beta}_1 \circ \cdots \circ \boldsymbol{\beta}_D$ and $\text{vec}(\boldsymbol{B}) = \boldsymbol{\beta}_D \otimes \cdots \otimes \boldsymbol{\beta}_1$. This reduces to modeling $y = \alpha + \boldsymbol{z}'\boldsymbol{\gamma} + \boldsymbol{\beta}_1'\boldsymbol{X}\boldsymbol{\beta}_2$ when $D = 2$, corresponding to the bilinear model considered in Hung and Wang (2013). Since only the single parameter vector $\boldsymbol{\beta}_j$ captures signal along the jth dimension, a rank-1 assumption severely limits flexibility, ruling out interactions among dimensions. Following Zhou et al. (2013), we use a more flexible rank-R parafac decomposition for $\boldsymbol{B} = \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)}$ introduced in (2) with $\boldsymbol{\beta}_j^{(r)} \in \Re^{p_j}$, $1 \leq j \leq D$, and $1 \leq r \leq R$. Expression (4) then becomes

$$\begin{aligned} y &= \alpha + \boldsymbol{z}'\boldsymbol{\gamma} + \left\langle \boldsymbol{X}, \sum_{r=1}^{R} \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)} \right\rangle + \epsilon \\ &= \alpha + \boldsymbol{z}'\boldsymbol{\gamma} + \sum_{(i_1,\ldots,i_D)} (\boldsymbol{X})_{i_1,\ldots,i_D} (\boldsymbol{B})_{i_1,\ldots,i_D} + \epsilon \end{aligned} \tag{5}$$

where voxel $(\boldsymbol{X})_{i_1,\ldots,i_D}$ of the tensor predictor has corresponding parameter

$$(\boldsymbol{B})_{i_1,\ldots,i_D} = \sum_{r=1}^{R}\prod_{j=1}^{D}\beta_{j,i_j}^{(r)}, \quad (i_1,\ldots,i_D) \in \mathcal{V}_{\boldsymbol{B}} = \otimes_{j=1}^{D}\{1,\ldots,p_j\}. \tag{6}$$

The model is therefore nonlinear in the parameters defining $\boldsymbol{B}$. A hierarchical specification is completed by placing priors over unknown model parameters. While placing priors over α and $\boldsymbol{\gamma}$ is straightforward, Section 3.2 focuses on specification of the prior over tensor parameters which is nontrivial and one of the main contributions of this work.

Under the assumed rank-R parafac decomposition for $\boldsymbol{B}$, model (5) requires estimating $p+2+R\sum_{j=1}^{D}p_j$ as opposed to $p+2+\prod_{j=1}^{D}p_j$ parameters for the unstructured vectorized (saturated) model. As we are interested in identifying geometric sub-regions of the tensor across which coefficients are not close to zero, with the remaining elements being very close to zero, one wonders whether such dramatic dimension reduction retains sufficient flexibility. Finally, we would like to accurately estimate coefficient values in these sub-regions. Consistent with our theoretical analysis in Section 4, extensive simulation studies in Section 6 confirm our ability to accomplish these goals.

2.3 Model Identifiability

From model (5) it is clear that only voxel-level coefficients are identified and not the individual tensor margins defining their product-sum given in (6). In the tensor setting, identifiability restrictions are understood in light of the following indeterminacies:

1. *Scale indeterminacy*: for each $r = 1,\ldots,R$, define $\boldsymbol{\lambda}_r = (\lambda_{1r},\ldots,\lambda_{Dr})$ such that $\prod_{j=1}^{D}\lambda_{jr} = 1$. Then replacing $\boldsymbol{\beta}_j^{(r)}$ by $\lambda_{jr}\boldsymbol{\beta}_j^{(r)}$ leaves the tensor parameter $\boldsymbol{B}$ unaltered.

2. *Permutation indeterminacy*: $\sum_{r=1}^{R}\circ_{j=1}^{D}\boldsymbol{\beta}_j^{(r)} = \sum_{r=1}^{R}\circ_{j=1}^{D}\boldsymbol{\beta}_j^{(P(r))}$ for any permutation $P(\cdot)$ of $\{1,2,\ldots,R\}$. In particular, this implies that $\circ_{j=1}^{D}\boldsymbol{\beta}_j^{(r)}$ are not identifiable for $r = 1,\ldots,R$.

3. *Orthogonal transformation indeterminacy* ($D = 2$ only): for any orthonormal matrix $\boldsymbol{O}$, one has $(\boldsymbol{\beta}_1^{(r)}\boldsymbol{O}) \circ (\boldsymbol{\beta}_2^{(r)}\boldsymbol{O}) = \boldsymbol{\beta}_1^{(r)} \otimes \boldsymbol{\beta}_2^{(r)}$.

For $D > 2$, imposing the following $(D-1)R$ constraints ensures identifiability of the margin parameters comprising the rank-R parafac decomposition:

$$\beta_{j,1}^{(r)} = 1,\ 1 \le j < D,\ 1 \le r \le R, \qquad \text{and} \qquad \beta_{D,1}^{(1)} > \cdots > \beta_{D,1}^{(R)}. \tag{7}$$

In the context of Bayesian tensor regression, our emphasis is on accurate estimation and inference on tensor parameter $\boldsymbol{B}$, and on state-of-the-art predictive performance. Importantly, $\boldsymbol{B}$ is always identifiable even if the tensor margins $\boldsymbol{\beta}_j^{(r)}$'s are not, hence we avoid imposing identifiability restrictions on the latter. As is evident from our simulation studies, non-identifiability of $\boldsymbol{\beta}_j^{(r)}$ does not appear to cause convergence issues and in-fact, simplifies the design of efficient computational algorithms.

3. Multiway Shrinkage Priors

This Section outlines the novel multiway shrinkage prior on the tensor coefficient.

3.1 Vector Shrinkage Priors

There has been recent interest in high-dimensional regression with vector predictors, choosing priors which shrink small coefficients towards zero while minimizing shrinkage of large coefficients. Many of these priors can be expressed as a global-local (GL) scale mixtures (Polson and Scott, 2012) with

$$\theta_j \sim \mathrm{N}(0,\psi_j\tau), \quad \psi_j \sim g, \quad \tau \sim h, \tag{8}$$

where $(\theta_1,\ldots,\theta_p)$ is a coefficient vector, τ is a global scale and ψ_j is a local-scale. When g is a mixture of two components, with one concentrated near zero and the other away from zero, a spike and slab prior is obtained. Many other choices of g and h have been considered. Although the GL family is widely used and versatile, Bhattacharya et al. (2015) note advantages in drawing the local scales jointly. In particular, they propose to let

$$\theta_j \sim \mathrm{DE}(\cdot|\phi_j\tau), \quad (\phi_1,...,\phi_p) \sim \mathrm{Dirichlet}(a,\ldots,a), \quad \tau \sim h.$$

where $\mathrm{DE}(\cdot)$ denote the double-exponential distribution. For small a and large p, the $\mathrm{Dirichlet}(a,\ldots,a)$ prior has the property of favoring many values close to zero with a few much larger values, but with $\sum_j \phi_j = 1$. Though we draw motivation from literature on vector shrinkage priors, our goal of proposing a shrinkage prior on tensor parameter $\boldsymbol{B}$ is fundamentally more challenging as discussed in forthcoming sections.

3.2 Multiway Priors

We propose a new class of multiway shrinkage priors in the generalized linear model setting with tensor valued predictors. Assuming tensor parameter $\boldsymbol{B}$ admits a rank-R parafac decomposition, model (5) results in cell-level coefficients that are a nonlinear function of the corresponding tensor margin parameters (see (6)). Moreover, this implies simultaneous shrinkage on each of the $\prod_{j=1}^{D}p_j$ cell coefficients as imposed by the prior over $R\sum_{j=1}^{D}p_j$ parameters. This necessitates careful prior specification on the tensor margins $\{\boldsymbol{\beta}_j^{(r)}; 1 \le j \le D, 1 \le r \le R\}$ such that the induced cell-level prior has adequate tails so as to prevent over shrinkage.

There are a number of desirable characteristics for a multiway prior on the tensor margins. The proposed multiway shrinkage prior must have a structure that facilitates efficient and reliable model fitting. In addition, it is important to ensure that

1. For each $r = 1,\ldots,R$, $(\beta_{1,i_1}^{(r)},\ldots,\beta_{D,i_D}^{(r)})$ and $(\beta_{1,k_1}^{(r)},\ldots,\beta_{D,k_D}^{(r)})$ are equal in distribution, for any $(i_1,\ldots,i_D),(k_1,\ldots,k_D) \in \mathcal{V}_{\boldsymbol{B}} \times \mathcal{V}_{\boldsymbol{B}}$ and $(i_1,\ldots,i_D) \neq (k_1,\ldots,k_D)$. This is to ensure that $(B)_{i_1,\ldots,i_D}$ and $(B)_{k_1,\ldots,k_D}$ have the same distribution apriori.

2. Shrinkage towards a low rank decomposition, with the model adapting to the complexity and signal in the data, effectively deleting unnecessary dimensions.

3. The prior should favor recovery of contiguous geometric subregions of the tensor across which the cell observations are predictive of the response.

3.3 The Multiway Dirichlet GDP Prior

There are many ways of specifying priors over tensor margins $\boldsymbol{\beta}_j^{(r)}$ to satisfy the criteria listed. We propose a particular choice called the *multiway Dirichlet generalized double Pareto* (M-DGDP) prior. This prior induces shrinkage across components in an exchangeable way, with global scale $\tau \sim \text{Ga}(a_\tau, b_\tau)$ adjusted in each component as $\tau_r = \phi_r \tau$ for $r = 1, \ldots, R$, where $\Phi = (\phi_1, \ldots, \phi_R) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_R)$ encourages shrinkage towards lower ranks in the assumed parafac decomposition. In addition, $\boldsymbol{W}_{jr} = \text{diag}(w_{jr,1}, \ldots, w_{jr,p_j})$, $j = 1, \ldots, D$ and $r = 1, \ldots, R$ are margin-specific scale parameters for each component. The hierarchical margin-level prior is given by

$$\boldsymbol{\beta}_j^{(r)} \sim \text{N}\big(0, (\phi_r \tau)\boldsymbol{W}_{jr}\big), \quad w_{jr,k} \sim \text{Exp}(\lambda_{jr}^2/2), \quad \lambda_{jr} \sim \text{Ga}(a_\lambda, b_\lambda). \tag{9}$$

Collapsing over element-specific scales, notice that $\beta_{j,k}^{(r)} | \lambda_{jr}, \phi_r, \tau \overset{\text{iid}}{\sim} \text{DE}(\lambda_{jr}/\sqrt{\phi_r \tau})$, $1 \le k \le p_j$. Prior (9) induces a GDP prior on the individual margin coefficients which in turn has the form of an adaptive Lasso penalty (Armagan et al., 2013a). Flexibility in estimating $\boldsymbol{B}_r = \{\boldsymbol{\beta}_j^{(r)}; 1 \le j \le D\}$ is accommodated by modeling within-margin heterogeneity via element-specific scaling $w_{jr,k}$. Common rate parameter λ_{jr} shares information between margin elements, encouraging shrinkage at the local scale.

The framework adpoted by Zhou et al. (2013) for Frequentist tensor regression starts by assuming the true parafac rank and relying on shrinkage through a global parameter for estimating the tensor coefficient. In contrast, M-DGDP prior proposes joint shrinkage on the global and local component parameters to achieve improved inference and estimation. Though rank-selection is not our purview, our prior also accomodates dimension reduction by favoring low-rank factorizations as discussed below.

4. Posterior Consistency for Tensor Regression

This Section details out theoretical properties of the tensor regression framework.

4.1 Notation and Framework

We establish convergence results for tensor regression model (5) under the simplifying assumptions that the intercept is omitted by centering the response and the error variance is $\sigma^2 = 1$. Since our main focus is on the tensor coefficient, we assume coefficients for ordinary scalar covariates to be known. Without loss of generality, we assume $\boldsymbol{\gamma} = (0, \ldots, 0)$. We consider an asymptotic setting in which the dimensions of the tensor grow with n. This paradigm attempts to capture the fact that tensor dimension $\prod_j p_{j,n}$ is typically substantially larger than sample size. This creates theoretical challenges, related to (but distinct from) those faced in showing posterior consistency for high dimensional regression (Armagan et al., 2013b) and multiway contingency tables (Zhou et al., 2015).

Suppose the data generating model comes from the same class of models where the fitted model belongs to, i.e., having true tensor parameter $\boldsymbol{B}_n^0 \in \otimes_{j=1}^D \Re^{p_j, n}$, error variance $\sigma_0^2 = 1$. We also assume that the true tensor coefficient $\boldsymbol{B}_n^0$ generating the data admits a

rank-R PARAFAC decomposition as below

$$\boldsymbol{B}_n^0 = \sum_{r=1}^R \boldsymbol{\beta}_{1,n}^{0(r)} \circ \cdots \circ \boldsymbol{\beta}_{D,n}^{0(r)}, \qquad \boldsymbol{\beta}_{j,n}^{0(r)} = (\beta_{j,n,1}^{0(r)}, \ldots, \beta_{j,n,p_{j,n}}^{0(r)})' \in \Re^{p_{j,n}}.$$

In addition, define $\boldsymbol{F}_n, \boldsymbol{F}_n^0 \in \Re^{R \sum_{j=1}^D p_{j,n}}$ as the vectorized parameters:

$$\boldsymbol{F}_n = \text{vec}\big(\boldsymbol{\beta}_{1,n}^{(1)}, \cdots, \boldsymbol{\beta}_{1,n}^{(R)}, \cdots, \boldsymbol{\beta}_{D,n}^{(1)}, \cdots, \boldsymbol{\beta}_{D,n}^{(R)}\big)$$
$$\boldsymbol{F}_n^0 = \text{vec}\big(\boldsymbol{\beta}_{1,n}^{0(1)}, \cdots, \boldsymbol{\beta}_{1,n}^{0(R)}, \cdots, \boldsymbol{\beta}_{D,n}^{0(1)}, \cdots, \boldsymbol{\beta}_{D,n}^{0(R)}\big).$$

Define a Kulback-Leibler (KL) neighborhood around the true tensor $\boldsymbol{B}_n^0$ as

$$\mathcal{B}_n = \left\{ \boldsymbol{B}_n : \frac{1}{n} \sum_{i=1}^n \text{KL}(f(y_i|\boldsymbol{B}_n^0), f(y_i|\boldsymbol{B}_n)) < \epsilon \right\}.$$

Denote $\text{KL}(f(y_i|\boldsymbol{B}_n^0), f(y_i|\boldsymbol{B}_n))$ as KL_i. The KL-distance between $\text{N}(\mu_1, \sigma_1^2)$ and $\text{N}(\mu_2, \sigma_2^2)$ is $\log(\sigma_2/\sigma_1) + \big(\sigma_1^2 + (\mu_1 - \mu_2)^2\big)/2\sigma_2^2 - \frac{1}{2}$, so it follows $\text{KL}_i = \text{KL}\left(\text{N}\big(\langle \boldsymbol{X}_i, \boldsymbol{B}_n \rangle, 1\big), \text{N}\big(\langle \boldsymbol{X}_i, \boldsymbol{B}_n^0 \rangle, 1\big)\right)$ $= \frac{1}{2}\left(\langle \boldsymbol{X}_i, \boldsymbol{B}_n^0 \rangle - \langle \boldsymbol{X}_i, \boldsymbol{B}_n \rangle\right)^2$. Hence, a KL neighborhood of radius ϵ around $\boldsymbol{B}_n^0$ can be re-expressed as $\mathcal{B}_n = \left\{ \boldsymbol{B}_n : \frac{1}{2n} \sum_{i=1}^n \left(\langle \boldsymbol{X}_i, \boldsymbol{B}_n^0 \rangle - \langle \boldsymbol{X}_i, \boldsymbol{B}_n \rangle\right)^2 < \epsilon \right\}$. Further, let π_n and Π_n denote prior and posterior densities with n observations, respectively, and

$$\Pi_n(\mathcal{B}_n^c) = \frac{\int_{\mathcal{B}_n^c} f(\boldsymbol{y}_n|\boldsymbol{B}_n)\pi_n(\boldsymbol{F}_n)}{\int f(\boldsymbol{y}_n|\boldsymbol{B}_n)\pi_n(\boldsymbol{F}_n)},$$

with $\boldsymbol{y}_n = (y_1, \ldots, y_n)'$ and $f(\boldsymbol{y}_n|\boldsymbol{B}_n)$ is the density of $\boldsymbol{y}_n$ under model (5). Posterior consistency is established by showing that

$$\Pi_n\left(\mathcal{B}_n^c\right) \to 0 \quad \text{under } \boldsymbol{B}_n^0 \quad \text{a.s. as } n \to \infty. \tag{10}$$

4.2 Main Result

Our main theorem is that (10) holds under a simple sufficient condition on the prior and the tensor predictors.

Theorem 1 *Let* $\zeta_n = n^{\frac{1+\rho_3}{2}}$ $(\rho_3 > 0)$, $M_n = \frac{1}{n}\sqrt{\sum_{i=1}^n ||\boldsymbol{X}_i||_2^2}$. *Given Lemma 6 in Appendix A, for any* $\epsilon > 0$, $\Pi_n(\boldsymbol{B}_n : \frac{1}{n}\sum_{i=1}^n KL_i > \epsilon) \to 0$ *a.s. under* $\boldsymbol{B}_n^0$, *for the prior* $\pi_n(\boldsymbol{B}_n)$ *that satisfies*

$$\pi_n\left(\boldsymbol{B}_n : ||\boldsymbol{B}_n - \boldsymbol{B}_n^0||_2 < \frac{2\eta}{3M_n\zeta_n}\right) > \exp(-dn), \ \textit{for all large } n \tag{11}$$

for any $d > 0$ *and* $\eta < \frac{\epsilon}{32} - d$. *That is, the model is posterior consistent when* (11) *holds.*

Lemma 6 in Appendix A verifies the existence of exponentially-consistent tests. The proof of the Lemma and Theorem are provided in the Appendix A. The proposed multiway shrinkage prior satisfies (11) and hence leads to posterior consistency under the following Theorem.

Theorem 2 *For fixed constants H_1, H_2, M_1, ρ_1 and $\rho_2 > 0$, the M-DGDP prior (9) on $\boldsymbol{B}_n$ satisfies (11), i.e. yields posterior consistency under conditions:*

(a) $H_1 n^{\rho_1} < M_n < H_2 n^{\rho_2}$

(b) $\sup_{l=1,\ldots,p_{j,n}} |\boldsymbol{\beta}^{0(r)}_{j,n,l}| < M_1 < \infty$, *for all* $j = 1, \ldots, D$, $r = 1, \ldots, R$

(c) $\sum_{j=1}^{D} p_{j,n} \log(p_{j,n}) = o(n)$.

Remark 3 *Condition (a) in Theorem 2 gives an upper and lower bound on the sum of the Frobenius norms of tensor predictors. In tensor predictors with $\{0,1\}$ entries (often observed for white/grey matter fMRI data), this condition simply imposes a restriction on the minimum and maximum number of 1s in a tensor predictor as a function of the sample size n. Condition (b) is mild, assuming the supremum of all entries in the tensor margins are bounded. Finally, condition (c) in Theorem 2 requires that $\sum_{j=D} p_{j,n}$ grows sub-linearly with sample size n. However, note that the number of cells $\prod_{j=1}^{D} p_{j,n}$ in the tensor can grow at a rate much faster than the sample size n; hence, the modeling framework allows large tensor covariates even for moderate sample sizes. Of course, condition (c) trivially holds when the dimension of the tensor is kept fixed as n grows.*

Remark 4 *Our Section on posterior consistency is based on the assumption that $\boldsymbol{\gamma} = (0, \ldots, 0)'$ and error variance 1, however the results are trivially extendable to cases with unknown $\boldsymbol{\gamma}$ and σ^2. In such a generalization, it is important to note that one would need to assume the dim($\boldsymbol{\gamma}$) is fixed and not growing with n.*

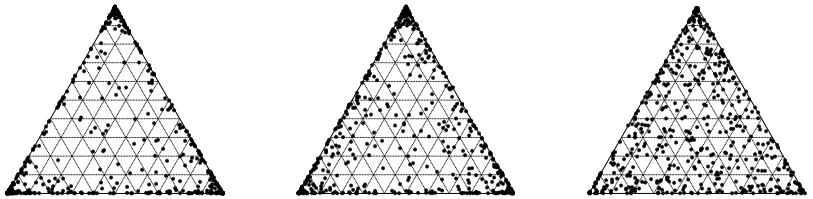
4.3 Prior Hyper-parameter Elicitation

The marginal distribution of cell coefficients (6) under the proposed M-DGDP prior (9) is not available in closed-form. To assess how a shrinkage prior on the margins induces prior on cell coefficients of the tensor, we turn to an expression for the cell-level variance:

$$\begin{aligned}
\text{var}(\boldsymbol{B}_{i_1,\ldots,i_D}) &= \mathbb{E}\Big(\text{var}\Big\{ \sum_{r=1}^{R} \prod_{j=1}^{D} \boldsymbol{\beta}^{(r)}_{j,i_d} \,|\, \boldsymbol{W}, \Lambda, \Phi, \tau \Big\} \Big) \\
&= \mathbb{E}_{\Phi}\Big(\sum_{r=1}^{R} \phi_r^D \, \mathbb{E}_{\tau}\{\tau^D\} \, \mathbb{E}_{\Lambda_{\cdot,r}}\Big\{ \mathbb{E}_{\boldsymbol{W}_r|\boldsymbol{\Lambda}_r}\Big(\prod_{j=1}^{D} w_{jr,i_j} \Big) \Big\} \Big) \\
&= \frac{\Gamma(\alpha_0 + D)}{\Gamma(\alpha_0)\, b_\tau^D} (2C_\lambda)^D \, \mathbb{E}_{\Phi}\Big(\sum_{r=1}^{R} \phi_r^D \Big).
\end{aligned}$$

The following Lemma provides lower and upper bounds on the variance that can be useful for elicitation of default hyperparameter values.

Lemma 5 *Under M-DGDP shrinkage prior (9) and for $D > 1$, if $\alpha_1 = \cdots = \alpha_R = c/R$, $c \in \mathbb{N}_+$, with constants $C_\lambda = b_\lambda^2/\big((a_\lambda - 1)(a_\lambda - 2)\big)$, $a_\lambda > 2$, $A_\tau = \exp((D^2 - 3D)/2)$, then the cell-level variance is bounded below by $R\alpha_1^D (2C_\lambda/b_\tau)^D$ and above by $A_\tau (2C_\lambda/b_\tau)^D \exp(\alpha_1 RD)$.*



(a) $\boldsymbol{\alpha} = (0.2, 0.2, 0.2)$ (b) $\boldsymbol{\alpha} = (0.3, 0.3, 0.3)$ (c) $\boldsymbol{\alpha} = (0.5, 0.5, 0.5)$

Figure 1: Visualization of points in the $\mathcal{S}^2$ probability simplex for 500 independent realizations of Dirichlet($\boldsymbol{\alpha}$). As $\alpha \downarrow 0$, points increasingly tend to concentrate around vertices of the $\mathcal{S}^{R-1}$ simplex. This notion of sparsity is made precise in Yang and Dunson (2014).

Hyperparameters of the Dirichlet component in multiway prior (9) play a key role in controlling dimensionality of the model, with smaller values favoring more component-specific scales $\tau_r \approx 0$, thus effectively collapsing on a low-rank tensor factorization. Figure 1 plots realizations from the Dirichlet distribution when $R = 3$ for different concentration parameters[2] α.

A discrete uniform prior is placed on α over a grid, $\mathcal{A}$. By default, grid values are chosen to be 10 equally spaced values in $[R^{-D}, R^{-0.10}]$, letting the data tune this parameter according to the degree of sparisty present. Armagan et al. (2013a) study various choices of $(a_\lambda, \zeta = b_\lambda/a_\lambda)$ that lead to desirable shrinkage properties, such as Cauchy-like tails for $\beta^{(r)}_{j,k}$ while retaining Laplace-like shrinkage near zero. Empirical results from simulation studies across a variety of settings in Section 6 reveal no strong sensitivity to choices for hyper-parameters a_λ, b_λ. From Lemma 5, setting $a_\lambda = 3$ and $b_\lambda = \sqrt[2D]{a_\lambda}$ avoids overly narrow variance of the induced prior on tensor elements, $\boldsymbol{B}_{i_1,\ldots,i_D}$. Table 1 provides various quantiles of the induced prior on these elements under these default hyperparameter settings as a function of the parafac rank-R and tensor dimension D.

5. Posterior Computation and Model Fitting

Letting $y \in \Re$ denote a response, and $\boldsymbol{z} \in \Re^p, \boldsymbol{X} \in \otimes_{j=1}^{D} \Re^{p_j}$ predictors, we let

$$\begin{aligned}
& y|\boldsymbol{\gamma}, \boldsymbol{B}, \sigma \sim \text{N}\big(\boldsymbol{z}'\boldsymbol{\gamma} + \langle \boldsymbol{X}, \boldsymbol{B} \rangle, \sigma^2\big) \\
& \boldsymbol{B} = \sum_{r=1}^{R} \boldsymbol{B}_r, \;\; \boldsymbol{B}_r = \boldsymbol{\beta}_1^{(r)} \circ \cdots \circ \boldsymbol{\beta}_D^{(r)} \\
& \sigma^2 \sim \pi_\sigma, \;\; \boldsymbol{\gamma} \sim \pi_\gamma, \;\; \boldsymbol{\beta}_j^{(r)} \sim \pi_{\boldsymbol{\beta}}.
\end{aligned} \tag{12}$$

2. For simplicity we assume $\alpha_1 = \cdots = \alpha_R = \alpha$.

	R	5%	25%	50%	75%	95%
$D=2$	1	0.001	0.011	0.057	0.254	1.729
	5	0.004	0.040	0.164	0.595	3.332
	10	0.005	0.058	0.237	0.852	4.635
$D=3$	1	0.000	0.001	0.010	0.072	0.917
	5	0.000	0.009	0.061	0.341	3.382
	10	0.001	0.017	0.111	0.608	5.996

Table 1: Percentiles for $|\boldsymbol{B}_{i_1,\ldots,i_D}|$ under the M-DGDP prior with default $a_\lambda = 3$, $b_\lambda = \sqrt[2D]{a_\lambda}$, $b_\tau = \alpha R^{1/D}\,(v=1)$ and $\alpha = 1/R$. Statistics are displayed as the parafac rank-R vary and dimension D of the tensor vary.

The noise variance is modeled using a conjugate inverse-gamma prior, $\sigma^2 \sim \text{IG}(v/2, vs_0^2/2)$, with $v=2$ and s_0^2 chosen by default so $\Pr(\sigma^2 \leq 1) = 0.95$ assuming a centered and scaled response. Regression coefficients are given a conjugate normal prior $\boldsymbol{\gamma} \sim \text{N}(0, \sigma^2\boldsymbol{\Sigma}_{0\boldsymbol{\gamma}})$ and the tensor predictor is normalized over all cells to have mean zero and variance 1, allowing one to assume default values for hyper-parameters in the proposed multiway prior.

5.1 Posterior Computation

The proposed multiway prior (9) leads to Gibbs sampling scheme for most parameters of the tensor regression model (12). We rely on marginalization and blocking to reduce auto-correlation for $\big\{(\boldsymbol{\beta}_j^{(r)}, w_{jr}; 1 \leq j \leq D, 1 \leq r \leq R), (\Phi, \tau), (\boldsymbol{\gamma}, \sigma)\big\}$, drawing in sequence from $[\alpha, \Phi, \tau|\boldsymbol{B}, W]$, $[\boldsymbol{B}, W|\Phi, \tau, \boldsymbol{\gamma}, \sigma, \boldsymbol{y}]$ and $[\boldsymbol{\gamma}, \sigma|\boldsymbol{B}, \boldsymbol{y}]$ as follows:

(1) Sample $[\alpha, \Phi, \tau|\boldsymbol{B}, \boldsymbol{W}]$ compositionally as $[\alpha|\boldsymbol{B}, \boldsymbol{W}][\Phi, \tau|\alpha, \boldsymbol{B}, \boldsymbol{W}]$:

(a) Sample from the conditional distribution of Dirichlet concentration parameter $[\alpha|\boldsymbol{B}, \boldsymbol{W}]$ via griddy-Gibbs: form a reference set by drawing M samples from $[\Phi, \tau|\alpha, \boldsymbol{B}, \boldsymbol{W}]$ for each $\alpha \in \mathcal{A}$. Set $w_{j,l} = \pi(\boldsymbol{B}|\alpha, \Phi_l, \tau_l, \boldsymbol{W})\pi(\Phi_l, \tau_l|\alpha)$, $1 \leq l \leq M$, $p(\alpha|\boldsymbol{B}, \boldsymbol{W}) = \pi(\alpha)\sum_{l=1}^M w_{j,l}/M$, and $\Pr(\alpha = \alpha_j|-) = p(\alpha_j|\boldsymbol{B}, \boldsymbol{W})/\sum_{\alpha\in\mathcal{A}} p(\alpha|\boldsymbol{B}, \boldsymbol{W})$.

(b) Sample component-specific scales as $[\Phi, \tau|\alpha^*, \boldsymbol{B}, \boldsymbol{W}] = [\Phi|\boldsymbol{B}, \boldsymbol{W}][\tau|\Phi, \boldsymbol{B}, \boldsymbol{W}]$; define $p_0 = \sum_{j=1}^D p_j$, and recall $a_\tau = \sum_{r=1}^R \alpha_r = R\alpha$ and $b_\tau = \alpha(R/v)^{1/D}$ (see Section 3.3), then

- draw $\psi_r \sim \text{giG}(\alpha - p_0/2, 2b_\tau, 2C_r)$, $C_r = \sum_{j=1}^D \boldsymbol{\beta}_j^{(r)T}\boldsymbol{W}_{jr}^{-1}\boldsymbol{\beta}_j^{(r)}$, and set $\phi_r = \psi_r/\sum_{l=1}^R \psi_l$ in parallel for $1 \leq r \leq R$ (see Appendix A for definition of 'giG')
- draw $\tau \sim \text{giG}(a_\tau - Rp_0/2, 2b_\tau, 2\sum_{r=1}^R D_r)$, $D_r = C_r/\phi_r$.

(2) Sample from $\big\{(\boldsymbol{\beta}_j^{(r)}, w_{jr}, \lambda_{jr}); 1 \leq j \leq D, 1 \leq r \leq R\big\}|\Phi, \tau, \boldsymbol{\gamma}, \sigma, \boldsymbol{y}$ using a back-fitting procedure to produce a sequence of draws from the margin-level conditional distributions across components. For $r = 1, \ldots, R$ and $j = 1, \ldots, D$, sample from conditional distribution $[(\boldsymbol{\beta}_j^{(r)}, w_{jr}, \lambda_{jr})|\boldsymbol{\beta}_{-j}^{(r)}, \boldsymbol{B}_{-r}, \Phi, \tau, \boldsymbol{\gamma}, \sigma, \boldsymbol{y}]$, where $\boldsymbol{\beta}_{-j}^{(r)} = \{\boldsymbol{\beta}_l^{(r)}, l \neq j\}$ and $\boldsymbol{B}_{-r} = \boldsymbol{B} \setminus \boldsymbol{B}_r$;

(a) draw $[w_{jr}, \lambda_{jr}|\boldsymbol{\beta}_j^{(r)}, \phi_r, \tau] = [w_{jr}|\lambda_{jr}, \boldsymbol{\beta}_j^{(r)}, \phi_r, \tau][\lambda_{jr}|\boldsymbol{\beta}_j^{(r)}, \phi_r, \tau]$:

- draw $\lambda_{jr} \sim \text{Ga}\big(a_\lambda + p_j, b_\lambda + ||\boldsymbol{\beta}_j^{(r)}||_1/\sqrt{\phi_r\tau}\big)$; and
- draw $w_{jr,k} \sim \text{giG}\big(\frac{1}{2}, \lambda_{jr}^2, \beta_{j,k}^{2\,(r)}/(\phi_r\tau)\big)$ independently for $1 \leq k \leq p_j$

(b) draw $\boldsymbol{\beta}_j^{(r)} \sim \text{N}(\boldsymbol{\mu}_{jr}, \boldsymbol{\Sigma}_{jr})$: define $h_{i,j,k}^{(r)} = \sum_{d_1=1,\ldots,d_D=1}^{p_1,\ldots,p_D} I(d_j = k)\, x_{d_1,\ldots,d_D}\big(\prod_{l\neq j}\beta_{l,i_l}^{(r)}\big)$, $\boldsymbol{H}_{i,j}^{(r)} = (h_{i,j,1}^{(r)}, \ldots, h_{i,j,p_j}^{(r)})'$, $\tilde{y}_i = y_i - \boldsymbol{z}_i'\boldsymbol{\gamma} - \sum_{l\neq r}\langle \boldsymbol{X}_i, \boldsymbol{B}_l\rangle$ for $1 \leq i \leq n$; then $\boldsymbol{\Sigma}_{jr} = \big(\boldsymbol{H}_j^{(r)T}\boldsymbol{H}_j^{(r)}/\sigma^2 + \boldsymbol{W}_{jr}^{-1}/(\phi_r\tau)\big)^{-1}$, $\boldsymbol{\mu}_{jr} = \boldsymbol{\Sigma}_{jr}\boldsymbol{H}_j^{(r)}\tilde{\boldsymbol{y}}/\sigma^2$

(3) Sample $[\boldsymbol{\gamma}, \sigma|\boldsymbol{B}, \boldsymbol{y}] = [\boldsymbol{\gamma}|\sigma, \tilde{\boldsymbol{y}}][\sigma^2|\tilde{\boldsymbol{y}}]$; define $\tilde{y}_i = y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}\rangle$ for $1 \leq i \leq n$, then

(a) draw $\sigma^2 \sim \text{IG}(a_\sigma, b_\sigma)$, $a_\sigma = (n+v)/2$, $b_\sigma = \big(vs_0^2 + ||\tilde{\boldsymbol{y}}||_2^2 - \tilde{\boldsymbol{y}}^T\boldsymbol{Z}\boldsymbol{\mu}_{\boldsymbol{\gamma}}\big)/2$

(b) draw $\boldsymbol{\gamma} \sim \text{N}\big(\boldsymbol{\mu}_{\boldsymbol{\gamma}}, \sigma^2\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}\big)$, $\boldsymbol{\Sigma}_{\boldsymbol{\gamma}} = \big(\boldsymbol{Z}^T\boldsymbol{Z} + \boldsymbol{\Sigma}_{0\boldsymbol{\gamma}}^{-1}\big)^{-1}$, $\boldsymbol{\mu}_{\boldsymbol{\gamma}} = \boldsymbol{\Sigma}_{\boldsymbol{\gamma}}\boldsymbol{Z}^T\tilde{\boldsymbol{y}}$.

6. Simulation Studies

To illustrate finite-sample performance of the proposed multiway priors, we show results from a simulation study with various dimensionality (p, R) and define $\bar{b} = \max|\boldsymbol{B}^0_{i_1,\ldots,i_D}|$ as the maximum signal size. Throughout, set $p_j = p$, true error variance $\sigma_0^2 = 1$ and $\bar{b} = 1$ for convenience. In addition, we set the true vector coefficient $\boldsymbol{\gamma}_0 = (0, \ldots, 0)$ and focus exclusively on inference for tensor parameter $\boldsymbol{B}$. The following simulated setups are considered:

1. "Generated" tensor: We construct tensor parameters having rank $R_0 = \{3, 5\}$ with $p = \{64, 100\}$ and $D = 2$.

2. "Ready made" tensor: We use three tensor (2D) images without generating them from a parafac decomposition with known rank.

Five replicated datasets with $n = 1000$ are generated according to (12) with $x_{i_1,\ldots,i_D} \sim \text{N}(0, 1)$. The tensor parameters considered are shown in Figure 2, where the magnitude of the non-zero cells is $\bar{b} = 1$. Examples are chosen to demonstrate recovery of cell-level coefficients across varying degrees of complexity (dimension, parafac rank) and sparsity (% of non-zero cells; see Figure 2). The performance of our method with M-DGDP prior (9) (BTR) is compared with (i) frequentist tensor regression with penalization (FTR)(Zhou et al., 2013); and (ii) Lasso (on the vectorized tensor predictor). Comparisons are based on (a) cell mean squared estimation error (true non-zero, true zero, and overall); and (b) frequentist coverage (and length) of 95% credible intervals.

By default, BTR uses $R = 10$ as an upper bound on the tensor parafac rank, minimizing effects of extra dimensions automatically and concentrating on a lower rank coefficient tensor as MCMC proceeds. We ran FTR with various choices of R and found equivalent performance for R between 3 to 15, thus the default value is set to $R = 10$ to ensure comparability with the BTR fitting. MCMC for BTR was run for 1300 iterations, with a 300 iteration burn-in and remaining samples thinned by 5. The latter was chosen to keep runtime between BTR and FTR similar for 3D simulation studies in Section 7. There, the total runtime using non-optimized R code on an x86×64 Intel(R) Core(TM) i7-3770 is between 6.2 - 7.5 hours. In simulation studies, the tuning parameter in FTR is selected

		R3-ex	R5-ex	Shapes	Eagle	Palmtree	Horse
$\|\text{cell}_0\| > 0$	BTR	$\mathbf{0.023}_{0.00}$	$\mathbf{0.021}_{0.00}$	$\mathbf{0.243}_{0.01}$	$\mathbf{0.226}_{0.02}$	$\mathbf{0.316}_{0.01}$	$\mathbf{0.278}_{0.01}$
	FTR	$0.035_{0.00}$	$0.030_{0.00}$	$0.415_{0.03}$	$0.354_{0.03}$	$0.435_{0.02}$	$0.391_{0.03}$
	Lasso	$0.628_{0.02}$	$0.822_{0.03}$	$0.619_{0.07}$	$0.665_{0.03}$	$0.698_{0.03}$	$0.888_{0.01}$
$\|\text{cell}_0\| = 0$	BTR	$\mathbf{0.011}_{0.00}$	$\mathbf{0.014}_{0.00}$	$\mathbf{0.071}_{0.00}$	$\mathbf{0.085}_{0.00}$	$\mathbf{0.100}_{0.01}$	$\mathbf{0.137}_{0.00}$
	FTR	$0.022_{0.00}$	$0.020_{0.00}$	$0.127_{0.02}$	$0.163_{0.03}$	$0.159_{0.00}$	$0.215_{0.02}$
	Lasso	$0.090_{0.00}$	$0.098_{0.02}$	$0.081_{0.01}$	$0.097_{0.00}$	$0.094_{0.01}$	$0.155_{0.02}$
Overall	BTR	**0.013**	**0.015**	**0.093**	**0.102**	**0.131**	**0.172**
	FTR	0.023	0.021	0.164	0.184	0.196	0.257
	Lasso	0.187	0.288	0.179	0.204	0.217	0.407

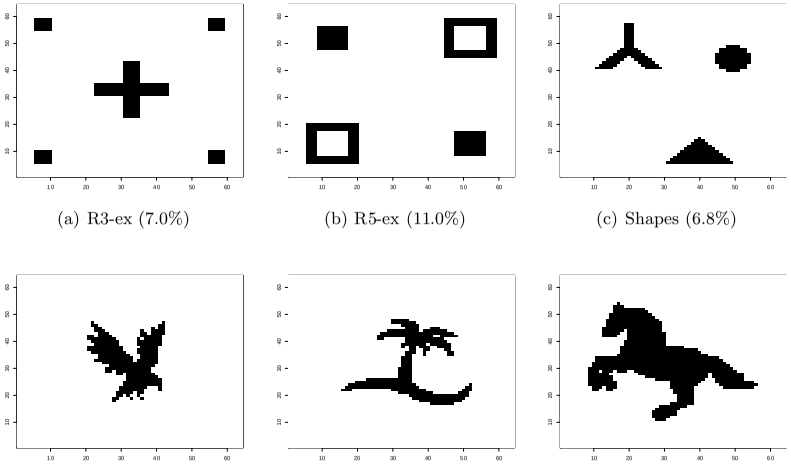
Table 2: Comparison of cell estimation as measured by root mean squared error (RMSE) for the six 2D tensor images portrayed in Figure 2. Results from FTR (Zhou et al., 2013) use $R = 10$. For BTR, $R = 10$ is used as an upper bound to the tensor parafac rank. Subscript shows the standard error over a few replicated simulations.

over a grid of values to minimize RMSE for the tensor predictor[3]. In real applications, cross validation instead would be used to select the tuning parameter that results in lowest hold-out predictive RMSE for FTR. Assuming 10-fold cross validation were used over a vector of 20 tuning parameters, FTR would have a runtime of approximately 8 hours. It also needs to be mentioned that the convergence of parameters in BTR is extremely rapid with an average effective sample size (ESS) ≈ 600 over 1000 iterations.

Cell-level RMSE reported in Table 2 demonstrates that our method (BTR) consistently out performs FTR. When the tensor parameter has a low-rank parafac decomposition ('R3-ex' and 'R5-ex'), BTR and FTR perform best, with BTR having lower RMSE on both true zero and non-zero cells. This validates empirically prior (9) along with our suggested default hyper-parameter choices in Section 3. In particular, the tensor coefficient in BTR has three different types of shrinkage parameters: global, local and shrinkage across ranks. Such a careful construction of shrinkage prior on $\boldsymbol{B}$ adapts to varying degrees of sparsity, shrinking many tensor coefficients close to zero while accurately estimating nonzero cells. FTR shrinkage being dependent on only local parameters suffers in terms of both inferential and predictive performances.

Table 3 demonstrate that BTR yields 95% credible intervals with good frequentist coverage across each of the simulated settings, both overall as well as on the true non-zero coefficients. Our method is one of the first to offer uncertainty quantification for tensor valued predictors.Finally, Table 4 provides evidence of the robustness of our method to increasing predictor dimension using two of the simulated examples. In both cases, RMSE for FTR worsens considerably on the true zero coefficients. For the true nonzero cells, RMSE increases for both methods as the margin dimension increases; however on a relative basis, FTR worsens considerably more, while on an absolute scale, BTR remains the clear winner.

3. To choose initial values, a preliminary analysis was run with a coarsened 16×16 image.



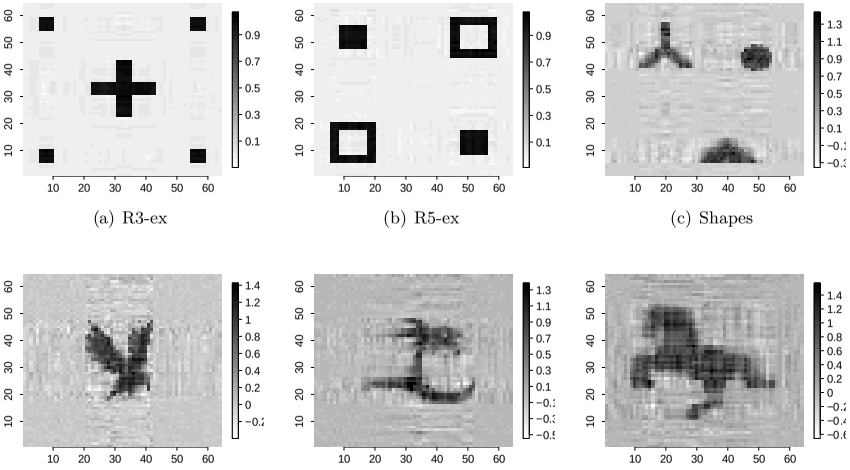
(a) R3-ex (7.0%) (b) R5-ex (11.0%) (c) Shapes (6.8%)

(d) Eagle (7.4%) (e) Palmtree (8.0%) (f) Horse (18.6%)

Figure 2: Simulated data with 64×64 2D tensor images ($p = 64$, $D = 2$). Row 1: The first two images (from left) have a rank-3 and rank-5 parafac decomposition; the third image is "regular", although does not have a low-rank parafac decomposition. Row 2: All three images are irregular, and do not have a low-rank parafac decomposition. Sparsity (% non-zero cells) are displayed in sub-captions.

		R3-ex	R5-ex	Shapes	Eagle	Palmtree	Horse
$\|\text{cell}_0\| > 0$	coverage	$0.986_{0.02}$	$0.946_{0.02}$	$0.747_{0.01}$	$0.731_{0.04}$	$0.677_{0.04}$	$0.795_{0.02}$
Overall	coverage	$0.995_{0.01}$	$0.970_{0.01}$	$0.965_{0.00}$	$0.940_{0.02}$	$0.948_{0.02}$	$0.927_{0.01}$
	length	$0.066_{0.01}$	$0.061_{0.01}$	$0.290_{0.00}$	$0.301_{0.03}$	$0.410_{0.03}$	$0.566_{0.02}$

Table 3: Row 1: Average coverage of 95% posterior credible intervals for all the cells of $\boldsymbol{B}$ for which the true cell coefficient is nonzero. Row 2: Average coverage of 95% posterior credible intervals for all the cells of $\boldsymbol{B}$. Row 3: Average length of 95% posterior credible intervals for all the cells of $\boldsymbol{B}$. Subscripts show standard errors over replicated simulations.



(a) R3-ex (b) R5-ex (c) Shapes

(d) Eagle (e) Palmtree (f) Horse

Figure 3: Recovered images for the 64×64 2D tensor images in Figure 2 using our proposed BTR method. Here, $R = 10$ is used as an upper bound to the tensor parafac rank.

		$\lvert\text{cell}_0\rvert$	R5-ex		Shapes	
			64	100	64	100
BTR	coverage	> 0	$0.946_{0.02}$	$0.991_{0.01}$	$0.747_{0.01}$	$0.590_{0.06}$
	length	> 0	$0.061_{0.01}$	$0.069_{0.01}$	$0.290_{0.00}$	$0.247_{0.01}$
	rmse	> 0	$0.021_{0.00}$	$0.032_{0.01}$	$0.243_{0.01}$	$0.320_{0.03}$
	rmse	$= 0$	$0.014_{0.00}$	$0.014_{0.00}$	$0.071_{0.00}$	$0.063_{0.00}$
FTR	rmse	> 0	$0.030_{0.00}$	$0.369_{0.06}$	$0.415_{0.03}$	$0.586_{0.14}$
	rmse	$= 0$	$0.020_{0.00}$	$0.111_{0.02}$	$0.127_{0.02}$	$0.135_{0.02}$

Table 4: Sensitivity analysis of cell estimation error (RMSE) as the tensor dimension increases; here $p_j = p \in \{64, 100\}$ for the 2D tensor images 'R5-ex' and 'Shapes'.

7. Simulated response with a real 3D brain image

We analyze data containing 3D MRI images for 550 adolescents, with information such as age and sex available. Age and sex are treated as ordinary scalar covariates while 3D MRI images act as tensor covariates. Let $\boldsymbol{X}$ denote a $30\times30\times30$ 3D MRI image, Z_1 be the age and Z_2 be the sex of an individual. The response is simulated using $y \sim \text{N}\left(\boldsymbol{Z}'\boldsymbol{\gamma} + \langle \boldsymbol{X}, \boldsymbol{B}_0\rangle, \sigma^2\right)$, where $\boldsymbol{Z}$ denotes $(Z_1, Z_2)'$, $\boldsymbol{\gamma} \in \mathcal{R}^2$ and $\boldsymbol{B}_0 \in \mathcal{R}^{30\times30\times30}$.

We assume the true $\boldsymbol{B}_0$ is a rank 2 tensor, with $\boldsymbol{B}_0 = \boldsymbol{a}_1 \circ \boldsymbol{a}_2 \circ \boldsymbol{a}_3 + \boldsymbol{b}_1 \circ \boldsymbol{b}_2 \circ \boldsymbol{b}_3$. Initialization and standardization of predictors follow exactly as prescribed in Section 5. By varying $\boldsymbol{a}_i$'s and $\boldsymbol{b}_i$'s, the following cases with varying degrees of sparsity in the tensor parameter $\boldsymbol{B}_0$ are considered:
Case 1: $\boldsymbol{b}_1 = \boldsymbol{b}_2 = (0, \ldots, 0, \sin((1:15) * \pi/4))$, $\boldsymbol{b}_3 = (\sin((1:10) * \pi/4), 0, \ldots, 0)$,
$\boldsymbol{a}_1 = (0, \ldots, 0, \sin((1:10) * \pi/4))$, $\boldsymbol{a}_2 = (0, \ldots, 0, \cos((1:15) * \pi/4))$,
$\boldsymbol{a}_3 = (\sin((1:15) * \pi/4), 0, \ldots, 0)$.
Case 2: $\boldsymbol{b}_1 = \boldsymbol{b}_2 = (0, \ldots, 0, \sin((1:15) * \pi/6))$, $\boldsymbol{b}_3 = (\sin((1:20) * \pi/6), 0, \ldots, 0)$,
$\boldsymbol{a}_1 = (0, \ldots, 0, \sin((1:15) * \pi/4))$, $\boldsymbol{a}_2 = (0, \ldots, 0, \cos((1:10) * \pi/6))$,
$\boldsymbol{a}_3 = (\sin((1:15) * \pi/6), 0, \ldots, 0)$.
Case 3: $\boldsymbol{b}_1 = \boldsymbol{b}_2 = (0, \ldots, 0, \sin((1:20) * \pi/6))$, $\boldsymbol{b}_3 = (\sin((1:20) * \pi/6), 0, \ldots, 0)$,
$\boldsymbol{a}_1 = (0, \ldots, 0, \sin((1:10) * \pi/4))$, $\boldsymbol{a}_2 = (0, \ldots, 0, \cos((1:20) * \pi/4))$,
$\boldsymbol{a}_3 = (\sin((1:20) * \pi/6), 0, \ldots, 0)$.

We implement BTR, FTR, and Lasso on the vectorized tensor. As before, we present results for FTR with $R = 10$ (See additional discussion in Section 6 on FTR default setup). Point estimates for coefficients corresponding to age and sex covariates are provided in Table 6. Table 5 summarizes RMSEs for the estimated tensor coefficients for each method. BTR shows at least a 15% improvement over FTR on simulated cases considered. Evidently BTR tends to outperform FTR and vectorized lasso by a greater margin in less sparse settings as well. Importantly, every parameter in BTR is auto-tuned, while the `TensorReg` toolbox used for FTR (Zhou et al., 2013) requires calibrating tuning parameter values specific to each setting. Note that rather than using cross validation, tuning parameters in these experiments were chosen to provide the lowest possible (most optimistic) RMSE for the tensor coefficient. FTR fixes $R = 10$ based on findings discussed in Section 6 while BTR sets $R = 10$ as an upper bound, concentrating on a lower dimension parafac rank via adaptive shrinkage. While vectorized lasso and FTR do not come equipped with parameter uncertainty estimates, Table 7 demonstrates how BTR consistently provides over 95% coverage across examples with varying degrees of sparsity.

Finally, Table 8 provides a measure of mixing efficiency for a single MCMC run in each of the simulated cases considered (post burn-in over the remaining 1000 MCMC samples). All reported RMSE and coverage statistics were computed over these draws as well (thinning by 5 as previously discussed in Section 6).

8. Brain Connectome Data Analysis

Brain connectome data are known by neuroscientists to have low signal-to-noise ratio, and effective modeling is often hindered as the sample size is often very limited compared to the

		Case 1	Case 2	Case 3
$\|\text{cell}_0\| > 0$	BTR	**0.39**	**0.30**	**0.34**
	FTR	0.46	0.41	0.43
	Lasso	0.46	0.42	0.44
$\|\text{cell}_0\| = 0$	BTR	**0.04**	**0.14**	**0.10**
	FTR	0.00	0.00	0.00
	Lasso	0.01	0.03	0.02
Overall	BTR	**0.13**	**0.20**	**0.17**
	FTR	0.15	0.22	0.18
	Lasso	0.15	0.23	0.18

Table 5: Comparison of cell estimation as measured by root mean squared error (RMSE) for the coefficients in case 1,2, 3 corresponding to 3D tensor images. Results from both BTR and FTR (Zhou et al., 2013) use $R = 10$.

		Case 1	Case 2	Case 3
γ_1 (truth = 0.5)	BTR	**0.57**	**0.54**	**0.33**
	FTR	0.46	0.85	0.95
γ_2 (truth = 2.0)	BTR	**2.00**	**2.04**	**1.86**
	FTR	1.87	0.22	3.30

Table 6: Point estimates for age and sex coefficients under BTR and FTR Zhou et al. (2013). True parameter values are also provided.

	Case 1	Case 2	Case 3
Coverage	0.98	0.96	0.99
Length	0.54	0.87	2.16

Table 7: Length and coverage of 95% credible intervals for BTR. Values are reported as averages over all voxels of the tensor coefficient.

		τ^2	Scalar predictor γ_i (Ave.)	Tensor predictor β_{ijk} (Ave.)
Case 1 (88% sparsity)	lag-2	0.04	0.34	0.45
	lag-4	0.01	0.11	0.22
Case 2 (82% sparsity)	lag-2	0.08	0.33	0.46
	lag-4	0.03	0.12	0.23
Case 3 (70% sparsity)	lag-2	0.09	0.43	0.53
	lag-4	0.02	0.30	0.40

Table 8: MCMC autocorrelation of the proposed BTR method on data studies of Section 7 generated using 3D brain MRI scans.

number of cells in the tensor predictor. In this setting, developing well calibrated predictive models is thus of key importance.

To investigate the performance of competing methods outside the class of fMRI brain image data, we present an analysis using a brain connectome dataset on structural connectivity. Data are extracted from diffusion tensor imaging (DTI) and consist of estimates of the number of "fibers" connecting pairs of brain regions for 109 individuals. For each individual, brain connections among 70 brain regions (following desikan atlas) are encoded by a 70×70 weighted adjacency matrix. The (i, j)-th off-diagonal entry in the adjacency matrix is the estimated number of fiber tracts connecting the i-th and j-th brain region. The data also provides 10 clinical covariates for every individual, including sex, age, openness, agreeableness and conscientiousness.

The focus of this study is on developing a predictive model with Creativity Composite Index (CCI) as a response fitted against clinical covariates and a tensor covariate (i.e., the weighted adjacency matrix). Implementing FTR on this data using the `TensorReg` package was attempted, however, functions in the toolbox require $n > R \times p$. In this example, because $n = 109$ and $p = 70$, it is only possible therefore to fit FTR with $R = 1$, which has previously been found to perform poorly by Zhou et al. (2013). We therefore compare our proposed method (BTR) to the vectorized Lasso on the basis of their predictive performance. To assess the predictive performance, the sample of $n = 109$ individuals are divided into 10 folds. Both vectorized lasso and BTR are fitted on 9 folds as training data and the remaining fold as the hold out sample. This is carried out for each of the 10 folds and predictive inferences are obtained for both vectorized lasso and BTR.

Table 9 reports the root mean squared error (RMSE) and correlation between observed and predicted responses, here average is over the 10 crossvalidated folds. For reference, average RMSE of the null model is 10.03 with a standard deviation of 2.40 across the

Method	avg(RMSE)	sd(RMSE)	avg(cov.)	sd(cov.)	avg(cor.)	sd(cor.)
Lasso	9.21	2.18	63%	20%	0.31	0.11
BTR	9.03	1.64	91%	10%	0.32	0.13

Table 9: mean and standard deviation of RMSEs and cor(y_{obs}, y_{pred}) of Lasso and BTR over 10 folds of the data. It also provides mean and standard deviation of coverages of Lasso and BTR over 10 folds of the data

folds. Given the very high degree of sparsity in the connectome adjacency matrix, it is not surprising that the Lasso is competitive to BTR. However, note that BTR detects this signal with far fewer effective parameters as compared to vectorized lasso. Finally, we measure coverage of 95% predictive intervals for all competitors. The latter is of course a byproduct of our fully Bayesian approach (BTR), while for the Lasso we use a two-staged approach. First we estimate the regression coefficients and subsequently construct approximate 95% predictive intervals based on the normal response-model centered on the predictive mean with variance equal to the estimated residual variance.

9. Discussion

This work develops a novel class of prior distributions on tensor valued predictors which substantially reduces dimensionality relative to vectorizing, providing a multiway analogue of vector shrinkage priors, and enabling high dimensional region selection. The prior on tensor coefficient constructed here imparts shrinkage of the tensor components at global and local levels, while also encouraging shrinkage towards low rank tensor decomposition. In contrast, existing penalization framework on the tensor coefficient shrinks only at the global level. Strong theoretical results are proved for the proposed class of multiway shrinkage priors and a computationally efficient MCMC algorithm is developed in the regression setting. We plan to extend methods developed here to settings where the measured response for each subject is binary (e.g., indicator of a heath outcome) or multivariate, i.e., $\boldsymbol{y} = (y_1, ..., y_d)$. Also, the current framework of Bayesian tensor regression fixes rank R of the PARAFAC decomposition at reasonably large value. It might be of interest to learn the PARAFAC rank R by adding a discrete prior distribution on R.

In various longitudinal studies, monitoring the evolution in the predictor response relationship (i.e., changes to the scalar and tensor parameters) is of fundamental interest. One application involves subjects receiving various treatments (e.g., chemotherapy), with tensor valued predictors corresponding to mRI (fMRI) scans obtained at regular intervals over a period of time. In such settings it is of crucial importance to monitor the progression of the disease in response to the treatment being administered. We plan to extend our method to such settings.

Acknowledgement

The authors would like to thank Joshua T. Vogelstein from Johns Hopkins university for gracefully allowing us to utilize their brain connectome dataset in the real data analysis of our proposed BTR method.

Appendix A

MCMC algorithm

The following derivations concern the M-DGDP prior (9) and the sampling algorithm outlined in Section 5.1.

For step (1b) Recall from Section 3.3 that $\tau \sim \text{Ga}(a_\tau, b_\tau)$ and $\Phi \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_R)$ and denote $p_0 = \sum_{j=1}^D p_j$. Then,

$$\begin{aligned}
\pi(\Phi|\boldsymbol{B}, \boldsymbol{\omega}) &\propto \pi(\Phi) \int_0^\infty \pi(\boldsymbol{B}|\boldsymbol{\omega}, \Phi, \tau)\pi(\tau)d\tau \\
&\propto \Big[\prod_{r=1}^R \phi_r^{\alpha_r - 1}\Big] \int_0^\infty \prod_{r=1}^R \Big[(\tau\phi_r)^{-p_0/2} \exp\Big(-\frac{1}{\tau\phi_r}\sum_{j=1}^d ||\boldsymbol{\beta}_{jr}||^2/(2\omega_{jr})\Big)\Big] \tau^{a_\tau - 1}\exp(-b_\tau\tau)d\tau \\
&\propto \Big[\prod_{r=1}^R \phi^{\alpha_r - \frac{p_0}{2} - 1}\Big] \int_0^\infty \tau^{a_\tau - R\frac{p_0}{2} - 1} \prod_{r=1}^R \exp\Big(-\frac{C_r}{\tau\phi_r} - b_\tau(\tau\phi_r)\Big)\, d\tau
\end{aligned}$$

with $C_r = \sum_{j=1}^d ||\boldsymbol{\beta}_{jr}||^2/(2\omega_{jr})$. When $a_\tau = \sum_{r=1}^R \alpha_r$, this contains the kernel of a generalized inverse Gaussian (gIG) distribution for $(\tau\phi_r)$. Recall: $X \sim f_X(x) = \text{giG}(p, a, b) \propto x^{p-1}\exp(-(ax + b/x)/2)$. Following Lemma 9 in the Appendix B, for independent random variable $T_r \sim f_r$ on $(0, \infty)$, the joint density of $\{\phi_r = T_r/\sum_{\tilde{r}} T_{\tilde{r}} : r = 1, \ldots, R\}$ has support on $\mathcal{S}^{R-1}$. In particular,

$$f(\phi_1, \ldots, \phi_{R-1}) = \int_0^\infty t^{R-1} \prod_{r=1}^R f_r(\phi_r t)\, dt, \quad \phi_R = 1 - \sum_{r<R} \phi_r.$$

Substituting $f_r(x) \propto x^{-\delta_r}\exp(-C_r/x)\exp(-b_\tau x)$ in the above expression yields

$$\begin{aligned}
f(\phi_1, \ldots, \phi_{R-1}) &\propto \int_0^\infty \tau^{R-1} \prod_{r=1}^R (\phi_r\tau)^{-\delta_r} \exp\Big(-\frac{C_r}{(\phi_r\tau)} - b_\tau(\phi_r\tau)\Big)d\tau \\
&= \Big[\prod_{r=1}^R \phi^{-\delta_r}\Big] \int_0^\infty \tau^{R - \sum_r \delta_r - 1} \prod_{r=1}^R \exp\Big(-\frac{C_r}{(\phi_r\tau)} - b_\tau(\phi_r\tau)\Big)d\tau.
\end{aligned}$$

Matching exponents between this expression and the preceding one implies (1) $a_\tau - R(p_0/2) - 1 = R - \sum_r \delta_r - 1$, and (2) $\delta_r = 1 + p_0/2 - \alpha_r$. Then,

$$a_\tau = R(1 + p_0/2) - \sum_r \delta_r = R(1 + p_0/2) - (R + Rp_0/2 - \sum_r \alpha_r) = \sum_r \alpha_r$$

as previously noted. Hence, draws from $[\Phi|\alpha, \boldsymbol{B}, \boldsymbol{W}]$ are obtained by sampling $T_r \sim f_r = \text{giG}(\alpha_r - p_0/2, 2b_\tau, 2C_r)$ independently for $r = 1, \ldots, R$, and renormalizing.

Proof of lemma 5

Proof Using priors defined in (9), one has $C_\lambda = \mathbb{E}_\lambda(1/\lambda^2) = \frac{b_\lambda^2}{(a_\lambda-1)(a_\lambda-2)}$ for any $a_\lambda > 2$. In addition, the following inequalities are useful to bound the latter quantity:

- If $\alpha_1 = c/R$, $c \in \mathbb{N}_+$, $\Gamma(\alpha_0+D)/\Gamma(\alpha_0) = \alpha_0(\alpha_0+1)\cdots(\alpha_0+D-1)$. Using the fact that $\log(x+1) \leq x$, $x \geq 0$, one has $\log(\alpha_0)+\cdots+\log(\alpha_0+D-1) \leq \alpha_0 D - 1 + \sum_{k=1\vee D-2}^{D-2} k$. Then $\alpha_0^D \leq \Gamma(\alpha_0+D)/\Gamma(\alpha_0) \leq A_\tau \exp(\alpha_0 D)$ where $A_\tau = \exp(-1+\sum_{k=1\vee D-2}^{D-2} k) = \exp\big((D^2-3D)/2\big)$, $D \geq 2$.
- Let $||x||_r$ denote the L^rth norm. Trivially, $||\Phi||_D^D \leq 1$; in addition, by Hölder's inequality, for any $x \in \Re^k$ and $0 < r < p$, one has $||x||_p \geq k^{-\left(\frac{1}{r}-\frac{1}{p}\right)}||x||_r$. In our setting, $D \geq 2$. Taking $r = 1$ in the latter yields $||\Phi||_D^D \geq R^{-(D-1)}$.

Recall $\alpha_0 = \sum_{r=1}^R \alpha_r = \alpha_1 R$. This leads to the lower and upper bounds for the prior voxel-level variance:

$$\begin{aligned}\text{var}(B_{i_1,\ldots,i_D}) &\geq (2C_\lambda)^D (\alpha_1 R)^D R^{-(D-1)}/b_\tau^D \;= (2C_\lambda)^D \alpha_1^D R/b_\tau^D \\ \text{var}(B_{i_1,\ldots,i_D}) &\leq A_\tau (2C_\lambda)^D \exp(\alpha_1 RD)/b_\tau^D.\end{aligned}$$

■

Consistency proofs

The proof of Theorem 1 relies in part on the existence of exponentially consistent tests.

Definition An exponentially consistent sequence of test functions $\Phi_n = I(\boldsymbol{y}_n \in \mathcal{C}_n)$ for testing $H_0 : \boldsymbol{B}_n = \boldsymbol{B}_n^0$ vs. $H_1 : \boldsymbol{B}_n \neq \boldsymbol{B}_n^0$ satisfies

$$\mathbb{E}_{\boldsymbol{B}_n^0}(\Phi_n) \leq c_1 \exp(-b_1 n), \qquad \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} \mathbb{E}_{\boldsymbol{B}_n}(1-\Phi_n) \leq c_2 \exp(-b_2 n)$$

for some $c_1, c_2, b_1, b_2 > 0$.

Lemma 6 *There exist an exponentially consistent sequence of tests Φ_n for testing H_0 : $\boldsymbol{B}_n = \boldsymbol{B}_n^0$ vs. $H_1 : \boldsymbol{B}_n \neq \boldsymbol{B}_n^0$.*

Proof We begin by stating that $\sum_{i=1}^n \left(y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n^0\rangle\right)^2 \sim \chi_n^2$ under $\boldsymbol{B}_n^0$. We choose the critical region of the test Φ_n as $\mathcal{C}_n = \left\{\boldsymbol{B}_n : \frac{1}{n}\sum_{i=1}^n \left(y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n^0\rangle\right)^2 > \epsilon/4\right\}$. Note that

$$\mathbb{E}_{\boldsymbol{B}_n^0}(\Phi_n) = P_{\boldsymbol{B}_n^0}\Big(\sum_{i=1}^n \left(y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n^0\rangle\right)^2 > n\epsilon/4\Big) \leq \exp\Big(-\frac{n\epsilon}{16}\Big), \text{ for large n,}$$

where the last line follows by simplifying Lemma 1 in Laurent and Massart (2000).

Now we will use the fact that

$$\begin{aligned}&\frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n^0\rangle)^2 \\ &= \frac{1}{n}\sum_{i=1}^n \left(y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle\right)^2 + \frac{1}{n}\sum_{i=1}^n \left(\langle \boldsymbol{X}_i, \boldsymbol{B}_n - \boldsymbol{B}_n^0\rangle\right)^2 + \frac{2}{n}\sum_{i=1}^n \left(y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle\right)\left(\langle \boldsymbol{X}_i, \boldsymbol{B}_n - \boldsymbol{B}_n^0\right) \\ &= \frac{1}{n}\sum_{i=1}^n \left(\langle \boldsymbol{X}_i, \boldsymbol{B}_n - \boldsymbol{B}_n^0\rangle\right)^2 + \frac{1}{n}\sum_{i=1}^n KL_i + \frac{2}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)\langle \boldsymbol{X}_i, \boldsymbol{B}_n - \boldsymbol{B}_n^0\rangle.\end{aligned}$$

Note that, under $\boldsymbol{B}_n$,

$$\frac{2}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)\langle \boldsymbol{X}_i, \boldsymbol{B}_n - \boldsymbol{B}_n^0\rangle \sim N(0, \frac{4}{n^2}\sum_{i=1}^n KL_i),$$

so that, $\frac{2}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)\langle \boldsymbol{X}_i, \boldsymbol{B}_n - \boldsymbol{B}_n^0\rangle = \sqrt{\frac{4}{n}\sum_{i=1}^n KL_i}\frac{Z}{\sqrt{n}}$, where $Z \sim N(0,1)$. Thus,

$$\begin{aligned}&\sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} \mathbb{E}_{\boldsymbol{B}_n}(1-\Phi_n) = \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} P_{\boldsymbol{B}_n}\left(\frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n^0\rangle)^2 \leq \epsilon/4\right) \\ &\leq \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} P_{\boldsymbol{B}_n}\left(\left|\left|\sqrt{\frac{4}{n}\sum_{i=1}^n KL_i}\frac{Z}{\sqrt{n}} + \frac{1}{n}\sum_{i=1}^n KL_i\right| - \left|\frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)^2\right|\right| \leq \epsilon/4\right) \\ &\leq \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} P_{\boldsymbol{B}_n}\left(\left|\sqrt{\frac{4}{n}\sum_{i=1}^n KL_i}\frac{Z}{\sqrt{n}} + \frac{1}{n}\sum_{i=1}^n KL_i\right| - \epsilon/4 \leq \left|\frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)^2\right|\right) \\ &\leq \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} P_{\boldsymbol{B}_n}\left(\left|\frac{1}{n}\sum_{i=1}^n KL_i + \sqrt{\frac{4\sum_{i=1}^n KL_i}{n}}\frac{Z}{\sqrt{n}}\right| - \epsilon/4 \leq \left|\frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)^2\right|\right)\end{aligned}$$

Let $\mathcal{T}_n = \left\{\left|\sqrt{\frac{4\sum_{i=1}^n KL_i}{n}}\frac{Z}{\sqrt{n}}\right| \leq \frac{1}{2n}\sum_{i=1}^n KL_i\right\}$. Using this fact we have

$$\begin{aligned}&\sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} \mathbb{E}_{\boldsymbol{B}_n}(1-\Phi_n) \\ &\leq \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} P_{\boldsymbol{B}_n}\left(\left\{\left|\frac{1}{n}\sum_{i=1}^n KL_i + \sqrt{\frac{4\sum_{i=1}^n KL_i}{n}}\frac{Z}{\sqrt{n}}\right| - \epsilon/4 \leq \left|\frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)^2\right|\right\} \cap \mathcal{T}_n\right) + \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} P_{\boldsymbol{B}_n}(\mathcal{T}_n) \\ &\leq \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} P_{\boldsymbol{B}_n}\left(\frac{1}{2n}\sum_{i=1}^n KL_i - \epsilon/4 \leq \left|\frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)^2\right|\right) + \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} P_{\boldsymbol{B}_n}\left(\left|\frac{Z}{\sqrt{n}}\right| \geq \frac{1}{4}\sqrt{\frac{1}{n}\sum_{i=1}^n KL_i}\right) \\ &\leq P_{\boldsymbol{B}_n}\left(\frac{3\epsilon}{4} \leq \left|\frac{1}{n}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)^2\right|\right) + P_{\boldsymbol{B}_n}\left(|Z| \geq \frac{1}{4}\sqrt{n\epsilon}\right) \\ &\leq P_{\boldsymbol{B}_n}\left(\frac{3n\epsilon}{4} \leq \chi_n^2\right) + P_{\boldsymbol{B}_n}\left(\chi_1^2 \geq \frac{n\epsilon}{4}\right) \leq \exp\left(-\frac{3n\epsilon}{16}\right) + \exp\left(-\frac{n\epsilon}{64}\right) \leq 2\exp\left(-\frac{n\epsilon}{64}\right),\end{aligned}$$

where the last line requires an application of Lemma 1 in Laurent and Massart (2000). ■

Theorem 1

Proof Under Lemma 6 one has

$$\Pi_n(\mathcal{B}_n^c) = \frac{\int_{\mathcal{B}_n^c} f(\boldsymbol{y}_n|\boldsymbol{B}_n)\pi_n(\boldsymbol{F}_n)}{\int f(\boldsymbol{y}_n|\boldsymbol{B}_n)\pi_n(\boldsymbol{F}_n)} = \frac{\int_{\mathcal{B}_n^c} \frac{f(\boldsymbol{y}_n|\boldsymbol{B}_n)}{f(\boldsymbol{y}_n|\boldsymbol{B}_n^0)}\pi_n(\boldsymbol{F}_n)}{\int \frac{f(\boldsymbol{y}_n|\boldsymbol{B}_n)}{f(\boldsymbol{y}_n|\boldsymbol{B}_n^0)}\pi_n(\boldsymbol{F}_n)} = \frac{N}{D} \leq \Phi_n + (1-\Phi_n)\frac{N}{D}.$$

Note that we have

$$P_{\boldsymbol{B}_n^0}\left(\Phi_n > \exp(-b_1 n/2)\right) \leq \mathbb{E}_{\boldsymbol{B}_n^0}\left(\Phi_n\right)\exp(b_1 n/2) \leq c_1 \exp(-b_1 n/2).$$

Therefore $\sum_{n=1}^{\infty} P_{\boldsymbol{B}_n^0}\left(\Phi_n > \exp(-b_1 n/2)\right) < \infty$. Using Borel-Cantelli lemma $P_{\boldsymbol{B}_n^0}\left(\Phi_n > \exp(-b_1 n/2) i.o.\right) = 0$. It follows that

$$\Phi_n \to 0 \quad a.s. \tag{13}$$

In addition, we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{B}_n^0}((1-\Phi_n)N) &= \int (1-\Phi_n)\int_{\mathcal{B}_n^c} \frac{f(\boldsymbol{y}_n|\boldsymbol{B}_n)}{f(\boldsymbol{y}_n|\boldsymbol{B}_n^0)}\pi_n(\boldsymbol{F}_n) f(\boldsymbol{y}_n|\boldsymbol{B}_n^0) \\
&= \int_{\mathcal{B}_n^c}\int (1-\Phi_n) f(\boldsymbol{y}_n|\boldsymbol{B}_n)\pi_n(\boldsymbol{F}_n) \\
&\leq \sup_{\boldsymbol{B}_n \in \mathcal{B}_n^c} \mathbb{E}_{\boldsymbol{B}_n}(1-\Phi_n) \leq c_2 \exp(-b_2 n).
\end{aligned}$$

Using a similar technique as above, $P_{\boldsymbol{B}_n^0}\left((1-\Phi_n)N\exp(nb_2/2) > \exp(-nb_2/4) i.o.\right) = 0$ so

$$\exp(bn)(1-\Phi_n)N \to 0 \quad a.s.. \tag{14}$$

By Lemma 6 and (13)-(14) it is enough to show that $M = \exp(\tilde{b}n)\int \frac{f(\boldsymbol{y}_n|\boldsymbol{B}_n)}{f(\boldsymbol{y}_n|\boldsymbol{B}_n^0)}\pi_n(\boldsymbol{F}_n) \to \infty$ for some $\tilde{b} \leq b = \frac{\epsilon}{256}$. We choose $\tilde{b} = b$. Consider the set $\mathcal{H}_n = \left\{\boldsymbol{B}_n : \frac{1}{n}\log\left[\frac{f(\boldsymbol{y}_n|\boldsymbol{B}_n^0)}{f(\boldsymbol{y}_n|\boldsymbol{B}_n)}\right] < \eta\right\}$, for some η which is chosen later.

$$\begin{aligned}
M &\geq \exp(\tilde{b}n)\int_{\mathcal{H}_n} \exp\left(-n\frac{1}{n}\log\frac{f(\boldsymbol{y}_n|\boldsymbol{B}_n^0)}{f(\boldsymbol{y}_n|\boldsymbol{B}_n)}\right)\pi_n(\boldsymbol{F}_n) \\
&\geq \exp((\tilde{b}-\eta)n)\pi_n(\mathcal{H}_n).
\end{aligned}$$

Note that

$$\begin{aligned}
&\frac{1}{n}\log\left[\frac{f(\boldsymbol{y}_n|\boldsymbol{B}_n)}{f(\boldsymbol{y}_n|\boldsymbol{B}_n^0)}\right] \\
&= \frac{1}{n}\left[-\frac{1}{2}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n\rangle)^2 + \frac{1}{2}\sum_{i=1}^n (y_i - \langle \boldsymbol{X}_i, \boldsymbol{B}_n^0\rangle)^2\right].
\end{aligned}$$

Let $\boldsymbol{y}_n = (y_1, ..., y_n)'$, $\boldsymbol{H}_n = (\langle \boldsymbol{X}_1, \boldsymbol{B}_n\rangle, \ldots, \langle \boldsymbol{X}_n, \boldsymbol{B}_n\rangle)$ and $\boldsymbol{H}_n^0 = \left(\langle \boldsymbol{X}_1, \boldsymbol{B}_n^0\rangle, \ldots, \langle \boldsymbol{X}_n, \boldsymbol{B}_n^0\rangle\right)$. Then

$$\begin{aligned}
&\pi_n\Big(\boldsymbol{B}_n : \frac{1}{n}\left[-||\boldsymbol{y}_n - \boldsymbol{H}_n^0||^2 + ||\boldsymbol{y}_n - \boldsymbol{H}_n||^2\right] < 2\eta\Big) \\
&\geq \pi_n\Big(\boldsymbol{B}_n : \frac{1}{n}\left|2||\boldsymbol{y}_n - \boldsymbol{H}_n^0||\left(||\boldsymbol{y}_n - \boldsymbol{H}_n|| - ||\boldsymbol{y}_n - \boldsymbol{H}_n^0||\right) + \left(||\boldsymbol{y}_n - \boldsymbol{H}_n|| - ||\boldsymbol{y}_n - \boldsymbol{H}_n^0||\right)^2\right| < 2\eta\Big) \\
&\geq \pi_n\Big(\boldsymbol{B}_n : \frac{1}{n}\left|2||\boldsymbol{y}_n - \boldsymbol{H}_n^0||||\boldsymbol{H}_n^0 - \boldsymbol{H}_n|| + ||\boldsymbol{H}_n - \boldsymbol{H}_n^0||^2\right| < 2\eta\Big) \\
&\geq \pi_n\Big(\boldsymbol{B}_n : \frac{1}{n}||\boldsymbol{H}_n^0 - \boldsymbol{H}_n|| < \frac{2\eta}{3\zeta_n}, ||\boldsymbol{y}_n - \boldsymbol{H}_n^0||^2 < \zeta_n^2\Big) \\
&\geq \pi_n(\mathcal{A}_{1n} \cap \mathcal{A}_{2n})
\end{aligned}$$

where $\mathcal{A}_{1n} = \left\{\frac{1}{n}||\boldsymbol{H}_n - \boldsymbol{H}_n^0|| < \frac{2\eta}{3\zeta_n}\right\}$, $\mathcal{A}_{2n} = \left\{||\boldsymbol{y}_n - \boldsymbol{H}_n^0||^2 < \zeta_n^2\right\}$.

We will show that $P_{\boldsymbol{B}_n^0}(\mathcal{A}_{2n}) = 1$ for all large n. Assume $\zeta_n = n^{(1+\rho_3)/2}$, $\rho_3 > 0$ so that $\zeta_n^2 > 8n$ for all large n. Then,

$$P_{\boldsymbol{B}_n^0}(\mathcal{A}'_{2n}) = P_{\boldsymbol{B}_n^0}(\chi_n^2 > \zeta_n^2) \leq \exp(-\zeta_n^2/2).$$

Therefore, using Borel-Cantelli lemma $P_{\boldsymbol{B}_n^0}(\mathcal{A}'_{2n}\ i.o.) = 0$. Hence $P_{\boldsymbol{B}_n^0}(\mathcal{A}_{2n}) = 1$ for all large n. It is enough to bound $\pi_n(\mathcal{A}_{1n})$. Let $M_n = \frac{1}{n}\sqrt{\sum_{i=1}^n ||\boldsymbol{X}_i||_2^2}$. Now use the fact that $\frac{1}{n}||\boldsymbol{H}_n - \boldsymbol{H}_n^0|| = \frac{1}{n}\sqrt{\sum_{i=1}^n (\langle \boldsymbol{X}_i, \boldsymbol{B}_n - \boldsymbol{B}_n^0\rangle)^2} \leq \left(\frac{1}{n}\sqrt{\sum_{i=1}^n ||\boldsymbol{X}_i||_2^2}\right)||\boldsymbol{B}_n - \boldsymbol{B}_n^0||_2$ to conclude

$$\left\{||\boldsymbol{B}_n - \boldsymbol{B}_n^0||_2 < \frac{2\eta}{3M_n\zeta_n}\right\} \subseteq \mathcal{A}_{1n}. \tag{15}$$

By (11) one has $\pi_n(\mathcal{A}_{1n}) \geq \pi_n\Big(||\boldsymbol{B}_n - \boldsymbol{B}_n^0||_2 < \frac{2\eta}{3M_n\zeta_n}\Big) \geq \exp(-dn)$ and hence $M \geq \exp\left((\tilde{b} - \eta - d)n\right) \to \infty$ as $n \to \infty$ proving the result. ■

Theorem 2

Proof Define $g : \mathbb{R} \to \mathbb{R}$ s.t.

$$g(\kappa) = R\kappa^D + \kappa^{D-1}\sum_{j=1}^D\sum_{r=1}^R ||\boldsymbol{\beta}_{j,n}^{0(r)}||_2 + \cdots + \kappa\sum_{j=1}^D\sum_{r=1}^R\prod_{l\neq j}||\boldsymbol{\beta}_{l,n}^{0(r)}||_2.$$

Let $\kappa_n > 0$ be s.t. $g(\kappa_n) = \frac{2\eta}{3M_n\zeta_n}$. Note that by Decarte's rule of sign, the equation $g(\kappa) - \frac{2\eta}{3M_n\zeta_n} = 0$ has a unique positive root. Further

$$\frac{1}{\kappa_n} < 1 + \max_{i=1,\ldots,D}\left|\frac{3\sum_{j_1\neq\cdots\neq j_i}\sum_{r=1}^R\prod_{l=1}^i ||\boldsymbol{\beta}_{j_l,n}^{0(r)}||_2}{2\eta/M_n\zeta_n}\right| \tag{16}$$

$$\kappa_n < 1 + \max\left\{\frac{2\eta}{3M_n\zeta_n R}, \max_{i=1,\ldots,D}\left|\frac{\sum_{j_1\neq\cdots\neq j_i}\sum_{r=1}^R\prod_{l=1}^i ||\boldsymbol{\beta}_{j_l,n}^{0(r)}||_2}{R}\right|\right\} \tag{17}$$

by Lemma 8 in Appendix B.

Using Lemma 7 in Appendix B, it is easy to see that

$$\left\{||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||_2\leq\kappa_n,\ j=1,...,D;\ r=1,...,R\right\}\subseteq\left\{||\boldsymbol{B}_n-\boldsymbol{B}_n^0||_2<\frac{2\eta}{3M_n\zeta_n}\right\}. \quad (18)$$

Using (15), $\pi_n\left(||\boldsymbol{B}_n-\boldsymbol{B}_n^0||_2<\frac{2\eta}{3M_n\zeta_n}\right)\geq\pi_n\left(\left\{||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||_2\leq\kappa_n,\ j=1,...,D;\ r=1,...,R\right\}\right)$. Note that

$$\begin{aligned}&\pi_n\left(\left\{||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||_2\leq\kappa_n,\ j=1,...,D;\ r=1,...,R\right\}|\{w_{jr,l}\}_{l=1}^{p_{j,n}},\{\lambda_{jr}\}_{j,r=1}^{D,R-1},\{\phi_r\}_{r=1}^{R-1},\tau\right)\\&\geq\left[\prod_{j=1}^{D}\prod_{r=1}^{R}\pi_n\left(||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||_2\leq\kappa_n|\{w_{jr,l}\}_{l=1}^{p_{j,n}},\{\lambda_{jr}\}_{j,r=1}^{D,R-1},\{\phi_r\}_{r=1}^{R-1},\tau\right)\right].\end{aligned}$$

Therefore, it is enough to bound $\pi_n(||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||\leq\kappa_n, j=1,...,D; r=1,...,R)$. For $j=1,...,D$, $r=1,...,R$,

$$\begin{aligned}&\pi_n(||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||\leq\kappa_n|\{w_{jr,l}\}_{l=1}^{p_{j,n}},\lambda_{jr},\{\phi_r\}_{r=1}^{R-1},\tau)\\&\quad\geq\prod_{l=1}^{p_{j,n}}\pi_n\left(|\beta_{j,n,l}^{(r)}-\beta_{j,n,l}^{0(r)}|\leq\frac{\kappa_n}{\sqrt{p_{j,n}}}|\{w_{jr,l}\}_{l=1}^{p_{j,n}},\lambda_{jr},\{\phi_r\}_{r=1}^{R-1},\tau\right)\\&\quad\geq\prod_{l=1}^{p_{j,n}}\left\{\left(\frac{2\kappa_n}{\sqrt{2p_{j,n}\pi w_{jr,l}\phi_r\tau}}\right)\exp\left(-\frac{|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}}{w_{jr,l}\phi_r\tau}\right)\right\},\end{aligned}$$

where the last step follows from the fact that $\int_a^b e^{-x^2/2}dx\geq e^{-(a^2+b^2)/2}(b-a)$. Thus,

$$\begin{aligned}&\pi_n(||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||\leq\kappa_n|\lambda_{jr},\{\phi_r\}_{r=1}^{R-1},\tau)\\&=\mathbb{E}\left[\pi_n(||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||\leq\kappa_n|\{w_{jr,l}\}_{l=1}^{p_{j,n}},\lambda_{jr},\{\phi_r\}_{r=1}^{R-1},\tau)\right]\\&\geq\left(\frac{2\kappa_n}{\sqrt{2p_{j,n}\pi\phi_r\tau}}\right)^{p_{j,n}}\prod_{l=1}^{p_{j,n}}E\left\{\frac{1}{\sqrt{w_{jr,l}}}\exp\left(-\frac{|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}}{w_{jr,l}\phi_r\tau}\right)\right\}\\&\geq\left(\frac{2\kappa_n\lambda_{jr}^2}{2\sqrt{2p_{j,n}\pi\phi_r\tau}}\right)^{p_{j,n}}\prod_{l=1}^{p_{j,n}}\int_{w_{jr,l}}\left\{\frac{1}{\sqrt{w_{jr,l}}}\exp\left(-\frac{|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}}{w_{jr,l}\phi_r\tau}-\frac{\lambda_{jr}^2w_{jr,l}}{2}\right)dw_{jr,l}\right\}.\end{aligned} \quad (19)$$

Use the change of variable $\frac{1}{w_{jr,l}}=z_{jr,l}$ and the normalizing constant from the inverse Gaussian density to deduce

$$\begin{aligned}&\int_{w_{jr,l}}\left\{\frac{1}{\sqrt{w_{jr,l}}}\exp\left(-\frac{|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}}{w_{jr,l}\phi_r\tau}-\frac{\lambda_{jr}^2w_{jr,l}}{2}\right)dw_{jr,l}\right\}\\&=\int_{z_{jr,l}}\left\{\frac{1}{\sqrt{z_{jr,l}^3}}\exp\left(-\frac{(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n})}{\phi_r\tau}z_{jr,l}-\frac{\lambda_{jr}^2}{2z_{jr,l}}\right)dz_{jr,l}\right\}\\&=\sqrt{\left(\frac{2\pi}{\lambda_{jr}^2}\right)}\exp\left(-\lambda_{jr}\frac{\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{\sqrt{\phi_r\tau}}\right).\end{aligned}$$

(19) can be written as

$$\begin{aligned}&\pi_n(||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||\leq\kappa_n|\lambda_{jr},\{\phi_r\}_{r=1}^{R-1},\tau)\\&\geq\left(\frac{2\kappa_\eta\lambda_{jr}^2}{2\sqrt{2p_{j,n}\pi\phi_r\tau}}\right)^{p_{j,n}}\prod_{l=1}^{p_{j,n}}\left[\sqrt{\left(\frac{2\pi}{\lambda_{jr}^2}\right)}\exp\left(-\lambda_{jr}\frac{\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{\sqrt{\phi_r\tau}}\right)\right]\\&=\left(\frac{2\kappa_n\lambda_{jr}}{2\sqrt{p_{j,n}\phi_r\tau}}\right)^{p_{j,n}}\exp\left(-\lambda_{jr}\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{\sqrt{\phi_r\tau}}\right).\end{aligned}$$

Therefore,

$$\begin{aligned}&\pi_n(||\boldsymbol{\beta}_{j,n}^{(r)}-\boldsymbol{\beta}_{j,n}^{0(r)}||\leq\kappa_n|\{\phi_r\}_{r=1}^{R-1},\tau)\\&\geq\left(\frac{2\kappa_n}{2\sqrt{p_{j,n}\phi_r\tau}}\right)^{p_{j,n}}\frac{b_{\lambda,r}^{a_{\lambda,r}}}{\Gamma(a_{\lambda,r})}\int_{\lambda_{jr}}\lambda_{jr}^{p_{j,n}+a_{\lambda,r}-1}\exp\left(-\lambda_{jr}\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{\sqrt{\phi_r\tau}}+b_{\lambda,r}\right]\right)d\lambda_{jr}\\&=\left(\frac{2\kappa_n}{2\sqrt{p_{j,n}\phi_r\tau}}\right)^{p_{j,n}}\frac{b_{\lambda,r}^{a_{\lambda,r}}}{\Gamma(a_{\lambda,r})}\frac{\Gamma(p_{j,n}+a_{\lambda,r})}{\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{\sqrt{\phi_r\tau}}+b_{\lambda,r}\right]^{p_{j,n}+a_{\lambda,r}}}\\&=\left(\frac{2\kappa_n}{2b_{\lambda,r}\sqrt{p_{j,n}\phi_r\tau}}\right)^{p_{j,n}}\frac{1}{\Gamma(a_{\lambda,r})}\frac{\Gamma(p_{j,n}+a_{\lambda,r})}{\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{b_{\lambda,r}\sqrt{\phi_r\tau}}+1\right]^{p_{j,n}+a_{\lambda,r}}}.\end{aligned}$$

The final expression as in the above yields

$$\pi_n(||\boldsymbol{\beta}_{j,n}^{(r)} - \boldsymbol{\beta}_{j,n}^{0(r)}|| \leq \kappa_n, j = 1, ..., D, r = 1, ..., R | \{\phi_r\}_{r=1}^{R-1}, \tau)$$
$$\geq \mathbb{E}\left\{ \prod_{j=1}^{D}\prod_{r=1}^{R}\left[\left(\frac{2\kappa_n}{2b_{\lambda,r}\sqrt{p_{j,n}\phi_r\tau}}\right)^{p_{j,n}} \frac{1}{\Gamma(a_{\lambda,r})}\lambda_{j,r}^{p_{j,n}+a_{\lambda,r}-1} \frac{\Gamma(p_{j,n}+a_{\lambda,r})}{\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{b_{\lambda,r}\sqrt{\phi_r\tau}}+1\right]^{p_{j,n}+a_{\lambda,r}}} \right] \right\}.$$

We will now use the fact that for $\phi_r \leq 1$,

$$\frac{1}{\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{b_{\lambda,r}\sqrt{\phi_r\tau}}+1\right]^{p_{j,n}+a_{\lambda,r}}} \geq \frac{1}{\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{b_{\lambda,r}\sqrt{\phi_r\tau}}+\frac{1}{\sqrt{\tau\phi_r}}\right]^{p_{j,n}+a_{\lambda,r}}} I_{\tau\in[0,1]}.$$

This inequality is critical to provide a lower bound on $\pi_n(||\boldsymbol{\beta}_{j,n}^{(r)} - \boldsymbol{\beta}_{j,n}^{0(r)}|| \leq \kappa_n, j = 1, ..., D, r = 1, ..., R)$ as following

$$\pi_n(||\boldsymbol{\beta}_{j,n}^{(r)} - \boldsymbol{\beta}_{j,n}^{0(r)}|| \leq \kappa_n, j = 1, ..., D, r = 1, ..., R)$$
$$\geq \frac{\lambda_2^{\lambda_1}\Gamma(Ra)}{\Gamma(\lambda_1)\Gamma(a)^R}\prod_{j=1}^{D}\prod_{r=1}^{R}\left[\left(\frac{\kappa_n}{\sqrt{p_{j,n}}b_{\lambda,r}}\right)^{p_{j,n}}\frac{\Gamma(p_{j,n}+a_{\lambda,r})}{\Gamma(a_{\lambda,r})}\right]\int_\tau \tau^{\lambda_1 - R\sum_{j=1}^{D}\frac{p_{j,n}}{2}-1}\exp(-\lambda_2\tau)$$
$$\int_{\boldsymbol{\phi}\in\mathcal{S}^{R-1}}\frac{\prod_{r=1}^{R}\phi_r^{a-1}}{\prod_{r=1}^{R}\phi_r^{\sum_{j=1}^{D}\frac{p_{j,n}}{2}}}\prod_{j=1}^{D}\prod_{r=1}^{R}\frac{1}{\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{b_{\lambda,r}\sqrt{\phi_r\tau}}+1\right]^{p_{j,n}+a_{\lambda,r}}}d\boldsymbol{\phi}d\tau$$
$$\geq \frac{\lambda_2^{\lambda_1}\Gamma(Ra)}{\Gamma(\lambda_1)\Gamma(a)^R}\prod_{j=1}^{D}\prod_{r=1}^{R}\left[\left(\frac{\kappa_n}{\sqrt{p_{j,n}}b_{\lambda,r}}\right)^{p_{j,n}}\frac{\Gamma(p_{j,n}+a_{\lambda,r})}{\Gamma(a_{\lambda,r})}\right]\prod_{j=1}^{D}\prod_{r=1}^{R}\frac{1}{\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{b_{\lambda,r}}+1\right]^{p_{j,n}+a_{\lambda,r}}}$$
$$\left(\int_{\tau=0}^{1}\tau^{\lambda_1+\sum_{r=1}^{R}a_{\lambda,r}\frac{D}{2}-1}\exp(-\tau\lambda_2)d\tau\right)\int_{\boldsymbol{\phi}\in\mathcal{S}^{R-1}}\prod_{r=1}^{R}\phi_r^{a+a_{\lambda,r}\frac{D}{2}-1}d\boldsymbol{\phi}$$
$$= \frac{\lambda_2^{\lambda_1}\Gamma(Ra)}{\Gamma(\lambda_1)\Gamma(a)^R}\prod_{j=1}^{D}\prod_{r=1}^{R}\left[\left(\frac{\kappa_n}{\sqrt{p_{j,n}}b_{\lambda,r}}\right)^{p_{j,n}}\frac{\Gamma(p_{j,n}+a_{\lambda,r})}{\Gamma(a_{\lambda,r})}\right]\prod_{j=1}^{D}\prod_{r=1}^{R}\frac{1}{\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{b_{\lambda,r}}+1\right]^{p_{j,n}+a_{\lambda,r}}}$$
$$\times\frac{\exp(-\lambda_2)}{(\lambda_1+\sum_{r=1}^{R}a_{\lambda,r}\frac{D}{2})}\frac{\prod_{r=1}^{R}\left[\Gamma(a+a_{\lambda,r}\frac{D}{2})\right]}{\Gamma(Ra+\frac{D}{2}\sum_{r=1}^{R}a_{\lambda,r})}.$$

Denote $C_6 = \frac{\lambda_2^{\lambda_1}\Gamma(Ra)}{\Gamma(\lambda_1)[\Gamma(a)]^R}\frac{\exp(-\lambda_2)}{(\lambda_1+\sum_{r=1}^{R}a_{\lambda,r}\frac{D}{2})}\frac{\prod_{r=1}^{R}\left[\Gamma(a+a_{\lambda,r}\frac{D}{2})\right]}{\Gamma(Ra+\sum_{r=1}^{R}a_{\lambda,r}\frac{D}{2})}$. Then the above expression gives us

$$-\log\left(||\boldsymbol{B}_n - \boldsymbol{B}_n^0||_2 < \frac{2\eta}{3M_n\zeta_n}\right)$$
$$\leq -\log(C_6) + \sum_{j=1}^{D}\sum_{r=1}^{R}p_{j,n}\left[-\log(\kappa_n) + \frac{1}{2}\log(p_{j,n}) + \log(b_{\lambda,r}) + \log(\Gamma(a_{\lambda,r})\right]$$
$$-\sum_{r=1}^{R}\sum_{j=1}^{D}\log(\Gamma(p_{n,j}+a_{\lambda,r})) + \sum_{j=1}^{D}\sum_{r=1}^{R}(p_{j,n}+a_{\lambda,r})\log\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{b_{\lambda,r}}+1\right]. \quad (20)$$

Using (16) and assumption (b), it is easy to see that $\frac{1}{\kappa_n} < G_5 n^{\rho_2+\frac{\rho_3+1}{2}}\prod_{j=1}^{D}p_{j,n}$ for a constant $G_5 > 0$ for all large n. Therefore, $\sum_{j=1}^{D}\sum_{r=1}^{R}p_{j,n}\left[\log\left(\frac{1}{\kappa_n}\right) + \frac{1}{2}\log(p_{j,n}) + \log(b_{\lambda,r}) + \log(\Gamma(a_{\lambda,r})\right] = o(n)$. Also, $\sum_{r=1}^{R}\sum_{j=1}^{D}\log(\Gamma(p_{j,n}+a_{\lambda,r}))] \leq \sum_{j=1}^{D}(p_{j,n}+a_{\lambda,r})\log(p_{j,n}+a_{\lambda,r}) = o(n)$, by assumption (c). Finally, $\sum_{j=1}^{D}\sum_{r=1}^{R}(p_{j,n}+a_{\lambda,r})\log\left[\frac{\sum_{l=1}^{p_{j,n}}\sqrt{2\left(|\beta_{j,n,l}^{0(r)}|^2+\kappa_n^2/p_{j,n}\right)}}{b_{\lambda,r}}+1\right] = o(n)$, by assumptions (b) and (c). Thus, $-\log\left(\pi_n(\boldsymbol{B}_n : ||\boldsymbol{B}_n - \boldsymbol{B}_n^0||_2 < \frac{2\eta}{3M_n\zeta_n})\right) < dn$ for all $d > 0$, for all large n. This proves the result. ■

Appendix B

This Section contains additional Lemmas relevant to the article.

Lemma 7 *Suppose $\boldsymbol{T} = T_1 \circ \cdots \circ T_D$ and $\boldsymbol{F} = F_1 \circ \cdots \circ F_D$ are two rank one tensors of same dimension. Then*

$$\boldsymbol{T} - \boldsymbol{F} = (T_1 - F_1) \circ \cdots \circ (T_D - F_D) + \sum_{l=1}^{D-1}\sum_{\mathcal{I}_1\cup\mathcal{I}_2=1:D, |\mathcal{I}_1|=l, |\mathcal{I}_2|=D-l}\gamma_1 \circ \cdots \circ \gamma_D,$$

where $\gamma_j = F_j$ if $j \in \mathcal{I}_2$; $= T_j - F_j$ if $j \in \mathcal{I}_1$.

Proof We will show it by induction. If $D = 2$ then,

$$\boldsymbol{T} - \boldsymbol{F} = T_1 \circ T_2 - F_1 \circ F_2 = (T_1 - F_1) \circ T_2 + F_1 \circ T_2 - F_1 \circ F_2$$
$$= (T_1 - F_1) \circ (T_2 - F_2) + (T_1 - F_1) \circ F_2 + F_1 \circ (T_2 - F_2).$$

Assume the result to hold for $D-1$. For D,

$$
\begin{aligned}
&T_1 \circ \cdots \circ T_D - F_1 \circ \cdots \circ F_D \\
&= (T_1 - F_1) \circ T_2 \circ \cdots \circ T_D + F_1 \circ [T_2 \circ \cdots \circ T_D - F_2 \circ \cdots \circ F_D] \\
&= (T_1 - F_1) \circ [(T_2 - F_2) \circ \cdots \circ (T_D - F_D) + F_2 \circ \cdots \circ F_D + \\
&\sum_{l=1}^{D-2} \sum_{\mathcal{I}_1 \cup \mathcal{I}_2, |\mathcal{I}_1|=l, |\mathcal{I}_2|=D-1-l} \gamma_2 \circ \cdots \circ \gamma_D] + \\
&F_1 \circ [(T_2 - F_2) \circ \cdots \circ (T_D - F_D) + \sum_{l=1}^{D-2} \sum_{\mathcal{I}_1 \cup \mathcal{I}_2, |\mathcal{I}_1|=l, |\mathcal{I}_2|=D-1-l} \gamma_2 \circ \cdots \circ \gamma_D] \\
&= (T_1 - F_1) \circ \cdots \circ (T_D - F_D) + \sum_{l=1}^{D-1} \sum_{\mathcal{I}_1 \cup \mathcal{I}_2, |\mathcal{I}_1|=l, |\mathcal{I}_2|=D-l} \gamma_1 \circ \cdots \circ \gamma_D].
\end{aligned}
$$

Hence proved. ■

Lemma 8 *Let x^* be a real root of the polynomial $P(x) = a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x - a_0$. Then $1/|x^*| < 1 + \max_{i=1,\ldots,k} \left|\frac{a_i}{a_0}\right|$.*

Proof Consider the polynomial $P_1(\zeta) = \zeta^k - \left(\frac{a_1}{a_0}\right)\zeta^{k-1} - \cdots - \left(\frac{a_k}{a_0}\right)$. By making a change of variable with $\zeta = \frac{1}{x}$, we obtain

$$
\begin{aligned}
P_1\left(\frac{1}{x}\right) &= \frac{1}{x^k} - \left(\frac{a_1}{a_0}\right)\frac{1}{x^{k-1}} - \cdots - \left(\frac{a_k}{a_0}\right) \\
&= -\frac{a_k x^k + \cdots + a_1 x - a_0}{a_0 x^k}.
\end{aligned}
$$

Note that $P_1\left(\frac{1}{x}\right) = 0$ is solved by $x = x^*$. Therefore, $P_1(\zeta) = 0$ is solved by $\zeta = \frac{1}{x^*}$. The result follows by using Cauchy bound on the roots of a polynomial. ■

Lemma 9 *Suppose $T_1, \ldots, T_m$ are independent random variables with T_j having density f_j supported in $(0, \infty)$. Let $\phi_j = \frac{T_j}{\sum_{l=1}^m T_m}$. Then the joint density of $(\phi_1, \ldots, \phi_{m-1})$ has a joint density supported on the simplex $\mathcal{S}^{m-1}$ and is given by*

$$
f(\phi_1, \ldots, \phi_{m-1}) = \int_{t=0}^{\infty} t^{m-1} \prod_{l=1}^{m} f_j(\phi_j t) dt,
$$

where $\phi_m = 1 - \sum_{l=1}^{m-1} \phi_l$.

Proof This result is well known in the theory of normalized random measures (Kruijer et al., 2010). ■

References

Artin Armagan, David B Dunson, and Jaeyong Lee. Generalized double pareto shrinkage. *Statistica Sinica*, 23(1):119–143, 2013a.

Artin Armagan, David B Dunson, Jaeyong Lee, Waheed U Bajwa, and Nate Strawn. Posterior consistency in linear models under shrinkage priors. *Biometrika*, 100(4):1011–1018, 2013b.

Anirban Bhattacharya, Debdeep Pati, Natesh S Pillai, and David B Dunson. Dirichlet–laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, 110 (512):1479–1490, 2015.

Brian S Caffo, Ciprian M Crainiceanu, Guillermo Verduzco, Suresh Joel, Stewart H Mostofsky, Susan Spear Bassett, and James J Pekar. Two-stage decompositions for the analysis of functional connectivity for fmri with application to alzheimer's disease risk. *NeuroImage*, 51(3):1140–1149, 2010.

David Gerard and Peter Hoff. Adaptive higher-order spectral estimators. *arXiv preprint arXiv:1505.02114*, 2015.

Jeff Goldsmith, Lei Huang, and Ciprian M Crainiceanu. Smooth scalar-on-image regression via spatial Bayesian variable selection. *Journal of Computational and Graphical Statistics*, 23(1):46–64, 2014.

Chris Hinrichs, Vikas Singh, Lopamudra Mukherjee, Guofan Xu, Moo K Chung, Sterling C Johnson, and Alzheimer's Disease Neuroimaging Initiative. Spatially augmented lpboosting for ad classification with evaluations on the adni dataset. *Neuroimage*, 48(1):138–149, 2009.

Peter D Hoff et al. Multilinear tensor regression for longitudinal relational data. *The Annals of Applied Statistics*, 9(3):1169–1193, 2015.

Hung Hung and Chen-Chien Wang. Matrix variate logistic regression model with application to eeg data. *Biostatistics*, 14(1):189–202, 2013.

Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

Willem Kruijer, Judith Rousseau, and Aad Van Der Vaart. Adaptive Bayesian density estimation with location-scale mixtures. *Electronic Journal of Statistics*, 4:1225–1257, 2010.

Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

Nicole Lazar. *The Statistical Analysis of Functional MRI Data*. Springer Science & Business Media, 2008.

Martin A Lindquist. The statistical analysis of fmri data. *Statistical Science*, 23(4):439–464, 2008.

Nicholas G Polson and James G Scott. Local shrinkage rules, lévy processes and regularized regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):287–311, 2012.

Peihua Qiu. Jump surface estimation, edge detection, and image restoration. *Journal of the American Statistical Association*, 102(478):745–756, 2007.

Philip T Reiss and R Todd Ogden. Functional generalized linear models with images as predictors. *Biometrics*, 66(1):61–69, 2010.

Srikanth Ryali, Kaustubh Supekar, Daniel A Abrams, and Vinod Menon. Sparse logistic regression for whole-brain classification of fmri data. *NeuroImage*, 51(2):752–764, 2010.

Taiji Suzuki. Convergence rate of Bayesian tensor estimatior and its minimax optimality. In *Proceedings of the 32nd International Conference on Machine Learning (Lille, 2015)*, pages 1273–1282, 2015.

Xuejing Wang, Bin Nan, Ji Zhu, and Robert Koeppe. Regularized 3d functional regression for brain image data via Haar wavelets. *The annals of applied statistics*, 8(2):1045, 2014.

Yun Yang and David B Dunson. Minimax optimal Bayesian aggregation. *arXiv preprint arXiv:1403.1345*, 2014.

Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

Jing Zhou, Anirban Bhattacharya, Amy H Herring, and David B Dunson. Bayesian factorizations of big sparse tensors. *Journal of the American Statistical Association*, 110(512): 1562–1576, 2015.

Journal of Machine Learning Research 18 (2017) 1-50 Submitted 8/16; Revised 5/17; Published 8/17

Robust Discriminative Clustering with Sparse Regularizers

Nicolas Flammarion[1] NICOLAS.FLAMMARION@ENS.FR
Balamurugan Palaniappan[2] PALANIAPPAN@TELECOM-PARISTECH.FR
Francis Bach[1] FRANCIS.BACH@ENS.FR
[1]*INRIA*
Département d'Informatique de l'ENS, École Normale Supérieure, CNRS, PSL Research University
Paris, France.

[2]*Signal, Statistique et Apprentissage (S^2A) Group, IDS Department, LTCI*
Telecom-ParisTech
Paris, France.

Editor: Sathiya Keerthi

Abstract

Clustering high-dimensional data often requires some form of dimensionality reduction, where clustered variables are separated from "noise-looking" variables. We cast this problem as finding a low-dimensional projection of the data which is well-clustered. This yields a one-dimensional projection in the simplest situation with two clusters, and extends naturally to a multi-label scenario for more than two clusters. In this paper, (a) we first show that this joint clustering and dimension reduction formulation is equivalent to previously proposed discriminative clustering frameworks, thus leading to convex relaxations of the problem; (b) we propose a novel sparse extension, which is still cast as a convex relaxation and allows estimation in higher dimensions; (c) we propose a natural extension for the multi-label scenario; (d) we provide a new theoretical analysis of the performance of these formulations with a simple probabilistic model, leading to scalings over the form $d = O(\sqrt{n})$ for the affine invariant case and $d = O(n)$ for the sparse case, where n is the number of examples and d the ambient dimension; and finally, (e) we propose an efficient iterative algorithm with running-time complexity proportional to $O(nd^2)$, improving on earlier algorithms for discriminative clustering with the square loss, which had quadratic complexity in the number of examples.

1. Introduction

Clustering is an important and commonly used pre-processing tool in many machine learning applications, with classical algorithms such as K-means (MacQueen, 1967), linkage algorithms (Gower and Ross, 1969) or spectral clustering (Ng et al., 2002). In high dimensions, these unsupervised learning algorithms typically have problems identifying the underlying optimal discrete nature of the data; for example, they are quickly perturbed by adding a few noisy dimensions. Clustering high-dimensional data thus requires some form of dimensionality reduction, where clustered variables are separated from non-informative "noise-looking" (e.g., Gaussian) variables.

Several frameworks aim at linearly separating noise from signal, that is finding projections of the data that extracts the signal and removes the noise. They differ in the ways signals and noise are defined. A line of work that dates back to projection pursuit (Friedman and Stuetzle, 1981) and independent component analysis (Hyvärinen et al., 2004) defines the noise as Gaussian while the signal is non-Gaussian (Blanchard et al., 2006; Le Roux and Bach, 2013; Diederichs et al., 2013). In this work, we follow De la Torre and Kanade (2006); Ding and Li (2007), along the alternative route where one defines the signal as being clustered while the noise is any non-clustered variable. In the simplest situation with two clusters, we may project the data into a one-dimensional subspace. Given a data matrix $X \in \mathbb{R}^{n\times d}$ composed of n d-dimensional points, the goal is to find a direction $w \in \mathbb{R}^d$ such that $Xw \in \mathbb{R}^n$ is well-clustered, e.g., by K-means. This is equivalent to identifying both a direction to project, represented as $w \in \mathbb{R}^d$ and the labeling $y \in \{-1,1\}^n$ that represents the partition into two clusters.

Most existing formulations are non-convex and typically perform a form of alternating optimization (De la Torre and Kanade, 2006; Ding and Li, 2007), where given $y \in \{-1,1\}^n$, the projection w is found by linear discriminant analysis (or any binary classification method), and given the projection w, the clustering is obtained by thresholding Xw or running K-means on Xw. As shown in Section 2, this alternating minimization procedure happens to be equivalent to maximizing the (centered) correlation between $y \in \{-1,1\}^n$ and the projection $Xw \in \mathbb{R}^d$, that is

$$\max_{w\in\mathbb{R}^d, y\in\{-1,1\}^n} \frac{(y^\top \Pi_n X w)^2}{\|\Pi_n y\|_2^2 \, \|\Pi_n X w\|_2^2},$$

where $\Pi_n = I_n - \frac{1}{n} 1_n 1_n^\top$ is the usual centering projection matrix (with $1_n \in \mathbb{R}^n$ being the vector of all ones, and I_n the $n \times n$ identity matrix). This correlation is equal to one when the projection is perfectly clustered (independently of the number of elements per cluster). Existing methods are alternating minimization algorithms with no theoretical guarantees.

In this paper, we relate this formulation to discriminative clustering formulations (Xu et al., 2004; Bach and Harchaoui, 2007), which consider the problem

$$\min_{v\in\mathbb{R}^d,\ b\in\mathbb{R},\ y\in\{-1,1\}^n} \frac{1}{n}\|y - Xv - b1_n\|_2^2, \tag{1}$$

with the intuition of finding labels y which are easy to predict by an affine function of the data. In particular, we show that given the relationship between the number of positive labels and negative labels (i.e., the squared difference between the respective number of elements), these two problems are equivalent, and hence discriminative clustering explicitly performs joint dimension reduction and clustering.

While the discriminative framework is based on convex relaxations and has led to interesting developments and applications (Zhang et al., 2009; Li et al., 2009; Joulin et al., 2010a,b; Wang et al., 2010; Niu et al., 2013; Huang et al., 2015), it has several shortcomings when used with the square loss: (a) the running-time complexity of the semi-definite formulations is at least quadratic in n, and typically much more, (b) no theoretical analysis has ever been performed, (c) no convex sparse extension has been proposed to handle data with many irrelevant dimensions, (d) balancing of the clusters remains an issue, as it typically adds an extra hyperparameter which may be hard to set. In this paper, we focus on addressing these concerns.

©2017 Nicolas Flammarion, Balamurugan Palaniappan and Francis Bach.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-429.html.

When there are more than two clusters, one considers either the *multi-label* or the *multi-class* settings. The multi-class problem assumes that the data are clustered into distinct classes, i.e., a single class per observation, whereas the multi-label problem assumes the data share different labels, i.e., multiple labels per observation. We show in this work that discriminative clustering framework extends more naturally to multi-label scenarios and that this extension has the same convex relaxation.

A summary of the contributions of this paper follows:

- In Section 2, we relate discriminative clustering with the square loss to a joint clustering and dimension reduction formulation. The proposed formulation takes care of the balancing hyperparameter implicitly.
- We propose in Section 3, a novel sparse extension to discriminative clustering and show that it can still be cast through a convex relaxation.
- When there are more than two clusters, we extend naturally the sparse formulation to a multi-label scenario in Section 4.
- We then proceed to provide a theoretical analysis of the proposed formulations with a simple probabilistic model in Section 5, which effectively leads to scalings over the form $d = O(\sqrt{n})$ for the affine invariant case and $d = O(n)$ for the 1-sparse case.
- Finally, we propose in Section 6 efficient iterative algorithms with running-time complexity for each step equal to $O(nd^2)$, the first to be linear in the number of observations n for discriminative clustering with the square loss.

Throughout this paper we assume that $X \in \mathbb{R}^{n\times d}$ is *centered*, a common pre-processing step in unsupervised (and supervised) learning. This implies that $X^\top 1_n = 0$ and $\Pi_n X = X$.

2. Joint Dimension Reduction and Clustering

In this section, we focus on the single binary label case, where we first study the usual non-convex formulation, before deriving convex relaxations based on semi-definite programming. Some of the following results are already known in the literature; however, we state them here for completeness.

2.1 Non-convex formulation

Following De la Torre and Kanade (2006); Ding and Li (2007); Ye et al. (2008), we consider a cost function which depends on $y \in \{-1,1\}^n$ and $w \in \mathbb{R}^d$, which is such that alternating optimization is exactly (a) running K-means with two clusters on Xw to obtain y given w (when we say "running K-means", we mean solving the vector quantization problem exactly), and (b) performing linear discriminant analysis to obtain w given y.

Proposition 1 (Joint clustering and dimension reduction for two clusters) *Given $X \in \mathbb{R}^{n\times d}$ such that $X^\top 1_n = 0$ and X has rank d, consider the optimization problem*

$$\max_{w\in\mathbb{R}^d, y\in\{-1,1\}^n} \frac{(y^\top Xw)^2}{\|\Pi_n y\|_2^2 \, \|Xw\|_2^2}. \tag{2}$$

Given y, the optimal w is obtained as $w = (X^\top X)^{-1}X^\top y$, while given w, the optimal y is obtained by running K-means on Xw.

This equivalence might be straightforward, however it has not been precisely stated in the literature to the best of our knowledge.

Proof Given y, we need to optimize the Rayleigh quotient $\frac{w^\top X^\top y y^\top Xw}{w^\top X^\top Xw}$ with a rank-one matrix in the numerator, which leads to $w = (X^\top X)^{-1}X^\top y$. Given w, we show in Appendix A, that the averaged distortion measure of K-means once the means have been optimized is exactly equal to $(y^\top Xw)^2/\|\Pi_n y\|_2^2$. ■

Algorithm. The proposition above leads to an alternating optimization algorithm. Note that K-means in one dimension may be run *exactly* in $O(n \log n)$ (Bellman, 1973). After having optimized with respect to w in Eq. (2), we then need to maximize with respect to y the function $\frac{y^\top X(X^\top X)^{-1}X^\top y}{\|\Pi_n y\|_2^2}$, which happens to be exactly performing K-means on the whitened data (which is now in high dimension and not in 1 dimension). At first, it seems that dimension reduction is *simply* equivalent to whitening the data and performing K-means; while this is a formally correct statement, the resulting K-means problem is not easy to solve as the clustered dimension is hidden in noise; for example, algorithms such as K-means++ (Arthur and Vassilvitskii, 2007), which have a multiplicative theoretical guarantee on the final distortion measure, are not provably effective here because the minimal final distortion is not small (since the clusters are corrupted by some noisy dimensions), and the multiplicative guarantee is then meaningless.

2.2 Convex relaxation and discriminative clustering

The discriminative clustering formulation in Eq. (1) may be optimized for any $y \in \{-1,1\}^n$ in closed form with respect to b as $b = \frac{1_n^\top (y - Xv)}{n} = \frac{1_n^\top y}{n}$ since X is centered. Substituting b in Eq. (1) leads us to

$$\min_{v\in\mathbb{R}^d} \frac{1}{n}\|\Pi_n y - Xv\|_2^2 = \frac{1}{n}\|\Pi_n y\|_2^2 - \max_{w\in\mathbb{R}^d} \frac{(y^\top Xw)^2}{\|Xw\|_2^2}, \tag{3}$$

where v is obtained from any solution w as $v = w\frac{y^\top Xw}{\|Xw\|_2^2}$. Thus, given

$$\frac{(y^\top 1_n)^2}{n^2} = \frac{1}{n^2}\big(\#\{i, y_i = 1\} - \#\{i, y_i = -1\}\big)^2 = \alpha \in [0,1], \tag{4}$$

which characterizes the asymmetry between clusters and with $\|\Pi_n y\|^2 = n(1-\alpha)$, we obtain from Eq. (3), an equivalent formulation to Eq. (2) (with the added constraint) as

$$\min_{y\in\{-1,1\}^n,\ v\in\mathbb{R}^d} \frac{1}{n}\|\Pi_n y - Xv\|_2^2 \ \text{ such that } \ \frac{(y^\top 1_n)^2}{n^2} = \alpha. \tag{5}$$

This is exactly equivalent to a discriminative clustering formulation with the square loss (Bach and Harchaoui, 2007) with an explicit cluster balance constraint. Consequently we

have formally established that the discriminative clustering formulation in Eq. (5) is related to the joint clustering and dimension reduction formulation in Eq. (2). Following Bach and Harchaoui (2007), we may optimize Eq. (5) in closed form with respect to v as $v = (X^\top X)^{-1} X^\top y$. Substituting v in Eq. (5) leads us to

$$\min_{y \in \{-1,1\}^n} \frac{1}{n} y^\top \big(\Pi_n - X(X^\top X)^{-1} X^\top \big) y \;\; \text{such that} \;\; \frac{(y^\top 1_n)^2}{n^2} = \alpha. \tag{6}$$

This combinatorial optimization problem is NP-hard in general (Karp, 1972; Garey et al., 1976). Hence in practice, it is classical to consider the following convex relaxation of Eq. (6) (Luo et al., 2010). For any admissible $y \in \{-1, +1\}^n$, the matrix $Y = yy^\top \in \mathbb{R}^{n \times n}$ is a rank-one symmetric positive semi-definite matrix with unit diagonal entries and conversely any such Y may be written in the form $Y = yy^\top$ such that y is admissible for Eq. (6). Moreover by rewriting Eq. (6) as

$$\min_{y \in \{-1,1\}^n} \frac{1}{n} \operatorname{tr} yy^\top \big(\Pi_n - X(X^\top X)^{-1} X^\top \big) \;\; \text{such that} \;\; \frac{1_n^\top (yy^\top) 1_n}{n^2} = \alpha,$$

we see that the objective and constraints are linear in the matrix $Y = yy^\top$ and Eq. (6) is equivalent to

$$\min_{Y \succcurlyeq 0,\ \operatorname{rank}(Y)=1,\ \operatorname{diag}(Y)=1} \frac{1}{n} \operatorname{tr} Y \big(\Pi_n - X(X^\top X)^{-1} X^\top \big) \text{ such that } \frac{1_n^\top Y 1_n}{n^2} = \alpha.$$

Then dropping the non-convex rank constraint leads us to the following classical convex relaxation:

$$\min_{Y \succcurlyeq 0,\ \operatorname{diag}(Y)=1} \frac{1}{n} \operatorname{tr} Y \big(\Pi_n - X(X^\top X)^{-1} X^\top \big) \text{ such that } \frac{1_n^\top Y 1_n}{n^2} = \alpha. \tag{7}$$

This is the standard (unregularized) formulation, which is cast as a semi-definite program. The complexity of interior-point methods is $O(n^7)$, but efficient algorithms in $O(n^2)$ for such problems have been developed due to the relationship with the max-cut problem (Journée et al., 2010; Wen et al., 2012). We note that convex relaxation techniques are also used for semi-supervised methods (De Bie and Cristianini, 2003).

Given the solution Y, one may traditionally obtain a candidate $y \in \{-1, 1\}^n$ by running K-means on the largest eigenvector of Y or by sampling (Goemans and Williamson, 1995). In this paper, we show in Section 5 that it may be advantageous to consider the first two eigenvectors.

2.3 Unsuccessful full convex relaxation

The formulation in Eq. (7) imposes an extra parameter α that characterises the cluster imbalance. It is tempting to find a direct relaxation of Eq. (2). It turns out to lead to a trivial relaxation, which we outline below.

When optimizing Eq. (2) with respect to w, we obtain the following optimization problem

$$\max_{y \in \{-1,1\}^n} \frac{y^\top X (X^\top X)^{-1} X^\top y}{y^\top \Pi_n y},$$

leading to a quasi-convex relaxation as

$$\max_{Y \succcurlyeq 0,\ \operatorname{diag}(Y)=1} \frac{\operatorname{tr} YX(X^\top X)^{-1} X^\top}{\operatorname{tr} \Pi_n Y},$$

whose solution is found by solving a sequence of convex problems (Boyd and Vandenberghe, 2004, Section 4.2.5). As shown in Appendix B, this may be exactly reformulated as a single convex problem:

$$\max_{M \succcurlyeq 0,\ \operatorname{diag}(M) = 1 + \frac{1_n^\top M 1}{n^2}} \operatorname{tr} MX(X^\top X)^{-1} X^\top.$$

Unfortunately, this relaxation always leads to trivial solutions, and we thus need to consider the relaxation in Eq. (7) for several values of $\alpha = 1_n^\top Y 1_n / n^2$ (and then the non-convex algorithm can be run from the rounded solution of the convex problem, using Eq. (2) as a final objective). Alternatively, we may solve the following *penalized* problem for several values of $\nu \geqslant 0$:

$$\min_{Y \succcurlyeq 0,\ \operatorname{diag}(Y)=1} \frac{1}{n} \operatorname{tr} Y \big(\Pi_n - X(X^\top X)^{-1} X^\top \big) + \frac{\nu}{n^2} 1_n^\top Y 1_n. \tag{8}$$

For $\nu = 0$, $Y = 1_n 1_n^\top$ is always a trivial solution. As outlined in our theoretical section and as observed in our experiments, it is sufficient to consider $\nu \in [0, 1]$.

By convex duality (Borwein and Lewis, 2000, Sec. 4.3), both constrained relaxation in Eq. (7) and penalized relaxation in Eq. (8) are formally equivalent for specific choices of constraint parameter α and penalization parameter ν. We will see in Section 6 that the formulation in Eq. (8) is more suitable for algorithmic design (Bach et al., 2012).

2.4 Equivalent relaxations

Optimizing Eq. (5) with respect to v in closed form as in Section 2.2 is feasible with no regularizer or with a quadratic regularizer. However, if one needs to add more complex regularizers, we need a different relaxation. Therefore, we now propose a new formulation of the discriminative clustering framework. We start from the penalized version of Eq. (5),

$$\min_{y \in \{-1,1\}^n,\ v \in \mathbb{R}^d} \frac{1}{n} \| \Pi_n y - Xv \|_2^2 + \nu \frac{(y^\top 1_n)^2}{n^2}, \tag{9}$$

which we expand as:

$$\min_{y \in \{-1,1\}^n,\ v \in \mathbb{R}^d} \frac{1}{n} \operatorname{tr} \Pi_n yy^\top - \frac{2}{n} \operatorname{tr} Xvy^\top + \frac{1}{n} \operatorname{tr} X^\top X vv^\top + \nu \frac{(y^\top 1_n)^2}{n^2}, \tag{10}$$

and relax as, using $Y = yy^\top$, $P = yv^\top$ and $V = vv^\top$,

$$\min_{V,P,Y} \frac{1}{n} \operatorname{tr} \Pi_n Y - \frac{2}{n} \operatorname{tr} P^\top X + \frac{1}{n} \operatorname{tr} X^\top X V + \nu \frac{1_n^\top Y 1_n}{n^2} \text{ s.t. } \begin{pmatrix} Y & P \\ P^\top & V \end{pmatrix} \succcurlyeq 0,\ \operatorname{diag}(Y) = 1. \tag{11}$$

When optimizing Eq. (11) with respect to V and P, we get exactly Eq. (8). Indeed, the optimum is attained for $V = (X^\top X)^{-1} X^\top Y X (X^\top X)^{-1}$ and $P = YX(X^\top X)^{-1}$ as shown in Appendix C.1. Therefore, the convex relaxation in Eq. (11) is equivalent to Eq. (8).

However, we get an interesting behavior when optimizing Eq. (11) with respect to P and Y also in closed form. For $\nu = 1$, we obtain, as shown in Appendix C.2, the following closed form expressions:

$$\begin{aligned} Y &= \operatorname{Diag}(\operatorname{diag}(XVX^\top))^{-1/2} XVX^\top \operatorname{Diag}(\operatorname{diag}(XVX^\top))^{-1/2} \\ P &= \operatorname{Diag}(\operatorname{diag}(XVX^\top))^{-1/2} XV, \end{aligned}$$

leading to the problem:

$$\min_{V \succcurlyeq 0} \quad 1 - \frac{2}{n} \sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} + \frac{1}{n} \operatorname{tr}(VX^\top X). \tag{12}$$

The formulation above in Eq. (12) is interesting for several reasons: (a) it is formulated as an optimization problem in $V \in \mathbb{R}^{d \times d}$, which will lead to algorithms whose running time will depend on n linearly (see Section 6), (b) it allows for easy adding of regularizers (see Section 3), which may be formulated as convex functions of $V = vv^\top$. At first sight this seems to be valid only for $\nu = 1$. However we now propose a reformulation which can handle all possible $\nu \in [0, 1)$ through a simple data augmentation.

Reformulation for any ν When $\nu \in [0, 1)$, we may reformulate the objective function in Eq. (9) as follows:

$$\begin{aligned} \frac{1}{n}\|\Pi_n y - Xv\|_2^2 + \nu \frac{(y^\top 1_n)^2}{n^2} &= \frac{1}{n}\|\Pi_n y - Xv + \nu \frac{y^\top 1_n}{n} 1_n\|_2^2 - \big(\nu \frac{y^\top 1_n}{n}\big)^2 + \nu \big(\frac{y^\top 1_n}{n}\big)^2 \\ &= \frac{1}{n}\|y - Xv - (1-\nu)\frac{y^\top 1_n}{n} 1_n\|_2^2 + \frac{\nu}{1-\nu}\big((1-\nu)\frac{y^\top 1_n}{n}\big)^2 \\ &= \min_{b \in \mathbb{R}} \frac{1}{n}\|y - Xv - b1_n\|_2^2 + \frac{\nu}{1-\nu} b^2, \end{aligned} \tag{13}$$

since $\frac{1}{n}\|y - Xv - b1_n\|_2^2 + \frac{\nu}{1-\nu} b^2$ can be optimized in closed form with respect to b as $b = (1-\nu)\frac{y^\top 1_n}{n}$. Note that the weighted imbalance ratio $(1-\nu)\frac{y^\top 1_n}{n}$ is made as an optimization variable in Eq. (13). Thus we have the following reformulation

$$\begin{aligned} &\min_{v \in \mathbb{R}^d,\ y \in \{-1,1\}^n} \frac{1}{n}\|\Pi_n y - Xv\|_2^2 + \nu \frac{(y^\top 1_n)^2}{n^2} \\ = &\min_{v \in \mathbb{R}^d,\ b \in \mathbb{R},\ y \in \{-1,1\}^n} \frac{1}{n}\|y - Xv - b1_n\|_2^2 + \frac{\nu}{1-\nu} b^2, \end{aligned} \tag{14}$$

which is a non-centered penalized formulation on a higher-dimensional problem in the variable $\binom{v}{b} \in \mathbb{R}^{d+1}$. In the rest of the paper, we will focus on the case $\nu = 1$ for ease of exposition. This enables the use of the formulation in Eq. (12), which is easier to optimize. It is worth noting that this is not an algorithmic restriction. Of course any problem with $\nu \in [0, 1)$ can be treated with equal ease by adding a constant term and a quadratic regularizer.

3. Regularization

There are several natural possibilities. We consider norms Ω such that $\Omega(w)^2 = \Gamma(ww^\top)$ for a certain convex function Γ; all norms have that form (Bach et al., 2012, Proposition 5.1). When $\nu = 1$, Eq. (12) then becomes

$$\max_{V \succcurlyeq 0} \frac{2}{n} \sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} - \frac{1}{n} \operatorname{tr}(VX^\top X) - \Gamma(V). \tag{15}$$

The quadratic regularizers $\Gamma(V) = \operatorname{tr} \Lambda V$ have already been tackled by Bach and Harchaoui (2007). They consider the regularized version of problem in Eq. (3)

$$\min_{v \in \mathbb{R}^d} \frac{1}{n}\|\Pi_n y - Xv\|_2^2 + v^\top \Lambda v, \tag{16}$$

optimize in closed form with respect to v as $v = (X^\top X + n\Lambda)^{-1} X^\top y$. Substituting v in Eq. (16) leads them to

$$\min_{Y \succcurlyeq 0,\ \operatorname{diag}(Y) = 1} \frac{1}{n} \operatorname{tr} Y\big(\Pi_n - X(X^\top X + n\Lambda)^{-1} X\big).$$

In this paper, we propose a novel sparse extension to discriminative clustering framework with the square loss. Specifically we formulate a non-trivial sparse regularizer which is a combination of weighted squared ℓ_1-norm and ℓ_2-norm. It leads to

$$\Gamma(V) = \operatorname{tr}[\operatorname{Diag}(a) V \operatorname{Diag}(a)] + \| \operatorname{Diag}(c) V \operatorname{Diag}(c)\|_1, \tag{17}$$

such that $\Gamma(vv^\top) = \sum_{i=1}^d a_i^2 v_i^2 + \big(\sum_{i=1}^d c_i |v_i|\big)^2$. This allows to treat all situations simultaneously, with $\nu = 1$ or with $\nu \in [0, 1)$. To be more precise, when $\nu \in [0, 1)$, we can consider in Eq. (14), a problem of size $d+1$ with a design matrix $[X, 1_n] \in \mathbb{R}^{n \times (d+1)}$, a direction of projection $\binom{v}{b} \in \mathbb{R}^{d+1}$ and different weights for the last variable with $a_{d+1} = \frac{\nu}{1-\nu}$ and $c_{d+1} = 0$.

Note that the sparse regularizers on V introduced in this paper are significantly different when compared to the sparse regularizers on variable v in Eq. (3), for example, considered by Wang et al. (2013). A straightforward sparse regularizer on v in Eq. (3), despite leading to a sparse projection, does not yield natural generalizations of the discriminative clustering framework in terms of theory or algorithms.

In our analysis and experiments for the balanced clusters (when $\nu = 1$), the sparse regularization $\Gamma(\cdot) = \lambda \|\cdot\|_1$, for $\lambda \in \mathbb{R}$ will often be considered. This is equivalent to setting $a = 0_d$ and $c = \sqrt{\lambda} 1_d$ in Eq. (17). The problem in Eq. (15) then becomes

$$\max_{V \succcurlyeq 0} \frac{2}{n} \sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} - \frac{1}{n} \operatorname{tr}(VX^\top X) - \lambda \|V\|_1. \tag{18}$$

The sparse regularizers considered in this paper have a significant algorithmic appeal for certain applications in computer vision (Bojanowski et al., 2013; Alayrac et al., 2016), audio processing (Lajugie et al., 2016) and natural language processing (Grave, 2014). They also lead to robust cluster recovery under minor assumptions as will be illustrated on a simple example in Section 5. The practical benefits of the sparse regularizers will be further demonstrated using empirical evaluation on synthetic and real data sets in Section 7.

4. Extension to Multiple Labels

The discussion so far has focussed on two clusters. Yet it is key in practice to tackle more clusters. It is worth noting that the discrete formulations in Eq. (2) and Eq. (5) extend directly to more than two clusters. However two different extensions of the initial problems Eq. (2) or Eq. (5) are conceivable. They lead to problems with different constraints on different optimization domains and, consequently, to different relaxations. We discuss these possibilities next.

One extension is the *multi-class* case. The multi-class problem which is dealt with by Bach and Harchaoui (2007) assumes that the data are clustered into K classes and the various partitions of the data points into clusters are represented by the K-class indicator matrices $y \in \{0,1\}^{n\times K}$ such that $y1_K = 1_n$. The constraint $y1_K = 1_n$ ensures that one data point belongs to only one cluster. However as discussed by Bach and Harchaoui (2007), by letting $Y = yy^\top$, it is possible to lift these K-class indicator matrices into the outer convex approximations $\mathcal{C}_K = \{Y \in \mathbb{R}^{n\times n} : Y = Y^\top, \operatorname{diag}(Y) = 1_n, Y \succcurlyeq 0, Y \preccurlyeq \frac{1}{K}1_n 1_n^\top\}$ (Frieze and Jerrum, 1995), which is different for all values of K. Note that letting $K = 2$ corresponds to the previous sections.

In this paper, we consider a different novel extension for discriminative clustering to the *multi-label* case. The multi-label problem assumes that the data share k labels and the data-label membership is represented by matrices $y \in \{-1,+1\}^{n\times k}$. In other words, the multi-class problem embeds the data in the extreme points of a simplex, while the multi-label problem does so in the extreme points of the hypercube.

The discriminative clustering formulation of the multi-label problem is

$$\min_{v\in\mathbb{R}^{d\times k},\ y\in\{-1,1\}^{n\times k}} \frac{1}{n}\|\Pi_n y - Xv\|_F^2, \tag{19}$$

where the Frobenius norm is defined for any vector or rectangular matrix as $\|A\|_F^2 = \operatorname{tr} AA^\top = \operatorname{tr} A^\top A$. Letting $k = 1$ here corresponds to the previous sections. The discrete ensemble of matrices $y \in \{-1,+1\}^{n\times k}$ can be naturally lifted into $\mathcal{D}_k = \{Y \in \mathbb{R}^{n\times n} : Y = Y^\top, \operatorname{diag}(Y) = k1_n, Y \succcurlyeq 0\}$, since $\operatorname{diag}(Y) = \operatorname{diag}(yy^\top) = \sum_{i=1}^k y_{i,i}^2 = k$. As the optimization problems in Eq. (7) and Eq. (8) have linear objective functions, we can change the variable from Y to $\tilde{Y} = Y/k$ to change the constraint $\operatorname{diag}(Y) = k1_n$ to $\operatorname{diag}(\tilde{Y}) = 1_n$ without changing the optimizer of the problem. Thus the problems can be solved over the relaxed domain $\mathcal{D} = \{Y \in \mathbb{R}^{n\times n} : Y = Y^\top, \operatorname{diag}(Y) = 1_n, Y \succcurlyeq 0\}$ which is independent of k.

Note that the domain $\mathcal{D}$ is similar to that considered in the problems in Eq. (8) and Eq. (11) and these convex relaxations are the same regardless of the value of k. Hence the multi-label problem is a more natural extension of the discriminative framework, with a slight change in how the labels y are recovered from the solution Y (we discuss this in Section 5.3).

5. Theoretical Analysis

In this section, we provide the first theoretical analysis for the discriminative clustering framework with the square loss. We start with the 2-clusters situation: the non-sparse case is considered first and analysis is provided for both balanced and imbalanced clusters. Our study for the sparse case currently only provides results for the simple 1-sparse solution. However, the analysis also yields valuable insights on the scaling between n and d. We then derive results for multi-label situation.

For ease of analysis, we consider the constrained problem in Eq. (7), the penalized problem in Eq. (8) or their equivalent relaxations in Eq. (12) or Eq. (18) under various scenarios, for which we use the same proof technique. We first try to characterize the low-rank solutions of these relaxations and then show in certain simple situations the uniqueness of such solutions, which are then non-ambiguously found by convex optimization. Perturbation arguments could extend these results by weakening our assumptions but are not within the scope of this paper, and hence we do not investigate them further in this section.

5.1 Analysis for two clusters: non-sparse problems

In this section, we consider several noise models for the problem, either adding irrelevant dimensions or perturbing the label vector with noise. We consider these separately for simplicity, but they could also be combined (with little extra insight).

5.1.1 Irrelevant dimensions

We consider an "ideal" design matrix $X \in \mathbb{R}^{n\times d}$ such that there exists a direction v along which the projection Xv is perfectly clustered into two distinct real values c_1 and c_2. Since Eq. (2) is invariant by affine transformation, we can rotate the design matrix X to have $X = [y, Z]$ with $y \in \{-1,1\}^n$, which is clustered into $+1$ or -1 along the direction $v = \binom{1}{0_{d-1}}$. Then after being centered, the design matrix is written as $X = [\Pi_n y, Z]$ with $Z = [z_1, \dots, z_{d-1}] \in \mathbb{R}^{n\times(d-1)}$. The columns of Z represent the noisy irrelevant dimensions added on top of the signal y.

5.1.2 Balanced problem

When the problem is well balanced ($y^\top 1_n = 0$), y is already centered and $\Pi_n y = y$. Thus the design matrix is represented as $X = [y, Z]$. We consider here the penalized formulation in Eq. (8) with $\nu = 1$ which is the only scenario where we are able to provide a theoretical analysis.

Let us assume that the columns $(z_i)_{i=1,\dots,d-1}$ of Z are i.i.d. with symmetric distribution z, with $\mathbb{E}z = \mathbb{E}z^3 = 0$ and such that $\|z\|_\infty$ is almost surely bounded by $R \geq 0$. We denote by $\mathbb{E}z^2 = m$ its second moment and by $\mathbb{E}z^4/(\mathbb{E}z^2)^2 = \beta$ its (unnormalized) kurtosis.

Surprisingly the clustered vector y happens to generate a solution $yy^\top$ of the relaxation Eq. (8) for all possible values of Z (see Lemma 11 in Appendix D.2). However the problem in Eq. (8) should have a *unique* solution in order to always recover the correct assignment y. Unfortunately the semidefinite constraint $Y \succcurlyeq 0$ of the relaxation makes the second-order information arduous to study. Due to this reason, we consider the other equivalent relaxation in Eq. (12) for which $V_* = vv^\top$ is also solution with $v \propto (X^\top X)^{-1}X^\top y$ (see Lemma 12 in Appendix D.3). Fortunately the semidefinite constraint $V \succcurlyeq 0$ of the problem in Eq. (12) may be ignored since the second-order information in V of the objective function

already provides unicity for the unconstrained problem. Hence we are able to ensure the uniqueness of the solution with high probability.

Proposition 2 *Let us assume $d \geq 3$, $\beta > 1$ and $m^2 \geq \frac{\beta-3}{2(d+\beta-4)}$:*
(a) If $n \geq d^2 R^4 \frac{1+(d+\beta)m^2}{m^2(\beta-1)}$, V_* is the unique solution of the problem in Eq. (12) with high probability.
(b) If $n \geq \frac{d^2R^4}{\min\{m^2(\beta-1), 2m^2, 2m\}}$, v is the principal eigenvector of any solution of the problem in Eq. (12) with high probability.

Let us make the following observations:

- **Proof technique**: The proof relies on a computation of the Hessian of $f(V) = \frac{2}{n}\sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} - \frac{1}{n}\operatorname{tr} X^\top XV$ which is the objective function in Eq. (12). We first derive the expectation of $\nabla^2 f(V)$ with respect to the distribution of X. By the law of large numbers, it amounts to have n going to infinity in $\nabla^2 f(V)$. Then we expand the spectrum of this operator $\mathbb{E}\nabla^2 f(V)$ to lower-bound its smallest eigenvalue. Finally we use concentration theory on matrices, following Tropp (2012), to bound the Hessian $\nabla^2 f(V)$ for finite n.

- **Effect of kurtosis**: We remind that $\beta \geqslant 1$, with equality if and only if z follows a Rademacher law ($\mathbb{P}(z = +1) = \mathbb{P}(z = -1) = 1/2$). Thus, if the noisy dimensions are clustered, then unsurprisingly, our guarantee is meaningless. Note that the constant β behaves like a distance of the distribution z to the Rademacher distribution. Moreover, $\beta = 3$ if z follows a standard normal distribution.

- **Scaling between d and n**: If the noisy variables are not evenly clustered between the same clusters $\{\pm 1\}$ (i.e., $\beta > 1$), we recover a rank-one solution as long as $n = O(d^3)$; while, as long as $n = O(d^2)$, the solution is not unique but its principal eigenvector recovers the correct clustering. Moreover, as explained in the proof, its spectrum would be very spiky.

- The assumption $m^2 \geq \frac{\beta-3}{2(d+\beta-4)}$ is generally satisfied for large dimensions. Note that $m^2 d$ is the total variance of the irrelevant dimensions, and when it is small, i.e., when $m^2 \leq \frac{\beta-3}{2(d+\beta-4)}$, the problem is particularly simple, and we can also show that V_* is the unique solution of the problem in Eq. (12) with high probability if $n \geq \frac{d^2R^4}{m^2}$. Finally, note that for sub-Gaussian distributions (where $\beta \leq 3$), the extra constraint is vacuous, while for super-Gaussian distributions (where $\beta \geq 3$), this extra constraint only appears for small m.

- This result provides the first guarantee for discriminative clustering. However similar theoretical results have been derived for K-means by Ostrovsky et al. (2006) and Gaussian mixtures by Kalai et al. (2010); Moitra and Valiant (2010), where separation conditions between the two clusters are derived, under which the clustering problem is efficiently solved. It would be of great interest to relate these separation conditions to our condition on n and d but this is outside the scope of this work.

5.1.3 Noise robustness for the one dimensional balanced problem

We assume now that the data are one-dimensional and are perturbed by some noise $\varepsilon \in \mathbb{R}^n$ such that $X = y + \varepsilon$ with $y \in \{-1, 1\}^n$. The solution of the relaxation in Eq. (8) recovers the correct y in this setting only when each component of y and $y + \varepsilon$ have the same sign (this is shown in Appendix D.5). This result comes out naturally from the information on whether the signs of y and $y + \varepsilon$ are the same or not. Further if we assume that y and ε are independent, this condition is equivalent to $\|\varepsilon\|_\infty < 1$ almost surely.

5.1.4 Unbalanced problem

When the clusters are imbalanced ($y^\top 1_n \neq 0$), the natural rank-one candidates $Y_* = yy^\top$ and $V_* = vv^\top$ are no longer solutions of the relaxations in Eq. (8) (for $\nu = 1$) and Eq. (12), as proved in Appendix D.6. Nevertheless we are able to characterize some solutions of the penalized relaxation in Eq. (8) for $\nu = 0$.

Lemma 3 *For $\nu = 0$ and for any non-negative $a, b \in \mathbb{R}$ such that $a + b = 1$,*

$$Y = ayy^\top + b1_n 1_n^\top$$

is solution of the penalized relaxation in Eq. (8).

Hence any eigenvector of this solution Y would be supported by the directions y and 1_n. Moreover when the value $\alpha_* = (\frac{1_n^\top y}{n})^2$ is known, it turns out that we can characterize some solutions of the constrained relaxation in Eq. (7), as stated in the following lemma.

Lemma 4 *For $\alpha \geq \alpha_*$,*

$$Y = \frac{1-\alpha}{1-\alpha_*} yy^\top + \Big(1 - \frac{1-\alpha}{1-\alpha_*}\Big) 1_n 1_n^\top$$

is a rank-2 solution of the constrained relaxation in Eq. (7) with constraint parameter α.

The eigenvectors of Y enable to recover y for $\alpha_* \leq \alpha < 1$. We conjecture (and checked empirically) that this rank-2 solution is unique under similar regimes to those considered for the balanced case. The proof would be more involved since, when $\nu \neq 1$, we are not able to derive an equivalent problem in V for the penalized relaxation in Eq. (8) similar to Eq. (12) for the balanced case. We also note that Lemmas 3 and 4 will be direct consequences of Lemma 8 in Section 5.3.

Thus Y being rank-2, one should really be careful and consider the first two eigenvectors when recovering y from a solution Y. This can be done by rounding the principal eigenvector of $\Pi_n Y \Pi_n = \frac{1-\alpha}{1-\alpha_*} \Pi_n y (\Pi_n y)^\top$ as discussed in the following lemma.

Lemma 5 *Let y_{ev} be the principal eigenvector of $\Pi_n Y \Pi_n$ where Y is defined in Lemma 4, then*

$$\operatorname{sign}(y_{ev}) = y.$$

Proof By definition of Y, $y_{ev} = \sqrt{\frac{1-\alpha}{1-\alpha_*}} \Pi_n y$ thus $\operatorname{sign}(y_{ev}) = \operatorname{sign}(\Pi_n y)$ and since $\alpha \leq 1$ then $\operatorname{sign}(\Pi_n y) = \operatorname{sign}(y - \sqrt{\alpha} 1_n) = y$. ■

In practice, contrary to the standard procedure, we should, for any ν, solve the penalized relaxation in Eq. (8) and then do K-means on the principal eigenvector of the centered solution $\Pi_n Y \Pi_n$ instead of the solution Y to recover the correct y. This procedure is followed in our experiments on real-world data in Section 7.2.

5.2 Analysis for two clusters: one-sparse problems

We assume here that the direction of projection v (such that $Xv = y$) is l-sparse (by l-sparse we mean $\|v\|_0 = l$). The ℓ_1-norm regularized problem in Eq. (18) is no longer invariant by affine transformation and we cannot consider that $X = [y, Z]$ without loss of generality. Yet the relaxation Eq. (18) seems experimentally to only have rank-one solutions for the simple $l = 1$ situation. Hence we are able to derive some theoretical analysis only for this case. It is worth noting the $l = 1$ case is simple since it can be solved in $O(d)$ by using K-means separately on all dimensions and ranking them. Nonetheless the proposed scaling also holds in practice for $l \geqslant 1$ (see Figure 1b).

Thereby we consider data $X = [y, Z]$ with $y \in \{-1, 1\}^n$ and $Z \in \mathbb{R}^{n \times (d-1)}$ which are clustered in the direction $v = [1, 0, \dots, 0]^\top \in \mathbb{R}^d$. When adding a ℓ_1-penalty, the initial problem in Eq. (5) for $\alpha = 0$ is

$$\min_{y \in \{-1,1\}^n,\ v \in \mathbb{R}^d} \frac{1}{n} \|y - Xv\|_2^2 + \lambda \|v\|_1^2. \tag{20}$$

When optimizing in v this problem is close to the Lasso (Tibshirani, 1996) and a solution is known to be $v_i^* = (y^\top y + n\lambda)^{-1} y^\top y = \frac{1}{1+\lambda}$, $\forall i \in J$ and $v_i^* = 0$, $\forall i \in \{1, 2, \dots, d\} \setminus J$, where J is the support of v^*. The candidate $V_* = v^* v^{*\top}$ is still a solution of the relaxation in Eq. (18) (see Lemma 15 in Appendix E.1) and we will investigate under which conditions on X this solution is unique. Let us assume as before $(z_i)_{i=1,\dots,d}$ are i.i.d. with distribution z symmetric with $\mathbb{E}z = \mathbb{E}z^3 = 0$, and denote by $\mathbb{E}z^2 = m$ and $\mathbb{E}z^4/(\mathbb{E}z^2)^2 = \beta$. We also assume that $\|z\|_\infty$ is almost surely bounded by $0 \leq R \leq 1$. We are able to ensure the uniqueness of the solution with high-probability.

Proposition 6 *Let us assume $d \geq 3$.*
(a) If $n \geq dR^2 \frac{1+(d+\beta)m^2}{m^2(\beta-1)}$, V_* is the unique solution of the problem Eq. (12) with high probability.
(b) If $n \geq \frac{dR^2}{m^2(\beta-1)}$, v^* is the principal eigenvector of any solution of the problem Eq. (12) with high probability.

The proof technique is very similar to the one of Proposition 2. With the function $g(V) = \frac{2}{n} \sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} - \lambda \|V\|_1 - \frac{1}{n} \operatorname{tr} X^\top XV$, we can certify that g will decrease around the solution V_* by analyzing the eigenvalues of its Hessian.

The rank-one solution V_* is recovered by the principal eigenvector of the solution of the relaxation Eq. (18) as long as $n = O(d)$. Thus we have a much better scaling when compared to the non-sparse setting where $n = O(d^2)$. We also conjecture a scaling of order $n = O(ld)$ for a projection in a l-sparse direction (see Figure 1b for empirical results).

The proposition does not state any particular value for the regularizer parameter λ. This makes sense since the proposition only holds for the simple situation when $l = 1$. We propose to use $\lambda = 1/\sqrt{n}$ by analogy with the Lasso.

5.3 Analysis for the multi-label extension

In this section, the signals share k labels which are corrupted by some extra noisy dimensions. We assume the centered design matrix to be $X = [\Pi_n y, Z]$ where $y \in \{-1, +1\}^{n \times k}$ and $Z \in \mathbb{R}^{n \times (d-k)}$. We also assume that y is full-rank[1]. We denote by $y = [y_1, \dots, y_k]$ and $\alpha_i = \left(\frac{y_i^\top 1_n}{n}\right)^2$ for $i = 1, \cdots, k$. We consider the discrete constrained problem

$$\min_{v \in \mathbb{R}^{d \times k},\ y \in \{-1,1\}^{n \times k}} \frac{1}{n} \|\Pi_n y - Xv\|_F^2 \text{ such that } \frac{1_n^\top y y^\top 1_n}{n^2} = \alpha^2, \tag{21}$$

and the discrete penalized problem for $\nu = 0$

$$\min_{v \in \mathbb{R}^{d \times k},\ y \in \{-1,1\}^{n \times k}} \frac{1}{n} \|\Pi_n y - Xv\|_F^2. \tag{22}$$

As explained in Section 4, these two discrete problems admit the same relaxations in Eq. (7) and Eq. (8) we have studied for one label. We now investigate when the solution of the problems in Eq. (21) and in Eq. (22) generate solutions of the relaxations in Eq. (7) and Eq. (8).

By analogy with Lemma 3, we want to characterize the solutions of these relaxations which are supported by the constant vector 1_n and the labels $(y_1, \dots, y_k)$. Their general form is $Y = \tilde{y} A \tilde{y}^\top$ where $A \in \mathbb{R}^{k \times k}$ is symmetric semi-definite positive and $\tilde{y} = [1_n, y]$. However the initial y is easily recovered from the solution Y only when A is diagonal. To that end the following lemma derives some condition under which the only matrix A such that the corresponding Y satisfies the constraint of the relaxations in Eq. (7) and Eq. (8) is diagonal.

Lemma 7 *The solutions of the matrix equation $\operatorname{diag}(\tilde{y} A \tilde{y}^\top) = 1_n$ with unknown variable A are diagonal if and only if the family $\{1_n, (y_i)_{1 \leq i \leq k}, (y_i \odot y_j)_{1 \leq i < j \leq k}\}$ is linearly independent where we denoted by $\odot$ the Hadamard (i.e., pointwise) product between matrices.*

In this way we are able to characterize the solution of relaxations in Eq. (7) and Eq. (8) with the following result:

Lemma 8 *Let us assume that the family $\{1_n, (y_i)_{1 \leq i \leq k}, (y_i \odot y_j)_{1 \leq i < j \leq k}\}$ is linearly independent. If $\alpha \geq \alpha_{\min} = \min_{1 \leq i \leq k} \{\alpha_i\}$ with $(\alpha_i)_{1 \leq i \leq k}$ defined above Eq. (21), the solutions of the constrained relaxation in Eq. (7) supported by the vectors $(1_n, y_1, \cdots, y_k)$ are of the form:*

$$Y = a_0^2 1_n 1_n^\top + \sum_{i=1}^k a_i^2 y_i y_i^\top,$$

where $(a_i)_{0 \leq i \leq k}$ satisfies $\sum_{i=0}^k a_i^2 = 1$ and $a_0^2 + \sum_{i=1}^k a_i^2 \alpha_i = \alpha$.

1. This assumption is fairly reasonable since the probability of a matrix whose entries are i.i.d. Rademacher random variables to be singular is conjectured to be $1/2 + o(1)$ (Bourgain et al., 2010).

Moreover the solutions of the penalized relaxation in Eq. (8) for $\nu = 0$ which are supported by the vectors $(1_n, y_1, \cdots, y_k)$ are of the form:

$$Y = a_0^2 1_n 1_n^\top + \sum_{i=1}^k a_i^2 y_i y_i^\top,$$

where $(a_i)_{0\leq i\leq k}$ satisfies $\sum_{i=0}^k a_i^2 = 1$.

In the *multi-label* case, some combinations of the constant matrix $1_n 1_n^\top$ and the rank-one matrices $y_i y_i^\top$ are solutions of constrained or penalized relaxations. Furthermore, under some assumptions on the labels $(y_i)_{1\leq i\leq k}$, these combinations are the only solutions which are supported by the vectors $(1_n, y_1, \cdots, y_k)$. And we conjecture (and checked empirically) that under assumptions similar to those made for the balanced one-label case, all the solutions of the relaxation are supported by the family $(1_n, y_1, \cdots, y_k)$ and consequently share the same form as in Lemma 8. Thus the eigenvector of the solution Y would be in the span of the directions $(1_n, y_1, \cdots, y_k)$.

Let us consider an eigenvalue decomposition of $Y = FF^\top = \sum_{i=0}^k \lambda_i e_i e_i^\top$ and denote by $M = [a_0 1_n, a_1 y_1, \cdots, a_k y_k]$ where $(a_i)_{0\leq i\leq k}$ are defined in Lemma 8. Since $MM^\top = FF^\top$, there is an orthogonal transformation R such that $FR = M$. We also denote the product FR by $FR = [\xi_0, \cdots, \xi_K]$. We propose now an alternating minimization procedure to recover the labels $(y_1, \cdots, y_k)$ from M.

Lemma 9 *Consider the optimization problem*

$$\min_{M\in\mathcal{M},\ R\in\mathbb{R}^{k\times k}:\ R^\top R = I_k} \|FR - M\|_F^2,$$

where $\mathcal{M} = \{[a_0 1_n, a_1 y_1, \cdots, a_k y_k], a \in \mathbb{R}^{k+1} : \|a\|_2 = 1, y_i \in \{\pm 1\}^n\}$.

Given M, the problem is equivalent to the orthogonal Procrustes problem (Schönemann, 1966). Denote by $U\Delta V^\top$ a singular value decomposition of $F^\top M$. The optimal R is obtained as $R = UV^\top$. While given R, the optimal M is obtained as

$$M = \frac{1}{\sqrt{\|\xi_1\|_1^2 + \|\xi_2\|_1^2 + \ldots + \|\xi_k\|_1^2}} [\|\xi_0\|_1 \operatorname{sign}(\xi_0), \cdots, \|\xi_k\|_1 \operatorname{sign}(\xi_k)].$$

Proof We give only the argument for the optimization problem with respect to M. Given R, the optimization problem in M is equivalent to $\max_{a\in\mathbb{R}^{k+1}:\|a\|_2=1,\ y\in\{-1,1\}^{n\times k}} \operatorname{tr}(FR)^\top M$ and $\operatorname{tr}(FR)^\top M = a_0 \xi_0^\top 1_n + \sum_{i=1}^k a_i \xi_i^\top y_i$. Thus by property of the dual norms the solution is given by $y_i = \operatorname{sign}(\xi_i)$ and $a_i = \frac{\|\xi_i\|_1}{\sqrt{\|\xi_1\|_1^2 + \|\xi_2\|_1^2 + \ldots + \|\xi_k\|_1^2}}$. ■

The minimization problem in Lemma 9 is non-convex; however we observe that performing few alternating optimizations is sufficient to recover the correct $(y_1, \ldots, y_k)$ from M.

5.4 Discussion

In this section we studied the tightness of convex relaxations under simple scenarios where the relaxed problem admits low-rank solutions generated by the solution of the original non-convex problem. Unfortunately the solutions lose the characterized rank when the initial problem is slightly perturbed since the rank of a matrix is not a continuous function. Nevertheless, the spectrum of the new solution is really spiked, and thus these results are quite conservative. We empirically observe that the principal eigenvectors keep recovering the correct information outside these scenarios. However this simple proof mechanism is not easily adaptable to handle perturbed problems in a straightforward way since it is difficult to characterize the properties of eigenvectors of the solution of a semi-definite program. Hence we are able to derive a proper theoretical study only for these simple models.

6. Algorithms

In this section, we present an optimization algorithm which is adapted to large n settings, and avoids the n-dimensional semidefinite constraint.

6.1 Reformulation

We aim to solve the general regularized problem which corresponds to Eq. (15)

$$\max_{V\succcurlyeq 0} \frac{2}{n}\sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} - \frac{1}{n}\operatorname{tr} V(X^\top X + n\operatorname{Diag}(a)^2) - \|\operatorname{Diag}(c)V\operatorname{Diag}(c)\|_1. \tag{23}$$

We consider a slightly different optimization problem:

$$\max_{V\succcurlyeq 0} \frac{1}{n}\sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} - \|\operatorname{Diag}(c)V\operatorname{Diag}(c)\|_1 \quad \text{s.t. } \operatorname{tr} V(\frac{1}{n}X^\top X + \operatorname{Diag}(a)^2) = 1. \tag{24}$$

When c is equal to zero, then Eq. (24) is exactly equivalent to Eq. (23); when c is small (as will typically be the case in our experiments), the solutions are very similar—in fact, one can show by Lagrangian duality that by a sequence of problems in Eq. (24), one may obtain the solution to Eq. (23).

6.2 Smoothing

By letting $A = \frac{X^\top X}{n} + \operatorname{Diag}(a)^2$, we consider a strongly-convex approximation of Eq. (24) as:

$$\begin{aligned} \max_{V\succcurlyeq 0} \quad & \frac{1}{n}\sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} - \|\operatorname{Diag}(c)V\operatorname{Diag}(c)\|_1 - \varepsilon \operatorname{tr}[(A^{\frac12}VA^{\frac12})\log(A^{\frac12}VA^{\frac12})] \\ \text{s.t.} \quad & \operatorname{tr}(A^{\frac12}VA^{\frac12}) = 1, \end{aligned} \tag{25}$$

where $-\operatorname{tr} M\log(M)$ is a spectral convex function called the von-Neumann entropy (von Neumann, 1927). The difference in the two problems is known to be $\varepsilon\log(d)$ (Nesterov, 2007). As shown in Appendix G.1, the dual problem is

$$\min_{u\in\mathbb{R}_+^n, C\in\mathbb{R}^{d\times d}:|C_{ij}|\leqslant c_i c_j} \frac{1}{2n}\sum_{i=1}^n \frac{1}{u_i} + \phi^\varepsilon\big(A^{-\frac12}\big(\frac{1}{2n}X^\top\operatorname{Diag}(u)X - C\big)A^{-\frac12}\big), \tag{26}$$

where $\phi^\varepsilon(M)$ is an ε-smooth approximation to the maximal eigenvalue of the matrix M.

6.3 Optimization algorithm

In order to solve Eq. (26), we split the objective function into a smooth part $F(u,C) = \phi^{\varepsilon}\big(A^{-\frac{1}{2}}\big(\frac{1}{2n}X^\top \operatorname{Diag}(u)X - C\big)A^{-\frac{1}{2}}\big)$ and a non-smooth part $H(u,C) = \mathbb{I}_{|C_{ij}|\leqslant c_i c_j} + \frac{1}{2n}\sum_{i=1}^n \frac{1}{u_i}$. We may then apply FISTA (Beck and Teboulle, 2009) updates to the smooth function $\phi^{\varepsilon}(A^{-\frac{1}{2}}(\frac{1}{2n}X^\top \operatorname{Diag}(u)X - C)A^{-\frac{1}{2}})$, along with a proximal operator for the non-smooth terms $\mathbb{I}_{|C_{ij}|\leqslant c_i c_j}$ and $\frac{1}{2n}\sum_{i=1}^n \frac{1}{u_i}$, which may be computed efficiently. See details in Appendix G.2.

Running-time complexity. Since we need to project on the SDP cone of size d at each iteration, the running-time complexity per iteration is $O(d^3 + d^2 n)$; given that often $n \geqslant d$, the dominating term is $O(d^2 n)$. It is still an open problem to make this linear in d. Our function being $O(1/\varepsilon)$-smooth, the convergence rate is of the form $O(1/(\varepsilon t^2))$. Since we stop when the duality gap is $\varepsilon \log(d)$ (as we use smoothing, it is not useful to go lower), the number of iterations is of order $1/(\varepsilon\sqrt{\log(d)})$. The proposed algorithm is a clear improvement over the existing approach by Bach and Harchaoui (2007) which is quadratic in n.

7. Experiments

We implemented the proposed algorithm in Matlab. The code has been made available in `https://drive.google.com/uc?export=download&id=0B5Bx9jrp7celMk5pOFI4UGt0ZEk`. Two sets of experiments were performed: one on synthetically generated data sets and the other on real-world data sets. The details about experiments follow.

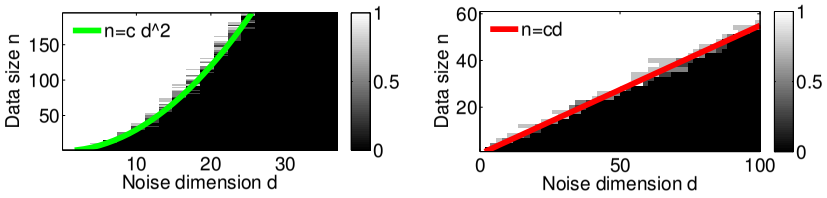
7.1 Experiments on synthetic data

In this section, we illustrate our theoretical results and algorithms on synthetic examples. The synthetic data were generated by assuming a fixed clustering with $\alpha_* \in [0,1]$, along a single direction and the remaining variables were whitened. We consider clustering error defined for a predictor $\bar{y}$ as $1 - (\bar{y}^\top y/n)^2$, with values in $[0,1]$ and equal to zero if and only if $y = \bar{y}$.

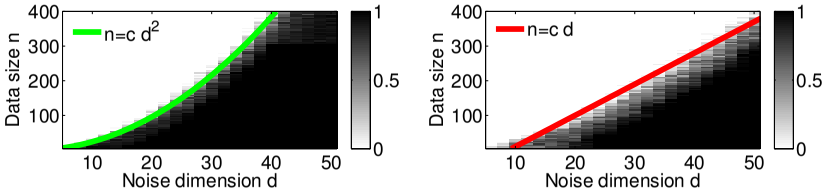
Phase transition. We first illustrate our theoretical results for the balanced case in Figure 1. We solve the relaxation in Eq. (12) and Eq. (18) for a large range of d and n using the `cvx` solver (Grant and Boyd, 2008, 2014). We show the results averaged over 4 replications and take $\lambda = 1/\sqrt{n}$ for the sparse problems. In Figure 1a, we investigate whether `cvx` finds a rank-one solution for a problem of size (n,d) (the value is 1 if the solution is rank-one and 0 otherwise). We compare the performance of the algorithms without ℓ_1-regularization in the affine invariant case and with ℓ_1-regularization in the 1-sparse case. We observe a phase transition with a scaling over the form $n = O(d^2)$ for the affine invariant case and $n = O(d)$ for the 1-sparse case. This is better than what is expected by the theory and corresponds rather to the performance of the principal eigenvector of the solution. It is worth noting that it may be uncertain to really distinguish between a rank-one solution and a spiked solution.

We also solve the relaxation for 4-sparse problems of different sizes d and n and plot the clustering error. We compare, in Figure 1b, the performance of the formulation in Eq. (12)

(without ℓ_1-regularization) which corresponds to the affine invariant case, against the ℓ_1-regularized formulation in Eq. (18). We notice a phase transition of the clustering error with a scaling over the form $n = O(d^2)$ for the affine invariant case and $n = O(d)$ for the 4-sparse case. It supports our conjecture on the scaling of order $n = O(ld)$ for l-sparse problems. Comparing left plots of Figure 1a and Figure 1b, we observe that the two phase-transitions occur at the same scaling between n and d. Thus there are few values of (n,d) for which the `cvx` solver finds a solution whose rank is strictly larger than one and whose principal eigenvector has a low clustering error. This illustrates, in practice, this solver aims to find a rank-one solution under the improved scaling $n = O(d^2)$.



(a) Phase transition for rank-one solution. Left: affine invariant case. Right: 1-sparse case.



(b) Phase transition for clustering error. Left: affine invariant case . Right: 4-sparse case.

Figure 1: Phase transition plots.

Unbalanced case. We generate an unbalanced problem for $d = 10$, $n = 80$ and $\alpha_* = 0.25$ and we average the results over 10 replications. We compare the clustering error for the constrained and the penalized relaxations in Eq. (7) and Eq. (8) when we consider the sign of the first or second eigenvector and when we use projection technique defined as $(\Pi_n Y_{(2)} \Pi_n)_{(1)}$ where $Y_{(k)}$ is the best rank-k approximation of Y, to extract the information of y. We see in Figure 2 that (a) for the constrained case, the range of α such that the sign of y is recovered is cut in two parts where one eigenvector is correct, whereas the projection method performs well on the whole set. (b) For the penalized case, the correct sign is

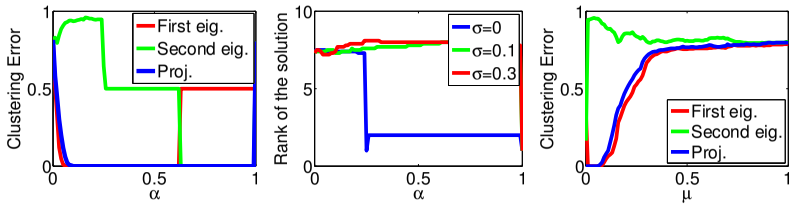


Figure 2: Unbalanced problem for $n = 80$, $d = 10$ and $\alpha_* = 0.25$. Left: Clustering error for the constrained relaxation. Middle: Rank of the solution for different level of noise σ. Right: Clustering error for the penalized relaxation.

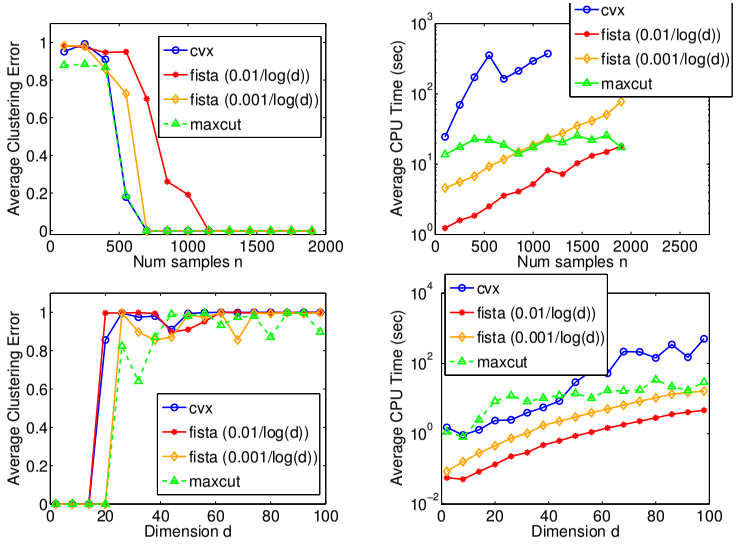
recovered for ν close to 0 by the first eigenvector and the projection method whereas the second one performs always badly. (c) When there is zero noise the rank of the solution is one for $\alpha \in \{\alpha_*, 1\}$, two for $\alpha \in (\alpha_*, 1)$ and greater otherwise. These findings confirm our analysis. However, when y is corrupted by some noise this result is no longer true.

Runtime experiments. We generated data with a k-sparse direction of projection v by adding $d - k$ noise variables to a randomly generated and rotated k-dimension data. The scalability of the FISTA based optimization algorithm illustrated in Section 6.3 to solve Eq. (24) (with $c = \sqrt{\lambda} 1_d$, $a = 0_d$) was compared against a benchmark `cvx` solver (which solves Eq. (18)). Experiments were performed for $\lambda = 0$ and $\lambda = 0.001$, the coefficient associated with the sparse $\|V\|_1$ term. For a fixed d, `cvx` breaks down for large n values (typically $n \geqslant 1000$). Similarly, the runtime required by `cvx` is generally high for $\lambda = 0$ and is comparable to our method for $\lambda = 0.001$. This behavior is illustrated in Figure 3.

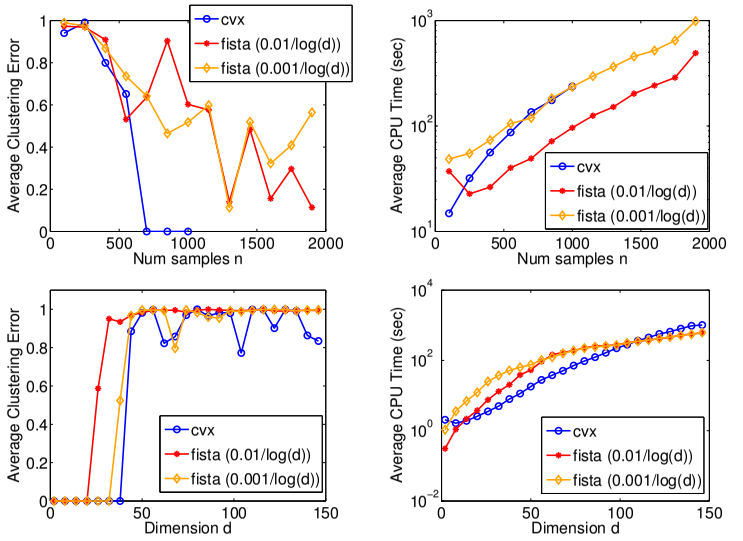
When $\lambda = 0$, the problem reduces exactly to the original Diffrac problem (Bach and Harchaoui, 2007). In the plots in Figure 3a our implementation using FISTA is compared to the baseline Diffrac which is solved with max-cut SDP (Boumal et al., 2014). We observed that our method is comparable in terms of runtime and clustering performance of low-rank methods for max-cut. However, for $\lambda > 0$, the equivalence with max-cut disappears.

The plots in these figures show the behavior of FISTA for two different stopping criteria: $\varepsilon = 10^{-2}/\log(d)$ and $\varepsilon = 10^{-3}/\log(d)$. It is observed that the choice $10^{-3}/\log(d)$ gives a better accurate solution at the cost of more number of iterations (and hence higher runtime). For sparse problems in Figure 3b, we see that cvx gets a better clustering performance (while crashing for large n); the difference would be reduced with a smaller duality gap for FISTA.

Clustering performance. Experiments comparing the proposed method (Eq. (24) with $c = \sqrt{\lambda} 1_d$ and $a = 0_d$ solved using FISTA based optimization algorithm, and Eq. (18) solved using benchmark `cvx` solver) with K-means and alternating optimization are given in Figure 4. K-means is run on the whitened variables in $\mathbb{R}^d$. Alternating optimization is another popular method proposed by Ye et al. (2008) for dimensionality reduction with clustering (where alternating optimization of w and y is performed to solve the non-convex formulation (2)). The plots show that both K-means and alternating optimization fail when only a few dimensions of noise variables are present. The plots also show that with the introduction of a sparse regularizer (corresponding to the non-zero λ) the proposed



(a) `cvx`, max-cut comparison with $\lambda = 0$. Top: n varied with $d = 50$, $k = 6$. `cvx` crashed for $n \approx 1000$. Bottom: d varied with $n = 100$, $k = 2$.

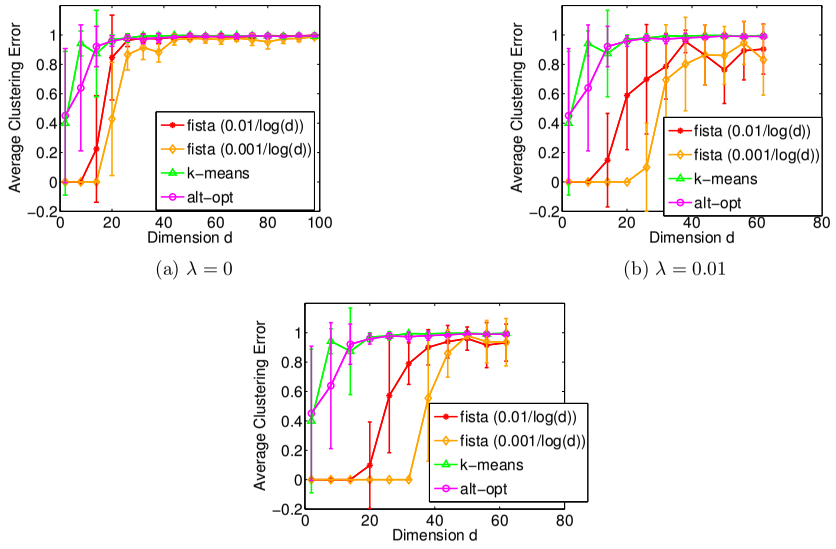


(b) `cvx` comparison with $\lambda = 0.001$. Top: n varied with $d = 50$, $k = 6$. `cvx` crashed for $n \approx 1000$. Bottom: d varied with $n = 100$, $k = 2$.

Figure 3: Scalability experiments.

method becomes more robust to noisy dimensions. As observed earlier, the performance of FISTA is also sensitive to the choice of ε.

Finally we give a comparison of sparse discriminative clustering (cvx and FISTA) with max-margin clustering (Li et al., 2009) in Figure 5. We note that square loss is used in our framework whereas hinge loss is used in max-margin clustering. We have also included the behavior of K-means and alternating optimization methods in Figure 5 for completeness. From this plot, it is clear that the max-margin clustering is sensitive to noisy dimensions present in the data. Sparse discriminative clustering with square loss is able to maintain zero cluster error for a large number of noisy dimensions, while the performance of max-margin clustering starts deteriorating after adding a few noisy dimensions. However, we note from Figure 5 that for large dimensions, the hinge loss used in max-margin clustering is observed to provide a better solution than the square loss used in our framework.



(a) $\lambda = 0$ (b) $\lambda = 0.01$

(c) $\lambda = 0.001$

Figure 4: Comparison with k-means and alternating optimization, $n = 100$.

7.2 Experiments on real-world data

Experiments on two-class data. Experiments were conducted on real two-class classification datasets[2] to compare the performance of sparse discriminative clustering against non-sparse discriminative clustering, alternating optimization, K-means and max-margin

2. The data sets were obtained from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

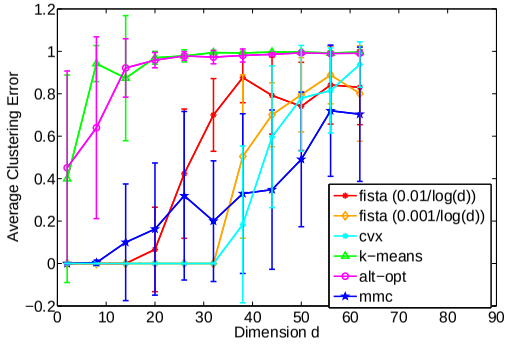


Figure 5: Comparison with k-means, alternating optimization and max-margin clustering (mmc), $n = 100$. The plots for FISTA, `cvx` and mmc correspond to the best choice of regularization parameters.

clustering algorithms. For sparse and non-sparse discriminative clustering, we consider the problem in Eq. (24) and the algorithm detailed in Section 6.3 (with the regularization $c = 0$ for the non-sparse case). The alternating optimization method is described in Proposition 1. For the two-class datasets, the clustering performance for a cluster $\bar{y} \in \{+1, -1\}^n$ obtained from an algorithm under comparison, was computed as $1-(\bar{y}^\top y/n)^2$, where y is the original labeling. Here we explicitly compare the output of clustering with the original labels of the data points.

The dataset details and clustering performance results are summarized in Table 1. The experiments for discriminative clustering were conducted for different values of $a, c \in \{10^{-3}, 10^{-2}, 10^{-1}\}1_d$ associated with the ℓ_2-regularizer and ℓ_1-regularizer respectively. The range of cluster imbalance parameter was chosen to be $\nu \in \{0.01, 0.25, 0.5, 0.75, 1\}$. Note that for $\nu \neq 1$, the reformulation given in Eq. (14) was used, as explained after Eq. (17) in Section 3. The results given in Table 1 pertain to the best choices of these parameters. Similarly, the values of regularization parameter for max-margin clustering (Li et al., 2009) were chosen from the set $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10\}$ and the cluster balance parameter was chosen from $\{0.1, 0.2, \ldots, 0.9\}$. The results for alternating optimization and K-means show the average cluster error (and standard deviation) over 10 different runs. These results show that the cluster error is quite high for many datasets. This is primarily due to the absence of an ambient low-dimensional clustering of the two-class data, which can be identified by the simple linear model presented in this paper. Since K-means does not provide explicit dimensionality reduction, it might not be able to take advantage of the existence of an ambient low-dimensional clustering of the two-class data and its performance is poor. The results show that max-margin clustering achieves best clustering performance on most datasets. This improved performance of max-margin clustering may be due to the use of hinge loss, as opposed to square loss used in discriminative clustering in this paper. However for the heart dataset, we note that the sparse version with the square loss performs

significantly better than the non-sparse version with the hinge loss (see additional experiments in Figure 5). The results also show that adding sparse regularizers to discriminative clustering helps in a better cluster identification when compared to the non-sparse case.

Table 1: Experiments on two-class datasets

Dataset	n	d	Cluster Error				
			Sparse Discriminative Clustering	Non-sparse Discriminative Clustering	Alternating Optimization	K-means	Max-margin Clustering
Heart	270	3	**0.52**	0.61	0.97 ± 0.03	0.91 ± 0.09	0.93
Diabetes	768	8	**0.88**	**0.88**	0.91 ± 0.05	0.93 ± 0.06	**0.88**
Breast-cancer	683	10	**0.15**	**0.15**	0.48 ± 0.17	0.68 ± 0.24	**0.15**
Australian	690	14	**0.5**	**0.5**	0.88 ± 0.17	0.87 ± 0.21	**0.5**
Liver-disorder	345	6	0.97	0.97	0.99 ± 0.01	0.99 ± 0.01	**0.73**
Sonar	208	60	**0.92**	0.95	0.98 ± 0.02	0.99 ± 0.01	**0.92**
DNA(1 vs 2,3)	1400	180	0.75	0.83	0.99 ± 0.01	0.98 ± 0.02	**0.71**
a1a	1605	113	0.74	0.75	0.98 ± 0.02	0.8 ± 0.08	**0.69**
w1a	2270	290	**0.11**	**0.11**	0.92 ± 0.08	0.16 ± 0.06	**0.11**

Experiments on real multi-label data. Experiments were also conducted on the Microsoft COCO dataset[3] to demonstrate the effectiveness of the proposed method in discovering multiple labels. We considered $n = 2000$ images from the dataset, each of which was labeled with a subset of $K = 80$ labels. The labels identified the objects in the images like person, car, chair, table, etc. and the corresponding features for each image were extracted from the last layer of a conventional convolutional neural network (CNN). The CNN was originally trained over the imagenet data (Krizhevsky et al., 2012).

For each image in the dataset, we obtained $d = 1000$ features. We then performed discriminative clustering on the 2000×1000 data matrix X and obtained the label matrix Y which was then subjected to the alternating optimization procedure (see Section 5.3).

It is clearly unlikely to recover perfect labels; therefore we now describe a way of measuring the amount of information which is recovered. In order to extract meaningful cluster information from the result so-obtained, we computed the correlation matrix $Y_k^\top \Pi_n Y_{true}$ where Y_{true} is the $n \times K$ label matrix containing actual labels and Π_n is the $n \times n$ centering matrix $I_n - \frac{1}{n} 1_n 1_n^\top$. The k predicted labels for the examples are present in the Y_k matrix of size $n \times k$. In order to choose an appropriate value of k, we plotted $\mathrm{Tr}(\Phi_{Y_{true}} \Phi_{Y_k})$ (shown in Figure 6 along with a K-means baseline), where $\Phi_{Y_k} = Y_k (Y_k{}^\top Y_k)^{-1} Y_k^\top$. From these plots, we chose $k = 30$ to be a suitable value for our interpretation purposes.

After choosing an arbitrary value of $k = 30$, we plotted the correlations between the actual and predicted labels. The heat map of the normalized absolute correlations is given in Figure 7, where the columns and rows corresponding to the 80 true labels and 30 predicted labels respectively, are ordered according to the sum of squared correlations (the top-scoring labels appear to the left-bottom). From this plot, we extract following highly correlated

3. Dataset obtained from `http://mscoco.org/dataset`

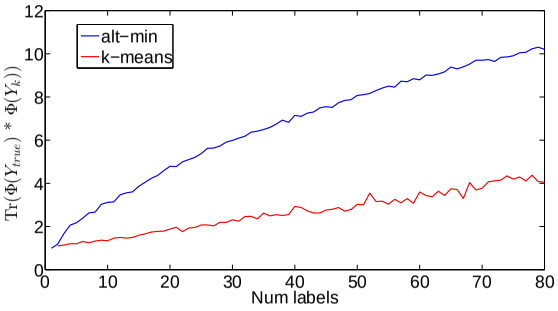


Figure 6: Plot of $\mathrm{Tr}(\Phi_{Y_{true}} \Phi_{Y_k})$.

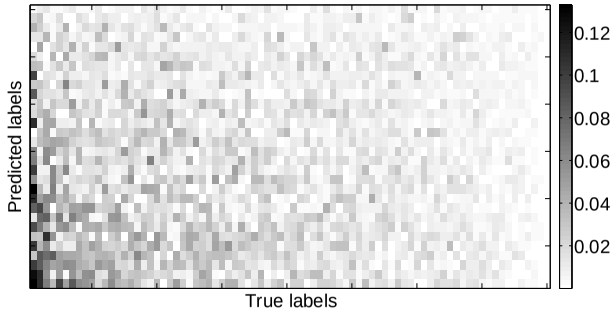


Figure 7: Heat map of correlations, $Y_k^\top \Pi_n Y_{true}$ with $k = 30$, with columns and rows ordered according to the sum of squared correlations.

labels: person, dining table, car, chair, cup, tennis racket, bowl, truck, fork, pizza, showing that these labels were partially recovered by our unsupervised technique (note that the CNN features are learned with supervision on the different dataset Imagenet, hence there is still some partial supervision).

8. Conclusion

In this paper, we provided a sparse extension of the discriminative clustering framework, and gave a first analysis of its theoretical performance in the totally unsupervised situation, highlighting provable scalings between ambient dimension d, number of observations and "clusterability" of irrelevant variables. We also proposed an efficient algorithm which is the first of its kind to be linear in the number of observations for discriminative clustering with the square loss. Our work could be extended in a number of ways, e.g., extending the sparse

analysis to l-sparse case with higher l, extending the framework to nonlinear clustering using kernels, considering related weakly supervised learning extensions (Joulin and Bach, 2012), going beyond uniqueness of rank-one solutions, and improving the complexity of our algorithm to $O(nd)$, for example using stochastic gradient techniques.

Acknowledgments

The authors would like to thank Nicolas Boumal for interesting discussions and for his advice on max-cut implementation, Piotr Bojanowski for his help with multi-label datasets and the reviewers for their constructive and helpful comments which helped to significantly improve the paper.

References

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

D. Arthur and S. Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007.

F. Bach and Z. Harchaoui. DIFFRAC : a discriminative and flexible framework for clustering. In *Advances in Advances in Neural Information Processing Systems (NIPS)*, 2007.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

R. Bellman. A note on cluster analysis and dynamic programming. *Mathematical Biosciences*, 1973.

G. Blanchard, M. Kawanabe, M. Sugiyama, V. Spokoiny, and K.-R. Müller. In search of non-Gaussian components of a high-dimensional distribution. *The Journal of Machine Learning Research*, 7:247–282, 2006.

Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Finding actors and actions in movies. In *Proc. IEEE International Conference on Computer Vision*, 2013.

J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization*, volume 3 of *CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC*. Springer-Verlag, 2000. Theory and examples.

N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab Toolbox for Optimization on Manifolds. *Journal of Machine Learning Research*, 2014.

J. Bourgain, V. H. Vu, and P. M. Wood. On the singularity probability of discrete random matrices. *Journal of Functional Analysis*, 258(2):559–603, 2010.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004.

T. De Bie and N. Cristianini. Convex methods for transduction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 73–80, 2003.

F. De la Torre and T. Kanade. Discriminative cluster analysis. In *Proceedings of the conference on machine learning (ICML)*, 2006.

E. Diederichs, A. Juditsky, A. Nemirovski, and V. Spokoiny. Sparse non-Gaussian component analysis by semidefinite programming. *Machine learning*, 91(2):211–238, 2013.

C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and K-means clustering. In *Proceedings of the conference on machine learning (ICML)*, 2007.

D. Freedman. *Statistical models: theory and practice*. Cambridge University Press, 2009.

J. H. Friedman and W. Stuetzle. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.

A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. In *Integer Programming and Combinatorial Optimization*. Springer, 1995.

M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoret. Comput. Sci.*, 1(3):237–267, 1976.

M. X. Goemans and D. P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM*, 42(6):1115–1145, November 1995.

J. C. Gower and G. J. S. Ross. Minimum spanning trees and single Linkage cluster analysis. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 18(1), 1969.

M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008.

M. Grant and S. Boyd. CVX: Matlab Software for Disciplined Convex Programming, version 2.1, March 2014.

Edouard Grave. A convex relaxation for weakly supervised relation extraction. In *EMNLP*, 2014.

G. Huang, J. Zhang, S. Song, and Z. Chen. Maximin separation probability clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

A. Joulin and F. Bach. A convex relaxation for weakly supervised classifiers. In *Proceedings of the conference on machine learning (ICML)*, 2012.

A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. In *Proc. CVPR*, 2010a.

A. Joulin, J. Ponce, and F. Bach. Efficient optimization for discriminative latent class models. In *Advances in Advances in Neural Information Processing Systems (NIPS)*, 2010b.

M. Journée, F. Bach, P-A Absil, and R. Sepulchre. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, 2010.

A. T. Kalai, A. Moitra, and G. Valiant. Efficiently learning mixtures of two Gaussians. In *International Symposium on Theory of Computing (STOC)*, pages 553–562, 2010.

R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Plenum, New York, 1972.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Advances in Neural Information Processing Systems (NIPS)*, 2012.

Rémi Lajugie, Piotr Bojanowski, Philippe Cuvillier, Sylvain Arlot, and Francis Bach. A weakly-supervised discriminative model for audio-to-score alignment. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2016.

N. Le Roux and F. Bach. Local component analysis. In *Proceedings of the International Conference on Learning Representations*, 2013.

Y.-F. Li, I. W. Tsang, J. T.-Y. Kwok, and Z-H. Zhou. Tighter and convex maximum margin clustering. In *AISTATS*, pages 344–351, 2009.

Z. Q. Luo, W. K. Ma, A. C. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *Signal Processing Magazine, IEEE*, 2010.

J. B. MacQueen. Some Methods for Classification and Analysis of MultiVariate Observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

A. Moitra and G. Valiant. Settling the polynomial learnability of mixtures of Gaussians. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 93–102, 2010.

Y. Nesterov. Smoothing technique and its applications in semidefinite optimization. *Math. Program.*, 2007.

A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an algorithm. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Advances in Neural Information Processing Systems (NIPS)*. 2002.

G. Niu, B. Dai, L. Shang, and M. Sugiyama. Maximum volume clustering: a new discriminative clustering approach. *Journal of Machine Learning Research*, 14(1):2641–2687, 2013.

R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy. The effectiveness of lloyd-type methods for the k-means problem. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 165–176, 2006.

P. H Schönemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31(1), 1966.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Roy. Statist. Soc. Ser. B*, 1996.

J. Tropp. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.*, 2012.

J. von Neumann. Thermodynamik quantummechanischer Gesamheiten. *Gött. Nach*, (1): 273–291, 1927.

F. Wang, B. Zhao, and C. Zhang. Linear time maximum margin clustering. *IEEE Transactions on Neural Networks*, 2010.

H. Wang, F. Nie, and H. Huang. Multi-View Clustering and Feature Learning via Structured Sparsity. In *Proceedings of the conference on machine learning (ICML)*, volume 28, 2013.

Z. Wen, D. Goldfarb, and K. Scheinberg. Block coordinate descent methods for semidefinite programming. In *Handbook on Semidefinite, Conic and Polynomial Optimization*. Springer, 2012.

L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *Advances in Advances in Neural Information Processing Systems (NIPS)*, 2004.

J. Ye, Z. Zhao, and M. Wu. Discriminative k-means for clustering. In *Advances in Advances in Neural Information Processing Systems (NIPS)*, 2008.

K. Zhang, I. W. Tsang, and J. T. Kwok. Maximum margin clustering made practical. *IEEE Transactions on Neural Networks*, 2009.

Appendix A. Joint clustering and dimension reduction

Given y, we need to optimize the Rayleigh quotient $\frac{w^\top X^\top y y^\top X w}{w^\top X^\top X w}$ with a rank-one matrix in the numerator, which leads to $w = (X^\top X)^{-1} X^\top y$. Given w, we will show that the averaged distortion measure of K-means once the means have been optimized is exactly equal to $(y^\top \Pi_n X w)^2 / \|\Pi_n y\|_2^2$. Given the data matrix $X \in \mathbb{R}^{n \times d}$, K-means to cluster the data into two components will tend to approximate the data points in X by the centroids $c_+ \in \mathbb{R}^d$ and $c_- \in \mathbb{R}^d$ such that

$$\begin{aligned} X &\approx \frac{(y+1_n)}{2} c_+^\top - \frac{(y-1_n)}{2} c_-^\top \text{ (since } y \in \{-1,1\}^n) \\ &= \frac{y}{2}(c_+^\top - c_-^\top) + \frac{1}{2} 1_n (c_+^\top + c_-^\top). \end{aligned}$$

The objective of K-means can now be written as problem $\mathcal{KM}$:

$$\begin{aligned} &\min_{y, c_+, c_-} \left\| X - \frac{y}{2}(c_+^\top - c_-^\top) - \frac{1}{2} 1_n (c_+^\top + c_-^\top) \right\|_F^2 \\ &= \min_{y, c_+, c_-} \left\| X - \frac{(y+1_n)}{2} c_+^\top - \frac{(1_n - y)}{2} c_-^\top \right\|_F^2 \\ &= \min_{y, c_+, c_-} \|X\|_F^2 + \|c_+^\top\|_F^2 \left\| \frac{(y+1_n)}{2} \right\|^2 + \|c_-^\top\|_F^2 \left\| \frac{(1-y_n)}{2} \right\|^2 + 2 c_-^\top c_+ \frac{(y+1_n)}{2}^\top \frac{(1_n - y)}{2} \\ &\qquad -2 \operatorname{tr} X^\top \left(\frac{(y+1_n)}{2} c_+^\top + \frac{(1_n - y)}{2} c_-^\top \right) \\ &= \min_{y, c_+, c_-} \|X\|_F^2 + \|c_+^\top\|_F^2 \frac{1}{2}(n + 1_n^\top y) + \|c_-^\top\|_F^2 \frac{1}{2}(n - 1_n^\top y) - 2 c_+^\top X^\top \left(\frac{y+1_n}{2} \right) \\ &\qquad -2 c_-^\top X^\top \left(\frac{1_n - y}{2} \right). \end{aligned}$$

Fixing y and minimizing with respect to c_+ and c_-, we get closed-form expressions for c_+ and c_- as

$$c_+ = \frac{X^\top (y + 1_n)}{(n + 1_n^\top y)} \quad \text{and} \quad c_- = \frac{X^\top (1_n - y)}{(n - 1_n^\top y)}.$$

Substituting these expressions in $\mathcal{KM}$, we have the following optimization problem in y:

$$\begin{aligned}
&\min_y \|X\|_F^2 - \frac{1}{2}\frac{\|X^\top(y+1_n)\|_F^2}{(n+1_n^\top y)} - \frac{1}{2}\frac{\|X^\top(1_n-y)\|_F^2}{(n-1_n^\top y)} \\
&= \min_y \|X\|_F^2 - \frac{1}{2}\frac{\operatorname{tr} XX^\top(y+1_n)(y+1_n)^\top}{(n+1_n^\top y)} - \frac{1}{2}\frac{\operatorname{tr} XX^\top(1_n-y)(1_n-y)^\top}{(n-1_n^\top y)} \\
&= \min_y \|X\|_F^2 - \frac{2}{(n+1_n^\top y)}\operatorname{tr} XX^\top\left(\frac{y+1_n}{2}\right)\left(\frac{y+1_n}{2}\right)^\top \\
&\qquad - \frac{2}{(n-1_n^\top y)}\operatorname{tr} XX^\top\left(\frac{1_n-y}{2}\right)\left(\frac{1_n-y}{2}\right)^\top \\
&= \min_y \ \operatorname{tr} XX^\top - \frac{2}{(n+1_n^\top y)}\operatorname{tr} XX^\top\left(\frac{y+1_n}{2}\right)\left(\frac{y+1_n}{2}\right)^\top \\
&\qquad - \frac{2}{(n-1_n^\top y)}\operatorname{tr} XX^\top\left(\frac{1_n-y}{2}\right)\left(\frac{1_n-y}{2}\right)^\top \\
&= \min_y \ \operatorname{tr} XX^\top\Big(I - \frac{1}{2(n+1_n^\top y)}(yy^\top + 1_n1_n^\top + y1_n^\top + 1_ny^\top) \\
&\qquad - \frac{1}{2(n-1_n^\top y)}(1_n1_n^\top + yy^\top - 1_ny^\top - y1_n^\top)\Big).
\end{aligned}$$

By the centering of X, we have $1_n^\top X = 0$ and hence $\operatorname{tr} XX^\top 1_n1_n^\top = \operatorname{tr} XX^\top 1_ny^\top = \operatorname{tr} XX^\top y1_n^\top = 0$. Therefore, we obtain

$$\begin{aligned}
&\min_y \ \operatorname{tr} XX^\top\Big(I - \frac{1}{2(n+1_n^\top y)}(yy^\top) - \frac{1}{2(n-1_n^\top y)}(yy^\top)\Big) \\
&= \min_y \ \operatorname{tr} XX^\top\Big(I - (yy^\top)\Big(\frac{1}{2(n+1_n^\top y)} + \frac{1}{2(n-1_n^\top y)}\Big)\Big) \\
&= \min_y \ \operatorname{tr} XX^\top\Big(I - (yy^\top)\Big(\frac{n}{n^2-(1_n^\top y)^2)}\Big)\Big) \\
&= \min_y \ \operatorname{tr} XX^\top\Big(I - \frac{nyy^\top}{n^2-(1_n^\top y)^2}\Big).
\end{aligned}$$

Thus we have the equivalent K-means problem as

$$\min_{y\in\{-1,1\}^n} \frac{1}{n}\operatorname{tr} Xww^\top X^\top\Big(I - \frac{n}{n^2-(y^\top 1)^2}yy^\top\Big) = 1 - \max_{y\in\{-1,1\}^n}\frac{(w^\top X^\top y)^2}{n^2-(y^\top 1)^2}.$$

Thus the averaged distortion measure of K-means with the optimized means is $\frac{(y^\top\Pi_nXw)^2}{\|\Pi_ny\|_2^2}$.

Appendix B. Full (unsuccessful) relaxation

It is tempting to find a direct relaxation of Eq. (2). It turns out to lead to a trivial relaxation, which we outline in this section. When optimizing Eq. (2) with respect to w, we obtain $\max_{y\in\{-1,1\}^n}\frac{y^\top X(X^\top X)^{-1}X^\top y}{y^\top\Pi_ny}$, leading to a quasi-convex relaxation as $\max_{\substack{Y\succcurlyeq 0,\\ \operatorname{diag}(Y)=1}}\frac{\operatorname{tr} YX(X^\top X)^{-1}X^\top}{\operatorname{tr}\Pi_nY}$. Unfortunately, this relaxation always leads to trivial solutions as described below.

Consider the quasi-convex relaxation

$$\max_{Y\succcurlyeq 0,\operatorname{diag}(Y)=1}\frac{\operatorname{tr} YX(X^\top X)^{-1}X^\top}{\operatorname{tr}\Pi_nY}. \tag{27}$$

By definition of Π_n this relaxation is equal to:

$$\max_{Y\succcurlyeq 0,\operatorname{diag}(Y)=1}\frac{1}{n}\frac{\operatorname{tr} YX(X^\top X)^{-1}X^\top}{1-\frac{1_n^\top Y1_n}{n^2}}.$$

Let $\mathcal{A} = \{Y \succcurlyeq 0,\ \operatorname{diag}(Y) = 1\}$ the feasible set of this problem and define $\mathcal{B} = \{M \succcurlyeq 0,\ \operatorname{diag}(M) = 1 + \frac{1_n^\top M1_n}{n^2}\}$. Let $Y \in \mathcal{A}$, then M defined by $M = \frac{Y}{1-\frac{1_n^\top Y1_n}{n^2}}$ belongs to $\mathcal{B}$ since $1 + \frac{1_n^\top M1_n}{n^2} = 1 + \frac{1_n^\top Y1_n}{n^2-1_n^\top Y1_n} = \frac{1}{1-\frac{1_n^\top Y1_n}{n^2}} = \operatorname{diag}(M)$. Reciprocally for $M \in \mathcal{B}$, we can define $Y = \frac{M}{1+\frac{1_n^\top M1_n}{n^2}}$, such that $\operatorname{diag}(Y) = 1$ and $Y \in \mathcal{A}$ and then verify that $M = \frac{Y}{1-\frac{1_n^\top Y1_n}{n^2}}$. Thus the problem Eq. (27) is equivalent to the relaxation

$$\max_{M\succcurlyeq 0,\operatorname{diag}(M)=1+\frac{1_n^\top M1_n}{n^2}}\frac{1}{n}\operatorname{tr} MX(X^\top X)^{-1}X^\top. \tag{28}$$

The Lagrangian function of this problem can be written as:

$$\begin{aligned}
L(\mu) &= \operatorname{tr} MX(X^\top X)^{-1}X^\top - \frac{\mu}{n}^\top[\operatorname{diag}(M) - 1_n - \frac{1_n^\top M1_n}{n^2}1_n] \\
&= \operatorname{tr} M[X(X^\top X)^{-1}X^\top - \operatorname{Diag}(\mu) + \frac{1_n^\top\mu}{n^2}1_n1_n^\top] + \frac{1}{n}\mu^\top 1_n.
\end{aligned}$$

Using $L(\mu)$ and the PSD constraint $M \succcurlyeq 0$, the dual problem is given by

$$\min_\mu \frac{\mu^\top 1_n}{n} \quad \text{s.t. } \operatorname{Diag}(\mu) - \frac{1_n^\top\mu}{n^2}1_n1_n^\top \succcurlyeq X(X^\top X)^{-1}X^\top.$$

Since $X(X^\top X)^{-1}X^\top \succcurlyeq 0$, this implies for the dual variable μ:

$$\begin{aligned}
\operatorname{Diag}(\mu) - \frac{1_n^\top\mu}{n^2}1_n1_n^\top \succcurlyeq 0 \ &\Leftrightarrow\ 1_n^\top\operatorname{Diag}(\mu)^{-1}1_n \le \frac{n^2}{\mu^\top 1_n} \\
&\Leftrightarrow\ \sum_{i=1}^n\frac{1}{\mu_i} \le \frac{n^2}{\sum_{i=1}^n\mu_i} \\
&\Leftrightarrow\ \frac{1}{n}\sum_{i=1}^n\frac{1}{\mu_i} \le \frac{1}{\frac{1}{n}\sum_{i=1}^n\mu_i}.
\end{aligned}$$

However for $\nu \in \mathbb{R}^n$, the harmonic mean $\big[\frac{1}{n}\sum_{i=1}^n\frac{1}{\nu_i}\big]^{-1}$ is always smaller than the arithmetic mean $\frac{1}{n}\sum_{i=1}^n\nu_i$ with equality if and only if $\nu = c1_n$ for $c \in \mathbb{R}$.

Thus the dual variable μ is constant and the diagonal constraint in problem Eq. (28) simplifies itself as a trace constraint. Therefore the problem is equivalent to the trivial relaxation below for which each eigenvector of $X(X^\top X)^{-1}X^\top$ is a solution:

$$\max_{M\succcurlyeq 0,\ \operatorname{tr}(M)=n+\frac{1_n^\top M1_n}{n}}\operatorname{tr} MX(X^\top X)^{-1}X^\top.$$

Appendix C. Equivalent relaxations

In this section, we give details about two equivalent relaxations.

C.1 First equivalent relaxation

We start from the penalized version of Eq. (5),

$$\min_{y\in\{-1,1\}^n,\ v\in\mathbb{R}^d} \frac{1}{n}\|\Pi_n y - Xv\|_2^2 + \nu\frac{(y^\top 1_n)^2}{n^2}, \tag{29}$$

which we expand as:

$$\min_{y\in\{-1,1\}^n,\ v\in\mathbb{R}^d} \frac{1}{n}\operatorname{tr}\Pi_n yy^\top - \frac{2}{n}\operatorname{tr}Xvy^\top + \frac{1}{n}\operatorname{tr}X^\top Xvv^\top + \nu\frac{(y^\top 1_n)^2}{n^2}, \tag{30}$$

and relax as, using $Y = yy^\top$, $P = yv^\top$ and $V = vv^\top$,

$$\min_{V,P,Y} \frac{1}{n}\operatorname{tr}\Pi_n Y - \frac{2}{n}\operatorname{tr}P^\top X + \frac{1}{n}\operatorname{tr}X^\top XV + \nu\frac{1_n^\top Y 1_n}{n^2} \text{ s.t. } \begin{pmatrix} Y & P \\ P^\top & V\end{pmatrix} \succcurlyeq 0,\ \operatorname{diag}(Y) = 1. \tag{31}$$

When optimizing Eq. (31) with respect to V and P, we get exactly Eq. (8). Indeed we solve this problem by fixing the matrix Y such that $Y = Y_0$ and $\operatorname{diag}(Y_0) = 1_n$. Then the Lagrangian function of the problem in Eq. (31) can be written as

$$\begin{aligned} L(A) &= \frac{1}{n}\operatorname{tr}\Pi_n Y - \frac{2}{n}\operatorname{tr}P^\top X + \frac{1}{n}\operatorname{tr}X^\top XV + \nu\frac{1_n^\top Y 1_n}{n^2} + \operatorname{tr}A(Y - Y_0) \\ &= \begin{pmatrix} Y & P \\ P^\top & V\end{pmatrix}\begin{pmatrix} \frac{1}{n}\Pi_n + \frac{\nu}{n^2}1_n 1_n^\top + A & \frac{-1}{n}X \\ \frac{-1}{n}X^\top & \frac{1}{n}X^\top X\end{pmatrix} - \operatorname{tr}AY_0. \end{aligned}$$

Using $L(A)$ and the psd constraint $\begin{pmatrix} Y & P \\ P^\top & V\end{pmatrix} \succcurlyeq 0$, we write the dual problem as

$$\min_A \operatorname{tr}AY_0 \text{ s.t. } \begin{pmatrix} \frac{1}{n}\Pi_n + \frac{\nu}{n^2}1_n 1_n^\top + A & \frac{-1}{n}X \\ \frac{-1}{n}X^\top & \frac{1}{n}X^\top X\end{pmatrix} \succcurlyeq 0.$$

From the Schur's complement condition of $\begin{pmatrix} \frac{1}{n}\Pi_n + \frac{\nu}{n^2}1_n 1_n^\top + A & \frac{-1}{n}X \\ \frac{-1}{n}X^\top & \frac{1}{n}X^\top X\end{pmatrix} \succcurlyeq 0$, we obtain $\frac{1}{n}\Pi_n + \frac{\nu}{n^2}1_n 1_n^\top + A \succcurlyeq \frac{1}{n}X(X^\top X)^{-1}X^\top$. Substituting the bound for A we get the optimal objective function value

$$\mathcal{D}^* = \frac{1}{n}\operatorname{tr}X(X^\top X)^{-1}X^\top Y_0 - \frac{1}{n}\operatorname{tr}\Pi_n Y_0 - \frac{\nu}{n^2}1_n^\top Y_0 1_n.$$

Note that the optimal dual objective value $\mathcal{D}^*$ corresponds to a fixed Y_0. Hence by maximizing with respect to Y we obtain exactly Eq. (8) and therefore, the convex relaxation in Eq. (11) is equivalent to Eq. (8). Moreover the Karush-Kuhn-Tucker (KKT) conditions gives

$$P^\top - X + VX^\top X = 0 \text{ and } -YX + PX^\top X = 0$$

Thus the optimum is attained for $P = YX(X^\top X)^{-1}$ and $V = (X^\top X)^{-1}X^\top YX(X^\top X)^{-1}$.

C.2 Second equivalent relaxation

For $\nu = 1$, we solve the problem in Eq. (31) by fixing the matrix $V = V_0$. Then the Lagrangian function of this problem can be written as

$$\begin{aligned} \hat{L}(\mu, B) &= \frac{1}{n}\operatorname{tr}\Pi_n Y - \frac{2}{n}\operatorname{tr}P^\top X + \frac{1}{n}\operatorname{tr}X^\top XV + \nu\frac{1_n^\top Y 1_n}{n^2} + \mu^\top(\operatorname{diag}(Y) - 1_n) + \operatorname{tr}B(V - V_0) \\ &= \begin{pmatrix} Y & P \\ P^\top & V\end{pmatrix}\begin{pmatrix} \frac{1}{n}I_n + \operatorname{diag}(\mu) & \frac{-1}{n}X \\ \frac{-1}{n}X^\top & \frac{1}{n}X^\top X + B\end{pmatrix} - \mu^\top 1_n - \operatorname{tr}BV_0. \end{aligned}$$

Using $\hat{L}(\mu, B)$ and the psd constraint $\begin{pmatrix} Y & P \\ P^\top & V\end{pmatrix} \succcurlyeq 0$, the dual problem is given by

$$\min_{\mu,B} \mu^\top 1_n + \operatorname{tr}BV_0 \text{ s.t.} \begin{pmatrix} \frac{1}{n}I_n + \operatorname{diag}(\mu) & \frac{-1}{n}X \\ \frac{-1}{n}X^\top & \frac{1}{n}X^\top X + B\end{pmatrix} \succcurlyeq 0.$$

From the Schur's complement condition of $\begin{pmatrix} \frac{1}{n}I_n + \operatorname{diag}(\mu) & \frac{-1}{n}X \\ \frac{-1}{n}X^\top & \frac{1}{n}X^\top X + B\end{pmatrix} \succcurlyeq 0$, we obtain $B \succcurlyeq \frac{1}{n^2}X^\top \operatorname{diag}(\mu + 1_n/n)^{-1}X - \frac{1}{n}X^\top X$. Substituting the bound for B we get the dual problem as

$$\begin{aligned} &\min_\mu \mu^\top 1_n + \frac{1}{n^2}\operatorname{tr}V_0X^\top \operatorname{diag}(\mu + 1_n/n)^{-1}X - \frac{1}{n}\operatorname{tr}V_0X^\top X \\ \implies &\min_\mu \sum_{i=1}^n \left(\mu_i + \frac{1}{n^2\mu_i + n}x_i^\top V_0 x_i\right) - \frac{1}{n}\operatorname{tr}V_0X^\top X. \end{aligned}$$

Solving for μ_i, we get

$$\mu_i^* = \frac{1}{n}\sqrt{x_i^\top V_0 x_i} - \frac{1}{n}.$$

Substituting μ_i^* into the dual objective function, we get the optimal objective function value

$$\hat{D} = \frac{2}{n}\sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} - 1 - \frac{1}{n}\operatorname{tr}V_0X^\top X.$$

Furthermore the KKT conditions give

$$Y\operatorname{diag}(\nu + 1_n/n) - \frac{1}{n}PX^\top = 0 \text{ and } P^\top \operatorname{diag}(\nu + 1_n/n) - \frac{1}{n}VX^\top = 0.$$

Thus we obtain the following closed form expressions:

$$\begin{aligned} P &= \operatorname{Diag}(\operatorname{diag}(XVX^\top))^{-1/2}XV \\ Y &= \operatorname{Diag}(\operatorname{diag}(XVX^\top))^{-1/2}XVX^\top \operatorname{Diag}(\operatorname{diag}(XVX^\top))^{-1/2}. \end{aligned}$$

The optimal dual objective value $\hat{D}$ corresponds to a fixed V_0. Therefore, maximizing with respect to V leads to the problem:

$$\min_{V\succcurlyeq 0}\ 1 - \frac{2}{n}\sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} + \frac{1}{n}\operatorname{tr}(VX^\top X). \tag{32}$$

Appendix D. Auxiliary results for Section 5.1

Here we provide auxiliary results for Section 5.1.

D.1 Auxiliary lemma

The matrix $X(X^\top X)^{-1}X^\top$ has the following properties (see e.g., (Freedman, 2009)).

Lemma 10 *The matrix $H = X(X^\top X)^{-1}X^\top$ is the orthogonal projection onto the column space of the design matrix X since:*

- *H is symmetric.*
- *H is idempotent ($H^2 = H$.*
- *X is invariant under H, that is $HX = X$.*

D.2 Rank-one solution of the relaxation Eq. (8)

We denote by $(x_i)_{i=1\dots n}$ the lines (or rows) of X.

Lemma 11 *The rank-one solution $Y_* = yy^\top$ is always a solution of the relaxation Eq. (8).*

Proof We give an elementary proof of this result without using convex optimization tools. Using Proposition 1 and Lemma 10 we have $Hy = y$, thus

$$\operatorname{tr} HY_* = \operatorname{tr} Hyy^\top = \operatorname{tr} yy^\top = n.$$

Moreover all $M \succcurlyeq 0$ can always be decomposed as $\sum_{i=1}^n \lambda_i u_i u_i^\top$ with $\lambda_i \geq 0$ and $(u_i)_{i=1,\dots,n}$ an orthonormal family. Since H is an orthogonal projection $(u_i)^\top H u_i = (Hu_i)^\top H u_i = \|Hu_i\|^2 \leq \|u_i\|^2 \leq 1$. Thus $\operatorname{tr} HM = \sum_{i=1}^n \lambda_i \operatorname{tr} Hu_i(u_i)^\top = \sum_{i=1}^n \lambda_i (u_i)^\top H u_i \leq \sum_{i=1}^n \lambda_i = \operatorname{tr} M$.

Then for all matrix M feasible we have $\operatorname{tr} HM \leq n$ since $\operatorname{diag}(M) = 1_n$ and $\operatorname{tr} HY_* = n$ which conclude the lemma. ■

D.3 Rank-one solution of the relaxation Eq. (12)

Lemma 12 *The rank-one solution $V_* = vv^\top$ is always a solution of the relaxation Eq. (12).*

Proof The Karush-Kuhn-Tucker (KKT) optimality conditions for the problem are for the dual variable $A \preccurlyeq 0$:

$$\frac{1}{n}\sum_{i=1}^n \frac{x_i x_i^\top}{\sqrt{x_i^\top V x_i}} - \frac{1}{n}XX^\top = A \text{ and } AV = 0 \quad \text{(Complementary Slackness)}.$$

Since $x_i^\top w = y_i$, $\sqrt{x_i^\top V_* x_i} = |y_i| = 1$, V_* and the dual variable $A = 0$ satisfy the KKT conditions and then V_* is solution of this problem. ■

D.4 Proof of Proposition 2

In the following lemma, we use a Taylor expansion to lower-bound f around its minimum.

Lemma 13 *For $d \geq 3$ and $\delta \in [0,1)$.*

If $\beta \geq 3$ and $m^2 \leq \frac{\beta-3}{2(d+\beta-4)}$, then with probability at least $1 - d\exp\big(-\frac{\delta^2 n m^2}{2R^4 d^2}\big)$, for any symmetric matrix Δ:

$$f(V_*) - f(V_* + \Delta) > 2(1-\delta)m^2\|\Delta\|_F^2 + o(\|\Delta\|^2) \geq 0.$$

Otherwise with probability at least $1 - d\exp\big(-\frac{\delta^2 n \mu_1}{4R^4 d^2}\big)$, for any symmetric matrix Δ:

$$f(V_*) - f(V_* + \Delta) > (1-\delta)\mu_1\|\Delta\|_F^2 + o(\|\Delta\|^2) \geq 0,$$

with $\mu_1 \geq \frac{m^2(\beta-1)}{1+(d+\beta-2)m^2}$. Moreover we also have with probability at least $1 - d\exp\big(-\frac{\delta^2 n \mu_2}{4R^4 d^2}\big)$, for any symmetric matrix $\Delta \in \Delta_{\min}^\perp$:

$$f(V_*) - f(V_* + \Delta) > (1-\delta)\mu_2\|\Delta\|_F^2 + o(\|\Delta\|^2) \geq 0,$$

where $\mu_2 = \min\{2m^2, m^2(\beta-1), 2m\}$ and $\Delta_{\min} = \begin{pmatrix} 1 & 0 \\ 0 & c_{\min} I_{d-1} \end{pmatrix}$ is defined in the proof and satisfies

$$|c_{\min}| \leq \frac{m}{|(d+\beta-2)m^2 - 1|}.$$

This lemma directly implies Proposition 2.

Proof

For $\Delta \in \mathcal{S}(d)$ and $\delta \in \mathbb{R}$ we compute for $f(V) = \frac{1}{n}\sum_{i=1}^n \sqrt{x_i^\top V x_i}$,

$$\frac{d^2}{d\delta^2} f(V + \delta\Delta) = -\frac{1}{4n}\sum_{i=1}^n \frac{(x_i^\top \Delta x_i)^2}{\sqrt{x_i^\top (V+\delta\Delta) x_i}^3}.$$

Thus the second directional derivative in $V = V_*$ along Δ is

$$\nabla^2_\Delta f(V_*) = \lim_{\delta\to 0} \frac{d^2}{d\delta^2} f(V + \delta\Delta) = -\frac{1}{4n}\sum_{i=1}^n (x_i^\top \Delta x_i)^2.$$

Let $\mathcal{T}_x$ be the semidefinite positive quadratic form of $\mathcal{S}(d)$ defined for $\Delta \in \mathcal{S}(d)$, by

$$\mathcal{T}_x : \Delta \mapsto (x^\top \Delta x)^2. \tag{33}$$

Then there exists a positive linear operator T_x from $\mathcal{S}(d)$ to $\mathcal{S}(d)$ such that $\mathcal{T}_x(\Delta) = \langle \Delta, T_x \Delta \rangle$.

Therefore the function f will be strictly concave if for all directions $\Delta \in \mathcal{S}(d)$

$$\frac{1}{n}\sum_{i=1}^n \mathcal{T}_{x_i}(\Delta) > 0. \tag{34}$$

We will bound the empirical expectation in Eq. (34) by first showing that its expectation remains away from 0. Then we will use a concentration inequality for matrices to control the distance between the sum in Eq. (34) and its expectation.

We first derive conditions so that the result is true in expectation, i.e., for the operator $\mathcal{T}$ defined by $\mathcal{T} = \mathbb{E}\mathcal{T}_x$ for x following the same law as $(y, z^\top)^\top$. We denote by $m = \mathbb{E}z^2$ and by $\beta = \mathbb{E}z^4/m^2$ its kurtosis.

We let $\Delta = \begin{pmatrix} a & b^\top \\ b & C \end{pmatrix}$ and then have $x^\top \Delta x = a + 2yb^\top z + z^\top Cz$. Thus

$$\mathcal{T}_x(\Delta) = a^2 + 4ayb^\top z + 2az^\top Cz + 4b^\top(zz^\top)b + (z^\top Cz)^2 + 4yb^\top z(z^\top Cz).$$

Therefore we can express the value of the operator $\mathcal{T}$ only in function of the elements of Δ:

$$\mathcal{T}(\Delta) = (a + m\operatorname{tr} C)^2 + 4m\|b\|_2^2 + 2m^2\|C - \operatorname{Diag}(\operatorname{diag}(C))\|_F^2 + m^2(\beta-1)\|\operatorname{diag}(C)\|^2,$$

where we have used

$$\begin{aligned}
\mathbb{E}(z^\top Cz)^2 &= \mathbb{E}\sum_{i,j,k,l} z_i z_j z_k z_l c_{i,j} c_{k,l} \\
&= \mathbb{E}\sum_i (z_i)^4 c_{i,i}^2 + \mathbb{E}\sum_{i,k\neq i} z_i^2 z_k^2 c_{i,i} c_{k,k} + 2\mathbb{E}\sum_{i,j\neq i} z_i^2 z_j^2 c_{i,j}^2 \\
&= \beta m^2 \sum_i c_{i,i}^2 + m^2 \sum_{i,k\neq i} c_{i,i} c_{k,k} + 2m^2 \sum_{i,j\neq i} c_{i,j}^2 \\
&= m^2(\beta-3)\sum_i c_{i,i}^2 + m^2 \sum_{i,k} c_{i,i} c_{k,k} + 2m^2 \sum_{i,j} c_{i,j}^2 \\
&= m^2(\beta-3)\|\operatorname{diag}(C)\|^2 + m^2\big(2\|C\|_F^2 + \operatorname{tr}(C)^2\big) \\
&= m^2(\beta-3)\|\operatorname{diag}(C)\|^2 + m^2\big(2\|C - \operatorname{Diag}(\operatorname{diag}(C))\|_F^2 + \operatorname{tr}(C)^2\big).
\end{aligned}$$

Since $\beta \geq 1$, we get

$$\mathcal{T}(\Delta) \geq (a + m\operatorname{tr} C)^2 + 4m\|b\|_2^2 + 2m^2(\|C\|_F^2 - \|\operatorname{diag}(C)\|^2).$$

Thus $\mathcal{T}(\Delta) = 0$ if and only if $\beta = 1$ with $b = 0_{d-1}$ and $C = \operatorname{diag}(c)$ with $c^\top 1_d = -\frac{a}{m_2}$. With the condition $\beta = 1$ meaning that $\operatorname{var}(z^2) = 0$ and thus z^2 is constant a.s., i.e., z follows a Rademacher law.

However we would like to bound $\mathcal{T}(\Delta)$ away from zero by some constant and for that we are looking for the smallest eigenvalue of the operator $\mathbb{E}T_x$. Unfortunately we are not able to solve the optimization problem

$$\min_{\Delta \in \mathcal{S}(d), \|\Delta\|_F^2 = 1} \mathcal{T}(\Delta),$$

and we have to compute all the spectrum of this operator to be able to find the smallest using $\mathbb{E}T_x\Delta = 1/2\nabla\mathcal{T}(\Delta)$.

We have

$$1/2\nabla\mathcal{T}(\Delta) = \begin{pmatrix} a + m\operatorname{tr}(C) & 2mb^\top \\ 2mb & (a + m\operatorname{tr}(C))m_2 I_{d-1} + 2m^2 C \\ & \quad + m^2(\beta-3)\operatorname{Diag}(\operatorname{diag}(C)) \end{pmatrix}.$$

- For all $b \in \mathbb{R}^{d-1}$ we have for $\Delta = \begin{pmatrix} 0 & b^\top \\ b & 0 \end{pmatrix}$, $1/2\nabla\mathcal{T}(\Delta) = 2m\Delta$. Thus $2m$ is an eigenvalue of multiplicity $d-1$.
- For all $C \in \mathbb{R}^{(d-1)\times(d-1)}$ with $\operatorname{diag}(C) = 0_{d-1}$ we have for $\Delta = \begin{pmatrix} 0 & 0 \\ 0 & C \end{pmatrix}$, $1/2\nabla\mathcal{T}(\Delta) = 2m^2\Delta$. Thus $2m^2$ is an eigenvalue of multiplicity $\frac{(d-1)(d-2)}{2}$.
- For all $c \in \mathbb{R}^{d-1}$ with $c^\top 1_{d-1} = 0$ we have for $\Delta = \begin{pmatrix} 0 & 0 \\ 0 & \operatorname{diag}(C) \end{pmatrix}$, $1/2\nabla\mathcal{T}(\Delta) = m^2(\beta-1)\Delta$. Thus $m^2(\beta-1)$ is an eigenvalue of multiplicity $d-2$.
- For all $a, c \in \mathbb{R}^2$ we have for $\Delta = \begin{pmatrix} a & 0 \\ 0 & cI_{d-1} \end{pmatrix}$,

$$\begin{aligned}
1/2\nabla\mathcal{T}(\Delta) &= \begin{pmatrix} a + m(d-1)c & 0 \\ 0 & [ma + m^2(d+\beta-2)c]I_{d-1} \end{pmatrix} \\
&= \operatorname{Diag}\Big[\begin{pmatrix} 1 & m1_{d-1}^\top \\ m1_{d-1} & (d+\beta-2)m^2 I_{d-1} \end{pmatrix}\begin{pmatrix} a \\ c1_{d-1} \end{pmatrix}\Big].
\end{aligned}$$

  Thus an eigenvalue of $\begin{pmatrix} 1 & (d-1)m \\ m & (d+\beta-2)m^2 \end{pmatrix}$ with an eigenvector $[a, c]^\top$ would be an eigenvalue of the operator $\mathbb{E}T_x$ with a corresponding eigenvector $\begin{pmatrix} a & 0 \\ 0 & cI_{d-1} \end{pmatrix}$. This matrix has two simple eigenvalues

$$\mu_\pm = \frac{1 + (d+\beta-2)m^2 \pm \sqrt{(1 + (d+\beta-2)m^2)^2 - 4m^2(\beta-1)}}{2}. \tag{35}$$

Moreover when we add all the multiplicity of the found eigenvalues we get $d-1+\frac{(d-1)(d-2)}{2} + d-2+2 = \frac{d(d+1)}{2}$ which is the dimension of $S(d)$, therefore we have found all the eigenvalues of the linear operator $\mathbb{E}T_x$.

We will prove now than the smallest eigenvalue is μ_- when the dimension d is large enough with regards to m^2 and $2m^2$ otherwise.

Lemma 14 *Let μ_1 and μ_2 be the two smallest eigenvalues of the operator $\mathbb{E}T_x$. Let us assume that $d \geq 3$ (the case $d = 2$ will also be done in the proof).*

If $\beta \geq 3$ and $m^2 \leq \frac{\beta-3}{2(d+\beta-4)}$ then

$$\mu_1 = 2m^2.$$

Otherwise

$$\mu_1 = \mu_- \geq \frac{m^2(\beta-1)}{1 + (d+\beta-2)m^2} \text{ and } \mu_2 = \min\{2m^2, m^2(\beta-1), 2m\}.$$

Moreover we denote by $\Delta_{\min} = \begin{pmatrix} 1 & 0 \\ 0 & c_{\min} I_{d-1} \end{pmatrix}$ the eigenvector associated to μ_- for which we have set without loss of generality the first component $a = 1$. Then

$$|c_{\min}| \leq \frac{m}{|(d+\beta-2)m^2 - 1|}.$$

Unfortunately μ_- can become small when the dimension increases as explained by the tight bound $\mu_- \geq \frac{m^2(\beta-1)}{1+(d+\beta-2)m^2}$. However the corresponding eigenvector has a particular structure we will be able to exploit.

Proof First we note that $\mu_- \leq m^2(\beta-1)$ and compute

$$\begin{aligned}
\mu_- \geq 2m^2 &\Leftrightarrow 1+(d+\beta-2)m^2 - \sqrt{(1+(d+\beta-2)m^2)^2 - 4m^2(\beta-1)} - 4m^2 \geq 0 \\
&\Leftrightarrow 1+(d+\beta-2)m^2 - 4m^2 \geq \sqrt{(1+(d+\beta-2)m^2)^2 - 4m^2(\beta-1)} \\
&\Leftrightarrow (1+(d+\beta-2)m^2-4m^2)^2 \geq (1+(d+\beta-2)m^2)^2 - 4m^2(\beta-1) \\
&\quad \text{and } 1+(d+\beta-6)m^2 \geq 0 \\
&\Leftrightarrow 16m^4 - 8m^2(1+(d+\beta-2)m^2) \geq -4m^2(\beta-1) \\
&\quad \text{and } 1+(d+\beta-6)m^2 \geq 0 \\
&\Leftrightarrow \underbrace{2(d+\beta-4)m^2 \leq \beta-3}_{\text{R1}} \text{ and } \underbrace{1+(d+\beta-6)m^2 \geq 0}_{\text{R2}}.
\end{aligned}$$

- If $d=2$,
  - If $\beta \leq 3$ we have necessary that $\beta \leq 2$ and the first term R1 implies $m^2 \geq \frac{3-\beta}{2(2-\beta)}$ and the second term R2 implies $m^2 \leq 1/(4-\beta)$. Thus we should have $(4-\beta)(3-\beta) \leq 2(2-\beta)$ which is not possible since the polynomial $\beta^2-5\beta+8 \geq 0$.
  - If $\beta \geq 3$, the first term R1 implies $m^2 \leq \frac{\beta-3}{2(\beta-2)} \leq 1$ and the second term R2 implies $m^2 \leq 1/(4-\beta) \leq \frac{\beta-3}{2(\beta-2)} \leq 1$ for $\beta \leq 4$ and is always satisfied otherwise.
- If $d \geq 3$, the first term R1 implies that $\beta \geq 3$ for which the second term R2 is always satisfied. It also implies that $m^2 \leq \frac{\beta-3}{2(d+\beta-4)} \leq 1$.

We denote by $\Delta_{\min} = \begin{pmatrix} 1 & 0 \\ 0 & c_{\min} I_{d-1} \end{pmatrix}$ the eigenvector for which we have set without loss of generality $a=1$ and

$$c_{\min} = \frac{-1}{2(d-1)m}\Big[\sqrt{((d+\beta-2)m^2-1)^2+4(d-1)m^2} - (d+\beta-2)m^2+1\Big].$$

Consequently $c_{\min} \leq 0$ and by convexity of the square root we have

$$\sqrt{((d+\beta-2)m^2-1)^2+4(d-1)m^2} \leq ((d+\beta-2)m^2-1) + \frac{2(d-1)m^2}{|(d+\beta-2)m^2-1|}.$$

Therefore

$$|c_{\min}| \leq \frac{m}{|(d+\beta-2)m^2-1|}.$$

■

We will control now the behavior of the empirical expectation by its expectation thanks to concentration theory. By definition T_x is a symmetric positive linear operator as its projection $T_x^\perp$ onto the orthogonal space of $\Delta_{\min}$. We can thus apply the Matrix Chernoff inequality from Tropp (2012, Theorem 5.1.1) to these two operators using $\|T_x\|_{op} \leq \|xx^\top\|^2 \leq \mathrm{tr}(xx^\top)^2 \leq \|x\|_2^4 \leq R^4 d^2$. Then:

$$\mathbb{P}\left(\lambda_{\min}\Big(\sum_{k=1} T_{x_k}\Big) \leq n\delta\mu_1\right) \leq d\Big[\frac{e^{-(1-\delta)}}{\delta^\delta}\Big]^{n\mu_1/(2R^4d^2)} \leq de^{-(1-\delta)^2 n\mu_1/(4R^4d^2)},$$

$$\mathbb{P}\left(\lambda_{\min}\Big(\sum_{k=1} T^\perp_{x_k}\Big) \leq n\delta\mu_2\right) \leq d\Big[\frac{e^{-(1-\delta)}}{\delta^\delta}\Big]^{n\mu_2/(2R^4d^2)} \leq de^{-(1-\delta)^2 n\mu_2/(4R^4d^2)},$$

For $m=1$ and $d \geq 3$ we have $\mu_1 = \mu_- \geq \frac{\beta-1}{\beta+d} \geq \min\{\frac{\beta-1}{2\beta}, \frac{\beta-1}{2d}\} \geq \min\{1/3, \frac{\beta-1}{2d}\}$. ■

D.5 Noise robustness for the one dimensional balanced problem

We want a condition on ε such that the solution of the relaxation Eq. (8) recovers the right y. We recall the dual problem of the relaxation Eq. (8)

$$\min \mu^\top 1_n \text{ s.t. } \mathrm{Diag}(\mu) \succcurlyeq X(X^\top X)^{-1}X^\top.$$

The KKT conditions are:

- Dual feasibility: $\mathrm{Diag}(\mu) \succcurlyeq X(X^\top X)^{-1}X^\top$.
- Primal feasibility: $\mathrm{Diag}(Y) = 1_n$ and $Y \succcurlyeq 0$.
- Complimentary slackness : $Y[\mathrm{Diag}(\mu) - X(X^\top X)^{-1}X^\top] = 0$

For $Y = yy^\top$ a rank one matrix, the last condition implies $\mathrm{Diag}(\mu)y = Hy$ and

$$\mu_i = \frac{(X(X^\top X)^{-1}X^\top y)_i}{y_i}.$$

For $X = y+\varepsilon$, we denote by $\tilde{y} = y+\varepsilon$, then $X(X^\top X)^{-1}X^\top = \frac{\tilde{y}\tilde{y}^\top}{\|\tilde{y}\|^2}$ and $X(X^\top X)^{-1}X^\top y = \frac{\tilde{y}^\top y}{\|\tilde{y}\|^2}\tilde{y}$. Thus

$$\mu_i = \frac{\tilde{y}^\top y}{\|\tilde{y}\|^2}\frac{\tilde{y}_i}{y_i}.$$

Assume that all $\tilde{y}_i y_i$ have the same sign, without loss of generality we assume $\tilde{y}_i y_i > 0$. By definition of μ, $\mu \geq 0$. To show the dual feasibility we have to show that $\mathrm{Diag}(\mu) \succcurlyeq H$ which is equivalent to $\mathrm{Diag}(\frac{\tilde{y}_i}{y_i}) \succcurlyeq \frac{\tilde{y}\tilde{y}^\top}{\tilde{y}^\top y}$, to $I_n - \mathrm{Diag}(\sqrt{\frac{y_i}{\tilde{y}_i}})\frac{\tilde{y}\tilde{y}^\top}{\tilde{y}^\top y}\mathrm{Diag}(\sqrt{\frac{y_i}{\tilde{y}_i}}) \succcurlyeq 0$ and to $\sum y_i\tilde{y}_i \leq \tilde{y}^\top y$ which is obviously true. Reciprocally if μ is dual feasible then $\mathrm{Diag}(\mu) \succcurlyeq 0$ and all the $\tilde{y}_i y_i$ have the same sign.

Therefore we have shown that y is solution of the relaxation Eq. (8) if and only if all the $\tilde{y}_i y_i$ have the same sign. If ε and y are independent this is equivalent to $\|\varepsilon\|_\infty \leq 1$ a.s.

D.6 The rank-one candidates are not solutions of the relaxation Eq. (12)

We assume now that $1_n^\top y \neq 0$ thus $y \neq \Pi_n y$, which means we do not have the same proportion in the two clusters. Let us assume that $\Pi_n y$ takes two values $\{\pi y_-, \pi y_+\}$ that is by definition of Π_n: $\pi y_+ = 1 - \frac{1_n^\top y}{n}$ and $\pi y_- = -1 - \frac{1_n^\top y}{n}$. For V_* defined as before, we get $x_i^\top V_* x_i = (\pi y_i)^2$ and with $I_\pm$ the set of indices such that $\Pi_n y_i = \pi y_\pm$ respectively, the KKT conditions for $V = V_*$ can be written as

$$\frac{1}{n}\Big[\sum_{i\in I_+}\Big(\frac{1}{\pi y_+}-1\Big)x_i x_i^\top + \sum_{i\in I_-}\Big(\frac{1}{-\pi y_-}-1\Big)x_i x_i^\top\Big] = A_n \preccurlyeq 0 \text{ and } A_n V_* = 0.$$

We check that with $n_\pm = \#\{I_\pm\}$:

$$\begin{aligned} w^\top A_n w = 0 &= \sum_{i\in I_+}\Big(\frac{1}{\pi y_+}-1\Big)(\pi y_+)^2 + \sum_{i\in I_-}\Big(\frac{1}{-\pi y_-}-1\Big)(\pi y_-)^2 \\ &= n_+\Big(\frac{1}{\pi y_+}-1\Big)(\pi y_+)^2 + n_-\Big(\frac{1}{-\pi y_-}-1\Big)(\pi y_-)^2 \\ &= n_+\pi y_+ - n_-\pi y_- - \big(n_+(\pi y_+)^2 + n_-(\pi y_-)^2\big) \\ &= y^\top \Pi_n y - (\Pi_n y)^\top \Pi_n y = y^\top \Pi_n y - y^\top \Pi_n y = 0. \end{aligned}$$

And $A_n = \frac{1}{2n}\big[\sum_{i\in I_+}\alpha_+ x_i x_i^\top + \sum_{i\in I_-}\alpha_- x_i x_i^\top\big]$ with $\alpha_+ = \big(\frac{1}{\pi y_+}-1\big)$ and $\alpha_- = \big(\frac{1}{-\pi y_-}-1\big)$. Unfortunately $\alpha_+\alpha_- \leq 0$, and A_n is not necessary negative. Even worse we will show that $\mathbb{E}A$ is not semi-definite negative which will conclude the proof since by the law of large number $\lim_{n\to\infty}\frac{1}{n}A_n = \mathbb{E}A$. Assume that the proportions of the two clusters stay constant with $n_\pm = \rho_\pm n$, then

$$\mathbb{E}A = \rho_+\alpha_+\begin{pmatrix}(\pi y_+)^2 & 0\\ 0 & I\end{pmatrix} + \rho_-\alpha_-\begin{pmatrix}(\pi y_-)^2 & 0\\ 0 & I\end{pmatrix}.$$

And $\rho_+\alpha_+(\pi y_+)^2 + \rho_-\alpha_-(\pi y_-)^2 = 0$ since $w^\top A_n w = 0$. Then

$$\begin{aligned} \rho_+\alpha_+ + \rho_-\alpha_- &= \frac{\rho_+\pi y_- - \rho_-\pi y_+ - \pi y_+\pi y_-}{\pi y_+\pi y_-} \\ &= \frac{-(\rho_+ + \rho_-) - \frac{1_n^\top y}{n}(\rho_+ - \rho_-) + (1-(1_n^\top y)^2)}{-(1-(\frac{1_n^\top y}{n})^2)} \\ &= \frac{\frac{1_n^\top y}{n}(\rho_+ - \rho_-) + (\frac{1_n^\top y}{n})^2)}{(1-(\frac{1_n^\top y}{n})^2)} = \frac{2(\frac{1_n^\top y}{n})^2}{(1-(\frac{1_n^\top y}{n})^2)} \geq 0. \end{aligned}$$

Thus $A = \frac{2(1_n^\top y)^2}{(n^2-(1_n^\top y)^2)}\begin{pmatrix}0 & 0\\ 0 & I\end{pmatrix}$ is not semi-definite negative and V_* is not solution of the relaxation Eq. (12).

Appendix E. Auxiliary results for sparse extension

In this section, we provide some results for sparse extension.

E.1 There is a rank-one solution of the relaxation Eq. (18)

Lemma 15 *The rank-one solution $V_* = v^* v^{*\top}$ is solution of the relaxation Eq. (18) if the design matrix X is such that $\frac{1}{n}X^\top X$ has all its diagonal entries less than one.*

Proof The KKT conditions for problem Eq. (18) are

$$\frac{1}{n}\sum_{i=1}^n \frac{x_i x_i^\top}{\sqrt{x_i^\top V x_i}} - \lambda U - \frac{1}{n}X^\top X = A \preccurlyeq 0 \text{ and } AV = 0,$$

with U such that $U_{ij} = \operatorname{sign}(V_{ij})$ if $V_{ij} \neq 0$ and $U_{ij} \in [-1,1]$ otherwise. For $V_* = v^* v^{*\top}$ this gives

$$A = \frac{(1+\lambda)}{n}X^\top X - \lambda U - \frac{1}{n}X^\top X = \lambda\Big[\frac{X^\top X}{n} - U\Big] \text{ with } U_{1,1} = 1 \text{ and } U_{i,j} \in [-1,1] \text{ otherwise.}$$

We check that $AV_* = 0$. If the design matrix X is such that $\frac{1}{n}X^\top X$ has all its diagonal entries less than one, we can choose a sub-gradient U such that the dual variable $A = 0$ and thus V_* is solution. Otherwise by property of semi-definite matrices, there is a diagonal entry of $\frac{1}{n}X^\top X$ which is bigger than 1 which prevents A to be semi-definite negative since the corresponding diagonal entry of $\frac{X^\top X}{n} - U$ will be positive. This shows that V_* does not solve the problem. ■

E.2 Proof of proposition 6

Lemma 16 *For $\delta \in [0,1)$, with probability $1 - 5d^2\exp\big(-\frac{\delta^2 n(\beta-1)}{2dR^4(1/m^2+\beta+d)}\big)$, for any direction Δ such that $V_* + \Delta \succcurlyeq 0$, we have:*

$$g(V_*) - g(V_*+\Delta) > (1-\delta)\Big[\lambda\|\Delta - \operatorname{Diag}(\Delta)\|_1 + \frac{\beta-1}{\beta+d+1/m^2}\frac{(1+\lambda)^3}{4}\|\operatorname{Diag}(\Delta)\|_2^2\Big] + o(\|\Delta\|^2) \geq 0.$$

Moreover we also have with probability at least $1 - 5d^2\exp\big(-\frac{\delta^2 n m^2(\beta-1)}{2dR^4}\big)$, for any symmetric matrix Δ such that $V_* + \Delta \succcurlyeq 0$ and $\operatorname{Diag}(\Delta) \in (e_{\min})^\perp$:

$$g(V_*) - g(V_*+\Delta) > (1-\delta)\Big[\lambda\|\Delta - \operatorname{Diag}(\Delta)\|_1 + m^2(\beta-1)\frac{(1+\lambda)^3}{4}\|\operatorname{Diag}(\Delta)\|_2^2\Big] + o(\|\Delta\|^2) \geq 0,$$

where $e_{\min} = [1, c_{\min} 1_{d-1}]$ is defined in the proof and $c_{\min}$ satisfies

$$|c_{\min}| \leq \frac{m}{|(d+\beta-2)m^2 - 1|}.$$

E.2.1 Proof outline

We will investigate under which conditions on X the solution is unique, first for a deterministic design matrix. We make the following deterministic assumptions on X for $\delta, \zeta \geq 0$ and $\mathcal{S} \subset \mathbb{R}^d$:

(A1) $\|\frac{X^\top X}{n}\|_\infty \leq 1$ **(A3)** $\|\frac{Z^\top Z}{n} - \mathrm{Diag}(\mathrm{diag}(\frac{1}{n}Z^\top Z))\|_\infty \leq \delta$
(A2) $\|\frac{Z^\top y}{n}\|_\infty \leq \delta$ **(A4)** $\lambda^{\mathcal{S}}_{\min}\big(\frac{X^{\odot 2}(X^{\odot 2})^\top}{n}\big) \geq \zeta > 0,$

where we denoted by $\odot$ the Hadamard (i.e., pointwise) product between matrices and $\lambda^{\mathcal{S}}_{\min}$ the minimum eigenvalue of a linear operator restricted to a subspace $\mathcal{S}$. Then with $g(V) = \frac{2}{n}\sum_{i=1}^n \sqrt{x_i^\top V x_i} - \lambda\|V\|_1 - \frac{1}{n}\operatorname{tr} X^\top XV$, we can certify that g will decrease around the solution V_*.

Lemma 17 *Let us assume that the noise matrix verifies assumptions (A1,A2,A3,A4), then for any direction Δ such that $V_* + \Delta \succcurlyeq 0$ and $\mathrm{diag}(\Delta) \in \mathcal{S}$ we have:*

$$g(V_*) - g(V_* + \Delta) \geq \lambda(1-\delta)\|\Delta - \mathrm{Diag}(\mathrm{diag}(\Delta))\|_1 + \zeta\frac{(1+\lambda)^3}{4}\|\mathrm{Diag}(\Delta)\|_2^2 + o(\|\Delta\|^2) > 0.$$

Let us assume now that $(z^i)_{i=1,.,d}$ are i.i.d of law z symmetric with $\mathbb{E}z = \mathbb{E}z^3 = 0$, $\mathbb{E}z^2 = m = 1$, $\mathbb{E}z^4/(\mathbb{E}z^2)^2 = \beta$ and such that $\|z\|_\infty$ is a.s. bounded by $0 \leq R \leq 1$. Then the matrix X satisfies a.s. assumption (A1). Using multiple Hoeffding's inequalities we will prove lemma 17.

Lemma 18 *If z does not follow a Rademacher law, the design matrix X satisfies assumptions (A1,A2,A3,A4) with probability greater than $1 - 8d^2\exp\big(-\frac{\delta^2 n(\beta-1)}{2d(\beta+d)R^4}\big)$ for $\mathcal{S} = \mathbb{R}^d$, and with probability greater than $1 - 8d^2\exp\big(-\frac{\delta^2 n\min\{\beta-1,2\}}{2dR^4}\big)$ for $\mathcal{S} = [1, c_{\min}1_{d-1}]^\perp$ where c_{min} is defined in the proof and satisfies*

$$|e_{\min}| \leq \frac{1}{d+\beta-3}.$$

This lemma concludes the proof of proposition 6. We will now prove these two lemmas.

E.2.2 Proof of lemma 17

Proof Since the dual variable A for the PSD constraint is 0 (see the proof of lemma 15), this constraint $V \succcurlyeq 0$ is not active and we will show that the function decreases in a set of directions Δ which include the one for which $V_* + \Delta \succcurlyeq 0$.

Therefore we consider a direction $\Delta = \begin{pmatrix} a & b^\top \\ b & C \end{pmatrix}$, with $C \succcurlyeq 0$, which is slightly more general than $V_* + \Delta \succcurlyeq 0$. We denote by $f(W) = \frac{2}{n}\sum_{i=1}^n \sqrt{x_i^\top W x_i} - \frac{1}{n}\operatorname{tr} X^\top XW$ the smooth part of g. By Taylor-Young, we have for all W:

$$f(W) - f(W+\Delta) = -\langle f'(W), \Delta\rangle - \frac{1}{2}\langle \Delta, f''(W)\Delta\rangle + o(\|\Delta\|^2).$$

Thus:

$$g(W) - g(W+\Delta) = -\langle f'(W), \Delta\rangle - \frac{1}{2}\langle \Delta, f''(W)\Delta\rangle + \lambda(\|W+\Delta\|_1 - \|W\|_1) + o(\|\Delta\|^2).$$

Letting $W = V_*$ this gives with $X^\top X = \begin{pmatrix} n & y^\top Z \\ Z^\top y & Z^\top Z \end{pmatrix}$,

$$\begin{aligned} g(W) - g(W+\Delta) &= -\lambda\langle \frac{X^\top X}{n}, \Delta\rangle - \frac{1}{2}\langle \Delta, f''(V_*)\Delta\rangle + \lambda(a + 2\|b\|_1 + \|C\|_1) + o(\|\Delta\|^2) \\ &= \lambda\big[2(\|b\|_1 - \frac{1}{n}b^\top Z^\top y) + \|C\|_1 - \frac{1}{n}\operatorname{tr}(Z^\top ZC)\big] - \frac{1}{2}\langle \Delta, f''(V_*)\Delta\rangle + o(\|\Delta\|^2). \end{aligned}$$

And with Hölder's inequality and assumption (A2)

$$\|b\|_1 - \frac{1}{n}b^\top Z^\top y \geq \|b\|_1(1 - \|\frac{1}{n}Z^\top y\|_\infty) \geq (1-\delta)\|b\|_1.$$

Nevertheless we will show in lemma 19 that $\|C\|_1 - \frac{1}{n}\operatorname{tr}(Z^\top ZC) \geq (1-\delta)\|C - \mathrm{diag}(C)\|_1$, thus

$$g(W) - g(W+\Delta) \geq \lambda(1-\delta)(2\|b\|_1 + \|C - \mathrm{diag}(C)\|_1) + o(\|\Delta\|^2). \tag{36}$$

However in Eq. (36), $g(W) - g(W+\Delta) = 0$ for $b = 0$ and C diagonal, therefore we have to investigate second order conditions, i.e., to show for $\Delta = \mathrm{diag}(e)$ with $e \in \mathbb{R}^d$ that $-\langle \Delta, f''(V_*)\Delta\rangle > 0$.

And with assumption (A4)

$$\begin{aligned} -\frac{4}{(1+\lambda)^3}\langle \mathrm{diag}(e), f''(V_*)\,\mathrm{diag}(e)\rangle &= \frac{1}{n}\sum_{i=1}^n (x_i^\top \mathrm{diag}(e) x_i)^2 \\ &= \frac{1}{n}\sum_{i=1}^n (\sum_{j=1}^d e_j (x_i^j)^2)^2 \\ &= \frac{1}{n}\sum_{i=1}^n e^\top [x_i^{\odot 2}(x_i^{\odot 2})^\top] e \\ &\geq \lambda_{\min}\big(\frac{X^{\odot 2}(X^{\odot 2})^\top}{n}\big)\|e\|^2 \geq \zeta\|e\|_2^2. \end{aligned}$$

Thus we can conclude:

$$g(W) - g(W+\Delta) \geq \lambda(1-\delta)(2\|b\|_1 + \|C - \mathrm{diag}(C)\|_1) + \zeta\frac{(1+\lambda)^3}{4}\|e\|_2^2 + o(\|\Delta\|^2).$$

■

E.2.3 Auxiliary lemma

Lemma 19 *For all matrix C symmetric semi-definite positive we have under assumptions (A1) and (A3):*

$$\operatorname{tr}\Big(S - \frac{Z^\top Z}{n}\Big)C \geq (1-\delta)\|C - \mathrm{diag}(C)\|_1 > 0.$$

Proof We denote by $\Sigma^n = \frac{Z^\top Z}{n}$. We always have $\|C\|_1 - \mathrm{tr}(\Sigma^n C) = \mathrm{tr}(S - \Sigma^n)C$ where $S_{i,j} = \mathrm{sign}(C_{i,j})$, thus if $\mathrm{diag}(C) > 0$ then $\mathrm{diag}(S) = 1$ and $\mathrm{diag}(S - \Sigma^n) \geq 0$ from assumption (A1). Moreover since $\Sigma^n{}_{i,j} \in [-1, 1]$ then $\mathrm{sign}(S - \Sigma^n) = \mathrm{sign}(S)$.

Thus $\mathrm{tr}(S-\Sigma^n)C = \sum_i C_{i,i}(S-\Sigma^n)_{i,i} + \sum_{i\neq j} C_{i,j}(S-\Sigma^n)_{i,j} \geq \sum_{i\neq j} C_{i,j}(S-\Sigma^n)_{i,j} \geq 0$. Furthermore from assumption (A3), $|(\Sigma^n)_{i,j}| \leq \delta$ for $i \neq j$. Therefore

$$\mathrm{tr}(S - \Sigma^n)C \geq \sum_{i\neq j} C_{i,j}(S - \Sigma^n)_{i,j} \geq \sum_{i\neq j} |C_{i,j}|(1 - \delta) \geq (1 - \delta)\|C - \mathrm{diag}(C)\|_1 > 0.$$

If there is a diagonal element of C which is 0, then the corresponding row and column in C will also be 0 and we can look at the same problem as before by erasing off from C and Σ^n the corresponding column and row. ■

E.2.4 Proof of lemma 18

Proof We will first show that the noise matrix Z satisfies assumptions (A2,A3). By Hoeffding's inequality we have with probability $1 - 2\exp(-\delta^2 n/(2R^2))$

$$\frac{1}{n}|\sum_{i=1}^n z_i^j| \leq \delta.$$

Then, since the law of z is symmetric $y_i z_i$ will have the same law as z_i and with probability $1 - 2\exp(-\delta^2 n/(2R^2))$, the design matrix Z satisfies assumption (A2):

$$\|\frac{Z^\top y}{n}\|_\infty \leq \delta.$$

Likewise we have with probability $1 - 2\exp(-\delta^2 n/(2R^4))$ that for $j \neq j'$

$$|\frac{1}{n}\sum_{i=1}^n z_i^j z_i^{j'}| \leq \delta.$$

Thus we also have with probability $1 - 2d^2\exp(-\delta^2 n/(2R^4))$ that Z satisfies assumption (A3):

$$\|\frac{1}{n}Z^\top Z - \mathrm{diag}(\frac{1}{n}Z^\top Z)\|_\infty \leq \delta.$$

Thus with probability $1 - 4d^2\exp(-\delta^2 n/(2R^4))$, the noise matrix Z satisfies assumptions (A1, A2, A3).

We proceed as in the proof of proposition 2 to show that X satisfies assumption (A4). We first derive a condition to have the result in expectation, then we use an inequality concentration on matrix to bound the empirical expectation. This will be very similar, but we will get a better scaling since Δ is diagonal.

Using the same arguments as in the proof of proposition 2 we have for the diagonal matrix $\Delta = \mathrm{diag}(e)$ with $e = (a, c) \in \mathbb{R}^d$:

$$e^\top \mathbb{E}(x^{\odot 2}(x^{\odot 2})^\top)e = \mathbb{E}(x^\top \Delta x)^2 = (a + mc^\top 1_{n-1})^2 + m^2(\beta - 1)\|c\|_2^2 > 0 \;\text{ if }\; \beta > 1.$$

We can show that $m^2(\beta - 1)$ is an eigenvalue of multiplicity $d - 2$ and $\mu_\pm$ are eigenvalues of multiplicity one of the operator $\Delta \mapsto \mathbb{E}(x^\top \Delta x)^2$ with eigenvectors $e_\pm$. Thus we have

$$\begin{aligned}\lambda_{\min}(\mathbb{E}x^{\odot 2}(x^{\odot 2})^\top) &= \frac{1 + (d + \beta - 2)m^2 - \sqrt{(1 + (d + \beta - 2)m^2)^2 - 4m^2(\beta - 1)}}{2} \quad (37)\\ &\geq \frac{m^2(\beta - 1)}{1 + (d + \beta - 2)m^2},\end{aligned}$$

and

$$\lambda_{\min}^{e_-^\perp}(\mathbb{E}x^{\odot 2}(x^{\odot 2})^\top) = m^2(\beta - 2).$$

Moreover

$$\lambda_{\max}\Big(x^{\odot 2}(x^{\odot 2})^\top\Big) = (x^{\odot 2})^\top x^{\odot 2} = \sum_{j=1}^d (x_i)^4 \leq dR^4.$$

Thus we can apply the Matrix Chernoff inequality from (Tropp, 2012) for $\mu_{\mathcal{S}} = \lambda_{\min}^{\mathcal{S}}(\mathbb{E}x^{\odot 2}(x^{\odot 2})^\top)$:

$$\mathbb{P}\left(\lambda_{\min}^{\mathcal{S}}\Big(\frac{X^{\odot 2}(X^{\odot 2})^\top}{n}\Big) \leq (1 - \delta)\mu_{\mathcal{S}}\right) \leq de^{-\delta^2 n\mu_{\mathcal{S}}/(2dR^4)}.$$

Thus with probability $1 - 5d^2\exp(-\delta^2 n\mu_-/(2dR^4))$ the design matrix X satisfies assumptions (A1,A2,A3,A4) with $\zeta = (1 - \delta)\mu_-$ and $\mathcal{S} = \mathbb{R}^d$. And with probability $1 - 5d^2\exp(-\delta^2 n\min\{\beta - 1, 2\}/(2dR^4))$ the design matrix X satisfies assumptions (A1,A2,A3,A4) with $\zeta = (1 - \delta)\min\{\beta - 1, 2\}$ and $\mathcal{S} = e_-^\perp$. ■

Appendix F. Proof of multi-label results

We first prove lemma 7:

Proof Let $A \in \mathbb{R}^{k\times k}$ symmetric semi-definite positive such that $\mathrm{diag}(\tilde{y}A\tilde{y}^\top) = 1_n$, then

$$\mathrm{diag}(\tilde{y}A\tilde{y}^\top) = \sum_{i=0}^k a_{i,i}1_n + 2\sum_{i=1}^k a_{0,i}y_i + 2\sum_{1\leq i<j\leq k} a_{i,j}y_i \odot y_j$$

thus

$$2\sum_{i=1}^k a_{0,i}y_i + 2\sum_{1\leq i<j\leq k} a_{i,j}y_i \odot y_j = (1 - \sum_{i=0}^k a_{i,i})1_n$$

And this system admits as unique solution 0_n if and only if the family $\{1_n, (y_i)_{1\leq i\leq k}, (y_i y_j)_{1\leq i<j\leq k}\}$ is *linearly independent.* ■

Then we prove the lemma 8:

Proof Since $a_0 + \sum_{i=1}^k a_i^2\alpha_i \geq \alpha_{\min}\sum_{i=0}^k a_i^2 = \alpha_{\min}$ we should have $\alpha \geq \alpha_{\min}$. We have already seen that such Y satisfies the constraint. The KKT conditions are: $B =$

$\operatorname{diag}(\mu) - H - \nu 11^\top \succcurlyeq 0$ and $BY = 0$. Since $y_i = \Pi_n y_i + \frac{(y_i^\top 1_n)}{n} 1_n$.

$$\begin{aligned} Hy_i &= H\Pi_n y_i + (y_i^\top 1_n) H 1_n \\ &= \Pi_n y \\ &= (y_i - \frac{1_n^\top y_i}{n} 1_n). \end{aligned}$$

Thus

$$\begin{aligned} HY &= \sum_{i=1}^k a_i^2 H y_i y_i^\top \\ &= \sum_{i=1}^k a_i^2 (y_i - \frac{1_n^\top y_i}{n} 1_n) y_i^\top \\ &= \sum_{i=1}^k a_i^2 (y_i y_i^\top - \frac{1_n^\top y_i}{n} 1_n y_i^\top) \end{aligned}$$

and $\operatorname{tr}(HY) = \sum_{i=1}^k a_i^2 (n - n\alpha_i) = n(1 - a_0^2 + a_0^2 - \alpha) = n(1-\alpha)$.

Furthermore since $1_n^\top \operatorname{diag}(Y) = n$ and $1_n^\top M 1_n = n^2\alpha$, for $\mu = 1_n$ and $\nu = 1/n$, $B.Y = n - n(1-\alpha) - n\alpha = 0$. And since $B = I_n - \frac{1}{n} 1_n 1_n^\top - H$, $B^2 = B$ and $B^\top = B$, thus B is a symmetric projection and consequently symmetric semi-definite positive.

Hence the primal variable Y and the dual variables $\mu = 1_n$ and $\nu = 1/n$ satisfy the KKT conditions, thus Y is solution of this problem. ■

Appendix G. Efficient optimization problem

We now give the details of an efficient optimization algorithm.

G.1 Dual computation

We consider the following strongly-convex approximation of Eq. (24), augmented with the von-Neumann entropy:

$$\max_{V \succcurlyeq 0} \frac{1}{n} \sum_{i=1}^n \sqrt{(XVX^\top)_{ii}} - \|\operatorname{Diag}(c) V \operatorname{Diag}(c)\|_1 - \varepsilon \operatorname{tr}[(A^{\frac{1}{2}} V A^{\frac{1}{2}}) \log (A^{\frac{1}{2}} V A^{\frac{1}{2}})] \text{ s.t. } \operatorname{tr}(A^{\frac{1}{2}} V A^{\frac{1}{2}}) = 1.$$

Introducing dual variables, we have

$$\begin{aligned} \min_{u \in \mathbb{R}_+^n, C : |C_{ij}| \leqslant c_i c_j} \max_{V \succcurlyeq 0} \quad & \frac{1}{2n} \sum_{i=1}^n \Big(u_i ((XVX^\top)_{ii}) + \frac{1}{u_i} \Big) - \operatorname{tr} CV - \varepsilon \operatorname{tr}[(A^{\frac{1}{2}} V A^{\frac{1}{2}}) \log (A^{\frac{1}{2}} V A^{\frac{1}{2}})] \\ \text{s.t.} \quad & \operatorname{tr}(A^{\frac{1}{2}} V A^{\frac{1}{2}}) = 1. \end{aligned}$$

By fixing u and C, and letting $Q = A^{\frac{1}{2}} V A^{\frac{1}{2}}$, we can write the max problem as

$$\begin{aligned} \max_{Q \succcurlyeq 0} \quad & \operatorname{tr} A^{-\frac{1}{2}} (\frac{1}{2n} X^\top \operatorname{Diag}(u) X - C) A^{-\frac{1}{2}} Q - \varepsilon \operatorname{tr}[Q \log(Q)] \\ \text{s.t.} \quad & \operatorname{tr} Q = 1. \end{aligned}$$

This problem is of the form

$$\begin{aligned} \max_{Q \succcurlyeq 0} \operatorname{tr} DQ - \varepsilon \sum_{i=1}^n \sigma_i(Q) \log \sigma_i(Q) \\ \text{s.t. } \operatorname{tr} Q = 1 \end{aligned}$$

where $D = A^{-\frac{1}{2}} (\frac{1}{2n} X^\top \operatorname{Diag}(u) X - C) A^{-\frac{1}{2}}$ and $\sigma_i(Q)$ denotes the i-th largest eigen value of the matrix Q. If we consider the matrix D to be of the form $D = U \operatorname{Diag}(\theta) U^\top$ with θ denoting the vector of ordered eigen values of D, then it turns out that at optimality Q has the form $Q = U \operatorname{Diag}(\sigma) U^\top$, with σ denoting the ordered vector of eigen values of Q.

Therefore the above optimization problem can be cast in terms of σ as:

$$\begin{aligned} \max_{\sigma \in \mathbb{R}^n} \theta^\top \sigma - \varepsilon \sum_{i=1}^n \sigma_i \log \sigma_i \\ \text{s.t. } \sum_{i=1}^n \sigma_i = 1. \end{aligned}$$

The solution of this problem is $\sigma_i = \frac{e^{\theta_i/\varepsilon}}{\sum_{j=1}^n e^{\theta_j/\varepsilon}}$, which leads to

$$\min_{\theta \in \mathbb{R}^n} \phi^\varepsilon(\theta) = \varepsilon \log \sum_{i=1}^n \Big(e^{\frac{\theta_i}{\varepsilon}} \Big).$$

In terms of the original matrix variables, we have

$$\min \phi^\varepsilon(D) = \varepsilon \log \operatorname{tr} e^{\frac{D}{\varepsilon}}.$$

Using the appropriate expansion of D, we have the overall optimization problem as

$$\min_{u \in \mathbb{R}_+^n, C : |C_{ij}| \leqslant c_i c_j} \frac{1}{2n} \sum_{i=1}^n \frac{1}{u_i} + \phi^\varepsilon (A^{-\frac{1}{2}} (\frac{1}{2n} X^\top \operatorname{Diag}(u) X - C) A^{-\frac{1}{2}}). \tag{38}$$

At optimality, we have

$$A^{\frac{1}{2}} V A^{\frac{1}{2}} = \Big(e^{\frac{(A^{-\frac{1}{2}} (\frac{1}{2n} X^\top \operatorname{Diag}(u) X - C) A^{-\frac{1}{2}})}{\varepsilon}} \Big) / \operatorname{tr} \Big(e^{\frac{(A^{-\frac{1}{2}} (\frac{1}{2n} X^\top \operatorname{Diag}(u) X - C) A^{-\frac{1}{2}})}{\varepsilon}} \Big).$$

The error of approximation is at most $\varepsilon \log d$ and the Lipschitz constant associated with the function $\phi^\varepsilon(\cdot)$ is $\frac{1}{\varepsilon}$.

G.2 Algorithm details

We write the optimization problem Eq. (38) as:

$$\min_{u \in \mathbb{R}_+^n} F(u, C) + H(u, C)$$

where

$$H(u,C) = \phi^{\varepsilon}(A^{-\frac{1}{2}}(\frac{1}{2n}X^{\top}\operatorname{Diag}(u)X - C)A^{-\frac{1}{2}})$$

is the smooth part and

$$F(u,C) = \mathbb{I}_{C:|C_{ij}|\leqslant c_i c_j} + \frac{1}{2n}\sum_{i=1}^{n}\frac{1}{u_i}$$

is the non-smooth part.

The gradient ∇_u of $H(u,C)$ with respect to u is

$$\nabla_u = \operatorname{diag}(B^{\top}U\operatorname{Diag}(\sigma)U^{\top}B).$$

where $B = \frac{1}{\sqrt{2n}}A^{-\frac{1}{2}}X^{\top}$ and the gradient of $H(u,C)$ with respect to C is

$$\nabla_C = (A^{-\frac{1}{2}}U\operatorname{Diag}(\sigma)U^{\top}A^{-\frac{1}{2}}).$$

The Lipschitz constant L associated with the gradient $\nabla H(u,C)$ is

$$L = \frac{2}{\varepsilon}\max\Big(\lambda_{max}(B^{\top}B \odot B^{\top}B), \lambda^2_{max}(A^{-1})\Big), \tag{39}$$

where $\lambda_{max}(M)$ denotes the maximum eigen value of matrix M. Computing L takes $O(\max(n,d)^3)$ time and L needs to be computed once at the beginning of the algorithm.

The resultant FISTA procedure is described in Algorithm 1. Note that the FISTA procedure first computes intermediate iterates $(\bar{u}^{k-\frac{1}{2}}, \bar{C}^{k-\frac{1}{2}})$ (Step 7, Algorithm 1) by taking descent steps along the respective gradient directions. Then two distinct problems in u and C (respectively Steps 8 and 9 in Algorithm 1) are solved. The sub-problem in u (Step 8) can be efficiently solved using a Newton procedure followed by a thresholding step, as illustrated in Algorithm 2. The sub-problem in C (Step 9) can also be solved using a simple thresholding step.

Algorithm 1 *FISTA Algorithm to solve Eq. (38)*

1: Input X.
2: Compute Lipschitz constant L.
3: Let (u^0, C^0) be an arbitrary starting point.
4: Let $(\bar{u}^0, \bar{C}^0) = (u^0, C^0)$, $t_0 = 1$.
5: Set the maximum iterations to be K.
6: **for** $k = 1, 2, \ldots, K$ **do** ▷ The loop can also be terminated based on duality gap.
7: $(\bar{u}^{k-\frac{1}{2}}, \bar{C}^{k-\frac{1}{2}}) = \Big(\bar{u}^k - \frac{1}{L}\nabla_{\bar{u}^k}, \bar{C}^k - \frac{1}{L}\nabla_{\bar{C}^k}\Big)$.
8: Obtain $u^k = \operatorname{argmin}_{u\in\mathbb{R}^n_+}\Big\{\frac{L}{2}\|u - \bar{u}^{k-\frac{1}{2}}\|^2 + \frac{1}{2n}\sum_{i=1}^{n}\frac{1}{u_i}\Big\}$ by Algorithm 2.
9: Obtain $C^k = \operatorname{argmin}_C\Big\{\mathbb{I}_{C:|C_{ij}|\leqslant c_i c_j} + \frac{L}{2}\|C - \bar{C}^{k-\frac{1}{2}}\|_F^2\Big\}$ by thresholding.
10: $t_k = \frac{1+\sqrt{1+4t_{k-1}^2}}{2}$.
11: $(\bar{u}^k, \bar{C}^k) = (u^k, C^k) + \frac{(t_{k-1}-1)}{t_k}\Big((u^k, C^k) - (u^{k-1}, C^{k-1})\Big)$.
12: **end for**
13: Output (u^K, C^K).

Algorithm 2 *Newton method to solve u sub-problem*

1: Input $u^{k-\frac{1}{2}}$, n, L.
2: $u_i^0 = \max(u_i^{k-\frac{1}{2}}, \frac{1}{(2nL)^{\frac{1}{3}}})$, $i = 1, 2, \ldots, n$.
3: Set $\mathcal{M}$ to be the max number of Newton steps.
4: **for** $t = 1, 2, \ldots, \mathcal{M}$ **do**
5: **for** $i = 1, 2, \ldots, n$ **do**
6: $u_i^t = \frac{2nL(u_i^{t-1})^3 u_i^{k-\frac{1}{2}} + 3u_i^t}{2(nL(u_i^{t-1})^3+1)}$.
7: **end for**
8: **end for**
9: Output $\max(u^{\mathcal{M}}, 0)$.

Journal of Machine Learning Research 18 (2017) 1-36 Submitted 9/16; Revised 3/17; Published 7/17

Making Decision Trees Feasible in Ultrahigh Feature and Label Dimensions

Weiwei Liu LIUWEIWEI863@GMAIL.COM
Centre for Artificial Intelligence
FEIT, University of Technology Sydney
NSW 2007, Australia
and
School of Computer Science and Engineering
The University of New South Wales, Sydney
NSW 2052, Australia

Ivor W. Tsang IVOR.TSANG@UTS.EDU.AU
Centre for Artificial Intelligence
FEIT, University of Technology Sydney
NSW 2007, Australia

Editor: Inderjit Dhillon

Abstract

Due to the non-linear but highly interpretable representations, decision tree (DT) models have significantly attracted a lot of attention of researchers. However, it is difficult to understand and interpret DT models in ultrahigh dimensions and DT models usually suffer from the curse of dimensionality and achieve degenerated performance when there are many noisy features. To address these issues, this paper first presents a novel data-dependent generalization error bound for the perceptron decision tree (PDT), which provides the theoretical justification to learn a sparse linear hyperplane in each decision node and to prune the tree. Following our analysis, we introduce the notion of budget-aware classifier (BAC) with a budget constraint on the weight coefficients, and propose a supervised budgeted tree (SBT) algorithm to achieve non-linear prediction performance. To avoid generating an unstable and complicated decision tree and improve the generalization of the SBT, we present a pruning strategy by learning classifiers to minimize cross-validation errors on each BAC. To deal with ultrahigh label dimensions, based on three important phenomena of real-world data sets from a variety of application domains, we develop a sparse coding tree framework for multi-label annotation problems and provide the theoretical analysis. Extensive empirical studies verify that 1) SBT is easy to understand and interpret in ultrahigh dimensions and is more resilient to noisy features. 2) Compared with state-of-the-art algorithms, our proposed sparse coding tree framework is more efficient, yet accurate in ultrahigh label and feature dimensions.

Keywords: Classification, Ultrahigh Feature Dimensions, Ultrahigh Label Dimensions, Perceptron Decision Tree

©2017 Weiwei Liu and Ivor W. Tsang.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-466.html.

1. Introduction

Decision tree (DT) (Moret, 1982; Safavian and Landgrebe, 1991) has been identified as one of the top 10 data mining and machine learning algorithms (Wu et al., 2008). Due to the non-linear, but highly interpretable representations (Murthy et al., 1994), DT has played a vital role in various data mining and machine learning applications, ranging from Agriculture (McQueen et al., 1995), Astronomy (Salzberg et al., 1995), Medicine (Czajkowski et al., 2014) and Molecular Biology (Shimozono et al., 1994) to Financial analysis (Mezrich, 1994).

Many early works focus on the DT, in which each node uses the value of a single feature (attribute) (Breiman et al., 1984; Quinlan, 1993) for decision. Since the decision of these methods at each node is equivalent to an axis-parallel hyperplane in the feature space, they are called axis-parallel DT, which suffer from the curse of dimensionality. To improve the performance of axis-parallel DT, researchers developed perceptron decision tree (PDT) (Bennett et al., 2000; Liu and Tsang, 2016) in which the test at each decision node uses the linear combinations of features. Clearly, PDT is the general form of axis-parallel DT. This paper first studies two emerging challenges for decision trees – ultrahigh feature dimensionality and ultrahigh label dimensionality. Then, we develop budget-aware classifier (BAC) and supervised budgeted tree to address ultrahigh feature dimensionality. Under the BAC framework, we further propose a sparse coding tree to address the second challenge.

1.1 Challenges of Ultrahigh Feature Dimensionality

Recently, the emerging trends of ultrahigh feature dimensionality have been analyzed in Mitchell et al. (2004), Fan et al. (2009) and Zhai et al. (2014). For example, the Web continues to generate quintillion bytes of data daily (Zikopoulos et al., 2012; Zhai et al., 2014), leading to three key challenges for PDT classification on the webspam data set with 16,609,143 features (which is collected from Wang et al. (2012)) :

1. It is difficult to understand and interpret PDT on this data set. How to make PDT interpretable in ultrahigh dimensions poses a critical challenge;

2. Many features in high dimensional space are usually non-informative or noisy. Those noisy features will decrease the generalization performance of PDT and derogate from their promising results for dealing with non-linear separable data, especially when there are many noisy features;

3. It is desired to identify a small set of features that is able to represent the original feature space in this ultrahigh dimensional data set.

To overcome these challenges, it is imperative to develop a sparse PDT with respect to input features and its corresponding theory.

1.2 Challenges of Ultrahigh Label Dimensionality

Annotation, which aims to tag objects with some labels to enhance the semantic understanding of the objects, has arisen in various applications. For example, in image annotation (Boutell et al., 2004; Guillaumin et al., 2009), one needs to predict some relevant keywords, such as *beach*, *sky* and *tree*, to describe a natural scene image. When annotating documents

(Schapire and Singer, 2000; Paralic and Bednar, 2003), one may need to classify them into different groups according to their annotations, such as *News*, *Finance* and *Education*. In video annotation (Song et al., 2005; Tang et al., 2008), some labels, such as *Government*, *Policy* and *Election* are needed to describe the subject of the video. In these real-world applications, the object needs to be annotated with multiple labels. Thus, it can be formulated as a multi-label annotation problem (Tsoumakas et al., 2010; Chen and Lin, 2012; Deng et al., 2014; Zhang and Zhou, 2014; Liu and Tsang, 2015b; Koyejo et al., 2015; Yen et al., 2016), which aims to annotate multiple labels for each input instance.

One central challenging issue for practical multi-label annotation is scalability in ultrahigh label and feature dimensions. Suppose one needs to annotate the presence or absence of q labels for each testing instance with m features, early approaches, such as Tsoumakas et al. (2010) and Read et al. (2011), scaling the number of annotations to be linear with the number of labels, take $\mathcal{O}(q \times m)$ time for annotation. We cannot get the results of these exhaustive approaches on the large-scale data sets within one week, such as amazon data set with 2,812,281 labels and 337,067 features, which is collected from McAuley et al. (2015a,b).

Recently, many works (Tsoumakas et al., 2008; Agrawal et al., 2013; Prabhu and Varma, 2014) have put more effort in exploring tree-based algorithms to minimize the number of annotations for multi-label annotations with ultrahigh label dimensions. Among them, FastXML (Prabhu and Varma, 2014) is the most recently advanced technique, which has shown state-of-the-art rapid annotations. However, there is no generalized performance guarantee for FastXML and our extensive empirical study verifies that FastXML generally underperforms in terms of annotation performance. Moreover, some literature (Tsoumakas et al., 2008; Madjarov et al., 2012) has shown Homer (Tsoumakas et al., 2008) achieves good annotation performance. Unfortunately, Homer requires at least $\mathcal{O}(q^2)$ time for training. Thus, we cannot get the results of Homer even on medium-sized data sets within one week. The question is: can we design some efficient, yet accurate multi-label annotation algorithms in ultrahigh label and feature dimensions?

1.3 Our Contributions and Organization

To address these challenges, we first revisit the PDT model. Specifically, we first present a novel generalization error bound for the PDT, which takes into account the distribution of the data, and provides the theoretical justification to learn a sparse hyperplane classifier in each decision node and prune the tree. From the analysis of our bound, the training error loss, the margin, the capacity of the kernel matrix defined on the training data, and the complexity of the tree are the dominant factors that affect the generalization performance of PDTs. In particular, our bound indicates that decreasing training error loss and the capacity of the kernel matrix, and enlarging the margin can yield better generalization performance. Suppose linear kernel is used, decreasing the capacity of the kernel matrix can be done via feature selection. Thereafter, we introduce the notion of budget-aware classifier (BAC) to perform feature selection and optimize training error loss and margin simultaneously, and then develop a supervised budgeted tree (SBT) to achieve non-linear prediction performance. To obtain a stable DT structure, instead of conducting ensemble, we propose an objective to learn classifiers and to identify a universal feature subset that

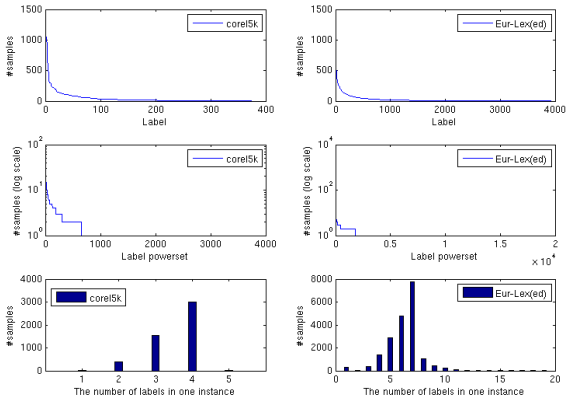


Figure 1: **Top**: Frequency of each label on the corel5k and Eur-Lex(ed) data sets. **middle**: Frequency of each label powerset on the corel5k and Eur-Lex(ed) data sets. **bottom**: Frequency of samples with the specific number of labels on the corel5k and Eur-Lex(ed) data sets.

minimizes the cross validation errors on each BAC. After that, we further diminish the generalization error by pruning the tree.

Based on BAC, we can design the tree-based algorithm to handle ultrahigh feature dimensions. How can one deal with ultrahigh label dimensions? To answer this question, we first use Figure 1 to demonstrate three important phenomena of real-world multi-label annotation data sets from various application domains[1]. We use corel5k and Eur-Lex(ed) data sets as examples. The details of these data sets are shown in Table 7.

Observations of Label Distribution: The labels of instances usually follow the distribution of Power Law (Barabási et al., 2000) which is shown in (Bhatia et al., 2015), as shown at the top of Figure 1, where many labels have very low label frequency. The bottom of Figure 1 shows the sparsity of labels in each instance, which means that the average number of labels in the data set is small. The middle of Figure 1 shows that very few label powersets have high frequency. Note that similar observations can also be found in image annotation tasks (Mao et al., 2013).

In real-world problems, although the number of labels would be very large, the first observation indicates that only a few labels appear very frequently; many others are rare. Based on this property, we design a tree-based algorithm to reduce the overall annotations as follows. Frequent labels should be predicted first in the decision tree, so there will be fewer annotations involved to predict the frequent labels, and more annotations for rare labels. To achieve this, we develop a novel annotation tree (AT) algorithm, which is analogous to the Shannon-Fano coding (Shannon, 1948) strategy, to deal with multi-label annotation problems. Moreover, the second observation shows that each instance typically

1. http://mulan.sourceforge.net

has few labels, which enables early stopping of our tree model. Thus, fewer annotations will be made along shorter paths in our tree model for most testing instances. Based on the property of the last observation, we also develop the powerset tree (PT) algorithm to deal with the transformed multi-class annotation task.

This paper is organized as follows. Section 2 reviews related work. Section 3 introduces a novel data-dependent generalization error bound for PDT. Based on our analysis, Section 4 develops the budget-aware classifier (BAC) to handle ultrahigh feature dimensions. Section 5 proposes the supervised budgeted tree (SBT) to construct a powerful non-linear classifier by arranging BAC in a tree-structure. Furthermore, based on BAC, Section 6 introduces the sparse coding tree framework for multi-label annotation with ultrahigh label and feature dimensions and provides generalization error bound for our proposed powerset tree (PT) and annotation tree (AT). Section 7 and Section 8 present the implementation issues and evaluate the performance of our proposed methods in ultrahigh dimensions, respectively. Lastly, Section 9 concludes the work.

2. Related Work

In this section, we introduce some backgrounds about the perceptron decision tree, multi-label annotation, multiple kernel learning, Huffman coding and Shannon-Fano coding.

2.1 Perceptron Decision Tree

OC1 (Murthy et al., 1994) is a popular PDT, which belongs to a randomized algorithm that performs a randomized hill-climbing search to learn the classifiers and builds the tree in a top-down approach. Murthy et al. (1994) show the competitive results of OC1 compared with C4.5 (Quinlan, 1993) (which is the famous axis-parallel DT) and other baselines. The authors in (Bennett et al., 2000) first provide the margin bound for the perceptron decision tree (PDT), then update the weights of OC1 with the maximum margin for each node. However, the results of Bennett et al. (2000) show that the performance of modified-OC1 with the maximum margin for each node is similar with OC1. The main reason may be the data dependency of the bound comes from the margin only.

The combination of bagging and building ensembles of trees (Breiman, 2001) is a popular strategy to control the generalization error of decision trees. Moreover, due to PDT's performance is sensitive to the tree structure, which is determined by the distribution of features, ensembles offer an effective technique for obtaining increased accuracy by combining the predictions of many different trees, such as random forest feature selection (RFFS) (Hastie et al., 2001) and Gradient boosted feature selection (GBFS) (Xu et al., 2014). GBFS is an advanced gradient boosted DT (Friedman, 2000), which employs gradient boosting to do feature selection in DT. GBFS, which belongs to axis-parallel DT, shows its state-of-the-art feature selection performance compared with l_1-regularized logistic regression (Lee et al., 2006), RFFS, and some other promising baselines. Cost-Sensitive Tree of Classifiers (CSTC) (Xu et al., 2013) is another state-of-the-art approach for feature selection in DT, which follows the PDT. Both are based on the l_1-regularized model (Zhu et al., 2003). Unfortunately, these l_1-regularized methods suffer from three major limitations:

1. Because the scale variation of weight parameters, it is non-trivial for these methods to control the number of features to be selected meanwhile to regulate the decision function;

2. Since l_1-norm regularization shrinks the regressors, the feature selection bias inevitably exists in the l_1-norm methods (Zhang and Huang, 2008; Zhang, 2010a,b);

3. l_1-norm regularized methods are inefficient or infeasible for handling the data sets with ultrahigh dimensions. Moreover, the theoretical and empirical analysis in Zhang and Huang (2008), Zhang (2010a), Zhang (2010b), and Tan et al. (2014) have shown that l_1-regularized methods achieve suboptimal performance in feature selection, compared with the sparse model with an explicit budget constraint on the weights.

Region-specific and locally linear models (Jose et al., 2013; Oiwa and Fujimaki, 2014) are an advanced technique for dealing with the non-linear problem. Local deep kernel learning (LDKL) (Jose et al., 2013) formulates the problem from a local kernel learning perspective where they learn tree-structured primal feature embedding and has shown its superior performance in (Jose et al., 2013). Partition-wise linear models (Oiwa and Fujimaki, 2014) assign linear models to partitions and represent region-specific linear models by linear combinations of partition-specific models. They first formalize the problem as l_0 penalizing problem and then apply a convex relaxation with a combination of l_1 penalties. The model is similar with GBFS and CSTC, which is different in penalizing terms. In this paper, we select GBFS and CSTC as the representatives of these kinds of methods for comparison.

2.2 Multi-Label Annotation

Much effort has been focused on annotation tasks, such as image annotation (Boutell et al., 2004) and video annotation (Song et al., 2005). Usually, annotation tasks can be formulated as a multi-label annotation problem (Tsoumakas et al., 2010; Chen and Lin, 2012; Deng et al., 2014; Zhang and Zhou, 2014; Koyejo et al., 2015; Liu and Tsang, 2015a; Gong et al., 2017). According to the annotation complexity, we divide these methods into several categories:

The first category includes some popular methods, such as Binary Relevance (BR) (Tsoumakas et al., 2010) and Classifier Chain (CC) (Read et al., 2011), the number of annotations scale linearly with the length of label vector. All these methods cannot handle ultrahigh label dimensions.

The second category includes the encoding-decoding strategy. For example, (Zhang and Schneider, 2011, 2012) first use different projection methods to transform the original label space into another space, and recover the original multiple labels using an expensive decoding process, which involves solving a quadratic programming (QP) problem on a space with a combinatorial nature.

The third category includes Label Powerset (LP) (Tsoumakas et al., 2010) and its variant (Tsoumakas and Vlahavas, 2007). LP reduces a multi-label annotation problem into a multi-class annotation problem by treating each distinct label set as one of the classes for a transformed multi-class learning task. One can then train a single multi-class classifier (Tsoumakas et al., 2010) or many binary classifiers (for example one-vs-all or one-vs-one) for the transformed multi-class annotation problem. The number of transformed classes

is upper bounded by $\min(n, 2^q)$ where n means the size of training data set and q means the number of labels. For large values of n and q, it imposes an extremely high training complexity.

The last category is tree-based algorithms, which have annotation costs that are logarithmic in the number of labels. FastXML (Prabhu and Varma, 2014) is one of the most advanced tree-based algorithms. However, there is no generalized performance guarantee for FastXML and our extensive empirical study verifies that FastXML generally underperforms in terms of annotation performance. Homer (Tsoumakas et al., 2008) is developed to use a divide-and-conquer-strategy to divide original problem into k sub-problems. Some literature (Tsoumakas et al., 2008; Madjarov et al., 2012) has shown Homer achieves good annotation performance. However, Homer requires at least $\mathcal{O}(q^2)$ time for training. Thus, we cannot get the results of Homer even on medium-sized data sets within one week.

2.3 Multiple Kernel Learning

By combining multiple sets of features, multiple kernel learning (MKL) (Lanckriet et al., 2004b; Rakotomamonjy et al., 2008) has become a powerful learning paradigm to enhance the interpretability of the decision function and improve performances of single kernel methods. MKL has been successfully applied in many tasks, such as object recognition (Bucak et al., 2014), signal processing (Subrahmanya and Shin, 2010), and genomic data fusion (Lanckriet et al., 2004a).

Many efficient MKL algorithms (Lanckriet et al., 2004b; Bach et al., 2004; Sonnenburg et al., 2006; Rakotomamonjy et al., 2008; Liu et al., 2015) are developed. SimpleMKL (Rakotomamonjy et al., 2008) is one of the most popular MKL solvers. SimpleMKL uses a sub-gradient method to solve the non-smooth optimization problem. However, it is expensive to calculate the sub-gradient for large-scale problems. Tan et al. (2014) develop an up-to-date MKL solver, which modifies accelerated proximal gradient method to solve the primal form of MKL and shows promising results. This paper will adapt this method to solve our subproblems.

2.4 Huffman Coding and Shannon-Fano Coding

Huffman coding (Huffman, 1952) is one of the most widespread bottom-up encoding algorithm for data compression. Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix code, that is, the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol. Given a set of symbols and their weights (usually proportional to probabilities), Huffman coding aims to find a set of prefix codewords with minimum expected codeword length. Based on the frequency of occurrence of each symbol, the principle of Huffman coding is to use a lower number of bits to encode the symbol that occurs more frequently. The detailed procedure of Huffman coding can be referred to Huffman (1952). This work first uses the idea of Huffman coding to deal with the label powerset annotation problems.

Shannon-Fano coding (Shannon, 1948) is a top-down encoding technique for constructing a prefix code based on a set of symbols and their weights. Shannon-Fano coding arranges the symbols in order from biggest weight to smallest weight, and then divides them into two sets whose total weights are roughly comparable. Then, we encode symbols in the first

Table 1: Important notations used in this paper.

Notation	description
n	number of instances
m	dimensionality of the input space
q	dimensionality of the output space
$\mathcal{X}$	the input space over $\mathbb{R}^m$
$\mathcal{Y}$	the output space ($\mathcal{Y} = \{-1, 1\}$ for binary classification, while $\mathcal{Y} \subseteq \{1, 2, \cdots, q\}$ for multi-label classification)
x_i	$x_i \in \mathbb{R}^m$ is a real vector representing the i-th input or instance
y_i	$y_i \in \{-1, 1\}$ is used to represent the i-th label for binary classification
$\mathbf{y}_i$	$\mathbf{y}_i \in \{0, 1\}^q$ is used to represent the i-th label set for multi-label classification, where $\mathbf{y}_i(j) = 1$ if and only if $j \in \mathcal{Y}_i$
X	instance matrix, where $X = [x_1, \cdots, x_n]'$
Y	binary label matrix, where $Y = [y_1, \cdots, y_n]'$
ϱ	$\varrho = [\varrho_1, \cdots, \varrho_m]' \in [0, 1]^m$ represents a feature selection vector, and $\sqrt{\varrho}$ denotes the elementwise square root operator
B	budget constraint on the weight coefficients
$\mathcal{A}$	domain of dual variable α, where $\mathcal{A} = \{\alpha \mid \sum_{i=1}^n \alpha_i \geq 1, 0 \leq \alpha_i \leq C, i = 1, \cdots, n\}$
$\mathcal{E}$	domain of ϱ, where $\mathcal{E} = \{\varrho \in [0, 1]^m \mid \sum_{j=1}^m \varrho_j \leq B\}$

set as zero and symbols in the second set as one. As long as any sets with more than one member remain, the same process is repeated on these sets. The detailed procedure of Shannon-Fano coding can be referred to Shannon (1948). Analogous to the Shannon-Fano coding strategy, this work first proposes a novel annotation tree algorithm to deal with multi-label annotation problems.

3. Data-Dependent Generalization Error Bound for PDT

Inspired by Bartlett and Mendelson (2002) and Shawe-Taylor and Cristianini (2004), this section provides the generalization error bound for the PDT, which takes into account the distribution of the data. We begin with some basic definitions. We denote the transpose of vector/matrix by the superscript $'$. Let $\odot$ represent the elementwise product sign. $\|\cdot\|$ denotes the l_2 norm. If $\mathcal{S}$ is a set, $|\mathcal{S}|$ denotes its cardinality. Let $\mathbb{R}$ represent real number, $\mathcal{X}$ be the input space over $\mathbb{R}^m$ and $\mathcal{Y} = \{-1, 1\}$ be the output space. Suppose $sign(\cdot)$ represents sign function; $tr(\cdot)$ denotes the trace of matrix argument; $ln(\cdot)$ represents the natural logarithm and $\Theta(\cdot)$ returns 1 if its argument is greater than 0, and zero otherwise. $\hat{E}_n$ denotes the empirical risk over n training samples $(x_1, y_1), \cdots, (x_n, y_n)$. Let F be a class of functions mapping from $\mathcal{X}$ to $\mathbb{R}$. We follow the definitions in Koltchinskii (2001), Bartlett and Mendelson (2002) and Mendelson (2002), and use $R_n(F)$ and $G_n(F)$ to denote the Rademacher and Gaussian complexity of F, respectively. The important notations in this paper are summarized in Table 1.

Definition 1 (Generalized Decision Trees (GDT), Bennett et al. (2000)) *Given $\mathcal{X} = \mathbb{R}^m$, and a set of Boolean functions $F_{GDT} = \{f_{GDT} : \mathcal{X} \to \{0, 1\}\}$, the class of Generalized Decision Trees over F_{GDT} is the set of functions that can be implemented using a binary*

tree where each internal node is labeled with an element of F_{GDT}, and each leaf is labeled with either 1 or 0.

Definition 2 (Perceptron Decision Tree (PDT), Bennett et al. (2000)) *Given $\mathcal{X} = \mathbb{R}^m$, and assume g is a linear real function over $\mathcal{X}$ and $w \in \mathbb{R}^m$, a Perceptron Decision Tree (PDT) is a GDT over*

$$F_{PDT} = \{h_w : h_w(x) = \Theta(g_w(x)), g_w(x) = w'x + b\}$$

Next, we introduce two important Theorems.

Theorem 3 (Bartlett and Mendelson (2002)) *Let D be a probability distribution on $\mathcal{X} \times \mathcal{Y}$ and let F be a set of real valued functions defined on $\mathcal{X}$, with $\sup\{|\mathfrak{F}(x)| : \mathfrak{F} \in F\}$ finite for all $x \in \mathcal{X}$. Suppose that $\phi : \mathbb{R} \to [0,1]$ satisfies $\phi(\alpha) \geq \Theta(-\alpha)$ and is Lipschitz with constant L, then with probability at least $1-\delta$ with respect to n training samples $(x_1, y_1), \cdots, (x_n, y_n)$ drawn independently according to D, every function in F satisfies*

$$P_D(y \neq sign(\mathfrak{F}(x))) \leq \hat{E}_n(\phi(y\mathfrak{F}(x))) + 2LR_n(F) + \sqrt{\frac{ln(2/\delta)}{2n}}$$

Theorem 4 (Shawe-Taylor and Cristianini (2004)) *Fix margin γ and let $\mathcal{F}$ be the class of functions mapping from $\mathcal{X} \times \mathcal{Y}$ to $\mathbb{R}$ given by $f(x,y) = -yg(x)$ ($x \in \mathcal{X}$, $y \in \mathcal{Y}$), where g is a linear function in a feature space induced by a kernel $k(\cdot,\cdot)$ with Euclidean norm at most 1. Then, with probability at least $1-\delta$ with respect to n training samples $(x_1, y_1), \cdots, (x_n, y_n)$ drawn independently according to D, we have*

$$P_D(y \neq sign(g(x))) \leq \frac{1}{n\gamma}\sum_{i=1}^{n}\xi_i + \frac{4}{n\gamma}\sqrt{tr(K)} + 3\sqrt{\frac{ln(2/\delta)}{2n}}$$

where K is a kernel matrix defined on the training data with $k(\cdot,\cdot)$ and $\xi_i = \max(0, \gamma - y_i g(x_i))$.

We provide the data-dependent generalization error bound for PDT as follows:

Theorem 5 *Let T be a PDT, where the Euclidean norm of the linear functions in T is at most 1. Suppose that the depth of T is no more than d, and it contains no more than M leaves. Let $(x_1, y_1), \cdots, (x_n, y_n)$ be n training samples, which is generated independently according to a probability distribution D. Then, we can bound the generalization error with probability greater than $1-\delta$ to be less than*

$$\sum_{l}\sum_{i=1}^{d_l}\mathcal{D}_i^l + \frac{5cdMG_n(T)}{2} + \big((3d+1)\sqrt{n}M + 1\big)\sqrt{\frac{ln(6/\delta)}{2n}}$$

where c is a universal small constant, $d_l (d_l \leq d)$ is the depth of leaf l, $\mathcal{D}_i^l = \frac{1}{n_l\gamma_i^l}\sum_{j=1}^{n_l}\xi_{i,j}^l + \frac{4}{n_l\gamma_i^l}\sqrt{tr(K_l)}$, n_l denotes the number of samples $(x_1^l, y_1^l), \cdots, (x_{n_l}^l, y_{n_l}^l)$ reaching leaf l, K_l is the kernel matrix for $\{x_1^l, \cdots, x_{n_l}^l\}$, $\xi_{i,j}^l = \max\big(0, \gamma_i^l - y_{i,j}^l g_i^l(x_j^l)\big)$, where $y_{i,j}^l$ represents the correct output for input x_j^l in decision node i and γ_i^l denotes the corresponding margin.

Proof For a tree of depth d, the indicator function of a leaf is a conjunction of no more than d decision functions. More specifically, if the decision tree consists of decision nodes chosen from a class T of Boolean functions, the indicator function of leaf l (which takes value 1 at a point x if x reaches l and 0 otherwise) is a conjunction of d_l functions from T, where d_l is the depth of leaf l. We can represent the function computed by the tree as follows:

$$f(x) = \sum_l \sigma_l \bigwedge_{i=1}^{d_l} h_{l,i}(x)$$

where the sum is over all leaves l, $\sigma_l \in \{\pm 1\}$ is the label of leaf l, $h_{l,i} \in T$, and the conjunction is understood to map to $\{0,1\}$. Let F be this class of functions. Choose a family $\{\phi_L : L \in \mathbb{N}\}$ of cost functions such that each ϕ_L dominates the step function $\Theta(-yf(x))$ and has a Lipschitz constant L. For each L, Theorem 3 implies that with probability at least $1-\delta_1$:

$$P(y \neq f(x)) \leq \hat{E}_n(\phi_L(yf(x))) + 2LR_n(F) + \sqrt{\frac{ln(2/\delta_1)}{2n}}$$

Define $\phi_L(\alpha)$ to be 1 if $\alpha \leq 0$, $1 - L\alpha$ if $0 < \alpha \leq 1/L$, and 0 otherwise. Let $\tilde{P}_n(l)$ denotes the proportion of all the samples in S which reach leaf l and are correctly classified. Assume $(x_1^l, y_1^l), \cdots, (x_{n_l}^l, y_{n_l}^l)$ reach leaf l with $|S_l| = n_l$. Then we have

$$\begin{aligned}
&\hat{E}_n(\phi_L(yf(x))) \\
&= \sum_l \frac{\sum_{(x^l,y^l)\in S_l}\Theta(-y^l f(x^l))}{n} + \sum_l \tilde{P}_n(l)\phi_L(1) \\
&= \sum_l \frac{\sum_{(x^l,y^l)\in S_l}\Theta(-y^l f(x^l))}{n} + \sum_l \tilde{P}_n(l)max(0, 1\text{–}L)
\end{aligned} \tag{1}$$

If no sample reaches leaf l, the sum term of leaf l in the Eq.(1) is zero. Here we consider $1 \leq n_l < n$. As $L \in \mathbb{N}$, the second term in Eq.(1) is zero. Following Theorem 5 in (Bartlett and Mendelson, 2002), with probability at least $1-\delta_2$, we get:

$$\begin{aligned}
&\frac{\sum_{(x^l,y^l)\in S_l}\Theta(-y^l f(x^l))}{n} \leq \frac{\sum_{(x^l,y^l)\in S_l}\Theta(-y^l f(x^l))}{n_l} \\
&\leq P(y^l \neq f(x^l)) + \frac{R_n(F)}{2} + \sqrt{\frac{ln(2/\delta_2)}{2n_l}}
\end{aligned} \tag{2}$$

The probability of x^l misclassified is equal to the probability of at least one decision node from the root to a leaf l making the errors. Suppose $\{g_1^l, \cdots, g_{d_l}^l\}$ are the linear functions with the Euclidean norm at most 1 and associated with each decision node from the root to a leaf l. The corresponding margins are $\{\gamma_1^l, \cdots, \gamma_{d_l}^l\}$. For input $x^l \in \{x_1^l, \cdots, x_{n_l}^l\}$, the correct output of each decision node from the root to a leaf l is denoted by $\{y_1^l, \cdots, y_{d_l}^l\}$, where $y_i^l \in \{y_{i,1}^l, \cdots, y_{i,n_l}^l\}$. By the union bound and Theorem 4, with probability at least

$1-\delta_3$, we get the following:

$$\begin{aligned}
&P(y^l \neq f(x^l)) \\
=&P\Big(\bigcup_{i=1}^{d_l}(y_i^l \neq sign(g_i^l(x^l)))\Big) \\
\leq&\sum_{i=1}^{d_l} P(y_i^l \neq sign(g_i^l(x^l))) \\
\leq&\sum_{i=1}^{d_l}\Big(\frac{1}{n_l\gamma_i^l}\sum_{j=1}^{n_l}\xi_{i,j}^l + \frac{4}{n_l\gamma_i^l}\sqrt{tr(K_l)}\Big) + 3d_l\sqrt{\frac{ln(2/\delta_3)}{2n_l}}
\end{aligned} \tag{3}$$

Let $\delta_1 = \delta_2 = \delta_3 = \delta/3$ and $L = M$. The results of Eq.(1), Eq.(2) and Eq.(3) imply that with probability at least $1-\delta$:

$$P(y \neq f(x)) \leq \sum_l\sum_{i=1}^{d_l}\mathcal{D}_i^l + \frac{5MR_n(F)}{2} + \big((3d+1)\sqrt{n}M+1\big)\sqrt{\frac{ln(6/\delta)}{2n}}$$

where $\mathcal{D}_i^l = \frac{1}{n_l\gamma_i^l}\sum_{j=1}^{n_l}\xi_{i,j}^l + \frac{4}{n_l\gamma_i^l}\sqrt{tr(K_l)}$. Let c is a constant. Theorem 12, Theorem 16 and Lemma 4 in (Bartlett and Mendelson, 2002) imply that $R_n(F) \leq cdG_n(T)$, which implies the result. ■

4. Budget-Aware Classifier

Theorem 5 reveals important factors to reduce the generalization error bound for the PDT model: the first is to decrease the training error loss and the capacity of the kernel matrix for each node, and the second is to enlarge the margin for each node. We first enforce an upper bound budget $\mathfrak{B}$ for the capacity of the kernel matrix, that is $tr(K_l) \leq \mathfrak{B}$. Suppose there are n samples $(x_1, y_1), \cdots, (x_n, y_n)$, $x_i \in \mathbb{R}^m$ ($x_i = [x_{i,1}, \cdots, x_{i,m}]'$). Let $X = [x_1, \cdots, x_n]'$ and $Y = [y_1, \cdots, y_n]'$. Since we use linear kernel here, decreasing the capacity of the kernel matrix can be transformed to a budget constraint on the weight coefficients, which leads to a feature selection problem for the linear decision function. Mathematically, let $\hat{K}$ denote the kernel matrix on the re-scaled instance: $\hat{x}_i = x_i \odot \sqrt{\varrho}$, where $\varrho = [\varrho_1, \cdots, \varrho_m]' \in [0,1]^m$ represents a feature selection vector and $\sqrt{\varrho}$ denotes the elementwise square root operator. The capacity constraint on the kernel matrix is $tr(\hat{K}) \leq \mathfrak{B}$, then $tr(\hat{K}) = \sum_{i=1}^n \varrho'(x_i \odot x_i) \leq n\tilde{x}_{max}\sum_{j=1}^m \varrho_j$, where $\tilde{x}_{max}$ be the maximum element in $[x_1 \odot x_1, \cdots, x_n \odot x_n]$. Let $\mathfrak{B}/n\tilde{x}_{max} = B$, where B is a budget constraint on the weight coefficients. Thus $tr(\hat{K}) \leq \mathfrak{B}$ can be transformed to $\sum_{j=1}^m \varrho_j \leq B, \varrho \in [0,1]^m$. Then we learn a sparse linear decision function on each node by minimizing the training error loss and maximizing the margin at the same time. Based on Theorem 5, we hereinafter introduce the notion of budget-aware classifier (BAC).

Definition 6 (Budget-Aware Classifier (BAC)) *Assume there are n samples $(x_1, y_1), \cdots, (x_n, y_n)$ drawn independently according to a probability distribution D, where $x_i \in \mathbb{R}^m$ and $y_i \in \{\pm 1\}$. $\varrho = [\varrho_1, \cdots, \varrho_m]' \in [0,1]^m$ represents a feature selection vector. Let $\hat{K}$ denote the kernel matrix on the re-scaled instance: $\hat{x}_i = x_i \odot \sqrt{\varrho}$ and the capacity constraint on the kernel matrix can be transformed as $\sum_{j=1}^m \varrho_j \leq B$ with a budget B. A budget-aware classifier contains a linear decision function $g_w(x) = w'(x_i \odot \sqrt{\varrho} + b), ||w||^2 = 1$, where the parameters are obtained by solving the following problem :*

$$\begin{aligned}
\min_{\varrho\in\mathcal{E}, w, \xi, \gamma} \; & -\gamma + C\sum_{i=1}^n \xi_i \\
s.t. \quad & y_i w'(x_i \odot \sqrt{\varrho} + b) \geq \gamma - \xi_i, \xi_i \geq 0, i = 1, \cdots, n \\
& ||w||^2 = 1, \gamma \geq 0
\end{aligned} \tag{4}$$

where $\mathcal{E} = \{\varrho \in [0,1]^m | \sum_{j=1}^m \varrho_j \leq B\}$ and B is a budget constraint on the weight coefficients.

4.1 Learning the BAC

For the sake of clarity, we consider the function without an offset, although the model can be extended to the function with the offset. For problem (4) with respect to w, ξ, γ, the corresponding Lagrangian function is: $\mathcal{L}(w, \xi, \gamma, \alpha, \lambda, \psi, \nu) = -\gamma + C\sum_{i=1}^n \xi_i - \sum_{i=1}^n \alpha_i(y_i w'(x_i \odot \sqrt{\varrho}) - \gamma + \xi_i) + \lambda(\|w\|^2 - 1) - \sum_{i=1}^n \psi_i\xi_i - \nu\gamma$, with $\alpha_i, \psi_i, \nu \geq 0$. Differentiating with respect to the primal variables and substituting the relations obtained into the primal, we obtain the following dual objective function: $\mathcal{L}(\alpha, \lambda, \varrho) = -\frac{1}{4\lambda}\big\|\sum_{i=1}^n \alpha_i y_i (x_i \odot \sqrt{\varrho})\big\|^2 - \lambda$. Optimizing λ gives $\lambda^* = \frac{1}{2}\|\sum_{i=1}^n \alpha_i y_i (x_i \odot \sqrt{\varrho})\|$, resulting in $\mathcal{L}(\alpha, \varrho) = -\big\|\sum_{i=1}^n \alpha_i y_i (x_i \odot \sqrt{\varrho})\big\|$. Optimizing the squared l_2-norm objective and l_2-norm objective yields the same solution. To simplify the optimization, we use the squared l_2-norm as the objective. Thus, problem (4) can be equivalently reformulated as the following problem:

$$\min_{\varrho\in\mathcal{E}}\max_{\alpha\in\mathcal{A}} -\frac{1}{2}\Big\|\sum_{i=1}^n \alpha_i y_i (x_i \odot \sqrt{\varrho})\Big\|^2 \tag{5}$$

where $\mathcal{A} = \{\alpha | \sum_{i=1}^n \alpha_i \geq 1, 0 \leq \alpha_i \leq C, i = 1, \cdots, n\}$.

Let $\mathbb{G}(\alpha, \varrho) = \frac{1}{2}\Big\|\sum_{i=1}^n \alpha_i y_i (x_i \odot \sqrt{\varrho})\Big\|^2$. According to the minimax saddle-point theorem (Sion, 1958), we have the following theorem:

Theorem 7 *According to the minimax saddle-point theorem in (Sion, 1958), the following equality holds:*

$$\min_{\varrho\in\mathcal{E}}\max_{\alpha\in\mathcal{A}} -\mathbb{G}(\alpha, \varrho) = \max_{\alpha\in\mathcal{A}}\min_{\varrho\in\mathcal{E}} -\mathbb{G}(\alpha, \varrho)$$

Proof Recall that both $\mathcal{A}$ and $\mathcal{E}$ are convex compact set. It can be easily verified that the objective function is concave with respect to α. As shown in Section 4.1.1, the objective function is linear with respect to ϱ, and also is convex in ϱ. Therefore, the conditions in minimax saddle-point theorem in (Sion, 1958) are satisfied here. ■

Algorithm 1 Cutting Plane Algorithm for Solving Problem (7)

Input: Data set $\{x_i, y_i\}_{i=1}^n$. Initialize α_0.

1: $t = 0$, $\mathcal{C}_t = \emptyset$.
2: **repeat**
3: $t = t + 1$.
4: Worst-case analysis: Finding the most violated ϱ_t based on α_{t-1} and set $\mathcal{C}_t = \mathcal{C}_{t-1} \cup \{\varrho_t\}$. The details can be referred to Section 4.1.1.
5: Master-problem optimization: Solving problem (8) over $\mathcal{C}_t$ and obtaining the optimal solution α_t. The details can be referred to Section 4.1.2.
6: **until** ϵ-optimal

Based on Theorem 7, instead of directly solving problem (5), we address the following problem instead:

$$\min_{\alpha\in\mathcal{A}} \max_{\varrho\in\mathcal{E}} \mathbb{G}(\alpha, \varrho) \tag{6}$$

where we drop the negative sign of problem (6) for the clarity of our optimization procedure. By introducing variable $\theta \in \mathbb{R}$, problem (6) can be further formulated as a semi-infinite programming (SIP) (Pee and Royset, 2011) problem:

$$\min_{\alpha\in\mathcal{A},\theta\in\mathbb{R}} \theta \;\; s.t. \;\; \theta \geq \mathbb{G}(\alpha, \varrho), \forall \varrho \in \mathcal{E} \tag{7}$$

Problem (7) involves infinite number of constraints. Inspired by the cutting plane algorithm (Kelley, 1960), which iteratively refines a feasible constraint set and then solves a sequence of approximating linear programs, we propose Algorithm 1 to solve problem (7). Specifically, given α_{t-1}, the worst-case analysis is to infer the most-violated ϱ_t, and add it into the active constraint set $\mathcal{C}_t$. Then, we update α_t by solving the following problem:

$$\min_{\alpha\in\mathcal{A},\theta\in\mathbb{R}} \theta \;\; s.t. \;\; \theta \geq \mathbb{G}(\alpha, \varrho_h), \forall \varrho_h \in \mathcal{C}_t \tag{8}$$

The following convergence theorem indicates that Algorithm 1 gradually approaches to the optimal solution:

Theorem 8 *Algorithm 1 stops after a finite number of steps with the globally optimal solution of Problem (7).*

The proof can be adapted from Tan et al. (2014). Algorithm 1 involves two major steps: **worst-case analysis** and **master-problem optimization**, which will be discussed in the following subsections:

4.1.1 Worst-case analysis

The worst-case analysis is to solve the following maximization problem:

$$\max_{\varrho} \frac{1}{2}\Big\|\sum_{i=1}^{n} \alpha_i y_i (x_i \odot \sqrt{\varrho})\Big\|^2 s.t. \sum_{j=1}^{m} \varrho_j \leq B, \varrho \in [0,1]^m \tag{9}$$

Note that $\Big\|\sum_{i=1}^{n} \alpha_i y_i (x_i \odot \sqrt{\varrho})\Big\|^2 = \sum_{j=1}^{m} (\sum_{i=1}^{n} \alpha_i y_i x_{i,j})^2 \varrho_j$. Let $\Xi = (\sum_{i=1}^{n} \alpha_i y_i x_i)^2 \in \mathbb{R}^m$. Then, problem (9) can be transformed to $\max_{\varrho} \frac{1}{2}\sum_{j=1}^{m} \Xi_j \varrho_j$ subjected to $\sum_{j=1}^{m} \varrho_j \leq B, \varrho \in [0,1]^m$, which is a simple linear programming (LP) problem. We obtain the optimal solution for this LP problem by first sorting Ξ_j's and then setting the first B numbers corresponding to ϱ_j to one and the rests to zero. This procedure takes $\mathcal{O}(mlogB)$ time complexity, which is very efficient.

4.1.2 Master-problem optimization

Note that any $\varrho_h \in \mathcal{C}_t$ indexes a set of features. For simplicity, we define $X_h = [x_{1,h}, \cdots, x_{n,h}]' \in \mathbb{R}^{n\times B}$, where $x_{i,h}$ denotes the i-th instance with the features indexed by ϱ_h. By introducing dual variable $\mu_h \geq 0$ for each constraint defined by ϱ_h, the Lagrangian function of problem (8) can be written as: $\mathcal{L}(\theta, \alpha, \mu) = \theta + \sum_{\varrho_h\in\mathcal{C}_t} \mu_h(\mathbb{G}(\alpha, \varrho_h) - \theta)$. By setting its derivative with respect to θ to zero, we have $\sum \mu_h = 1$. By applying the minimax saddle-point theorem, problem (8) with the negative sign can be transformed to the following problem:

$$\max_{\alpha\in\mathcal{A}} \min_{\mu\in\mathcal{M}} -\sum_{\varrho_h\in\mathcal{C}_t} \mu_h \mathbb{G}(\alpha, \varrho_h) = \min_{\mu\in\mathcal{M}} \max_{\alpha\in\mathcal{A}} -\frac{1}{2}(\alpha \odot Y)'(\sum_{\varrho_h\in\mathcal{C}_t} \mu_h X_h X_h')(\alpha \odot Y) \tag{10}$$

where $\mathcal{M} = \{\mu | \sum \mu_h = 1, \mu_h \geq 0\}$.

Problem (10) is a multiple kernel learning (MKL) problem (Lanckriet et al., 2004b; Rakotomamonjy et al., 2008; Li et al., 2013), where the kernel matrix $\sum_{\varrho_h\in\mathcal{C}_t} \mu_h X_h X_h'$ is a convex combination of $|\mathcal{C}_t|$ base kernel matrices $X_h X_h'$. The resultant MKL problem can be solved efficiently by the modified accelerating proximal gradient (APG) method developed in (Tan et al., 2014), which takes $\mathcal{O}(n\varsigma B)$ time complexity, where ς is the number of the worst-case analyses. Thus, the time complexity of Algorithm 1 is $\mathcal{O}(n\varsigma B + mlogB)$, which is computationally efficient for large-scale and very high dimensional problems. Next sections will present what can we do with BAC.

5. Supervised Budgeted Tree

Due to the non-linear but highly interpretable representations, PDT has played a vital role in various data mining and machine learning applications. Unfortunately, PDT suffers from three limitations on ultrahigh dimensions which are described in introduction. To break the bottlenecks of PDT on ultrahigh dimensions, we first perform supervised learning based on our theoretical results and BAC, and construct a powerful non-linear classifier by arranging BAC in a tree-structure. In this section, we first present an algorithm for supervised budgeted tree (SBT), and propose pruning strategy to improve the generalization of SBT.

5.1 Algorithm for SBT

We build a SBT in a top-down approach. Starting from the root, we train the BAC on each node and use it to split the training data into two child nodes. This process continues until the remaining data at the node cannot be further split by the induced classifier. The details are stated in Algorithm 2.

Algorithm 2 Supervised Budgeted Tree (SBT)

Input: Training data $(x_1, y_1), \cdots, (x_n, y_n)$, where $x_i \in \mathbb{R}^m$ and $y_i \in \{\pm 1\}$.
Output: A supervised budgeted tree

while BAC contains more than one class **do**
  train the BAC on this node.
  split the training data of this node into left child if these instances are predicted to be positive by the BAC and call Algorithm 2 with split data as the input.
  split the rest of training data of this node into right child and call Algorithm 2 with the remaining data as the input.
end while
Return the complete supervised budgeted tree.

5.2 SBT Pruning

Theorem 5 reveals that the generalized bound also depends on the depth and leaves of the tree, which appear in the second and third terms of the bound. Inspired by this, we can reduce the depth and leaves of the tree by pruning the tree to further diminish the generalization error. We follow the pruning strategy in (Xu et al., 2013) to conduct pruning for SBT. To this end, we present to use the cross validation (CV) for each node to estimate the generalization error. If the sum of CV errors of child nodes is bigger than that of the parent node, we remove these child nodes.

It is obvious that any slight changes in the data distribution result in generating different feature subsets for many feature selection methods, such as (Xu et al., 2013, 2014). To identify a stable and unique feature subset across different folds of CV, we introduce the universal feature selection vector ϱ^{uni}, which is selected by the following formulation:

$$\min_{w^v, \varrho^{uni}} \sum_v BAC(w^v, \varrho^{uni}, S^v) \tag{11}$$

where S^v represents the $v-th$ fold of training data, w^v is the corresponding weight vector, and $BAC(w^v, \varrho^{uni}, S^v)$ denotes problem (4). To solve problem (11), we first conduct the worst-case analysis using the sum of objectives of each data fold. Then, we get the universal feature selection vector ϱ^{uni} and based on this, we adapt the efficient Master-problem solvers in (Tan et al., 2014) to get the w^v for each data fold separately. We repeat these procedures until convergence. This framework is called embedded cross validation. By using this method, we can find an intrinsic set of features.

For prediction, we aggregate the outputs from classifiers, which is generated from each data fold, by means of voting.

6. Sparse Coding Tree

In this section, based on BAC, we present sparse coding tree framework for multi-label annotation problem and the transformed multi-class (label powerset) annotation tasks.

Algorithm 3 Powerset Tree (PT) for Label Powerset Annotation

Input: Given ϖ transformed classes S, corresponding frequencies P, and $T_{s_i}(i = 1, \cdots, \varpi)$.
Output: Powerset Tree.

Initialize a priority queue $Q = \emptyset$ with the ascending order of the frequency.
Create a leaf node for each transformed class s_i and insert it to Q.
while $|Q| > 1$ **do**
  remove the two nodes s_i and s_j of the lowest frequency from Q.
  create a new internal node s_k with s_i and s_j as children and its frequency equals to the sum of the two nodes' frequency.
  set the instance T_{s_i} as positive sample and T_{s_j} as negative sample, then construct a BAC for the node s_k.
  insert the new node s_k to Q.
end while
The remaining node is the root node and the tree is complete.

6.1 Label Powerset Annotation

Our goal is to minimize the number of classifiers for binary decisions to enable rapid label powerset annotation for a number of testing instances. Recall that, to minimize the expected codeword length of information, Huffman coding uses shorter codewords to encode more frequent symbols and longer codewords to encode less frequent symbols. The idea of Huffman coding can be easily adapted to deal with the label powerset annotation problems.

Given a random n multi-labeled training sample, we first transform multi-label annotation to multi-class annotation by treating each unique set of labels as one of the classes for transformed multi-class learning task and record the frequency of each transformed class. Let ϖ be the number of transformed classes. A powerset tree is built based on the ϖ classes and their corresponding frequencies. Leaf nodes of this powerset tree are the set of transformed classes, denoted by $S = \{s_1, \cdots, s_\varpi\}$, and the corresponding frequencies are represented by $P = \{p(s_1), \cdots, p(s_\varpi)\}$. $T_{s_i}(i = 1, \cdots, \varpi)$ denotes the training instances associated with each transformed class. In each decision node, we construct a BAC and then build a powerset tree dealing with the label powerset case in a bottom-up manner. The details of the algorithm are presented in Algorithm 3.

More frequent classes are close to the root and less frequent classes are close to the leaf node layer. Thus, the annotation for most testing instances only require a few binary decisions along some shorter paths in our algorithm.

6.2 Multi-label Annotation

For multi-label annotation problems, the hypothesis space is exponential and we cannot observe some combinations of labels in the training data. Thus Huffman coding cannot be used to tackle the multi-label annotation problem. To reduce the number of annotations, we develop a novel annotation tree algorithm, which is analogous to the Shannon-Fano coding strategy. We first construct a BAC to predict the frequent label that have more examples. Then, we can leverage the annotations of such classifier to divide the remaining labels into two sets whose cardinality are roughly comparable. As long as any sets with

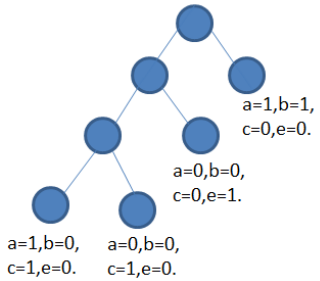


Figure 2: Schematic illustration of PT. We use a simple example to demonstrate our powerset tree. Assume one needs to annotate the presence or absence of four labels (a, b, c, e) for 8 input instances. Suppose $(1, 1, 0, 0)$, $(1, 0, 1, 0)$, $(0, 0, 0, 1)$ and $(0, 0, 1, 0)$ appear 3, 1, 3 and 1 times, respectively. Following Algorithm 3, we build the tree structure as shown in Figure 2.

more than one member remain, the same process is repeated on these sets. Our algorithm makes annotations for the ordered labels which accord to the arrangement with the label frequency ranging from high to low. The second observation shows that each instance has very few labels, based on our algorithm, so only a few annotations will be made for most testing instances. We start with some definitions associated with the specific nodes in a tree.

Let g represent the node in a tree, assume the label set associated with the node g is $G_g = \{1, 2, \cdots, q\}$, given a set of training instances $T_g = \{x_i\}_{i=1}^n$ and the corresponding labels $L_g = \{\mathbf{y}_i\}_{i=1}^n$.

Definition 9 *(Zero Label Set) refers to the situation where all the training instances at the node g do not belong to this label set. It defined as: $Zero_g = \{j|$ where all the training instances at the node g do not belong to label j $\}$.*

Definition 10 *(One Label Set) refers to the situation where all the training instances at the node g belong to this label set. It defined as: $One_g = \{j|$ where all the training instances at the node g belong to label j $\}$.*

To accelerate the learning process, these zero and one label sets will be thrown away. Thus we define the working label set for node g: $W_g = G_g - Zero_g - One_g$. Now, we introduce the learning process for the tree in three cases.

1. **Root node:** Assume g is the root node. We select label $l_g \in W_g$ with the highest frequency among W_g and build a BAC for l_g. g's left and right child are denoted as g_1 and g_0, respectively. The training instances of node g will be partitioned into left child g_1 if the values of label l_g of these instances is 1, otherwise the instances will be partitioned into right child g_0. The label set associated with node g_1 and g_0 will be reduced as $W_g \backslash l_g$. The training instance sets for g_1 and g_0 are denoted by T_{g_1} and T_{g_0}.

2. **Internal node:** The training procedure for internal (child) nodes are the same with that for the root node but uses a subset of training instances. The training instances are split recursively into two smaller subsets until our algorithm moves down to the leaf nodes.

3. **Leaf node:** If the working set of the node is empty, then it is a leaf node and stop training for the leaf nodes.

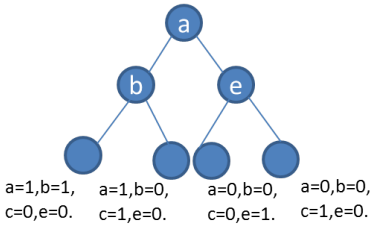


Figure 3: Schematic illustration of AT. We use the same example, which is aforementioned in Figure 2, to illustrate our annotation tree. Following Algorithm 4, we build a BAC for label a in the root node. For internal nodes, we build two BAC for label b and e in the left and right child nodes of the root, respectively. The working set in leaf nodes is empty, so we do not train any classifiers for the leaf nodes.

We predict the multi-label output for a new testing instance from the root to leaves and assign the labels of the leaf node to this testing instance. The schematic illustration of our tree is shown in Figure 3 and the details of algorithm is presented as follows:

6.3 Testing Time Complexity Analysis and Generalization Error Bound

In this section, we study the testing time complexity and generalization error bound for both label powerset and multi-label annotations. We denote logarithms to base 2 by log. Let $\mathbf{T}$ and $\mathbb{T}$ represent PT and AT, respectively.

6.3.1 Testing Time Complexity Analysis

As the annotation efficiency critically depends on the number of annotation times, we first study the average number of annotations for the both cases.

We first introduce the noiseless coding theorem in (Roman, 1992). Let $C = \{c_1, \cdots, c_n\}$ be the symbol set of size n. $p(c_i)$ is the probability that symbol c_i occurs. After encoding of symbols in C, $l(c_i)$ denotes the length of code c_i and the average codeword length is defined as $AveLen(c_1, \cdots, c_n) = \sum_{i=1}^n p(c_i) l(c_i)$. The entropy of $\{p(c_1), \cdots, p(c_n)\}$ is de-

Algorithm 4 Annotation Tree (AT) for Multi-label Annotation

Input: Given node g and its corresponding G_g, T_g and L_g.
Output: Annotation Tree

Compute $Zero_g$, One_g and W_g
while $W_g \neq \emptyset$ **do**
  select label $l_g \in W_g$ with the highest frequency
  build a BAC for l_g
  set the label set of left and right child as $W_g \backslash l_g$
  split the training instances of node g into left child g_1 if the values of label l_g of these instances are 1 and call Algorithm 4 with input ($W_g \backslash l_g$, T_{g_1})
  split the rest of training instances of node g into right child g_0 and call Algorithm 4 with input ($W_g \backslash l_g$, T_{g_0})
end while
The tree is complete

fined as $H(p(c_1), \cdots, p(c_n)) = \sum_{i=1}^{n} p(c_i) \log(1/p(c_i))$. We get the following noiseless coding theorem:

Theorem 11 *(The Noiseless Coding Theorem) Given $\{p(c_1), \cdots, p(c_n)\}$, we have*

$$H(p(c_1), \cdots, p(c_n)) \leq AveLen(c_1, \cdots, c_n) < H(p(c_1), \cdots, p(c_n)) + 1$$

Label powerset annotation: We define $H(P_{lp}) = \sum_{i=1}^{\varpi} p(s_i) \log(1/p(s_i))$. $l(s_i)$ denotes the number of annotations for class s_i and the average number of annotations is defined as $mean(S) = \sum_{i=1}^{\varpi} p(s_i) l(s_i)$. Following Theorem 11, we obtain the bound of the average number of annotations for the label powerset annotation problem using Algorithm 3.

Theorem 12 *Assume we transform a random n multi-labeled sample to a multi-class sample which has ϖ classes. We obtain the bound of the average number of annotations for the transformed multi-class annotation problem based on* **T***:*

$$H(P_{lp}) \leq mean(S) < H(P_{lp}) + 1$$

Multi-label annotation: Assume there are Φ leaf nodes in a AT. These leaf nodes denoted by $S' = \{s'_1, \cdots, s'_\Phi\}$. Assume the frequency p'_i of each unique label set can be associated with leaf node s'_i. We define $H(P_l) = \sum_{i=1}^{\Phi} p(s'_i) \log(1/p(s'_i))$. Similar to label powerset annotation problem, we use $mean(S')$ to represent the average number of annotations of Algorithm 4. We obtain the bound of $mean(S')$ for multi-label annotation.

Theorem 13 *Given a random n multi-labeled sample which has Φ unique label set. We obtain the bound of the average number of annotations for multi-label annotation problem based on $\mathbb{T}$:*

$$H(P_l) \leq mean(S') < H(P_l) + 1$$

Table 2: Testing time complexity comparison between the main methods used in this paper. $\Upsilon, \Psi, \iota, \mathfrak{D}$: # clusters, the average # instances in each cluster, # learners and the dimension of the embedding space used in SLEEC. κ_w denotes the average number of non-zero weights of PD-Sparse.

Method	**Worst Testing Time**	**Average Testing Time**
BR	$\mathcal{O}(mq)$	$\mathcal{O}(mq)$
LP	$\mathcal{O}(m\vartheta)$	-
Homer	-	$\mathcal{O}(m \log_k(q))$
FastXML	-	$\mathcal{O}(\mathcal{T} m \log(q))$
SLEEC	-	$\mathcal{O}(\mathfrak{D}\Upsilon\iota + \mathfrak{D}\Psi\iota + m\mathfrak{D})$
PD-Sparse	-	$\mathcal{O}(\kappa_w q)$
PT	$\mathcal{O}(\varsigma B \log(\vartheta))$	$\mathcal{O}(\varsigma B H(P_{lp}))$
AT	$\mathcal{O}(\varsigma B q)$	$\mathcal{O}(\varsigma B H(P_l))$

Proof Assume there are n multi-labeled samples which have Φ unique label set. We build a hierarchical tree based on n sample according to Algorithm 4. Each node is associated with one label, so the labels associated with nodes in each path of the tree are one of the unique label set, so there are Φ number of paths and leaf nodes. The frequency of each unique label set can be associated with each leaf node. After this transformation, we apply Theorem 12 to obtain the above bounds. ■

Therefore, in the label powerset and multi-label annotation algorithms, we can expect that the number of annotations for each instance are around $H(P_{lp})$ and $H(P_l)$, which are much fewer than $\min(n, 2^q)$ and q, respectively.

Algorithm 4 and Algorithm 3 are denoted as AT and PT, respectively. We provide the testing time complexity analysis for each testing instance. The number of annotations for BR are equal to the number of labels. Each time Homer divides the problem into k sub-problems. Following (Tsoumakas et al., 2008), it takes $\mathcal{O}(m \log_k(q))$ on testing time. According to (Prabhu and Varma, 2014), the average cost of prediction for FastXML is $\mathcal{O}(\mathcal{T} m \log(q))$, where $\mathcal{T}$ is the number of trees in the FastXML ensemble. In the worst case, LP requires $\min(n, 2^q)$ times of annotations respectively. The number of annotations for PT and AT are $\log(\min(n, 2^q))$ and q, respectively. As shown in our theoretical analysis, on average, the number of annotations can be further reduced to $H(P_{lp})$ and $H(P_l)$. The testing time complexity for the main methods used in this paper is presented in Table 2, where $\vartheta = \min(n, 2^q)$. From Table 2, we can see that the average testing time cost of our proposed PT and AT is lower than other methods.

6.3.2 Generalization Error Bound

Label powerset annotation: This subsection aims to bound the generalization error for a specific transformed class k in terms of the nodes in which instances in class k are

used to train the classifiers. We define the probability of error for transformed class k as $\epsilon_k(\mathbf{T}) = P\{x : x \in \mathbb{R}^m, x \text{ in class } k \text{ and } x \text{ is misclassified by } \mathbf{T} \text{ or } x \text{ is misclassified as class } k \text{ by } \mathbf{T}\}$. We first state some important definition, lemma and theorem.

We begin with the definition of the fat shattering dimension.

Definition 14 (Kearns and Schapire (1990)) *Let $\mathcal{H}$ be a set of real valued functions. We say that a set of points $\mathfrak{P}$ is γ-shattered by $\mathcal{H}$ relative to $r = (r_p)_{p\in\mathfrak{P}}$ if there are real numbers r_p indexed by $p \in \mathfrak{P}$ such that for all binary vectors b indexed by $\mathfrak{P}$, there is a function $f_b \in \mathcal{H}$ satisfying*

$$f_b(p) = \begin{cases} \geq r_p + \gamma & \text{if } b_p = 1 \\ \leq r_p - \gamma & \text{otherwise} \end{cases}$$

The fat shattering dimension $fat(\gamma)$ of the set $\mathcal{H}$ is a function from the positive real numbers to the integers which maps a value γ to the size of the largest γ-shattered set, if this is finite, or infinity otherwise.

Then, we provide the following lemma and theorem:

Lemma 15 *Given a specified $\mathbf{T}$ model with K decision nodes with margins $\gamma^1, \gamma^2, \cdots, \gamma^K$ at the decision nodes satisfying $k_i = fat(\gamma^i/8)$, where fat is continuous from the right. If $\mathbf{T}$ has correctly classified n multi-labeled examples $\boldsymbol{s}$ generated independently according to the unknown (but fixed) distribution D and $\bar{\boldsymbol{s}}$ is a set of another n multi-labeled examples, then we can bound the following probability to be less than δ: $P\{\boldsymbol{s}\bar{\boldsymbol{s}} : \exists\, \mathbf{T}$, it correctly classifies $\boldsymbol{s}$, fraction of $\bar{\boldsymbol{s}}$ misclassified $> \epsilon(n, K, \delta)\} < \delta$, where $\epsilon(n, K, \delta) = \frac{1}{n}(\mathfrak{Q}\log(8n) + \log\frac{2^K}{\delta})$ and $\mathfrak{Q} = \sum_{i=1}^{K} k_i \log(\frac{4en}{k_i})$.*

The proof can be adapted from Shawe-Taylor and Cristianini (1997).

Theorem 16 (Bartlett and Shawe-Taylor (1998)) *Let $\mathcal{H}$ be restricted to points in a ball of $\mathbf{M}$ dimensions of radius R about the origin, then*

$$fat_{\mathcal{H}}(\gamma) \leq \min\left\{\frac{R^2}{\gamma^2}, \mathbf{M} + 1\right\} \tag{12}$$

Based on Lemma 15 and Theorem 16, we derive the following theorem:

Theorem 17 *Assume we transform a random n multi-labeled sample to multi-class sample which has ϖ classes. Suppose transformed class k can be correctly classified by a $\mathbf{T}$, where the internal nodes are K decision nodes with margins γ^i at node i. In addition, suppose the number of nodes in which instances in class k are used to train the classifiers is $\mathbf{N}(k)$. Then with probability greater than $1 - \delta$,*

$$\epsilon_k(\mathbf{T}) \leq \frac{130R^2}{n}(D'_k \log(4en)\log(4n) + \mathbf{N}(k)\log(2n) + \log\frac{2}{\delta})$$

where $D'_k = \sum_{i\in k-nodes} \frac{1}{(\gamma^i)^2}$ (k-nodes are the set of nodes in which instances in class k are used to train the classifiers) and R is the radius of a ball containing the distributions support.

The proof can be adapted from Platt et al. (1999).

We discuss the implications of Theorem 17. Theorem 17 reveals one important factors to reduce generalization error bound for a specific transformed class k: it is to control the complexity of the model by reducing the average number of decision nodes. We build a $\mathbf{T}$ for the transformed multi-class classification problem. So, we obtain shorter path for higher frequency classes than that for less frequent classes. Based on Theorem 12, we can see that the average number of annotations of Algorithm 3 is around $H(P_{lp})$. Thus, PT is able to reduce the generalization error.

Multi-label annotation: This subsection analyzes the generalization error bound of multi-label annotation problem using Algorithm 4. Let $Path_{k-nodes}$ be the set of decision nodes along k-th path in $\mathbb{T}$. Assume $\epsilon_k(\mathbb{T})$ is the probability that any of node in $Path_{k-nodes}$ makes mistake. We provide the following generalization error bound for $\epsilon_k(\mathbb{T})$:

Theorem 18 *Given random n multi-labeled sample, suppose there is no mistake made by any node in $Path_{k-nodes}$ of $\mathbb{T}$, where the internal nodes are K decision nodes with margins γ^i at node i. Denote $\mathbb{N}(k) = |Path_{k-nodes}|$, that is the number of decision nodes along k-th path in $\mathbb{T}$. Then with probability greater than $1 - \delta$,*

$$\epsilon_k(\mathbb{T}) \leq \frac{130R^2}{n}(D'_k \log(4en)\log(4n) + \mathbb{N}(k)\log(2n) + \log\frac{2}{\delta})$$

where $D'_k = \sum_{i\in Path_{k-nodes}} \frac{1}{(\gamma^i)^2}$ and R is the radius of a ball containing the distributions support.

The proof can be adapted from Bennett et al. (2000).

We discuss the implications of Theorem 18. Theorem 18 reveals one important factors to reduce generalization error bound for some decision nodes along a specific path in $\mathbb{T}$: it is to control the complexity of the model by reducing the average number of decision nodes. Based on Theorem 13, we can see that the average number of annotations of Algorithm 4 is around $H(P_l)$. Thus, our proposed AT is able to tighten the generalization error bound.

7. Practical Issues

Accessing features in sparse format is very expensive for high dimensional data sets. This paper proposes two efficient auxiliary data structures to reduce the computation cost of indexing and accessing features. We first propose a modified direct indexing method for small and medium-sized data sets.

Let ε be the selected feature set of the tree and $\mathcal{Q}_{|\varepsilon|}$ represent the natural number set which contains $|\varepsilon|$ numbers from 0 to $|\varepsilon|-1$. Then, we build an injection table: $\mathcal{I} : \varepsilon \mapsto \mathcal{Q}_{|\varepsilon|}$ to compress the whole selected feature set to a continuous index set. To support direct indexing for every instance, we maintain an array with size $|\varepsilon|$. For each feature that belongs to ε, we use $\mathcal{I}$ to find out the index number, and store the feature value at that index in the array.

To handle very high dimensional data sets, in which the data is usually stored in sparse format, we maintain a hash table to store the feature ID as the key and the feature value for each instance. The method takes $\mathcal{O}(1)$ time to access the features.

Table 3: Data sets used for SBT

Data Set	# Features	# Training	# Testing
colon	2,000	32	30
pcmac	3,289	1,555	388
gisette	5,000	6,000	1,000
epsilon	2,000	21,767	14,511
rcv	47,236	12,146	8,096
news20	1,355,191	15,997	3,999
webspam	16,609,143	28,000	7,000

8. Experiment

In this section, we conduct some experiments to evaluate the performance of our proposed methods.

8.1 Evaluation on SBT

In this subsection, we conduct comprehensive experiments to evaluate the proposed method on both synthetic and high dimensional real-world data sets. Most data sets are collected from LIBSVM website[2]. pcmac data set is from (Xu et al., 2014). The training/testing partition is either predefined or the data is randomly split into 80% training and 20% testing. The statistics of these data sets are described in Table 3.

We compare SBT with a number of baseline methods: standard SVM (l_2-SVM)[3], CART-LC (Breiman et al., 1984), OC1 (Murthy et al., 1994), LDKL (Jose et al., 2013), GBFS (Xu et al., 2014), CSTC (Xu et al., 2013) and FGM (Tan et al., 2014). We use the linear classification/regression package LIBLINEAR (Fan et al., 2008) with L2-regularized square hinge loss (primal) to implement standard SVM. LDKL uses composite non-linear kernels, and achieves significantly better classification accuracies as compared to localized multiple kernel learning (Gönen and Alpaydin, 2008) (which is competitive with RBF-SVM) and other state-of-the-art non-linear methods. Therefore, we do not compare with other non-linear SVMs. The codes of other baseline methods are provided by their authors.

We use 5-fold cross validation to prune SBT. Following the parameter settings in (Tan et al., 2014), B is chosen in a range of $\{2, 5, 10, 20, 50, 100, 150, 200, 250, 400\}$ for rcv, news20 and webspam data sets, and $\{0.01m, 0.02m, \cdots, 0.09m\}$ for other data sets. C is selected using 5-fold cross validation over the range $\{0.001, 0.01, 0.1, 5, 10\}$ for the first three data sets and we fix $C = 5$ for other data sets. Following the settings in (Oiwa and Fujimaki, 2014), the tree-depth is fixed to 3 in LDKL.

8.1.1 Experiments on Synthetic Data

We compare the performance of different methods on the XOR synthetic data with different numbers of noisy features, which is non-linearly separable. Following the strategy in (Tan et al., 2014), we generate the synthetic data as follows: At first, we generate 10,000 training instances, 2,000 testing instances and 200 features. The first two dimensions are informative features, which are independently sampled from Gaussian distribution $\mathcal{N}(0, 1)$. The remaining dimension are noisy features, which are independently sampled from Uniform distribution $\mathcal{U}(0, 1)$. The output y is decided by these two informative features. For instance, if the first two features have the same sign, then y is 1 and -1 otherwise. Then, we gradually increase the noisy features to the data set and test whether the considered methods can successfully identify the former two informative features and have stable and accurate classification performance. B is chosen in a range of $\{1, 3, 5, 10\}$ using cross validation for SBT. Figure 4 (left panel) shows two informative features.

2. https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/
3. https://www.csie.ntu.edu.tw/∼cjlin/liblinear/

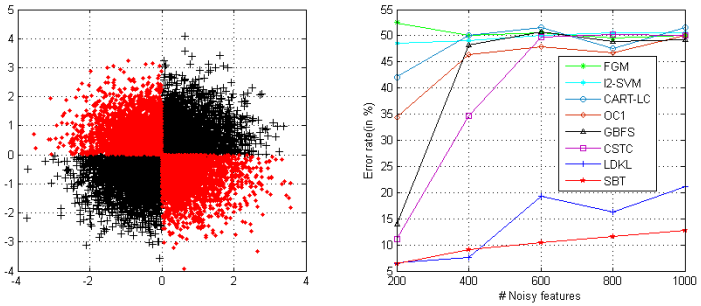


Figure 4: Testing error rates (in %) on the XOR synthetic data with different # noisy features.

Figure 4 (right panel) shows that: 1) l_2-SVM cannot work on the XOR data set. 2) FGM has shown impressive results in (Tan et al., 2014), but it cannot work on the XOR data set as well. 3) When the number of noisy features equals 200, CART-LC and OC1 are a bit better than l_2-SVM; GBFS and CSTC are much better than other baselines. However, CART-LC, OC1, GBFS and CSTC are sensitive to the number of noisy features. 4) LDKL and SBT are most successful, while our model generates more stable and accurate results.

8.1.2 Pruning Performance

To evaluate the pruning performance, this subsection compares the classification performance of our method with pruning and without pruning on epsilon and rcv data sets. The results are shown in Table 4. From this table, we can see that our proposed pruning strategy significantly improves the generalization performance, which corroborates our theoretical findings.

Table 4: Testing error rates (in %) for our method with pruning and without pruning. The best results are in bold.

Data set	SBT without pruning	SBT with pruning
epsilon	17.23	**12.45**
rcv	6.19	**3.85**

Table 5: Testing error rates (in %) for different methods. Numbers in parentheses indicate the depth of tree. The best results are in bold.

Data set	l_2-SVM	CART-LC	OC1	LDKL	SBT
colon	20.00	40.91(2)	33.33(2)	16.67	**13.33**(1)
pcmac	7.47	16.22(2)	9.68(6)	7.99	**7.22**(2)
gisette	2.30	48.17(5)	18.83(6)	2.80	**2.20**(2)
epsilon	13.24	49.40(2)	47.66(4)	15.36	**12.45**(8)
rcv	4.05	17.13(6)	48.68(3)	3.90	**3.85**(5)

8.1.3 Classification Performance

To verify the effectiveness of conducting feature selection in the decision tree, we evaluate the classification performance of SBT on real-world data sets compared with l_2-SVM, CART-LC, OC1 and LDKL. The results are shown in Table 5.

The results of Table 5 show that: 1) CART-LC and OC1 generally underperform on all data sets. The results support the argument of this paper: DT models usually suffer from the curse of dimensionality and achieve degenerated performance on high dimensional data sets, which boosts our approach. 2) SBT gets a lower error rate than l_2-SVM and LDKL. It is important to generate different hyperplanes with maximum-margin based on different sets of features. Thus, the results verify the effectiveness of conducting feature selection in the DT. 3) SBT has the best performance compared with baselines, while generating a small tree.

8.1.4 Feature Selection Performance

We evaluate the feature selection performance of SBT on pcmac, gisette, epsilon and rcv data sets compared with sparse classifier models, such as GBFS, CSTC and FGM. CSTC runs out of memory on rcv. The results of Figure 5 show that: 1) FGM and SBT clearly outperform GBFS and CSTC in terms of feature selection performance, which verifies that the sparse models with an explicit budget constraint on the weights are more effective than l_1-SVM based approaches. 2) SBT outperforms FGM. Thus, it is imperative to use the tree-structure to conduct feature selection.

Table 6: Training time (in second) for GBFS and our method on rcv data set. The faster results are in bold.

# Selected features	GBFS	SBT
100	874.2s	**5.9**s
500	5214.8s	**12.0**s
900	8523.3s	**41.9**s

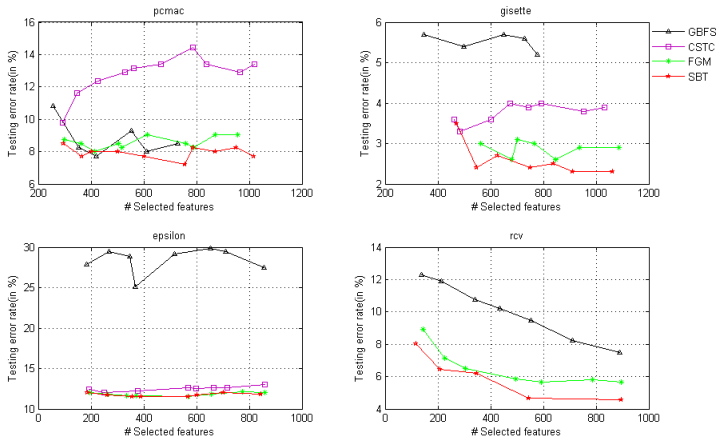


Figure 5: Testing error rates (in %) vs. # selected features for different methods on pcmac, gisette, epsilon and rcv data sets.

We next use rcv data set for training time comparison. The results are shown in Table 6. Since CART-LC, OC1 and CSTC are very slow on this data set, we do not report their training time here. From Table 6, we can see that, the training of our method is much faster than that of GBFS, and it is also much faster than LDKL, which takes 8019.3 seconds(s).

8.1.5 Results on Very High Dimensional Data Sets

We evaluate our method on news20 and webspam data sets with millions of dimensions. As CART-LC, OC1, LDKL, GBFS and CSTC are very slow on these data sets, we compare our method with l_2-SVM and FGM only on these data sets. The results are reported in

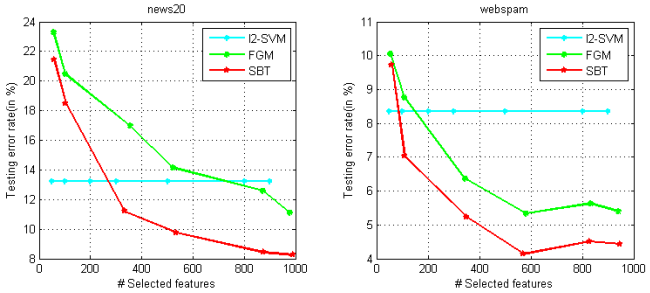


Figure 6: Testing error rates (in %) vs. # selected features for l_2-SVM, FGM and our method on news20 and webspam data sets. l_2-SVM uses all features.

Figure 6. From Figure 6, we can see that SBT consistently outperforms FGM and l_2-SVM on very high dimensional data sets.

The representation of SBT ($B = 2$) on webspam data set with 16,609,143 features is shown in Figure 7. In each decision node of Figure 7, we can see that our BAC can identify more informative and compact features. Thus, our SBT is easy to understand and interpret in ultrahigh dimensions.

8.2 Evaluation on PT and AT

In this subsection, we evaluate the performance of our proposed algorithms for multi-label annotation on a variety of real-world data sets from different application domains, which are collected from website[4][5]. The data sets fall into two categories. The first category contains five small and medium-sized data sets. The second category contains three large-scale data sets with very high label and feature dimensions. The details of data sets are shown in Table 7.

We compare our algorithms with several state-of-the-art multi-label annotation methods: 1) Flat methods: BR and LP; 2) Tree-based methods: Homer and FastXML; 3) Embedding-based method: Sparse local embedding for extreme classification (SLEEC) (Bhatia et al., 2015) is one of the most advanced embedding based method. 4) Sparse model: PD-Sparse (Yen et al., 2016) is a very recently proposed multiclass and multi-label approach which uses L1 regularization. The codes are provided by the respective authors. According to (Tsoumakas et al., 2008), k is chosen in a range of $\{2, 3, \cdots, 8\}$ using cross validation for Homer. Following (Prabhu and Varma, 2014), $\mathcal{T}$ is fixed to 50 for FastXML. We use the linear classification/regression package LIBLINEAR (Fan et al., 2008) with L2-regularized square hinge loss (primal) to train the classifier for both BR and LP. We use the default parameter in LIBLINEAR. According to the original setting in (Bhatia et al., 2015), we set

4. http://mulan.sourceforge.net
5. http://manikvarma.org/#Julian15a

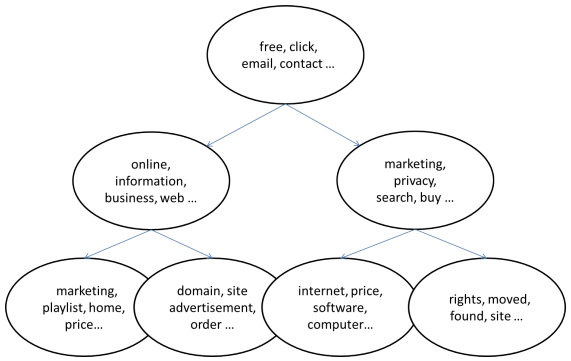


Figure 7: Representation of SBT on webspam data set.

Table 7: Data sets used for PT and AT

Data Set	# Labels	# Features	# Instances	Domain
mediamill	101	120	43,907	video
cal500	174	68	502	music
corel5k	374	499	5,000	image
Eur-Lex (dc)	412	5,000	19,348	text
Eur-Lex (ed)	3,993	5,000	19,348	text
wiki	30,938	101,938	20,762	web
delicious	205,443	782,585	26,701	web
amazon	2,812,281	337,067	10,482	advertisement

the number of the clusters as $\lfloor n/6000 \rfloor$, the number of learners as 15 and the dimension of the embedding space as 50 for small and medium-sized data sets, and 100 for very high dimensional data sets. We set B according to parameter settings in Section 8.1.

To measure the annotation performance of our methods and baseline methods fairly, this paper considers the standard F1 evaluation measurement (Du et al., 2017), which computes the F1 score for all the labels of each testing sample and takes the average over these samples. We use the publicly available split of the training and testing sets for very high dimensional data sets and perform 10-fold cross-validations on small and medium-sized data sets and report the mean and standard error of the F1 measure.

8.2.1 Results on Small and Medium-Sized Data Sets

This subsection studies the performance of various methods on mediamill, cal500, corel5k, Eur-Lex (dc) and Eur-Lex (ed) data sets. In our experiment, we cannot get the results of LP and Homer on Eur-Lex (ed) data set within one week. The F1 results of various methods on small and medium-sized data sets are listed in Table 8. From this table, we observe that: 1) Tree-based methods, such as Homer and FastXML, generally underperforms on all data sets.

Table 8: F1 Results of various methods on small and medium-sized data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" indicates that we cannot get the results of baselines within one week.

Data Set	BR	LP	Homer	FastXML	SLEEC	PD-Sparse	PT	AT
mediamill	0.414±0.00[3]	0.394±0.01[8]	0.411±0.00[4]	0.408±0.00[7]	0.411±0.00[4]	0.409±0.00[6]	0.422±0.01[2]	**0.462**±0.00[1]
cal500	0.317±0.02[6]	0.345±0.02[2]	0.321±0.01[5]	0.259±0.01[8]	0.310±0.03[7]	0.322±0.01[4]	0.324±0.02[3]	**0.362**±0.01[1]
corel5k	0.100±0.02[6]	0.138±0.01[2]	0.113±0.01[4]	0.056±0.01[8]	0.084±0.01[7]	0.110±0.02[5]	0.123±0.01[3]	**0.141**±0.01[1]
EUR-Lex(dc)	0.713±0.01[5]	0.659±0.01[6]	0.625±0.01[7]	0.516±0.01[8]	0.716±0.01[3]	0.718±0.01[2]	0.715±0.01[4]	**0.727**±0.01[1]
EUR-Lex(ed)	0.268±0.01[5]	-	-	0.254±0.01[6]	0.281±0.01[3]	0.302±0.01[2]	0.274±0.01[4]	**0.341**±0.01[1]
Ave. Rank	5.0	4.5	5.0	7.4	4.8	3.8	3.2	1.0

Table 9: F1 results of various methods on very high dimensional data sets. The best results are in bold.

Data Set	FastXML	SLEEC	PD-Sparse	PT	AT
wiki	0.1272	0.1364	0.1395	0.1422	**0.1657**
delicious	0.0077	0.0095	0.0086	0.0111	**0.0157**
amazon	0.0023	0.0148	0.0151	0.0166	**0.0348**

2) SLEEC, which is one of the most advanced embedding method, generally underperforms on small data sets, while it obtains competitive results on medium-sized data sets, such as Eur-Lex (dc) and Eur-Lex (ed). 3) PT and AT are two most successful methods, which significantly outperform tree-based algorithms, such as Homer and FastXML. The results verify our theoretical analysis.

Figure 8 shows the testing time of various methods spent on all testing instances per data set. From Figure 8.(a), we observe that: 1) SLEEC is slower than other baselines on all data sets. 2) Tree-based method, such as FastXML, is faster than BR, LP and SLEEC on medium-sized data sets. 3) PT and AT are faster than other baselines on all small and medium-sized data sets. For example, PT obtains around sixty times speedup over BR and LP, and about ten times speedup over FastXML on medium-sized data sets, such as Eur-Lex (dc). The results verify our testing time complexity analysis.

We next use Eur-Lex (ed) data set for training time comparison. BR, FastXML, SLEEC and PD-Sparse take 37652.1s, 269.9s, 5230.8s and 451.3s, respectively, while PT and AT take 839.8s and 1088.6s, respectively. Because BR and SLEEC require an expensive training process for each label and time-consuming embedding procedure, respectively, our proposed tree-based algorithms are able to achieve faster training than BR and SLEEC.

8.2.2 Results on Very High Dimensional Data Sets

In this subsection, we evaluate the performance of various methods on three very high dimensional data sets: wiki, delicious and amazon. In our experiment, we cannot get the

Table 10: F1@5 results of FastXML, SLEEC, PT and AT on wiki, delicious and amazon data sets. The best results are in bold.

Data Set	FastXML	SLEEC	PT	AT
wiki	0.5487	0.6067	0.6211	**0.6284**
delicious	0.4442	0.4631	0.4695	**0.4711**
amazon	0.4906	0.5077	0.5143	**0.5296**

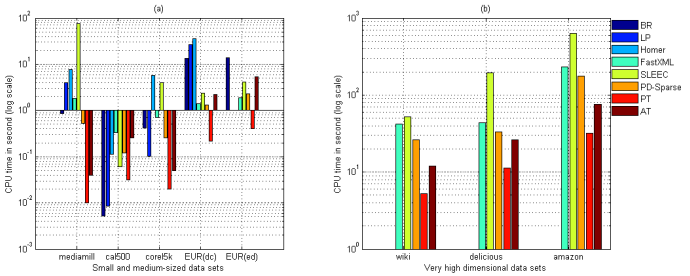


Figure 8: Testing time of various methods on all data sets (EUR-Lex is abbreviated to EUR).

results of BR, LP and Homer on these data sets within one week. The F1 results of various methods on three very high dimensional data sets are listed in Table 9. From Table 9 and Figure 8.(b), we can see that: 1) SLEEC outperforms FastXML, while FastXML is faster than SLEEC. 2) PD-Sparse is faster and more accurate than SLEEC and FastXML, which is consistent with the empirical studies in (Yen et al., 2016). 3) PT and AT are several times faster than FastXML, SLEEC and PD-Sparse. Moreover, PT and AT are much more accurate than FastXML, SLEEC and PD-Sparse on the data sets with millions of labels and features. Finally, we conclude that our proposed PT and AT are effective and efficient in ultrahigh label and feature dimensions.

Using Precision@5 and Recall@5, Table 10 shows the F1@5 results of FastXML, SLEEC, PT and AT on wiki, delicious and amazon data sets. From this table, we can see that our methods also outperform FastXML and SLEEC in terms of F1@5 measurement.

9. Conclusion

In this paper, we first develop a novel data-dependent generalization error bound for the PDT, where the data dependency of the bound comes from the training error loss, the

margin, the capacity of the kernel matrix defined on the training data, and the complexity of the tree. It provides the theoretical justification to learn a sparse linear hyperplane in each decision node and to prune the SBT. Our analysis reveals a new insight for the design of decision tree algorithms. Based on BAC and three important phenomena of real-world data sets from a variety of application domains, we develop the sparse coding tree framework for multi-label annotation with ultrahigh label and feature dimensions and provide generalized performance guarantee for our proposed PT and AT. Compared with state-of-the-art baselines, the results on the synthetic data set show that SBT is more resilient to noisy features, and empirical studies on real-world data sets demonstrate that SBT is easier to understand and interpret in ultrahigh dimensions and can identify more informative features than state-of-the-art feature selection methods. Moreover, our proposed PT and AT algorithms are able to annotate as well as possible, without performing unnecessary annotations in ultrahigh label and feature dimensions.

Acknowledgments

We would like to thank the action editor and reviewers for their valuable comments and constructive suggestions on our paper. This project is supported by the ARC Future Fellowship FT130100746, ARC grant LP150100671, DP170101628, DP150102728 and DP150103071.

References

Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for web pages. In *WWW*, pages 13–24, May 2013.

Francis R. Bach, Gert R. G. Lanckriet, and Michael I. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *ICML*, page 6, 2004.

A.-L. Barabási, R. Albert, H. Jeong, and G. Bianconi. Power-law distribution of the world wide web. *Science*, 287(54561):2115, 2000.

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3:463–482, 2002.

Peter L. Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In *Advances in Kernel Methods - Support Vector Learning*, pages 43–54. MIT Press, Cambridge, MA, USA, 1998.

Kristin P. Bennett, Nello Cristianini, John Shawe-Taylor, and Donghui Wu. Enlarging the margins in perceptron decision trees. *Machine Learning*, 41(3):295–313, 2000.

Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *NIPS*, pages 730–738, 2015.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. CA: Wadsworth International Group, 1984.

Serhat Selcuk Bucak, Rong Jin, and Anil K. Jain. Multiple kernel learning for visual object recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(7):1354–1369, 2014.

Yao-Nan Chen and Hsuan-Tien Lin. Feature-aware label space dimension reduction for multi-label classification. In *NIPS*, pages 1538–1546, 2012.

Marcin Czajkowski, Marek Grzes, and Marek Kretowski. Multi-test decision tree and its application to microarray data classification. *Artificial Intelligence in Medicine*, 61(1): 35–44, 2014.

Jia Deng, Olga Russakovsky, Jonathan Krause, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Scalable multi-label annotation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3099–3102, 2014.

Bo Du, Mengfei Zhang, Lefei Zhang, Ruimin Hu, and Dacheng Tao. PLTD: patch-based low-rank tensor decomposition for hyperspectral images. *IEEE Trans. Multimedia*, 19(1): 67–79, 2017.

Jianqing Fan, Richard Samworth, and Yichao Wu. Ultrahigh dimensional feature selection: Beyond the linear model. *JMLR*, 10:2013–2038, 2009.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Lin Chih-Jen. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

Mehmet Gönen and Ethem Alpaydin. Localized multiple kernel learning. In *ICML*, pages 352–359, 2008.

Chen Gong, Dacheng Tao, Wei Liu, Liu Liu, and Jie Yang. Label propagation via teaching-to-learn and learning-to-teach. *IEEE Trans. Neural Netw. Learning Syst.*, 28(6):1452–1465, 2017.

Matthieu Guillaumin, Thomas Mensink, Jakob Verbeek, and Cordelia Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*, pages 309–316, 2009.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. New York: Springer New York Inc, 2001.

David A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the Institute of Radio Engineers*, 40(9):1098–1101, 1952.

Cijo Jose, Prasoon Goyal, Parv Aggrwal, and Manik Varma. Local deep kernel learning for efficient non-linear svm prediction. In *ICML*, pages 486–494, 2013.

Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts. In *Proceedings of the 31st Symposium on the Foundations of Computer Science*, pages 382–391, 1990.

J. E. Kelley. The cutting plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Trans. Information Theory*, 47(5):1902–1914, 2001.

Oluwasanmi Koyejo, Nagarajan Natarajan, Pradeep Ravikumar, and Inderjit S. Dhillon. Consistent multilabel classification. In *NIPS*, pages 3321–3329, 2015.

Gert R. G. Lanckriet, Tijl De Bie, Nello Cristianini, Michael I. Jordan, and William Stafford Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004a.

Gert R. G. Lanckriet, Nello Cristianini, Peter L. Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004b.

Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient l1 regularized logistic regression. In *The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference*, pages 401–403, 2006.

Yu-Feng Li, Ivor W. Tsang, James T. Kwok, and Zhi-Hua Zhou. Convex and scalable weakly labeled svms. *Journal of Machine Learning Research*, 14(1):2151–2188, 2013.

Weiwei Liu and Ivor W. Tsang. Large margin metric learning for multi-label prediction. In *AAAI*, pages 2800–2806, 2015a.

Weiwei Liu and Ivor W. Tsang. On the optimality of classifier chain for multi-label classification. In *NIPS*, pages 712–720, 2015b.

Weiwei Liu and Ivor W. Tsang. Sparse perceptron decision tree for millions of dimensions. In *AAAI*, pages 1881–1887, 2016.

Xinwang Liu, Lei Wang, Jianping Yin, Yong Dou, and Jian Zhang. Absent multiple kernel learning. In *AAAI*, pages 2807–2813, 2015.

Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Saso Dzeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9): 3084–3104, Sep 2012.

Qi Mao, Ivor Wai-Hung Tsang, and Shenghua Gao. Objective-guided image annotation. *IEEE Transactions on Image Processing*, 22(4):1585–1597, 2013.

Julian J. McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *SIGKDD*, pages 785–794, 2015a.

Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52, 2015b.

Robert J. McQueen, Garner S.R., Nevill-Manning C.G., and Witten I.H. Applying machine learning to agricultural data. *Computers and Electronics in Agriculture*, 12(4):275–293, 1995.

Shahar Mendelson. Rademacher averages and phase transitions in glivenko-cantelli classes. *IEEE Trans. Information Theory*, 48(1):251–263, 2002.

Joseph J. Mezrich. When is a tree a hedge? *Financial Analysts Journal*, pages 75–81, 1994.

Tom M. Mitchell, Rebecca A. Hutchinson, Radu Stefan Niculescu, Francisco Pereira, Xuerui Wang, Marcel Adam Just, and Sharlene D. Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57(1-2):145–175, 2004.

Bernard M. E. Moret. Decision trees and diagrams. *Computing Surveys*, 14(4):593–623, 1982.

Sreerama K. Murthy, Simon Kasif, and Steven Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

Hidekazu Oiwa and Ryohei Fujimaki. Partition-wise linear models. In *NIPS*, pages 3527–3535, 2014.

Jan Paralic and Peter Bednar. Text mining for documents annotation and ontology support, 2003.

E. Y. Pee and Johannes O. Royset. On solving large-scale finite minimax problems using exponential smoothing. *Journal of Optimization Theory and Applications*, 148(2):390–421, 2011.

John C. Platt, Nello Cristianini, and John Shawe-Taylor. Large margin dags for multiclass classification. In *NIPS*, pages 547–553, Denver, Colorado, November 1999. MIT Press.

Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *SIGKDD*, pages 263–272, August 2014.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. USA: Morgan Kaufmann Publishers Inc, 1993.

Alain Rakotomamonjy, Francis R. Bach, Stéphane Canu, and Yves Grandvalet. Simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

Jesse Read, Bernhard Pfahringer, Geoffrey Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.

Steven Roman. *Coding and Information Theory*. Springer Verlag, New York, 1992.

S. Rasoul Safavian and David A. Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man and Cybernetics*, 21(3):660–674, 1991.

Steven Salzberg, Rupali Chandar, Holland Ford, Sreerama K. Murthy, and Richard L. White. Decision trees for automated identification of cosmic-ray hits in hubble space telescope images. *Publications of the Astronomical Society of the Pacific*, 107:1–10, 1995.

Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

Claude Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

John Shawe-Taylor and Nello Cristianini. Data-dependent structural risk minimization for perceptron decision trees. In *NIPS*, pages 336–342, 1997.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. New York: Cambridge University Press, 2004.

Shinichi Shimozono, Ayumi Shinohara, Takeshi Shinohara, Satoru Miyano, Satoru Kuhara, and Setsuo Arikawa. Knowledge acquisition from amino acid sequences by machine learning system bonsai. *Transactions of the Information Processing Society of Japan*, 35(10): 2009–2018, 1994.

Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.

Yan Song, Xian-Sheng Hua, Li-Rong Dai, and Meng Wang. Semi-automatic video annotation based on active learning with multiple complementary predictors. In *Multimedia Information Retrieval*, pages 97–104, 2005.

Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

Niranjan A. Subrahmanya and Yung C. Shin. Sparse multiple kernel learning for signal processing applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(5):788–798, 2010.

Mingkui Tan, Ivor W. Tsang, and Li Wang. Towards ultrahigh dimensional feature selection for big data. *JMLR*, 15(1):1371–1429, 2014.

Jinhui Tang, Xian-Sheng Hua, Guo-Jun Qi, Yan Song, and Xiuqing Wu. Video annotation based on kernel linear neighborhood propagation. *IEEE Transactions on Multimedia*, 10 (4):620–628, 2008.

Grigorios Tsoumakas and Ioannis P. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *ECML*, pages 406–417, Warsaw, Poland, 2007. Springer-Verlag.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, 2008.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis P. Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer US, 2010.

De Wang, Danesh Irani, and Calton Pu. Evolutionary study of web spam: Webb spam corpus 2011 versus webb spam corpus 2006. In *8th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 40–49, 2012.

Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus F. M. Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

Zhixiang Eddie Xu, Matt J. Kusner, Kilian Q. Weinberger, and Minmin Chen. Cost-sensitive tree of classifiers. In *ICML*, pages 133–141, 2013.

Zhixiang Eddie Xu, Gao Huang, Kilian Q. Weinberger, and Alice X. Zheng. Gradient boosted feature selection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 522–531, 2014.

Ian En-Hsu Yen, Xiangru Huang, Pradeep Ravikumar, Kai Zhong, and Inderjit S. Dhillon. Pd-sparse : A primal and dual sparse approach to extreme multiclass and multilabel classification. In *ICML*, pages 3069–3077, 2016.

Yiteng Zhai, Yew-Soon Ong, and Ivor W. Tsang. The emerging "big dimensionality". *IEEE Computational Intelligence Magazine*, 9(3):14–26, 2014.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2):894–942, 2010a.

Cun-Hui Zhang and Jian Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *Annals of Statistics*, 36(4):1567–1594, 2008.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *TKDE*, 26(8):1819–1837, Aug 2014.

Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010b.

Yi Zhang and Jeff G. Schneider. Multi-label output codes using canonical correlation analysis. In *Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 873–882. JMLR.org, 2011.

Yi Zhang and Jeff G. Schneider. Maximum margin output coding. In *ICML*, pages 1575–1582, Edinburgh, Scotland, July 2012. Omnipress.

Ji Zhu, Saharon Rosset, Trevor Hastie, and Robert Tibshirani. 1-norm support vector machines. In *NIPS*, pages 49–56, 2003.

Paul C. Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, David Corrigan, and James Giles. *Harness the Power of Big Data – The IBM Big Data Platform*. Mcgraw-Hill, New York, 2012.

Journal of Machine Learning Research 18 (2017) 1-37 Submitted 10/16; Revised 6/17; Published 8/17

Learning Scalable Deep Kernels with Recurrent Structure

Maruan Al-Shedivat ALSHEDIVAT@CS.CMU.EDU
Carnegie Mellon University

Andrew Gordon Wilson ANDREW@CORNELL.EDU
Cornell University

Yunus Saatchi SAATCHI@CANTAB.NET

Zhiting Hu ZHITINGH@CS.CMU.EDU
Carnegie Mellon University

Eric P. Xing EPXING@CS.CMU.EDU
Carnegie Mellon University

Editor: Neil Lawrence

Abstract

Many applications in speech, robotics, finance, and biology deal with sequential data, where ordering matters and recurrent structures are common. However, this structure cannot be easily captured by standard kernel functions. To model such structure, we propose expressive closed-form kernel functions for Gaussian processes. The resulting model, GP-LSTM, fully encapsulates the inductive biases of long short-term memory (LSTM) recurrent networks, while retaining the non-parametric probabilistic advantages of Gaussian processes. We learn the properties of the proposed kernels by optimizing the Gaussian process marginal likelihood using a new provably convergent semi-stochastic gradient procedure, and exploit the structure of these kernels for scalable training and prediction. This approach provides a practical representation for Bayesian LSTMs. We demonstrate state-of-the-art performance on several benchmarks, and thoroughly investigate a consequential autonomous driving application, where the predictive uncertainties provided by GP-LSTM are uniquely valuable.

1. Introduction

There exists a vast array of machine learning applications where the underlying datasets are sequential. Applications range from the entirety of robotics, to speech, audio and video processing. While neural network based approaches have dealt with the issue of *representation learning* for sequential data, the important question of modeling and propagating uncertainty across time has rarely been addressed by these models. For a robotics application such as a self-driving car, however, it is not just desirable, but essential to have complete predictive densities for variables of interest. When trying to stay in lane and keep a safe following distance from the vehicle front, knowing the uncertainty associated with lanes and lead vehicles is as important as the point estimates.

©2017 Maruan Al-Shedivat, Andrew Gordon Wilson, Yunus Saatchi, Zhiting Hu and Eric P. Xing.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-498.html.

Recurrent models with long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) have recently emerged as the leading approach to modeling sequential structure. The LSTM is an efficient gradient-based method for training recurrent networks. LSTMs use a memory cell inside each hidden unit and a special gating mechanism that stabilizes the flow of the back-propagated errors, improving the learning process of the model. While the LSTM provides state-of-the-art results on speech and text data (Graves et al., 2013; Sutskever et al., 2014), quantifying uncertainty or extracting full predictive distributions from deep models is still an area of active research (Gal and Ghahramani, 2016a).

In this paper, we quantify the predictive uncertainty of deep models by following a Bayesian nonparametric approach. In particular, we propose kernel functions which fully encapsulate the structural properties of LSTMs, for use with Gaussian processes. The resulting model enables Gaussian processes to achieve state-of-the-art performance on *sequential regression tasks*, while also allowing for a principled representation of uncertainty and non-parametric flexibility. Further, we develop a provably convergent semi-stochastic optimization algorithm that allows mini-batch updates of the recurrent kernels. We empirically demonstrate that this semi-stochastic approach significantly improves upon the standard non-stochastic first-order methods in runtime and in the quality of the converged solution. For additional scalability, we exploit the algebraic structure of these kernels, decomposing the relevant covariance matrices into Kronecker products of circulant matrices, for $\mathcal{O}(n)$ training time and $\mathcal{O}(1)$ test predictions (Wilson et al., 2015; Wilson and Nickisch, 2015). Our model not only can be interpreted as a Gaussian process with a recurrent kernel, but also as a deep recurrent network with probabilistic outputs, infinitely many hidden units, and a utility function robust to overfitting.

Throughout this paper, we assume basic familiarity with Gaussian processes (GPs). We provide a brief introduction to GPs in the background section; for a comprehensive reference, see, e.g., Rasmussen and Williams (2006). In the following sections, we formalize the problem of learning from sequential data, provide background on recurrent networks and the LSTM, and present an extensive empirical evaluation of our model. Specifically, we apply our model to a number of tasks, including system identification, energy forecasting, and self-driving car applications. Quantitatively, the model is assessed on the data ranging in size from hundreds of points to almost a million with various signal-to-noise ratios demonstrating state-of-the-art performance and linear scaling of our approach. Qualitatively, the model is tested on consequential self-driving applications: lane estimation and lead vehicle position prediction. Indeed, the main focus of this paper is on achieving state-of-the-art performance on consequential applications involving sequential data, following straightforward and scalable approaches to building highly flexible Gaussian process.

We release our code as a library at: `http://github.com/alshedivat/keras-gp`. This library implements the ideas in this paper as well as deep kernel learning (Wilson et al., 2016a) via a Gaussian process layer that can be added to *arbitrary* deep architectures and deep learning frameworks, following the Keras API specification. More tutorials and resources can be found at `https://people.orie.cornell.edu/andrew/code`.

2. Background

We consider the problem of learning a regression function that maps sequences to real-valued target vectors. Formally, let $\overline{\mathbf{X}} = \{\overline{\mathbf{x}_i}\}_{i=1}^n$ be a collection of sequences, $\overline{\mathbf{x}_i} = [\mathbf{x}_i^1, \mathbf{x}_i^2, \cdots, \mathbf{x}^{l_i}]$, each with corresponding length, l_i, where $\mathbf{x}_i^j \in \mathcal{X}$, and $\mathcal{X}$ is an arbitrary domain. Let $\mathbf{y} = \{\mathbf{y}_i\}_{i=1}^n$, $\mathbf{y}_i \in \mathbb{R}^d$, be a collection of the corresponding real-valued target vectors. Assuming that only the most recent L steps of a sequence are predictive of the targets, the goal is to learn a function, $f : \mathcal{X}^L \mapsto \mathbb{R}^d$, from some family, $\mathcal{F}$, based on the available data.

As a working example, consider the problem of estimating position of the lead vehicle at the next time step from LIDAR, GPS, and gyroscopic measurements of a self-driving car available for a number of previous steps. This task is a classical instance of the *sequence-to-reals regression*, where a temporal sequence of measurements is regressed to the future position estimates. In our notation, the sequences of inputs are vectors of measurements, $\overline{\mathbf{x}_1} = [\mathbf{x}^1]$, $\overline{\mathbf{x}_2} = [\mathbf{x}^1, \mathbf{x}^2]$, $\ldots$, $\overline{\mathbf{x}_n} = [\mathbf{x}^1, \mathbf{x}^2, \cdots \mathbf{x}^n]$, are indexed by time and would be of growing lengths. Typically, input sequences are considered up to a finite-time horizon, L, that is assumed to be predictive for the future targets of interest. The targets, $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$, are two-dimensional vectors that encode positions of the lead vehicle in the ego-centric coordinate system of the self-driving car.

Note that the problem of learning a mapping, $f : \mathcal{X}^L \mapsto \mathbb{R}^d$, is challenging. While considering whole sequences of observations as input features is necessary for capturing long-term temporal correlations, it virtually blows up the dimensionality of the problem. If we assume that each measurement is p-dimensional, i.e., $\mathcal{X} \subseteq \mathbb{R}^p$, and consider L previous steps as distinct features, the regression problem will become $(L \times p)$-dimensional. Therefore, to avoid overfitting and be able to extract meaningful signal from a finite amount of data, it is crucial to exploit the sequential nature of observations.

Recurrent models. One of the most successful ways to exploit sequential structure of the data is by using a class of recurrent models. In the sequence-to-reals regression scenario, such a model expresses the mapping $f : \mathcal{X}^L \mapsto \mathbb{R}^d$ in the following general recurrent form:

$$\mathbf{y} = \boldsymbol{\psi}(\mathbf{h}^L) + \boldsymbol{\epsilon}^t,\ \mathbf{h}^t = \boldsymbol{\phi}(\mathbf{h}^{t-1}, \mathbf{x}^t) + \boldsymbol{\delta}^t,\ t = 1, \ldots, L, \tag{1}$$

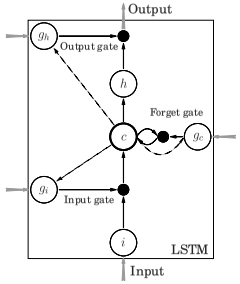
where $\mathbf{x}^t$ is an input observation at time t, $\mathbf{h}^t$ is a corresponding latent representation, and $\mathbf{y}$ is a target vector. Functions $\boldsymbol{\phi}(\cdot)$ and $\boldsymbol{\psi}(\cdot)$ specify model transitions and emissions, respectively, and $\boldsymbol{\delta}^t$ and $\boldsymbol{\epsilon}^t$ are additive noises. While $\boldsymbol{\phi}(\cdot)$ and $\boldsymbol{\psi}(\cdot)$ can be arbitrary, they are typically time-invariant. This strong but realistic assumption incorporated into the structure of the recurrent mapping significantly reduces the complexity of the family of functions, $\mathcal{F}$, regularizes the problem, and helps to avoid severe overfitting.

Recurrent models can account for various patterns in sequences by *memorizing* internal representations of their dynamics via adjusting $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$. Recurrent neural networks (RNNs) model recurrent processes by using linear parametric maps followed by nonlinear activations:

$$\mathbf{y} = \boldsymbol{\psi}(W_{hy}^\top \mathbf{h}^{t-1}),\ \mathbf{h}^t = \boldsymbol{\phi}(W_{hh}^\top \mathbf{h}^{t-1}, W_{xh}^\top \mathbf{x}^{t-1}),\ t = 1, \ldots, L, \tag{2}$$

where W_{hy}, W_{hh}, W_{xh} are weight matrices to be learned[1] and $\boldsymbol{\phi}(\cdot)$ and $\boldsymbol{\psi}(\cdot)$ here are some fixed element-wise functions. Importantly and contrary to the standard hidden Markov models (HMMs), *the state* of an RNN at any time t is *distributed* and effectively represented by an entire hidden sequence, $[\mathbf{h}^1, \cdots, \mathbf{h}^{t-1}, \mathbf{h}^t]$. A major disadvantage of the vanilla RNNs is that their training is nontrivial due to the so-called *vanishing gradient* problem (Bengio et al., 1994): the error back-propagated through t time steps diminishes exponentially which makes learning long-term relationships nearly impossible.

LSTM. To overcome vanishing gradients, Hochreiter and Schmidhuber (1997) proposed a long short-term memory (LSTM) mechanism that places a *memory cell* into each hidden unit and uses differentiable gating variables. The update rules for the hidden representation at time t have the following form (here $\sigma(\cdot)$ and $\tanh(\cdot)$ are element-wise sigmoid and hyperbolic tangent functions, respectively):



$$\begin{aligned}
\mathbf{i}^t &= \tanh\left(W_{xc}^\top \mathbf{x}^t + W_{hc}^\top \mathbf{h}^{t-1} + \mathbf{b}_c\right),\\
\mathbf{g}_i^t &= \sigma\left(W_{xi}^\top \mathbf{x}^t + W_{hi}^\top \mathbf{h}^{t-1} + W_{ci}^\top \mathbf{c}^{t-1} + \mathbf{b}_i\right),\\
\mathbf{c}^t &= \mathbf{g}_c^t\ \mathbf{c}^{t-1} + \mathbf{g}_i^t\ \mathbf{i}^t,\\
\mathbf{g}_c^t &= \sigma\left(W_{xf}^\top \mathbf{x}^t + W_{hf}^\top \mathbf{h}^{t-1} + W_{cf}^\top \mathbf{c}^{t-1} + \mathbf{b}_f\right),\\
\mathbf{o}^t &= \tanh\left(\mathbf{c}^t\right),\\
\mathbf{g}_o^t &= \sigma\left(W_{xo}^\top \mathbf{x}^t + W_{ho}^\top \mathbf{h}^{t-1} + W_{co}^\top \mathbf{c}^t + \mathbf{b}_o\right),\\
\mathbf{h}^t &= \mathbf{g}_o^t\ \mathbf{o}^t.
\end{aligned} \tag{3}$$

As illustrated above, $\mathbf{g}_i^t$, $\mathbf{g}_c^t$, and $\mathbf{g}_o^t$ correspond to the input, forget, and output gates, respectively. These variables take their values in $[0, 1]$ and when combined with the internal states, $\mathbf{c}^t$, and inputs, $\mathbf{x}^t$, in a multiplicative fashion, they play the role of soft gating. The gating mechanism not only improves the flow of errors through time, but also, allows the the network to decide whether to keep, erase, or overwrite certain memorized information based on the forward flow of inputs and the backward flow of errors. This mechanism adds stability to the network's memory.

Gaussian processes. The Gaussian process (GP) is a Bayesian nonparametric model that generalizes the Gaussian distributions to functions. We say that a random function f is drawn from a GP with a mean function μ and a covariance kernel k, $f \sim \mathcal{GP}(\mu, k)$, if for any vector of inputs, $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$, the corresponding vector of function values is Gaussian:

$$[f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)] \sim \mathcal{N}(\boldsymbol{\mu}, K_{X,X}),$$

with mean $\boldsymbol{\mu}$, such that $\boldsymbol{\mu}_i = \mu(\mathbf{x}_i)$, and covariance matrix $K_{X,X}$ that satisfies $(K_{X,X})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. GPs can be seen as distributions over the reproducing kernel Hilbert space (RKHS) of functions which is uniquely defined by the kernel function, k (Schölkopf and Smola, 2002).

1. The bias terms are omitted for clarity of presentation.

GPs with RBF kernels are known to be universal approximators with prior support to within an arbitrarily small epsilon band of any continuous function (Micchelli et al., 2006).

Assuming additive Gaussian noise, $y \mid \mathbf{x} \sim \mathcal{N}\left(f(\mathbf{x}), \sigma^2\right)$, and a GP prior on $f(\mathbf{x})$, given training inputs $\mathbf{x}$ and training targets $\mathbf{y}$, the predictive distribution of the GP evaluated at an arbitrary test point $\mathbf{x}_*$ is:

$$\mathbf{f}_* \mid \mathbf{x}_*, \mathbf{x}, \mathbf{y}, \sigma^2 \sim \mathcal{N}\left(\mathbb{E}[\mathbf{f}_*], \mathrm{Cov}[\mathbf{f}_*]\right), \tag{4}$$

where

$$\begin{aligned} \mathbb{E}[\mathbf{f}_*] &= \boldsymbol{\mu}_{X_*} + K_{X_*,X}[K_{X,X} + \sigma^2 I]^{-1}\mathbf{y}, \\ \mathrm{Cov}[\mathbf{f}_*] &= K_{X_*,X_*} - K_{X_*,X}[K_{X,X} + \sigma^2 I]^{-1}K_{X,X_*}. \end{aligned} \tag{5}$$

Here, $K_{X_*,X}$, K_{X,X_*}, $K_{X,X}$, and K_{X_*,X_*} are matrices that consist of the covariance function, k, evaluated at the corresponding points, $\mathbf{x} \in \mathbf{X}$ and $\mathbf{x}_* \in \mathbf{X}_*$, and $\boldsymbol{\mu}_{X_*}$ is the mean function evaluated at $\mathbf{x}_* \in \mathbf{X}_*$. GPs are fit to the data by optimizing *the evidence*—the marginal probability of the data given the model—with respect to kernel hyperparameters. The evidence has the form:

$$\log \mathbb{P}\left(\mathbf{y} \mid \mathbf{x}\right) = -\left[\mathbf{y}^\top (K + \sigma^2 I)^{-1}\mathbf{y} + \log\det(K + \sigma^2 I)\right] + \text{const}, \tag{6}$$

where we use a shorthand K for $K_{X,X}$, and K implicitly depends on the kernel hyperparameters. This objective function consists of a *model fit* and a *complexity penalty* term that results in an automatic Occam's razor for realizable functions (Rasmussen and Ghahramani, 2001). By optimizing the evidence with respect to the kernel hyperparameters, we effectively learn the the structure of the space of functional relationships between the inputs and the targets. For further details on Gaussian processes and relevant literature we refer interested readers to the classical book by Rasmussen and Williams (2006).

Turning back to the problem of learning from sequential data, it seems natural to apply the powerful GP machinery to modeling complicated relationships. However, GPs are limited to learning only pairwise correlations between the inputs and are unable to account for long-term dependencies, often dismissing complex temporal structures. Combining GPs with recurrent models has potential to addresses this issue.

3. Related work

The problem of learning from sequential data, especially from temporal sequences, is well known in the control and dynamical systems literature. Stochastic temporal processes are usually described either with generative *autoregressive models* (AM) or with *state-space models* (SSM) (Van Overschee and De Moor, 2012). The former approach includes nonlinear auto-regressive models with exogenous inputs (NARX) that are constructed by using, *e.g.*, neural networks (Lin et al., 1996) or Gaussian processes (Kocijan et al., 2005). The latter approach additionally introduces unobservable variables, *the state*, and constructs

autoregressive dynamics in the latent space. This construction allows to represent and propagate uncertainty through time by explicitly modeling the signal (via the state evolution) and the noise. Generative SSMs can be also used in conjunction with discriminative models via the Fisher kernel (Jaakkola and Haussler, 1999).

Modeling time series with GPs is equivalent to using *linear-Gaussian* autoregressive or SSM models (Box et al., 1994). Learning and inference are efficient in such models, but they are not designed to capture long-term dependencies or correlations beyond pairwise. Wang et al. (2005) introduced GP-based state-space models (GP-SSM) that use GPs for transition and/or observation functions. These models appear to be more general and flexible as they account for uncertainty in the state dynamics, though require complicated approximate training and inference, which are hard to scale (Turner et al., 2010; Frigola et al., 2014).

Perhaps the most recent relevant work to our approach is *recurrent Gaussian processes* (RGP) (Mattos et al., 2015). RGP extends the GP-SSM framework to regression on sequences by using a recurrent architecture with GP-based activation functions. The structure of the RGP model mimics the standard RNN, where every parametric layer is substituted with a Gaussian process. This procedure allows one to propagate uncertainty throughout the network for an additional cost. Inference is intractable in RGP, and efficient training requires a sophisticated *approximation* procedure, the so-called *recurrent variational Bayes*. In addition, the authors have to turn to RNN-based approximation of the variational mean functions to battle the growth of the number of variational parameters with the size of data. While technically promising, RGP seems problematic from the application perspective, especially in its implementation and scalability aspects.

Our model has several distinctions with prior work aiming to regress sequences to reals. Firstly, one of our goals is to keep the model as simple as possible while being able to represent and quantify *predictive uncertainty*. We maintain an analytical objective function and refrain from complicated and difficult-to-diagnose inference schemes. This simplicity is achieved by giving up the idea of propagating signal through a chain GPs connected in a recurrent fashion. Instead, we propose to directly *learn kernels* with recurrent structure via joint optimization of a simple functional composition of a standard GP with a recurrent model (*e.g.*, LSTM), as described in detail in the following section. Similar approaches have recently been explored and proved to be fruitful for non-recurrent deep networks (Wilson et al., 2016a,b; Calandra et al., 2016). We remark that combinations of GPs with nonlinear functions have also been considered in the past in a slightly different setting of warped regression targets (Snelson et al., 2003; Wilson and Ghahramani, 2010; Lázaro-Gredilla, 2012). Additionally, uncertainty over the recurrent parts of our model is represented via dropout, which is computationally cheap and turns out to be equivalent to approximate Bayesian inference in a deep Gaussian process (Damianou and Lawrence, 2013) with particular intermediate kernels (Gal and Ghahramani, 2016a,b). Finally, one can also view our model as a standalone flexible Gaussian process, which leverages learning techniques that scale to massive datasets (Wilson and Nickisch, 2015; Wilson et al., 2015).

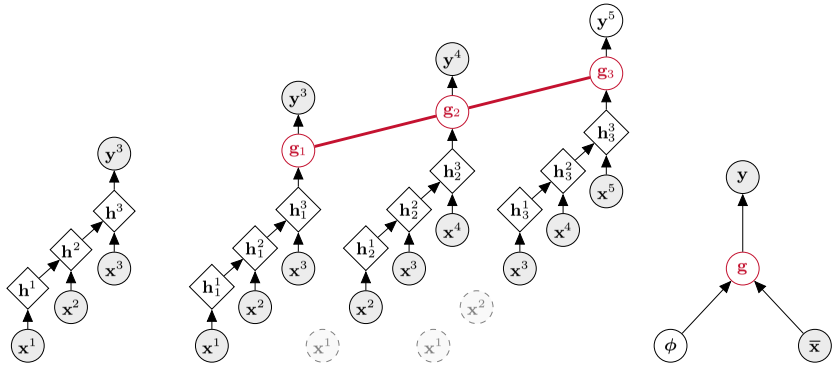


Figure 1: (a) Graphical representation of a recurrent model (RNN/LSTM) that maps an input sequence to a target value in one-step-ahead prediction manner. Shaded variables are observable, diamond variables denote deterministic dependence on the inputs. (b) Graphical model for GP-LSTM with a time lag, $L = 3$, two training time points, $t = 3$ and $t = 4$, and a testing time point, $t = 5$. Latent representations are mapped to the outputs through a Gaussian field (denoted in red) that globally correlates predictions. Dashed variables represent data instances unused at the given time step. (c) Graphical representation of a GP with a kernel structured with a parametric map, $\boldsymbol{\phi}$.

4. Learning recurrent kernels

Gaussian processes with different kernel functions correspond to different structured probabilistic models. For example, some special cases of the Matérn class of kernels induce models with Markovian structure (Stein, 1999). To construct deep kernels with recurrent structure we transform the original input space with an LSTM network and build a kernel directly in the transformed space, as shown in Figure 1b.

In particular, let L $\boldsymbol{\phi} : \mathcal{X}^L \mapsto \mathcal{H}$ be an arbitrary deterministic transformation of the input sequences into some latent space, $\mathcal{H}$. Next, let $k : \mathcal{H}^2 \mapsto \mathbb{R}$ be a real-valued kernel defined on $\mathcal{H}$. The decomposition of the kernel and the transformation, $\tilde{k} = k \circ \boldsymbol{\phi}$, is defined as

$$\tilde{k}(\overline{\mathbf{x}}, \overline{\mathbf{x}}') = k(\boldsymbol{\phi}(\overline{\mathbf{x}}), \boldsymbol{\phi}(\overline{\mathbf{x}}')), \text{ where } \overline{\mathbf{x}}, \overline{\mathbf{x}}' \in \mathcal{X}^L, \text{ and } \tilde{k} : \left(\mathcal{X}^L\right)^2 \mapsto \mathbb{R}. \tag{7}$$

It is trivial to show that $\tilde{k}(\overline{\mathbf{x}}, \overline{\mathbf{x}}')$ is a valid kernel defined on $\mathcal{X}^L$ (MacKay, 1998, Ch. 5.4.3). In addition, if $\boldsymbol{\phi}(\cdot)$ is represented by a neural network, the resulting model can be viewed as the same network, but with an additional GP-layer and the negative log marginal likelihood (NLML) objective function used instead of the standard mean squared error (MSE).

Input embedding is well-known in the Gaussian process literature (e.g. MacKay, 1998; Hinton and Salakhutdinov, 2008). Recently, Wilson et al. (2016a,b) have successfully scaled the approach and demonstrated strong results in regression and classification tasks for kernels based on feedforward and convolutional architectures. In this paper, we apply the same technique to learn kernels with recurrent structure by transforming input sequences with a recurrent neural network that acts as $\boldsymbol{\phi}(\cdot)$. In particular, a (multi-layer) LSTM architecture is used to embed L steps of the input time series into a single vector in the hidden space, $\mathcal{H}$. For the embedding, as common, we use the last hidden vector produced by the recurrent network. Note that however, any variations to the embedding (e.g., using other types of recurrent units, adding 1-dimensional pooling layers, or attention mechanisms) are all fairly straightforward[2]. More generally, the recurrent transformation can be random itself (Figure 1), which would enable direct modeling of uncertainty *within* the recurrent dynamics, but would also require inference for $\boldsymbol{\phi}$ (e.g., as in Mattos et al., 2015). In this study, we limit our consideration of random recurrent maps to only those induced by dropout.

Unfortunately, once the the MSE objective is substituted with NLML, it no longer factorizes over the data. This prevents us from using the well-established stochastic optimization techniques for training our recurrent model. In the case of feedforward and convolutional networks, Wilson et al. (2016a) proposed to pre-train the input transformations and then fine-tune them by jointly optimizing the GP marginal likelihood with respect to hyperparameters and the network weights using *full-batch* algorithms. When the transformation is recurrent, stochastic updates play a key role. Therefore, we propose a semi-stochastic block-gradient optimization procedure which allows mini-batching weight updates and fully joint training of the model from scratch.

4.1 Optimization

The negative log marginal likelihood of the Gaussian process has the following form:

$$\mathcal{L}(K) = \mathbf{y}^\top (K_y + \sigma^2 I)^{-1} \mathbf{y} + \log\det(K_y + \sigma^2 I) + \text{const}, \tag{8}$$

where $K_y + \sigma^2 I$ ($\triangleq K$) is the Gram kernel matrix, K_y, is computed on $\{\boldsymbol{\phi}(\overline{\mathbf{x}}_i)\}_{i=1}^N$ and implicitly depends on the base kernel hyperparameters, $\boldsymbol{\theta}$, and the parameters of the recurrent neural transformation, $\boldsymbol{\phi}(\cdot)$, denoted W and further referred as the *transformation hyperparameters*. Our goal is to optimize $\mathcal{L}$ with respect to both $\boldsymbol{\theta}$ and W.

The derivative of the NLML objective with respect to $\boldsymbol{\theta}$ is standard and takes the following form (Rasmussen and Williams, 2006):

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \frac{1}{2} \mathbf{tr}\left(\left[K^{-1} \mathbf{y}\mathbf{y}^\top K^{-1} - K^{-1} \right] \frac{\partial K}{\partial \boldsymbol{\theta}} \right), \tag{9}$$

2. Details on the particular architectures used in our empirical study are discussed in the next section.

where $\partial K/\partial \boldsymbol{\theta}$ is depends on the kernel function, $k(\cdot,\cdot)$, and usually has an analytic form. The derivative with respect to the l-th transformation hyperparameter, W_l, is as follows:[3]

$$\frac{\partial \mathcal{L}}{\partial W_l} = \frac{1}{2}\sum_{i,j}\left(K^{-1}\mathbf{y}\mathbf{y}^\top K^{-1} - K^{-1}\right)_{ij}\left\{\left(\frac{\partial k(\mathbf{h}_i,\mathbf{h}_j)}{\partial \mathbf{h}_i}\right)^\top \frac{\partial \mathbf{h}_i}{\partial W_l} + \left(\frac{\partial k(\mathbf{h}_i,\mathbf{h}_j)}{\partial \mathbf{h}_j}\right)^\top \frac{\partial \mathbf{h}_j}{\partial W_l}\right\}, \tag{10}$$

where $\mathbf{h}_i = \boldsymbol{\phi}(\overline{\mathbf{x}}_i)$ corresponds to the latent representation of the the i-th data instance. Once the derivatives are computed, the model can be trained with any first-order or quasi-Newton optimization routine. However, application of the stochastic gradient method—the *de facto* standard optimization routine for deep recurrent networks—is not straightforward: neither the objective, nor its derivatives factorize over the data[4] due to the kernel matrix inverses, and hence convergence is not guaranteed.

Semi-stochastic alternating gradient descent. Observe that once the kernel matrix, K, is *fixed*, the expression, $\left(K^{-1}\mathbf{y}\mathbf{y}^\top K^{-1} - K^{-1}\right)$, can be precomputed on the full data and fixed. Subsequently, Eq. (10) turns into a weighted sum of independent functions of each data point. This observation suggests that, given a *fixed kernel matrix*, one could compute a stochastic update for W on a mini-batch of training points by only using the corresponding sub-matrix of K. Hence, we propose to optimize GPs with recurrent kernels in a semi-stochastic fashion, alternating between updating the kernel hyperparameters, $\boldsymbol{\theta}$, on the full data first, and then updating the weights of the recurrent network, W, using stochastic steps. The procedure is given in Algorithm 1.

Semi-stochastic alternating gradient descent is a special case of block-stochastic gradient iteration (Xu and Yin, 2015). While the latter splits the variables into arbitrary blocks and applies Gauss–Seidel type stochastic gradient updates to each of them, our procedure alternates between applying deterministic updates to $\boldsymbol{\theta}$ and stochastic updates to W of the form $\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} + \lambda_{\boldsymbol{\theta}}^{(t)}\mathbf{g}_{\boldsymbol{\theta}}^{(t)}$ and $W^{(t+1)} \leftarrow W^{(t)} + \lambda_W^{(t)}\mathbf{g}_W^{(t)}$[5]. The corresponding Algorithm 1 is provably convergent for convex and non-convex problems under certain conditions. The following theorem adapts results of Xu and Yin (2015) to our optimization scheme.

Theorem 1 (informal) *Semi-stochastic alternating gradient descent converges to a fixed point when the learning rate, λ_t, decays as $\Theta(1/t^{\frac{1+\delta}{2}})$ for any $\delta \in (0,1]$.*

Applying alternating gradient to our case has a catch: the kernel matrix (and its inverse) has to be updated each time W and $\boldsymbol{\theta}$ are changed, *i.e.*, on every mini-batch iteration (marked red in Algorithm 1). Computationally, this updating strategy defeats the purpose of stochastic gradients because we have to use the entire data on each step. To deal with the issue of computational efficiency, we use ideas from asynchronous optimization.

3. Step-by-step derivations are given in Appendix B.
4. Cannot be represented as sums of independent functions of each data point.
5. In principle, stochastic updates of $\boldsymbol{\theta}$ are also possible. As we will see next, we choose in practice to keep the kernel matrix fixed while performing stochastic updates. Due to sensitivity of the kernel to even small changes in $\boldsymbol{\theta}$, convergence of the fully stochastic scheme is fragile.

Algorithm 1 Semi-stochastic alternating gradient descent.

input Data – $(X, \mathbf{y})$, kernel – $k_{\boldsymbol{\theta}}(\cdot,\cdot)$, recurrent transformation – $\phi_{\mathbf{w}}(\cdot)$.
1: Initialize $\boldsymbol{\theta}$ and $\mathbf{w}$; compute initial K.
2: **repeat**
3: **for all** mini-batches X_b in X **do**
4: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \texttt{update}_{\boldsymbol{\theta}}(X, \boldsymbol{\theta}, K)$. and $\mathbf{w} \leftarrow \mathbf{w} + \texttt{update}_{\mathbf{w}}(X_b, \mathbf{w}, K)$.
5: Update the kernel matrix, K.
6: **end for**
7: **until** Convergence
output Optimal $\boldsymbol{\theta}^*$ and $\mathbf{w}^*$

Algorithm 2 Semi-stochastic asynchronous gradient descent.

input Data – $(X, \mathbf{y})$, kernel – $k_{\boldsymbol{\theta}}(\cdot,\cdot)$, recurrent transformation – $\phi_{\mathbf{w}}(\cdot)$.
1: Initialize $\boldsymbol{\theta}$ and $\mathbf{w}$; compute initial K.
2: **repeat**
3: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \texttt{update}_{\boldsymbol{\theta}}(X, \mathbf{w}, K)$.
4: **for all** mini-batches X_b in X **do**
5: $\mathbf{w} \leftarrow \mathbf{w} + \texttt{update}_{\mathbf{w}}(X_b, \mathbf{w}, K^{\text{stale}})$.
6: **end for**
7: Update the kernel matrix, K.
8: **until** Convergence
output Optimal $\boldsymbol{\theta}^*$ and $\mathbf{w}^*$

Asynchronous techniques. One of the recent trends in parallel and distributed optimization is applying updates in an asynchronous fashion (Agarwal and Duchi, 2011). Such strategies naturally require some tolerance to delays in parameter updates (Langford et al., 2009). In our case, we modify Algorithm 1 to allow *delayed kernel matrix updates*.

The key observation is very intuitive: when the stochastic updates of W are small enough, K does not change much between mini-batches, and hence we can perform multiple stochastic steps for W before re-computing the kernel matrix, K, and still converge. For example, K may be updated once at the end of each pass through the entire data (see Algorithm 2). To ensure convergence of the algorithm, it is important to strike the balance between (a) the learning rate for W and (b) the frequency of the kernel matrix updates. The following theorem provides convergence results under certain conditions.

Theorem 2 (informal) *Semi-stochastic gradient descent with τ-delayed kernel updates converges to a fixed point when the learning rate, λ_t, decays as $\Theta(1/\tau t^{\frac{1+\delta}{2}})$ for any $\delta \in (0,1]$.*

Formal statements, conditions, and proofs for Theorems 1 and 2 are given in Appendix C.2.

Why stochastic optimization? GPs with recurrent kernels can be also trained with full-batch gradient descent, as proposed by Wilson et al. (2016a). However, stochastic gradient methods have been proved to attain better generalization (Hardt et al., 2016) and often demonstrate superior performance in deep and recurrent architectures (Wilson and Martinez, 2003). Moreover, stochastic methods are 'online', i.e., they update model parameters based on subsets of an incoming data stream, and hence can scale to very large datasets. In our experiments, we demonstrate that GPs with recurrent kernels trained with Algorithm 2 converge faster (i.e., require fewer passes through the data) and attain better performance than if trained with full-batch techniques.

Stochastic variational inference. Stochastic variational inference (SVI) in Gaussian processes (Hensman et al., 2013) is another viable approach to enabling stochastic optimization for GPs with recurrent kernels. Such method would optimize a variational lower bound on the original objective that factorizes over the data by construction. Recently, Wilson et al. (2016b) developed such a stochastic variational approach in the context of deep kernel learning. Note that unlike all previous existing work, our proposed approach does not require a variational approximation to the marginal likelihood to perform mini-batch training of Gaussian processes.

4.2 Scalability

Learning and inference with Gaussian processes requires solving a linear system involving an $n \times n$ kernel matrix, $K^{-1}y$, and computing a log determinant over K. These operations typically require $\mathcal{O}(n^3)$ computations for n training data points, and $\mathcal{O}(n^2)$ storage. In our approach, scalability is achieved through semi-stochastic training and structure-exploiting inference. In particular, asynchronous semi-stochastic gradient descent reduces both the total number of passes through the data required for the model to converge and the number of calls to the linear system solver; exploiting the structure of the kernels significantly reduces the time and memory complexities of the linear algebraic operations.

More precisely, we replace all instances of the covariance matrix $K_{\mathbf{y}}$ with $WK_{U,U}W^{\top}$, where W is a sparse interpolation matrix, and $K_{U,U}$ is the covariance matrix evaluated over m latent inducing points, which decomposes into a Kronecker product of circulant matrices (Wilson and Nickisch, 2015; Wilson et al., 2015). This construction makes inference and learning scale as $\mathcal{O}(n)$ and test predictions be $\mathcal{O}(1)$, while preserving model structure. For the sake of completeness, we provide an overview of the underlying algebraic machinery in Appendix A.

At a high level, because W is sparse and $K_{U,U}$ is structured it is possible to take extremely fast matrix vector multiplications (MVMs) with the approximate covariance matrix $K_{X,X}$. One can then use methods such as linear conjugate gradients, which only use MVMs, to efficiently solve linear systems. MVM or scaled eigenvalue approaches (Wilson and Nickisch, 2015; Wilson et al., 2015) can also be used to efficiently compute the log determinant and its derivatives. Kernel interpolation (Wilson et al., 2015) also enables fast predictions, as we describe further in the Appendix.

5. Experiments

We compare the proposed Gaussian processes with recurrent kernels based on RNN and LSTM architectures (GP-RNN/LSTM) with a number of baselines on datasets of various complexity and ranging in size from hundreds to almost a million of time points. For the datasets with more than a few thousand points, we use a massively scalable version of GPs (see Section 4.2) and demonstrate its scalability during inference and learning. We carry

Table 1: Statistics for the data used in experiments. SNR was determined by assuming a certain degree of smoothness of the signal, fitting kernel ridge regression with RBF kernel to predict the targets from the input time series, and regarding the residuals as the noise. Tasks with low average correlation between inputs and targets and lower SNR are harder prediction problems.

Dataset	Task	# time steps	# dim	# outputs	Abs. corr.	SNR
Drives	system ident.	500	1	1	0.7994	25.72
Actuator		1,024	1	1	0.0938	12.47
GEF	power load	38,064	11	1	0.5147	89.93
	wind power	130,963	16	1	0.1731	4.06
Car	speed	932,939	6	1	0.1196	159.33
	gyro yaw		6	1	0.0764	3.19
	lanes		26	16	0.0816	—
	lead vehicle		9	2	0.1099	—

out a number of experiments that help to gain empirical insights about the convergence properties of the proposed optimization procedure with delayed kernel updates. Additionally, we analyze the regularization properties of GP-RNN/LSTM and compare them with other techniques, such as dropout. Finally, we apply the model to the problem of lane estimation and lead vehicle position prediction, both critical in autonomous driving applications.

5.1 Data and the setup

Below, we describe each dataset we used in our experiments and the associated prediction tasks. The essential statistics for the datasets are summarized in Table 1.

System identification. In the first set of experiments, we used publicly available nonlinear system identification datasets: *Actuator*[6] (Sjöberg et al., 1995) and *Drives*[7] (Wigren, 2010). Both datasets had one dimensional input and output time series. *Actuator* had the size of the valve opening as the input and the resulting change in oil pressure as the output. *Drives* was from a system with motors that drive a pulley using a flexible belt; the input was the sum of voltages applied to the motors and the output was the speed of the belt.

Smart grid data[8]. We considered the problem of forecasting for the smart grid that consisted of two tasks (Figure 2). The first task was to predict power load from the historical temperature data. The data had 11 input time series coming from hourly measurements of temperature on 11 zones and an output time series that represented the cumulative hourly power load on a U.S. utility. The second task was to predict power generated by wind farms from the wind forecasts. The data consisted of 4 different hourly forecasts of the wind and hourly values of the generated power by a wind farm. Each wind forecast was a

6. http://www.iau.dtu.dk/nnbook/systems.html
7. http://www.it.uu.se/research/publications/reports/2010-020/NonlinearData.zip.
8. The smart grid data were taken from Global Energy Forecasting Kaggle competitions organized in 2012.

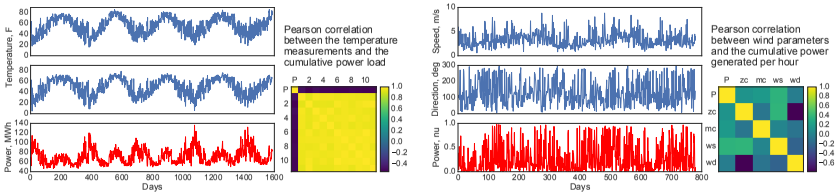


Figure 2: **Left:** Visualization of the *GEF-power* time series for two zones and the cumulative load with the time resolution of 1 day. Cumulative power load is generally negatively correlated with the temperature measurements on all the zones. **Right:** Visualization of the *GEF-wind* time series with the time resolution of 1 day.

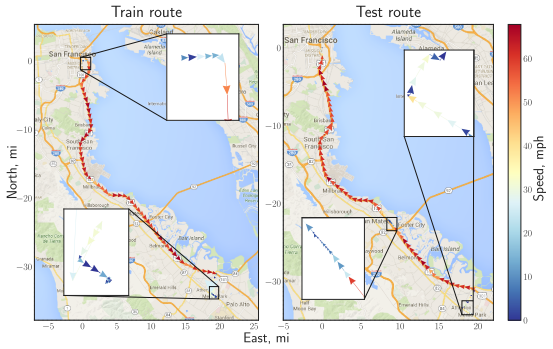
4-element vector that corresponded to *zonal component*, *meridional component*, *wind speed* and *wind angle*. In our experiments, we concatenated the 4 different 4-element forecasts, which resulted in a 16-dimensional input time series.

Self-driving car dataset[9]**.** One of the main target applications of the proposed model is prediction for autonomous driving. We considered a large dataset coming from sensors of a self-driving car that was recorded on two car trips with discretization of 10 *ms*. The data featured two sets of GPS ECEF locations, ECEF velocities, measurements from a fiber-optic gyro compass, LIDAR, and a few more time series from a variety of IMU sensors. Additionally, locations of the left and right lanes were extracted from a video stream for each time step as well as the position of the lead vehicle from the LIDAR measurements. We considered the data from the first trip for training and from the second trip for validation and testing. A visualization of the car routes with 25 second discretization in the ENU coordinates are given in Figure 3. We consider four tasks, the first two of which are more of proof-of-concept type variety, while the final two are fundamental to good performance for a self-driving car:

1. Speed prediction from noisy GPS velocity estimates and gyroscopic inputs.
2. Prediction of the angular acceleration of the car from the estimates of its speed and steering angle.
3. Point-wise prediction of the lanes from the estimates at the previous time steps, and estimates of speed, gyroscopic and compass measurements.
4. Prediction of the lead vehicle location from its location at the previous time steps, and estimates of speed, gyroscopic and compass measurements.

We provide more specific details on the smart grid data and self-driving data in Appendix D.

9. The dataset is proprietary. It was released in part for public use under the Creative Commons Attribution 3.0 license: http://archive.org/details/comma-dataset. More about the self-driving car: http://www.bloomberg.com/features/2015-george-hotz-self-driving-car/.



	Train	Test
Time	42.8 min	46.5 min
Speed of the self-driving car		
Min	0.0 mph	0.0 mph
Max	80.3 mph	70.1 mph
Average	42.4 mph	38.4 mph
Median	58.9 mph	44.8 mph
Distance to the lead vehicle		
Min	23.7 m	29.7 m
Max	178.7 m	184.7 m
Average	85.4 m	72.6 m
Median	54.9 m	53.0 m

Figure 3 & Table 2: **Left:** Train and test routes of the self-driving car in the ENU coordinates with the origin at the starting location. Arrows point in the direction of motion; color encodes the speed. Insets zoom selected regions of the routes. Best viewed in color. **Right:** Summary of the data collected on the train and test routes.

Models and metrics. We used a number of classical baselines: NARX (Lin et al., 1996), GP-NARX models (Kocijan et al., 2005), and classical RNN and LSTM architectures. The kernels of our models, GP-NARX/RNN/LSTM, used the ARD base kernel function and were structured by the corresponding baselines.[10] As the primary metric, we used root mean squared error (RMSE) on a held out set and additionally negative log marginal likelihood (NLML) on the training set for the GP-based models.

We train all the models to perform one-step-ahead prediction in an autoregressive setting, where targets at the future time steps are predicted from the input and target values at a fixed number of past time steps. For the system identification task, we additionally consider the non-autoregressive scenario (i.e., mapping only input sequences to the future targets), where we are performing prediction in the *free simulation mode*, and included recurrent Gaussian processes (Mattos et al., 2015) in our comparison. In this case, none of the future targets are available and the models have to re-use their own past predictions to produce future forecasts).

A note on implementation. Recurrent parts of each model were implemented using *Keras*[11] library. We extended *Keras* with the GP layer and developed a backed engine based on the *GPML* library[12]. Our approach allows us to take full advantage of the functionality available in *Keras* and *GPML*, *e.g.*, use automatic differentiation for the recurrent part of the model. Our code is available at http://github.com/alshedivat/keras-gp/.

10. GP-NARX is a special instance of our more general framework and we trained it using the proposed semi-stochastic algorithm.
11. http://www.keras.io
12. http://www.gaussianprocess.org/gpml/code/matlab/doc/

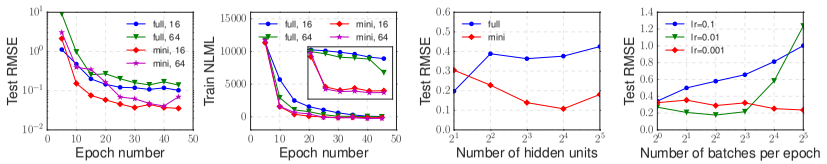


Figure 4: *Two charts on the left:* Convergence of the optimization in terms of RMSE on test and NLML on train. The inset zooms the region of the plot right beneath it using log scale for the vertical axis. *full* and *mini* denote full-batch and mini-batch optimization procedures, respectively, while 16 and 64 refer to models with the respective number of units per hidden layer. *Two charts on the right:* Test RMSE for a given architecture trained with a specified method and/or learning rate.

5.2 Analysis

This section discusses quantitative and qualitative experimental results. We only briefly introduce the model architectures and the training schemes used in each of the experiments. We provide a comprehensive summary of these details in Appendix E.

5.2.1 Convergence of the optimization

To address the question of whether stochastic optimization of recurrent kernels is necessary and to assess the behavior of the proposed optimization scheme with delayed kernel updates, we conducted a number of experiments on the Actuator dataset (Figure 4).

First, we constructed two GP-LSTM models with 1 recurrent hidden layer and 16 or 64 hidden units and trained them with (non-stochastic) full-batch iterative procedure (similar to the proposal of Wilson et al. (2016a)) and with our semi-stochastic optimizer with delayed kernel updates (Algorithm 2). The convergence results are given on the first two charts. Both in terms of the error on a held out set and the NLML on the training set, the models trained with mini-batches converged faster and demonstrated better final performance.

Next, we compared the two optimization schemes on the same GP-LSTM architecture with different sizes of the hidden layer ranging from 2 to 32. It is clear from the third chart that, even though full-batch approach seemed to find a better optimum when the number of hidden units was small, the stochastic approach was clearly superior for larger hidden layers.

Finally, we compared the behavior of Algorithm 2 with different number of mini-batches used for each epoch (equivalently, the number of steps between the kernel matrix updates) and different learning rates. The results are give on the last chart. As expected, there is a fine balance between the number of mini-batches and the learning rate: if the number of mini-batches is large (*i.e.*, the delay between the kernel updates becomes too long) while the learning rate is high enough, optimization does not converge; at the same time, an appropriate combination of the learning rate and the mini-batch size leads better generalization than the default batch approach of Wilson et al. (2016a).

Table 3: Average performance of the models in terms of RMSE on the system identification tasks. The averages were computed over 5 runs; the standard deviation is given in the parenthesis. Results for the RGP model are as reported by Mattos et al. (2015), available only for the free simulation.

	Drives			**Actuator**		
	regression	auto-regression	free simulation	regression	auto-regression	free simulation
NARX	0.33 (0.02)	0.19 (0.03)	0.38 (0.03)	0.49 (0.05)	0.18 (0.01)	0.57 (0.04)
RNN	0.53 (0.02)	0.17 (0.04)	0.56 (0.03)	0.56 (0.03)	0.17 (0.01)	0.68 (0.05)
LSTM	0.29 (0.02)	**0.14** (0.02)	0.40 (0.02)	0.40 (0.03)	0.19 (0.01)	0.44 (0.03)
GP-NARX	0.28 (0.02)	0.16 (0.04)	0.28 (0.02)	0.46 (0.03)	**0.14** (0.01)	0.63 (0.04)
GP-RNN	0.37 (0.04)	0.16 (0.03)	0.45 (0.03)	0.49 (0.02)	**0.15** (0.01)	0.55 (0.04)
GP-LSTM	**0.25** (0.02)	**0.13** (0.02)	0.32 (0.03)	0.36 (0.01)	**0.14** (0.01)	0.43 (0.03)
RGP	—	—	**0.249**	—	—	**0.368**

5.2.2 Regression, Auto-regression, and Free Simulation

In this set of experiments, our main goal is to provide a comparison between three different modes of one-step-ahead prediction, referred to as (i) regression, (ii) autoregression, and (iii) free simulation, and compare performance of our models with RGP—a classical RNN with every parametric layer substituted with a Gaussian process (Mattos et al., 2015)—on the Actuator and Drives datasets. The difference between the prediction modes consists in whether and how the information about the past targets is used. In the regression scenario, inputs and targets are separate time series and the model learns to map input values at a number of past time points to a target value at a future point in time. Autoregression, additionally, uses the *true* past target values as inputs; in the free simulation mode, the model learns to map past inputs and its own past predictions to a future target.

In the experiments in autoregression and free simulation modes, we used short time lags, $L = 10$, as suggested by Mattos et al. (2015). In the regression mode, since the model does not build the recurrent relationships based on the information about the targets (or their estimates), it generally requires larger time lags that can capture the state of the dynamics. Hence we increased the time lag to 32 in the regression mode. More details are given in Appendix E.

We present the results in Table 3. We note that GP-based architectures consistently yielded improved predictive performance compared to their vanilla deep learning counterparts on both of the datasets, in each mode. Given the small size of the datasets, we attribute such behavior to better regularization properties of the negative log marginal likelihood loss function. We also found out that when GP-based models were initialized with weights of pre-trained neural networks, they tended to overfit and give overly confident predictions on these tasks. The best performance was achieved when the models were trained from a random initialization (contrary to the findings of Wilson et al., 2016a). In free simulation mode RGP performs best of the compared models. This result is expected—RGP was particularly designed to represent and propagate uncertainty through a recurrent process.

Table 4: Average performance of the best models in terms of RMSE on the GEF and Car tasks. The averages were computed over 5 runs; the standard deviation is given in the parenthesis.

	GEF		Car			
	power load	wind power	speed	gyro yaw	lanes	lead vehicle
NARX	0.54 (0.02)	0.84 (0.01)	0.114 (0.010)	0.19 (0.01)	0.13 (0.01)	0.41 (0.02)
RNN	0.61 (0.02)	0.81 (0.01)	0.152 (0.012)	0.22 (0.01)	0.33 (0.02)	0.44 (0.03)
LSTM	0.45 (0.01)	**0.77** (0.01)	0.027 (0.008)	0.13 (0.01)	0.08 (0.01)	0.40 (0.01)
GP-NARX	0.78 (0.03)	0.83 (0.02)	0.125 (0.015)	0.23 (0.02)	0.10 (0.01)	0.34 (0.02)
GP-RNN	0.24 (0.02)	0.79 (0.01)	0.089 (0.013)	0.24 (0.01)	0.46 (0.08)	0.41 (0.02)
GP-LSTM	**0.17** (0.02)	**0.76** (0.01)	**0.019** (0.006)	**0.08** (0.01)	**0.06** (0.01)	**0.32** (0.02)

Our framework focuses on using recurrence to build expressive kernels for regression on sequences.

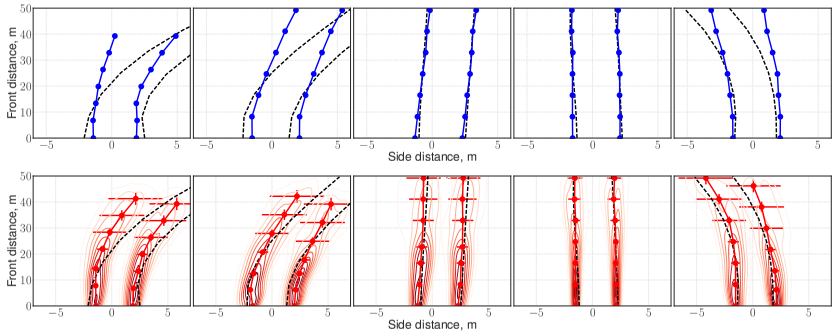
The suitability of each prediction mode depends on the task at hand. In many applications where the future targets become readily available as the time passes (e.g., power estimation or stock market prediction), the autoregression mode is preferable. We particularly consider autoregressive prediction in the further experiments.

5.2.3 Prediction for smart grid and self-driving car applications

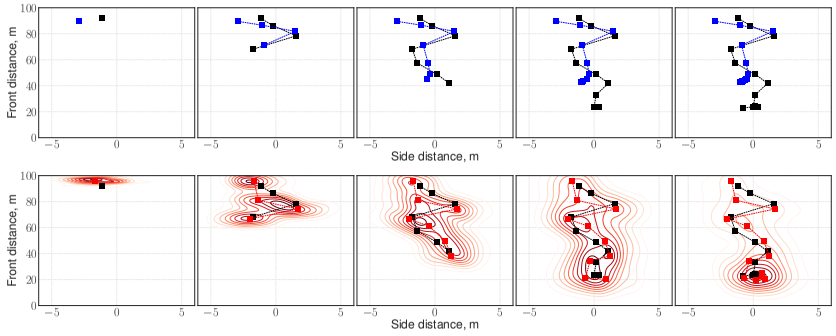
For both smart grid prediction tasks we used LSTM and GP-LSTM models with 48 hour time lags and were predicting the target values one hour ahead. LSTM and GP-LSTM were trained with one or two layers and 32 to 256 hidden units. The best models were selected on 25% of the training data used for validation. For autonomous driving prediction tasks, we used the same architectures but with 128 time steps of lag (1.28 s). All models were regularized with dropout (Srivastava et al., 2014; Gal and Ghahramani, 2016b). On both GEF and self-driving car datasets, we used the scalable version of Gaussian process (MSGP) (Wilson et al., 2015). Given the scale of the data and the challenge of nonlinear optimization of the recurrent models, we initialized the recurrent parts of GP-RNN and GP-LSTM with pre-trained weights of the corresponding neural networks. Fine-tuning of the models was performed with Algorithm 2. The quantitative results are provided in Table 4 and demonstrate that GPs with recurrent kernels attain the state-of-the-art performance.

Additionally, we investigated convergence and regularization properties of LSTM and GP-LSTM models on the GEF-power dataset. The first two charts of Figure 6 demonstrate that GP-based models are less prone to overfitting, even when the data is not enough. The third panel shows that architectures with a particular number of hidden units per layer attain the best performance on the power prediction task. An additional advantage of the GP-layers over the standard recurrent networks is that the best architecture could be identified based on the negative log likelihood of the model as shown on the last chart.

Finally, Figure 5 qualitatively demonstrates the difference between the predictions given by LSTM vs. GP-LSTM on point-wise lane estimation (Figure 5a) and the front vehicle



(a) Point-wise predictions of the lanes made by LSTM (upper) and by GP-LSTM (lower). Dashed lines correspond to the ground truth extracted from video sequences and used for training.



(b) LSTM (upper) and by GP-LSTM (lower) position predictions of the lead vehicle. Black markers and dashed lines are the ground truth; blue and red markers with solid lines correspond to predictions.

Figure 5: Qualitative comparison of the LSTM and GP-LSTM predictions on self-driving tasks. Predictive uncertainty of the GP-LSTM model is showed by contour plots and error-bars; the latter denote one standard deviation of the predictive distributions.

tracking (Figure 5b) tasks. We note that GP-LSTM not only provides a more robust fit, but also estimates the uncertainty of its predictions. Such information can be further used in downstream prediction-based decision making, *e.g.*, such as whether a self-driving car should slow down and switch to a more cautious driving style when the uncertainty is high.

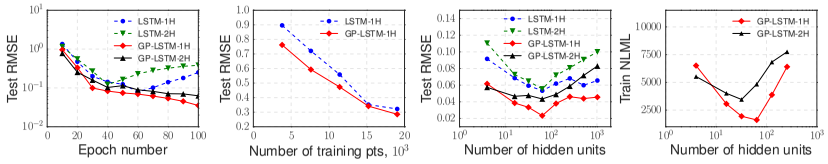


Figure 6: *Left to right*: RMSE vs. the number of training points; RMSE vs. the number model parameters per layer; NLML vs. the number model parameters per layer for GP-based models. All metrics are averages over 5 runs with different random initializations, computed on a held-out set.

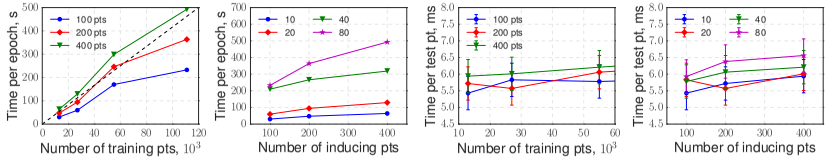


Figure 7: The charts demonstrate scalability of learning and inference of MSGP with an LSTM-based recurrent kernel. Legends with points denote the number of inducing points used. Legends with percentages denote the percentage of the training dataset used learning the model.

5.2.4 Scalability of the model

Following Wilson et al. (2015), we performed a generic scalability analysis of the MSGP-LSTM model on the car sensors data. The LSTM architecture was the same as described in the previous section: it was transforming multi-dimensional sequences of inputs to a two-dimensional representation. We trained the model for 10 epochs on 10%, 20%, 40%, and 80% of the training set with 100, 200, and 400 inducing points per dimension and measured the average training time per epoch and the average prediction time per testing point. The measured time was the total time spent on both LSTM optimization and MSGP computations. The results are presented in Figure 7.

The training time per epoch (one full pass through the entire training data) grows linearly with the number of training examples and depends linearly on the number of inducing points (Figure 7, two left charts). Thus, given a fixed number of inducing points per dimension, the time complexity of MSGP-LSTM learning and inference procedures is linear in the number of training examples. The prediction time per testing data point is virtually constant and does not depend on neither on the number of training points, nor on the number of inducing points (Figure 7, two right charts).

6. Discussion

We proposed a method for learning kernels with recurrent long short-term memory structure on sequences. Gaussian processes with such kernels, termed the GP-LSTM, have the structure and learning biases of LSTMs, while retaining a probabilistic Bayesian nonparametric representation. The GP-LSTM outperforms a range of alternatives on several sequence-to-reals regression tasks. The GP-LSTM also works on data with low and high signal-to-noise ratios, and can be scaled to very large datasets, all with a straightforward, practical, and generally applicable model specification. Moreover, the semi-stochastic scheme proposed in our paper is provably convergent and efficient in practical settings, in conjunction with structure exploiting algebra. In short, the GP-LSTM provides a natural mechanism for Bayesian LSTMs, quantifying predictive uncertainty while harmonizing with the standard deep learning toolbox. Predictive uncertainty is of high value in robotics applications, such as autonomous driving, and could also be applied to other areas such as financial modeling and computational biology.

There are several exciting directions for future research. The GP-LSTM quantifies predictive uncertainty but does not model the propagation of uncertainty in the inputs through a recurrent structure. Treating free simulation as a structured prediction problem and using online corrective algorithms, e.g., DAGGER (Ross et al., 2011), are likely to improve performance of GP-LSTM in the free prediction mode. This approach would not require explicitly modeling and propagating uncertainty through the recurrence and would maintain the high computational efficiency of our method.

Alternatively, it would be exciting to have a probabilistic treatment of all parameters of the GP-LSTM kernel, including all LSTM weights. Such an extension could be combined with stochastic variational inference, to enable both classification and non-Gaussian likelihoods as in Wilson et al. (2016b), but also open the doors to stochastic gradient Hamiltonian Monte Carlo (Chen et al., 2014) (SG-HMC) for efficient inference over kernel parameters. Indeed, SG-HMC has recently been used for efficient inference over network parameters in the Bayesian GAN (Saatchi and Wilson, 2017). A Bayesian approach to marginalizing the weights of the GP-LSTM kernel would also provide a principled probabilistic mechanism for learning model hyperparameters.

One could relax several additional assumptions. We modeled each output dimension with independent GPs that shared a recurrent transformation. To capture the correlations between output dimensions, it would be promising to move to a multi-task formulation. In the future, one could also learn the time horizon in the recurrent transformation, which could lead to major additional performance gains.

Finally, the semi-stochastic learning procedure naturally complements research in asynchronous optimization (e.g., Deisenroth and Ng, 2015). In combination with stochastic variational inference, the semi-stochastic approach could be used for parallel kernel learning, side-stepping the independence assumptions in prior work. We envision that such efforts for Gaussian processes will harmonize with current progress in Bayesian deep learning.

7. Acknowledgements

The authors thank Yifei Ma for helpful discussions and the anonymous reviewers for the valuable comments that helped to improve the paper. This work was supported in part by NIH R01GM114311, AFRL/DARPA FA87501220324, and NSF IIS-1563887.

Appendix A. Massively scalable Gaussian processes

Massively scalable Gaussian processes (MSGP) (Wilson et al., 2015) is a significant extension of the kernel interpolation framework originally proposed by Wilson and Nickisch (2015). The core idea of the framework is to improve scalability of the inducing point methods (Quinonero-Candela and Rasmussen, 2005) by (1) placing the virtual points on a regular grid, (2) exploiting the resulting Kronecker and Toeplitz structures of the relevant covariance matrices, and (3) do local cubic interpolation to go back to the kernel evaluated at the original points. This combination of techniques brings the complexity down to $\mathcal{O}(n)$ for training and $\mathcal{O}(1)$ for each test prediction. Below, we overview the methodology. We remark that a major difference in philosophy between MSGP and many classical inducing point methods is that the points are *selected and fixed* rather than optimized over. This allows to use significantly more virtual points which typically results in a better approximation of the true kernel.

A.1 Structured kernel interpolation

Given a set of m inducing points, the $n \times m$ cross-covariance matrix, $K_{X,U}$, between the training inputs, X, and the inducing points, $\mathbf{U}$, can be approximated as $\tilde{K}_{X,U} = W_X K_{U,U}$ using a (potentially sparse) $n \times m$ matrix of interpolation weights, W_X. This allows to approximate $K_{\mathbf{X},\mathbf{Z}}$ for an arbitrary set of inputs $\mathbf{Z}$ as $K_{\mathbf{X},\mathbf{Z}} \approx \tilde{K}_{X,U} W_{\mathbf{Z}}^\top$. For any given kernel function, K, and a set of inducing points, $\mathbf{U}$, *structured kernel interpolation* (SKI) procedure (Wilson and Nickisch, 2015) gives rise to the following approximate kernel:

$$K_{\text{SKI}}(\mathbf{x}, \mathbf{z}) = W_X K_{U,U} W_z^\top, \tag{A.1}$$

which allows to approximate $K_{X,X} \approx W_X K_{U,U} W_X^\top$. Wilson and Nickisch (2015) note that standard inducing point approaches, such as subset of regression (SoR) or fully independent training conditional (FITC), can be reinterpreted from the SKI perspective. Importantly, the efficiency of SKI-based MSGP methods comes from, first, a clever choice of a set of inducing points that allows to exploit algebraic structure of $K_{U,U}$, and second, from using very sparse local interpolation matrices. In practice, local cubic interpolation is used (Keys, 1981).

A.2 Kernel approximations

If inducing points, U, form a regularly spaced P-dimensional grid, and we use a stationary product kernel (e.g., the RBF kernel), then $K_{U,U}$ decomposes as a Kronecker product of Toeplitz matrices:

$$K_{U,U} = \mathbf{T}_1 \otimes \mathbf{T}_2 \otimes \cdots \otimes \mathbf{T}_P. \tag{A.2}$$

The Kronecker structure allows to compute the eigendecomposition of $K_{U,U}$ by separately decomposing $\mathbf{T}_1, \ldots, \mathbf{T}_P$, each of which is much smaller than $K_{U,U}$. Further, to efficiently

eigendecompose a Toeplitz matrix, it can be approximated by a circulant matrix[13] which eigendecomposes by simply applying discrete Fourier transform (DFT) to its first column. Therefore, an approximate eigendecomposition of each $\mathbf{T}_1, \ldots, \mathbf{T}_P$ is computed via the fast Fourier transform (FFT) and requires only $\mathcal{O}(m \log m)$ time.

A.3 Structure exploiting inference

To perform inference, we need to solve $(K_{\text{SKI}} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$; kernel learning requires evaluating $\log\det(K_{\text{SKI}} + \sigma^2\mathbf{I})$. The first task can be accomplished by using an iterative scheme—linear conjugate gradients—which depends only on matrix vector multiplications with $(K_{\text{SKI}} + \sigma^2\mathbf{I})$. The second is done by exploiting the Kronecker and Toeplitz structure of $K_{U,U}$ for computing an approximate eigendecomposition, as described above.

A.4 Fast Test Predictions

To achieve constant time prediction, we approximate the latent mean and variance of $\mathbf{f}_*$ by applying the same SKI technique. In particular, for a set of n_* testing points, X_*, we have

$$\begin{aligned} \mathbb{E}[\mathbf{f}_*] &= \boldsymbol{\mu}_{X_*} + K_{\mathbf{X}_*,\mathbf{X}}\left[K_{X,X} + \sigma^2 I\right]^{-1}\mathbf{y} \\ &\approx \boldsymbol{\mu}_{X_*} + \tilde{K}_{\mathbf{X}_*,\mathbf{X}}\left[\tilde{K}_{X,X} + \sigma^2 I\right]^{-1}\mathbf{y}, \end{aligned} \tag{A.3}$$

where $\tilde{K}_{X,X} = W K_{U,U} W^\top$ and $\tilde{K}_{\mathbf{X}_*,\mathbf{X}} = W_* K_{U,U} W^\top$, and W and W_* are $n \times m$ and $n_* \times m$ sparse interpolation matrices, respectively. Since $K_{U,U}W^\top[\tilde{K}_{X,X} + \sigma^2\mathbf{I}]^{-1}\mathbf{y}$ is precomputed at training time, at test time, we only multiply the latter with W_* matrix which results which costs $\mathcal{O}(n_*)$ operations leading to $\mathcal{O}(1)$ operations per test point. Similarly, approximate predictive variance can be also estimated in $\mathcal{O}(1)$ operations (Wilson et al., 2015).

Note that the fast prediction methodology can be readily applied to any trained Gaussian process model as it is agnostic to the way inference and learning were performed.

Appendix B. Gradients for GPs with recurrent kernels

GPs with deep recurrent kernels are trained by minimizing the negative log marginal likelihood objective function. Below we derive the update rules.

By applying the chain rule, we get the following first order derivatives:

$$\frac{\partial\mathcal{L}}{\partial\gamma} = \frac{\partial\mathcal{L}}{\partial K}\cdot\frac{\partial K}{\partial\boldsymbol{\gamma}}, \quad \frac{\partial\mathcal{L}}{\partial W} = \frac{\partial\mathcal{L}}{\partial K}\cdot\frac{\partial K}{\partial\boldsymbol{\phi}}\cdot\frac{\partial\boldsymbol{\phi}}{\partial W}. \tag{B.4}$$

The derivative of the log marginal likelihood w.r.t. to the kernel hyperparameters, $\boldsymbol{\theta}$, and the parameters of the recurrent map, W, are generic and take the following form (Rasmussen and Williams, 2006, Ch. 5, Eq. 5.9):

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}} = \tfrac{1}{2}\mathbf{tr}\left(\left[K^{-1}\mathbf{y}\mathbf{y}^\top K^{-1} - K^{-1}\right]\tfrac{\partial K}{\partial\boldsymbol{\theta}}\right), \tag{B.5}$$

$$\frac{\partial\mathcal{L}}{\partial W} = \tfrac{1}{2}\mathbf{tr}\left(\left[K^{-1}\mathbf{y}\mathbf{y}^\top K^{-1} - K^{-1}\right]\tfrac{\partial K}{\partial W}\right). \tag{B.6}$$

The derivative $\partial K/\partial\boldsymbol{\theta}$ is also standard and depends on the form of a particular chosen kernel function, $k(\cdot,\cdot)$. However, computing each part of $\partial\mathcal{L}/\partial W$ is a bit subtle, and hence we elaborate these derivations below.

Consider the ij-th entry of the kernel matrix, K_{ij}. We can think of K as a matrix-valued function of all the data vectors in d-dimensional transformed space which we denote by $H \in \mathbb{R}^{N\times d}$. Then K_{ij} is a scalar-valued function of H and its derivative w.r.t. the l-th parameter of the recurrent map, W_l, can be written as follows:

$$\frac{\partial K_{ij}}{\partial W_l} = \mathbf{tr}\left(\left(\frac{\partial K_{ij}}{\partial H}\right)^\top\frac{\partial H}{\partial W_l}\right). \tag{B.7}$$

Notice that $\partial K_{ij}/\partial H$ is a derivative of a scalar w.r.t. to a matrix and hence is a matrix; $\partial H/\partial W_l$ is a derivative of a matrix w.r.t. to a scalar which is taken element-wise and also gives a matrix. Also notice that K_{ij} is a function of H, but it only depends the i-th and j-th elements for which the kernel is being computed. This means that $\partial K_{ij}/\partial H$ will have only non-zero i-th row and j-th column and allows us to re-write (B.7) as follows:

$$\begin{aligned} \frac{\partial K_{ij}}{\partial W_l} &= \left(\frac{\partial K_{ij}}{\partial\mathbf{h}_i}\right)^\top\frac{\partial\mathbf{h}_i}{\partial W_l} + \left(\frac{\partial K_{ij}}{\partial\mathbf{h}_j}\right)^\top\frac{\partial\mathbf{h}_j}{\partial W_l} \\ &= \left(\frac{\partial K(\mathbf{h}_i,\mathbf{h}_j)}{\partial\mathbf{h}_i}\right)^\top\frac{\partial\mathbf{h}_i}{\partial W_l} + \left(\frac{\partial K(\mathbf{h}_i,\mathbf{h}_j)}{\partial\mathbf{h}_j}\right)^\top\frac{\partial\mathbf{h}_j}{\partial W_l}. \end{aligned} \tag{B.8}$$

Since the kernel function has two arguments, the derivatives must be taken with respect of each of them and evaluated at the corresponding points in the hidden space, $\mathbf{h}_i = \boldsymbol{\phi}(\mathbf{x}_i)$ and $\mathbf{h}_j = \boldsymbol{\phi}(\mathbf{x}_j)$. When we plug this into (B.6), we arrive at the following expression:

$$\frac{\partial\mathcal{L}}{\partial W_l} = \frac{1}{2}\sum_{i,j}\left(K^{-1}\mathbf{y}\mathbf{y}^\top K^{-1} - K^{-1}\right)_{ij}\left\{\left(\frac{\partial K(\mathbf{h}_i,\mathbf{h}_j)}{\partial\mathbf{h}_i}\right)^\top\frac{\partial\mathbf{h}_i}{\partial W_l} + \left(\frac{\partial K(\mathbf{h}_i,\mathbf{h}_j)}{\partial\mathbf{h}_j}\right)^\top\frac{\partial\mathbf{h}_j}{\partial W_l}\right\}. \tag{B.9}$$

The same expression can be written in a more compact form using the *Einstein notation*:

$$\frac{\partial\mathcal{L}}{\partial W_l} = \frac{1}{2}\left(K^{-1}\mathbf{y}\mathbf{y}^\top K^{-1} - K^{-1}\right)_i^j\left(\left[\frac{\partial K}{\partial\mathbf{h}}\right]_i^{jd} + \left[\frac{\partial K}{\partial\mathbf{h}}\right]_j^{id}\right)\left[\frac{\partial\mathbf{h}}{\partial W}\right]_i^{dl} \tag{B.10}$$

where d indexes the dimensions of the $\mathbf{h}$ and l indexes the dimensions of W.

13. Wilson et al. (2015) explored 5 different approximation methods known in the numerical analysis literature.

In practice, deriving a computationally efficient analytical form of $\partial K/\partial \mathbf{h}$ might be too complicated for some kernels (*e.g.*, the spectral mixture kernels (Wilson and Adams, 2013)), especially if the grid-based approximations of the kernel are enabled. In such cases, we can simply use a finite difference approximation of this derivative. As we remark in the following section, numerical errors that result from this approximation do not affect convergence of the algorithm.

Appendix C. Convergence results

Convergence results for the semi-stochastic alternating gradient schemes with and without delayed kernel matrix updates are based on (Xu and Yin, 2015). There are a few notable differences between the original setting and the one considered in this paper:

1. Xu and Yin (2015) consider a stochastic program that minimizes the expectation of the objective w.r.t. some distribution underlying the data:

$$\min_{x\in\mathcal{X}} f(x) := \mathbb{E}_\xi F(x;\xi), \tag{C.11}$$

   where every iteration a new ξ is sampled from the underlying distribution. In our case, the goal is to minimize the negative log marginal likelihood on a particular given dataset. This is equivalent to the original formulation (C.11), but with the expectation taken w.r.t. the empirical distribution that corresponds to the given dataset.

2. The optimization procedure of Xu and Yin (2015) has access to only a single random point generated from the data distribution at each step. Our algorithm requires having access to the entire training data each time the kernel matrix is computed.

3. For a given sample, Xu and Yin (2015) propose to loop over a number of coordinate blocks and apply Gauss–Seidel type gradient updates to each block. Our semi-stochastic scheme has only two parameter blocks, $\boldsymbol{\theta}$ and W, where $\boldsymbol{\theta}$ is updated deterministically on the entire dataset while W is updated with stochastic gradient on samples from the empirical distribution.

Noting these differences, we first adapt convergence results for the smooth non-convex case (Xu and Yin, 2015, Theorem 2.10) to our scenario, and then consider the variant with delaying kernel matrix updates.

C.1 Semi-stochastic alternating gradient

As shown in Algorithm 1, we alternate between updating $\boldsymbol{\theta}$ and W. At step t, we get a mini-batch of size N_t, $\mathbf{x}_t \equiv \{\bar{\mathbf{x}}_i\}_{i\in\mathcal{I}_t}$, which is just a selection of points from the full set, $\mathbf{x}$. Define the gradient errors for $\boldsymbol{\theta}$ and W at step t as follows:

$$\delta^t_{\boldsymbol{\theta}} := \tilde{\mathbf{g}}^t_{\boldsymbol{\theta}} - \mathbf{g}^t_{\boldsymbol{\theta}}, \quad \delta^t_W := \tilde{\mathbf{g}}^t_W - \mathbf{g}^t_W, \tag{C.12}$$

where $\mathbf{g}^{\mathbf{T}}_{\boldsymbol{\theta}}$ and $\mathbf{g}^{\mathbf{T}}_W$ are the true gradients and $\tilde{\mathbf{g}}^{\mathbf{T}}_{\boldsymbol{\theta}}$ and $\tilde{\mathbf{g}}^{\mathbf{T}}_W$ are estimates[14]. We first update $\boldsymbol{\theta}$ and then W, and hence the expressions for the gradients take the following form:

$$\tilde{\mathbf{g}}^t_{\boldsymbol{\theta}} \equiv \mathbf{g}^t_{\boldsymbol{\theta}} = \frac{1}{N}\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^t, W^t) = \frac{1}{2N}\sum_{i,j}\left(K_t^{-1}\mathbf{y}\mathbf{y}^\top K_t^{-1} - K_t^{-1}\right)_{ij}\left(\frac{\partial K_t}{\partial \boldsymbol{\theta}}\right)_{ij} \tag{C.13}$$

$$\mathbf{g}^t_W = \frac{1}{N}\nabla_W\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^t) = \frac{1}{2N}\sum_{i,j}\left(K_{t+1}^{-1}\mathbf{y}\mathbf{y}^\top K_{t+1}^{-1} - K_{t+1}^{-1}\right)_{ij}\left(\frac{\partial K_{t+1}}{\partial W}\right)_{ij} \tag{C.14}$$

$$\tilde{\mathbf{g}}^t_W = \frac{1}{2N_t}\sum_{i,j\in\mathcal{I}_t}\left(K_{t+1}^{-1}\mathbf{y}\mathbf{y}^\top K_{t+1}^{-1} - K_{t+1}^{-1}\right)_{ij}\left(\frac{\partial K_{t+1}}{\partial W}\right)_{ij} \tag{C.15}$$

Note that as we have shown in Appendix B, when the kernel matrix is fixed, $\mathbf{g}_W$ and $\tilde{\mathbf{g}}_W$ factorize over $\mathbf{x}$ and $\mathbf{x}_t$, respectively. Further, we denote all the mini-batches sampled before t as $\mathbf{x}_{[t-1]}$.

Lemma 1 *For any step t, $\mathbb{E}[\delta^t_W \mid \mathbf{x}_{[t-1]}] = \mathbb{E}[\delta^t_{\boldsymbol{\theta}} \mid \mathbf{x}_{[t-1]}] = 0$.*

Proof First, $\delta^t_{\boldsymbol{\theta}} \equiv 0$, and hence $\mathbb{E}[\delta^t_{\boldsymbol{\theta}} \mid \mathbf{x}_{[t-1]}] = 0$ is trivial. Next, by definition of δ^t_W, we have $\mathbb{E}[\delta^t_W \mid \mathbf{x}_{[t-1]}] = \mathbb{E}[\tilde{\mathbf{g}}^t_W - \mathbf{g}^t_W \mid \mathbf{x}_{[t-1]}]$. Consider the following:

- Consider $\mathbb{E}[\mathbf{g}^t_W \mid \mathbf{x}_{[t-1]}]$: $\mathbf{g}^t_W$ is a deterministic function of $\boldsymbol{\theta}^{t+1}$ and W^t. $\boldsymbol{\theta}^{t+1}$ is being updated deterministically using $\mathbf{g}^{t-1}_{\boldsymbol{\theta}}$, and hence only depends on W^t and $\boldsymbol{\theta}^t$. Therefore, it is independent of $\mathbf{x}_t$, which means that $\mathbb{E}[\mathbf{g}^t_W \mid \mathbf{x}_{[t-1]}] \equiv \mathbf{g}^t_W$.

- Now, consider $\mathbb{E}[\tilde{\mathbf{g}}^t_W \mid \mathbf{x}_{[t-1]}]$: Noting that the expectation is taken w.r.t. the empirical distribution and K_{t+1} does not depend on the current mini-batch, we can write:

$$\begin{aligned}\mathbb{E}[\tilde{\mathbf{g}}^t_W \mid \mathbf{x}_{[t-1]}] &= \mathbb{E}\left[\frac{1}{2N_t}\sum_{i,j\in\mathcal{I}_t}\left(K_{t+1}^{-1}\mathbf{y}\mathbf{y}^\top K_{t+1}^{-1} - K_{t+1}^{-1}\right)_{ij}\left(\frac{\partial K_{t+1}}{\partial W}\right)_{ij}\right] \\ &= \frac{N_t}{2NN_t}\sum_{i,j}\left(K_{t+1}^{-1}\mathbf{y}\mathbf{y}^\top K_{t+1}^{-1} - K_{t+1}^{-1}\right)_{ij}\left(\frac{\partial K_{t+1}}{\partial W}\right)_{ij} \\ &\equiv \mathbf{g}^t_W.\end{aligned} \tag{C.16}$$

Finally, $\mathbb{E}[\delta^t_W \mid \mathbf{x}_{[t-1]}] = \mathbf{g}^t_W - \mathbf{g}^t_W = 0$. ■

In other words, semi-stochastic gradient descent that alternates between updating $\boldsymbol{\theta}$ and W computes unbiased estimates of the gradients on each step.

14. Here, we consider the gradients and their estimates scaled by the number of full data points, N, and the mini-batch size, N_t, respectively. These constant scaling is introduced for sake of having cleaner proofs.

Remark 1 *Note that in case if $(\partial K_{t+1}/\partial W)$ is computed approximately, Lemma 1 still holds since both $\mathbf{g}_W^t$ and $\mathbb{E}[\tilde{\mathbf{g}}_W^t \mid \mathbf{x}_{[t-1]}]$ will contain exactly the same numerical errors.*

Assumption 1 *For any step t, $\mathbb{E}\|\delta_{\boldsymbol{\theta}}^t\|^2 \leq \sigma_t^2$ and $\mathbb{E}\|\delta_W^t\|^2 \leq \sigma_t^2$.*

Lemma 1 and Assumption 1 result into a stronger condition than the original assumption given by Xu and Yin (2015). This is due to the semi-stochastic nature of the algorithm, it simplifies the analysis, though it is not critical. Assumptions 2 and 3 are straightforwardly adapted from the original paper.

Assumption 2 *The objective function, $\mathcal{L}$, is lower bounded and its partial derivatives w.r.t. $\boldsymbol{\theta}$ and W are uniformly Lipschitz with constant $L > 0$:*

$$\|\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, W) - \nabla_{\boldsymbol{\theta}}\mathcal{L}(\tilde{\boldsymbol{\theta}}, W)\| \leq L\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|, \quad \|\nabla_W\mathcal{L}(\boldsymbol{\theta}, W) - \nabla_W\mathcal{L}(\boldsymbol{\theta}, \tilde{W})\| \leq L\|W - \tilde{W}\|. \tag{C.17}$$

Assumption 3 *There exists a constant ρ such that $\|\boldsymbol{\theta}^t\|^2 + \mathbb{E}\|W^t\|^2 \leq \rho^2$ for all t.*

Lemma 2 *Under Assumptions 2 and 3,*

$$\mathbb{E}\|W^t\|^2 \leq \rho^2, \quad \mathbb{E}\|\boldsymbol{\theta}^t\|^2 \leq \rho^2, \quad \mathbb{E}\|\mathbf{g}_W^t\|^2 \leq M_\rho^2, \quad \mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2 \leq M_\rho^2 \quad \forall t,$$

where $M_\rho^2 = 4L^2\rho^2 + 2\max\{\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^0, W^0), \nabla_W\mathcal{L}(\boldsymbol{\theta}^0, W^0)\}$.

Proof The inequalities for $\mathbb{E}\|\boldsymbol{\theta}\|^2$ and $\mathbb{E}\|W\|^2$ are trivial, and the ones for $\mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2$ and $\mathbb{E}\|\mathbf{g}_W^t\|^2$ are merely corollaries from Assumption 2. ■

Negative log marginal likelihood of a Gaussian process with a structured kernel is a nonconvex function of its arguments. Therefore, we can only show that the algorithm converges to a stationary point, *i.e.*, a point at which the gradient of the objective is zero.

Theorem 1 *Let $\{(\boldsymbol{\theta}^t, W^t)\}$ be a sequence generated from Algorithm 1 with learning rates for $\boldsymbol{\theta}$ and W being $\lambda_{\boldsymbol{\theta}}^t = c_{\boldsymbol{\theta}}^t\alpha_t$ and $\lambda_W^t = c_W^t\alpha_t$, respectively, where $c_{\boldsymbol{\theta}}^t, c_W^t, \alpha_t$ are positive scalars, such that*

$$0 < \inf_t c_{\{\boldsymbol{\theta},W\}}^t \leq \sup_t c_{\{\boldsymbol{\theta},W\}}^t < \infty, \quad \alpha_t \geq \alpha_{t+1}, \quad \sum_{t=1}^{\infty}\alpha_t = +\infty, \quad \sum_{t=1}^{\infty}\alpha_t^2 < +\infty, \quad \forall t. \tag{C.18}$$

Then, under Assumptions 1 through 3 and if $\sigma = \sup_t \sigma_t < \infty$, we have

$$\lim_{t\to\infty}\mathbb{E}\|\nabla\mathcal{L}(\boldsymbol{\theta}^t, W^t)\| = 0. \tag{C.19}$$

Proof The proof is an adaptation of the one given by Xu and Yin (2015) with the following three adjustments: (1) we have only two blocks of coordinates and (2) updates for $\boldsymbol{\theta}$ are deterministic which zeros out their variance terms, (3) the stochastic gradients are unbiased.

- From the Lipschitz continuity of $\nabla_W\mathcal{L}$ and $\nabla_{\boldsymbol{\theta}}\mathcal{L}$ (Assumption 2), we have:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^{t+1}) - \mathcal{L}(\boldsymbol{\theta}^{t+1}, W^t) \\
\leq\ & \langle \mathbf{g}_W^t, W^{t+1} - W^t\rangle + \frac{L}{2}\|W^{t+1} - W^t\|^2 \\
=\ & -\lambda_W^t\langle \mathbf{g}_W^t, \tilde{\mathbf{g}}_W^t\rangle + \frac{L}{2}(\lambda_W^t)^2\|\tilde{\mathbf{g}}_W^t\|^2 \\
=\ & -\left(\lambda_W^t - \frac{L}{2}(\lambda_W^t)^2\right)\|\mathbf{g}_W^t\|^2 + \frac{L}{2}(\lambda_W^t)^2\|\delta_W^t\|^2 - \left(\lambda_W^t - L(\lambda_W^t)^2\right)\langle \mathbf{g}_W^t, \delta_W^t\rangle \\
=\ & -\left(\lambda_W^t - \frac{L}{2}(\lambda_W^t)^2\right)\|\mathbf{g}_W^t\|^2 + \frac{L}{2}(\lambda_W^t)^2\|\delta_W^t\|^2 \\
& -\left(\lambda_W^t - L(\lambda_W^t)^2\right)\left(\langle \mathbf{g}_W^t - \nabla_W\mathcal{L}(\boldsymbol{\theta}^t, W^t), \delta_W^t\rangle + \langle \nabla_W\mathcal{L}(\boldsymbol{\theta}^t, W^t), \delta_W^t\rangle\right) \\
\leq\ & -\left(\lambda_W^t - \frac{L}{2}(\lambda_W^t)^2\right)\|\mathbf{g}_W^t\|^2 + \frac{L}{2}(\lambda_W^t)^2\|\delta_W^t\|^2 - \left(\lambda_W^t - L(\lambda_W^t)^2\right)\langle \nabla_W\mathcal{L}(\boldsymbol{\theta}^t, W^t), \delta_W^t\rangle \\
& +\frac{L}{2}\lambda_{\boldsymbol{\theta}}^t\left(\lambda_W^t + L(\lambda_W^t)^2\right)\left(\|\delta_W^t\|^2 + \|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2\right),
\end{aligned}$$

  where the last inequality comes from the following (note that $\mathbf{g}_W^t := \nabla_W\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^t)$):

$$\begin{aligned}
& -\left(\lambda_W^t - L(\lambda_W^t)^2\right)\langle \nabla_W\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^t) - \nabla_W\mathcal{L}(\boldsymbol{\theta}^t, W^t), \delta_W^t\rangle \\
\leq\ & \left|\lambda_W^t - L(\lambda_W^t)^2\right|\|\delta_W^t\|\|\nabla_W\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^t) - \nabla_W\mathcal{L}(\boldsymbol{\theta}^t, W^t)\| \\
\leq\ & L\lambda_{\boldsymbol{\theta}}^t\left|\lambda_W^t - L(\lambda_W^t)^2\right|\|\delta_W^t\|\|\mathbf{g}_{\boldsymbol{\theta}}^t\| \\
\leq\ & \frac{L}{2}\lambda_{\boldsymbol{\theta}}^t\left(\lambda_W^t + L(\lambda_W^t)^2\right)\left(\|\delta_W^t\|^2 + \|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2\right).
\end{aligned}$$

  Analogously, we derive a bound on $\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^t) - \mathcal{L}(\boldsymbol{\theta}^t, W^t)$:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^t) - \mathcal{L}(\boldsymbol{\theta}^t, W^t) \\
\leq\ & -\left(\lambda_{\boldsymbol{\theta}}^t - \frac{L}{2}(\lambda_{\boldsymbol{\theta}}^t)^2\right)\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2 + \frac{L}{2}(\lambda_{\boldsymbol{\theta}}^t)^2\|\delta_{\boldsymbol{\theta}}^t\|^2 - \left(\lambda_{\boldsymbol{\theta}}^t - L(\lambda_{\boldsymbol{\theta}}^t)^2\right)\langle \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^t, W^t), \delta_{\boldsymbol{\theta}}^t\rangle
\end{aligned}$$

- Since $(\boldsymbol{\theta}^t, W^t)$ is independent from the current mini-batch, $\mathbf{x}_t$, using Lemma 1, we have that $\mathbb{E}\langle \nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}^t, W^t), \delta_{\boldsymbol{\theta}}^t\rangle = 0$ and $\mathbb{E}\langle \nabla_W\mathcal{L}(\boldsymbol{\theta}^t, W^t), \delta_W^t\rangle = 0$.

Summing the two obtained inequalities and applying Assumption 1, we get:

$$\begin{aligned}
&\mathbb{E}[\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^{t+1}) - \mathcal{L}(\boldsymbol{\theta}^t, W^t)] \\
\leq\ & -\left(\lambda_W^t - \frac{L}{2}(\lambda_W^t)^2\right)\mathbb{E}\|\mathbf{g}_W^t\|^2 + \frac{L}{2}(\lambda_W^t)^2\sigma + \frac{L}{2}\lambda_{\boldsymbol{\theta}}^t\left(\lambda_W^t + L(\lambda_W^t)^2\right)(\sigma + \mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2) \\
& -\left(\lambda_{\boldsymbol{\theta}}^t - \frac{L}{2}(\lambda_{\boldsymbol{\theta}}^t)^2\right)\mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2 + \frac{L}{2}(\lambda_{\boldsymbol{\theta}}^t)^2\sigma \\
\leq\ & -\left(c\alpha_t - \frac{L}{2}C^2\alpha_t^2\right)\mathbb{E}\|\mathbf{g}_W^t\|^2 + LC^2\sigma\alpha_t + \frac{L}{2}C^2\alpha_t^2(1 + LC\alpha_t)(\sigma + \mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2) \\
& -\left(c\alpha_t - \frac{L}{2}C^2\alpha_t^2\right)\mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2 + \frac{L}{2}C^2\alpha_t^2\sigma,
\end{aligned}$$

where we denoted $c = \min\{\inf_t c_W^t, \inf_t c_{\boldsymbol{\theta}}^t\}$, $C = \max\{\sup_t c_W^t, \sup_t c_{\boldsymbol{\theta}}^t\}$. Note that from Lemma 2, we also have $\mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2 \leq M_\rho^2$ and $\mathbb{E}\|\mathbf{g}_W^t\|^2 \leq M_\rho^2$. Therefore, summing the right hand side of the final inequality over t and using (C.18), we have:

$$\lim_{t\to\infty}\mathbb{E}\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^{t+1}) - \mathbb{E}\mathcal{L}(\boldsymbol{\theta}^0, W^0) \leq -c\sum_{t=1}^{\infty}\alpha_t\left(\mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2 + \mathbb{E}\|\mathbf{g}_W^t\|^2\right). \tag{C.20}$$

Since the objective function is lower bounded, this effectively means:

$$\sum_{t=1}^{\infty}\alpha_t\mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2 < \infty, \quad \sum_{t=1}^{\infty}\alpha_t\mathbb{E}\|\mathbf{g}_W^t\|^2 < \infty.$$

- Finally, using Lemma 2, our assumptions and Jensen's inequality, it follows that

$$\left|\mathbb{E}\|\mathbf{g}_W^{t+1}\|^2 - \mathbb{E}\|\mathbf{g}_W^t\|^2\right| \leq 2LM_\rho C\alpha_t\sqrt{2(M_\rho^2 + \sigma^2)}.$$

According to Proposition 1.2.4 of (Bertsekas, 1999), we have $\mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\|^2 \to 0$ and $\mathbb{E}\|\mathbf{g}_W^t\|^2 \to 0$ as $t \to \infty$, and hence

$$\begin{aligned}
\mathbb{E}\|\nabla\mathcal{L}(\boldsymbol{\theta}^t, W^t)\| &\leq \mathbb{E}\|\nabla_W\mathcal{L}(\boldsymbol{\theta}^t, W^t) - \mathbf{g}_{\boldsymbol{\theta}}^t\| + \mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\| + \mathbb{E}\|\nabla_W\mathcal{L}(\boldsymbol{\theta}^t, W^t) - \mathbf{g}_{\boldsymbol{\theta}}^t\| + \mathbb{E}\|\mathbf{g}_W^t\| \\
&\leq 2LC\sqrt{2(M_\rho^2 + \sigma^2)}\alpha + \mathbb{E}\|\mathbf{g}_{\boldsymbol{\theta}}^t\| + \mathbb{E}\|\mathbf{g}_W^t\| \to 0 \text{ as } t \to \infty,
\end{aligned}$$

where the first term of the last inequality follows from Lemma 2 and Jensen's inequality. ■

C.2 Semi-stochastic gradient with delayed kernel matrix updates

We show that given a bounded delay on the kernel matrix updates, the algorithm is still convergent. Our analysis is based on computing the change in $\delta_{\boldsymbol{\theta}}^t$ and δ_W^t and applying the same argument as in Theorem 1. The only difference is that we need to take into account the perturbations of the kernel matrix due to the introduced delays, and hence we have to impose certain assumptions on its spectrum.

Assumption 4 *Recurrent transformations, $\boldsymbol{\phi}_W(\bar{\mathbf{x}})$, is L-Lipschitz w.r.t. W for all $\bar{\mathbf{x}} \in \mathcal{X}^L$:*

$$\|\boldsymbol{\phi}_{\tilde{W}}(\bar{\mathbf{x}}) - \boldsymbol{\phi}_W(\bar{\mathbf{x}})\| \leq L\|\tilde{W} - W\|.$$

Assumption 5 *The kernel function, $k(\cdot,\cdot)$, is uniformly G-Lipschitz and its first partial derivatives are uniformly J-Lipschitz:*

$$\left\|k(\tilde{\mathbf{h}}_1, \mathbf{h}_2) - k(\mathbf{h}_1, \mathbf{h}_2)\right\| \leq G\|\tilde{\mathbf{h}}_1 - \mathbf{h}_1\|, \ \left\|k(\mathbf{h}_1, \tilde{\mathbf{h}}_2) - k(\mathbf{h}_1, \mathbf{h}_2)\right\| \leq G\|\tilde{\mathbf{h}}_2 - \mathbf{h}_2\|,$$
$$\left\|\partial_1 k(\mathbf{h}_1, \mathbf{h}_2) - \partial_1 k(\tilde{\mathbf{h}}_1, \mathbf{h}_2)\right\| \leq J\|\tilde{\mathbf{h}}_1 - \mathbf{h}_1\|, \ \left\|\partial_2 k(\mathbf{h}_1, \mathbf{h}_2) - \partial_2 k(\mathbf{h}_1, \tilde{\mathbf{h}}_2)\right\| \leq J\|\tilde{\mathbf{h}}_2 - \mathbf{h}_2\|.$$

Assumption 6 *For any collection of data representations, $\{\mathbf{h}_i\}_{i=1}^N$, the smallest eigenvalue of the corresponding kernel matrix, K, is lower bounded by a positive constant $\gamma > 0$.*

Note that not only the assumptions are relevant to practice, Assumptions 5 and 6 can be also controlled by choosing the class of kernel functions used in the model. For example, the smallest eigenvalue of the kernel matrix, K, can be controlled by the smoothing properties of the kernel (Rasmussen and Williams, 2006).

Consider a particular stochastic step of Algorithm 2 at time t for a given mini-batch, $\mathbf{x}_t$, assuming that the kernel was last updated τ steps ago. The stochastic gradient will take the following form:

$$\hat{\mathbf{g}}_W^t = \frac{1}{N_t}\nabla_W\mathcal{L}(\boldsymbol{\theta}^{t+1}, W^t, K_{t-\tau}) = \frac{1}{2N_t}\sum_{i,j\in\mathcal{I}_t}\left(K_{t-\tau}^{-1}\mathbf{y}\mathbf{y}^\top K_{t-\tau}^{-1} - K_{t-\tau}^{-1}\right)_{ij}\left(\frac{\partial K_{t-\tau}}{\partial W}\right)_{ij}. \tag{C.21}$$

We can define $\hat{\delta}_W^t = \hat{\mathbf{g}}_W^t - \mathbf{g}_W^t$ and uniformly bound $\|\hat{\delta}_W^t - \delta_W^t\|$ in order to enable the same argument as in Theorem 1. To do that, we simply need to understand effect of perturbation of the kernel matrix on $\tilde{\mathbf{g}}_W^t$.

Lemma 3 *Under the given assumptions, the following bound holds for all $i, j = 1, \ldots, N$:*

$$\left|\left(K_{t-\tau}^{-1}\mathbf{y}\mathbf{y}^\top K_{t-\tau}^{-1} - K_{t-\tau}^{-1}\right)_{ij} - \left(K_t^{-1}\mathbf{y}\mathbf{y}^\top K_t^{-1} - K_t^{-1}\right)_{ij}\right| \leq D^2\|y\|^2 + D, \tag{C.22}$$

where $D = \gamma^{-1} + \left(\gamma - 2GL\tau\lambda\sigma\sqrt{N}\right)^{-1}$.

Proof The difference between $K_{t-\tau}^{-1}$ and K_t^{-1} is simply due to that the former has been computed for $W^{t-\tau}$ and the latter for W^t. To prove the bound, we need multiple steps.

- First, we need to bound the element-wise difference between $K_{t-\tau}$ and K_t. This is done by using Assumptions 4 and 5 and the triangular inequality:

$$\begin{aligned} |(K_t)_{ij} - (K_{t-\tau})_{ij}| &\leq G\left(\|\boldsymbol{\phi}_{W^t}(\bar{\mathbf{x}}_i) - \boldsymbol{\phi}_{W^{t-\tau}}(\bar{\mathbf{x}}_i)\| + \|\boldsymbol{\phi}_{W^t}(\bar{\mathbf{x}}_j) - \boldsymbol{\phi}_{W^{t-\tau}}(\bar{\mathbf{x}}_j)\|\right) \\ &\leq 2GL\|W^t - W^{t-\tau}\| \\ &= 2GL\|\sum_{s=1}^{\tau} \lambda_W^{t-\tau+s} \hat{\mathbf{g}}_W^{t-\tau+s}\| \\ &\leq 2GL\tau\lambda\sigma \end{aligned}$$

- Next, since each element of the perturbed matrix is bounded by $2GL\tau\lambda\sigma$, we can bound its spectral norm as follows:

$$\|K_t - K_{t-\tau}\| \leq \|K_t - K_{t-\tau}\|_F \leq 2GL\tau\lambda\sigma\sqrt{N},$$

  which means that the minimal singular value of the perturbed matrix is at least $\sigma_1 = \gamma - 2GL\tau\lambda\sigma\sqrt{N}$ due to Assumption 6.

- The spectral norm of the expression of interest can be bounded (quite pessimistically!) by summing up together the largest eigenvalues of the matrix inverses:

$$\begin{aligned} &\left|\left(K_{t-\tau}^{-1}\mathbf{y}\mathbf{y}^\top K_{t-\tau}^{-1} - K_{t-\tau}^{-1}\right)_{ij} - \left(K_t^{-1}\mathbf{y}\mathbf{y}^\top K_t^{-1} - K_t^{-1}\right)_{ij}\right| \\ \leq &\left\|\left(K_{t-\tau}^{-1} - K_t^{-1}\right)\mathbf{y}\mathbf{y}^\top\left(K_{t-\tau}^{-1} - K_t^{-1}\right) - \left(K_{t-\tau}^{-1} - K_t^{-1}\right)\right\| \\ \leq &\left(\gamma^{-1} + \left(\gamma - 2GL\tau\lambda\sigma\sqrt{N}\right)^{-1}\right)^2 \|\mathbf{y}\|^2 + \gamma^{-1} + \left(\gamma - 2GL\tau\lambda\sigma\sqrt{N}\right)^{-1}. \end{aligned}$$

Each element of a matrix is bounded by the largest eigenvalue. ■

Using Lemma 3, it is straightforward to extend Theorem 1 to Algorithm 2.

Theorem 2 *Let $\{(\boldsymbol{\theta}^t, W^t)\}$ be a sequence generated from Algorithm 2 with learning rates for $\boldsymbol{\theta}$ and W being $\lambda_{\boldsymbol{\theta}}^t = c_{\boldsymbol{\theta}}^t\alpha_t$ and $\lambda_W^t = c_W^t\alpha_t$, respectively, where $c_{\boldsymbol{\theta}}^t, c_W^t, \alpha_t$ are positive scalars, such that*

$$0 < \inf_t c_{\{\boldsymbol{\theta},W\}}^t \leq \sup_t c_{\{\boldsymbol{\theta},W\}}^t < \infty, \quad \alpha_t \geq \alpha_{t+1}, \quad \sum_{t=1}^{\infty}\alpha_t = +\infty, \quad \sum_{t=1}^{\infty}\alpha_t^2 < +\infty, \quad \forall t. \tag{C.23}$$

Then, under Assumptions 1 through 6 and if $\sigma = \sup_t \sigma_t < \infty$, we have

$$\lim_{t\to\infty} \mathbb{E}\|\nabla\mathcal{L}(\boldsymbol{\theta}^t, W^t)\| = 0. \tag{C.24}$$

Proof The proof is identical to the proof of Theorem 1. The only difference is in the following upper bound on the expected gradient error: $\mathbb{E}\|\delta_W^t\| \leq \sigma + 2ND(D\|y\|^2 + 1)J\rho$, where D is as given in Lemma 3. ■

Remark 1 *Even though the provided bounds are crude due to pessimistic estimates of the perturbed kernel matrix spectrum*[15]*, we still see a fine balance between the delay, τ, and the learning rate, λ, as given in the expression for the D constant.*

Appendix D. Details on the datasets

The datasets varied in the number of time steps (from hundreds to a million), input and output dimensionality, and the nature of the estimation tasks.

D.1 Self-driving car

The following is description of the input and target time series used in each of the autonomous driving tasks (dimensionality is given in parenthesis).

- **Car speed estimation:**
  - *Features:* GPS velocity (3), fiber gyroscope (3).
  - *Targets:* speed measurements from the car speedometer.
- **Car yaw estimation:**
  - *Features:* acceleration (3), compass measurements (3).
  - *Targets:* yaw in the car-centric frame.
- **Lane sequence prediction:** Each lane was represented by 8 cubic polynomial coefficients [4 coefficients for x (front) and 4 coefficients for y (left) axes in the car-centric frame]. Instead of predicting the coefficients (which turned out to lead to overall less stable results), we discretized the lane curves using 7 points (initial, final and 5 equidistant intermediate points).
  - *Features:* lanes at a previous time point (16), GPS velocity (3), fiber gyroscope (3), compass (3), steering angle (1).
  - *Targets:* coordinates of the lane discretization points (7 points per lane resulting in 28 total output dimensions).

15. Tighter bounds can be derived by inspecting the effects of perturbations on specific kernels as well as using more specific assumptions about the data distribution.

Table 5: Summary of the feedforward and recurrent neural architectures and the corresponding hyperparameters used in the experiments. GP-based models used the same architectures as their non-GP counterparts. Activations are given for the hidden units; vanilla neural nets used linear output activations.

Name	Data	Time lag	Layers	Units*	Type	Regularizer**	Optimizer
NARX	Actuator	32	1	256	ReLU	dropout(0.5)	Adam(0.01)
	Drives	16	1	128			
	GEF-power	48	1	256			
	GEF-wind	48	1	16			
	Car	128	1	128			
RNN	Actuator	32	1	64	tanh	dropout(0.25), rec_dropout(0.05)	Adam(0.01)
	Drives	16	1	64			
	GEF-power	48	1	16			
	GEF-wind	48	1	32			
	Car	128	1	128			
LSTM	Actuator	32	1	256	tanh	dropout(0.25), rec_dropout(0.05)	Adam(0.01)
	Drives	16	1	128			
	GEF-power	48	2	256			
	GEF-wind	48	1	64			
	Car	128	2	64			

*Each layer consisted of the same number of units given in the table.
**rec_dropout denotes the dropout rate of the recurrent weights (Gal and Ghahramani, 2016b).

- **Estimation of the nearest front vehicle position:** (x, y) coordinates in the car-centric frame.
  - *Features:* x and y at a previous time point (2), GPS velocity (3), fiber gyroscope (3), compass (3), steering angle (1).
  - *Targets:* x and y coordinates.

Appendix E. Neural architectures

Details on the best neural architectures used for each of the datasets are given in Table 5.

References

Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 873–881, 2011.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 1994.

Roberto Calandra, Jan Peters, Carl Edward Rasmussen, and Marc Peter Deisenroth. Manifold gaussian processes for regression. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 3338–3345. IEEE, 2016.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

Andreas C Damianou and Neil D Lawrence. Deep gaussian processes. In *AISTATS*, pages 207–215, 2013.

Marc Peter Deisenroth and Jun Wei Ng. Distributed gaussian processes. In *International Conference on Machine Learning (ICML)*, volume 2, page 5, 2015.

Roger Frigola, Yutian Chen, and Carl Rasmussen. Variational gaussian process state-space models. In *Advances in Neural Information Processing Systems*, pages 3680–3688, 2014.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1050–1059, 2016a.

Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1019–1027, 2016b.

Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1225–1234, 2016.

James Hensman, Nicolò Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 282–290. AUAI Press, 2013.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Using deep belief nets to learn covariance kernels for gaussian processes. In *Advances in neural information processing systems*, pages 1249–1256, 2008.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

Tommi S Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999.

Robert Keys. Cubic convolution interpolation for digital image processing. *IEEE transactions on acoustics, speech, and signal processing*, 29(6):1153–1160, 1981.

Juš Kocijan, Agathe Girard, Blaž Banko, and Roderick Murray-Smith. Dynamic systems identification with gaussian processes. *Mathematical and Computer Modelling of Dynamical Systems*, 11(4):411–424, 2005.

John Langford, Alex J Smola, and Martin Zinkevich. Slow learners are fast. *Advances in Neural Information Processing Systems*, 22:2331–2339, 2009.

Miguel Lázaro-Gredilla. Bayesian warped gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1619–1627, 2012.

Tsungnam Lin, Bil G Horne, Peter Tiňo, and C Lee Giles. Learning long-term dependencies in narx recurrent neural networks. *Neural Networks, IEEE Transactions on*, 7(6):1329–1338, 1996.

David JC MacKay. Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166, 1998.

César Lincoln C Mattos, Zhenwen Dai, Andreas Damianou, Jeremy Forth, Guilherme A Barreto, and Neil D Lawrence. Recurrent gaussian processes. *arXiv preprint arXiv:1511.06644*, 2015.

Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *The Journal of Machine Learning Research*, 7:2651–2667, 2006.

Joaquin Quinonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6: 1939–1959, 2005.

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. *Advances in neural information processing systems*, pages 294–300, 2001.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. The MIT Press, 2006.

Stéphane Ross, Geoffrey J Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, volume 1, page 6, 2011.

Yunus Saatchi and Andrew Gordon Wilson. Bayesian gan. *arXiv preprint arXiv:1705.09558*, 2017.

Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Jonas Sjöberg, Qinghua Zhang, Lennart Ljung, Albert Benveniste, Bernard Delyon, Pierre-Yves Glorennec, Håkan Hjalmarsson, and Anatoli Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

Edward Snelson, Carl Edward Rasmussen, and Zoubin Ghahramani. Warped gaussian processes. In *NIPS*, pages 337–344, 2003.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

ML Stein. Interpolation of spatial data, 1999.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Ryan D Turner, Marc P Deisenroth, and Carl E Rasmussen. State-space inference and learning with gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 868–875, 2010.

Peter Van Overschee and Bart De Moor. *Subspace identification for linear systems: Theory — Implementation — Applications*. Springer Science & Business Media, 2012.

Jack Wang, Aaron Hertzmann, and David M Blei. Gaussian process dynamical models. In *Advances in neural information processing systems*, pages 1441–1448, 2005.

Torbjörn Wigren. Input-output data sets for development and benchmarking in nonlinear identification. *Technical Reports from the department of Information Technology*, 20: 2010–020, 2010.

Andrew Wilson and Zoubin Ghahramani. Copula processes. In *Advances in Neural Information Processing Systems*, pages 2460–2468, 2010.

Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1067–1075, 2013.

Andrew Gordon Wilson and Hannes Nickisch. Kernel interpolation for scalable structured gaussian processes (kiss-gp). In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1775–1784, 2015.

Andrew Gordon Wilson, Christoph Dann, and Hannes Nickisch. Thoughts on massively scalable gaussian processes. *arXiv preprint arXiv:1511.01870*, 2015.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016a.

Andrew Gordon Wilson, Zhiting Hu, Ruslan R Salakhutdinov, and Eric P Xing. Stochastic variational deep kernel learning. In *Advances in Neural Information Processing Systems*, pages 2586–2594, 2016b.

D Randall Wilson and Tony R Martinez. The general inefficiency of batch training for gradient descent learning. *Neural Networks*, 16(10):1429–1451, 2003.

Yangyang Xu and Wotao Yin. Block stochastic gradient iteration for convex and nonconvex optimization. *SIAM Journal on Optimization*, 25(3):1686–1716, 2015.

Journal of Machine Learning Research 18 (2017) 1-52 Submitted 10/16; Revised 6/17; Published 7/17

Convolutional Neural Networks Analyzed via Convolutional Sparse Coding

Vardan Papyan* VARDANP@CAMPUS.TECHNION.AC.IL
Department of Computer Science
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

Yaniv Romano* YROMANO@TX.TECHNION.AC.IL
Department of Electrical Engineering
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

Michael Elad ELAD@CS.TECHNION.AC.IL
Department of Computer Science
Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel

Editor: Kevin Murphy

Abstract

Convolutional neural networks (CNN) have led to many state-of-the-art results spanning through various fields. However, a clear and profound theoretical understanding of the forward pass, the core algorithm of CNN, is still lacking. In parallel, within the wide field of sparse approximation, Convolutional Sparse Coding (CSC) has gained increasing attention in recent years. A theoretical study of this model was recently conducted, establishing it as a reliable and stable alternative to the commonly practiced patch-based processing. Herein, we propose a novel multi-layer model, ML-CSC, in which signals are assumed to emerge from a cascade of CSC layers. This is shown to be tightly connected to CNN, so much so that the forward pass of the CNN is in fact the thresholding pursuit serving the ML-CSC model. This connection brings a fresh view to CNN, as we are able to attribute to this architecture theoretical claims such as uniqueness of the representations throughout the network, and their stable estimation, all guaranteed under simple local sparsity conditions. Lastly, identifying the weaknesses in the above pursuit scheme, we propose an alternative to the forward pass, which is connected to deconvolutional and recurrent networks, and also has better theoretical guarantees.

Keywords: Deep Learning, Convolutional Neural Networks, Forward Pass, Sparse Representation, Convolutional Sparse Coding, Thresholding Algorithm, Basis Pursuit

∗. The authors contributed equally to this work.

©2017 Vardan Papyan, Yaniv Romano and Michael Elad.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-505.html.

1. Introduction

Deep learning (LeCun et al., 2015), and in particular CNN (LeCun et al., 1990, 1998; Krizhevsky et al., 2012), has gained a copious amount of attention in recent years as it has led to many state-of-the-art results spanning through many fields – including speech recognition (Bengio et al., 2003; Hinton et al., 2012; Mikolov et al., 2013), computer vision (Farabet et al., 2013; Simonyan and Zisserman, 2014; He et al., 2015), signal and image processing (Gatys et al., 2015; Ulyanov et al., 2016; Johnson et al., 2016; Dong et al., 2016), to name a few. In the context of CNN, the forward pass is a multi-layer scheme that provides an end-to-end mapping, from an input signal to some desired output. Each layer of this algorithm consists of three steps. The first convolves the input with a set of learned filters, resulting in a set of feature (or kernel) maps. These then undergo a point wise non-linear function, in a second step, often resulting in a sparse outcome (Glorot et al., 2011). A third (and optional) down-sampling step, termed *pooling*, is then applied on the result in order to reduce its dimensions. The output of this layer is then fed into another one, thus forming the multi-layer structure, often termed *forward pass*.

Despite its marvelous empirical success, a clear and profound theoretical understanding of this scheme is still lacking. A few preliminary theoretical results were recently suggested. In (Mallat, 2012; Bruna and Mallat, 2013) the Scattering Transform was proposed, suggesting to replace the learned filters in the CNN with predefined Wavelet functions. Interestingly, the features obtained from this network were shown to be invariant to various transformations such as translations and rotations. Other works have studied the properties of deep and fully connected networks under the assumption of independent identically distributed random weights (Giryes et al., 2015; Saxe et al., 2013; Arora et al., 2014; Dauphin et al., 2014; Choromanska et al., 2015). In particular, in (Giryes et al., 2015) deep neural networks were proven to preserve the metric structure of the input data as it propagates through the layers of the network. This, in turn, was shown to allow a stable recovery of the data from the features obtained from the network.

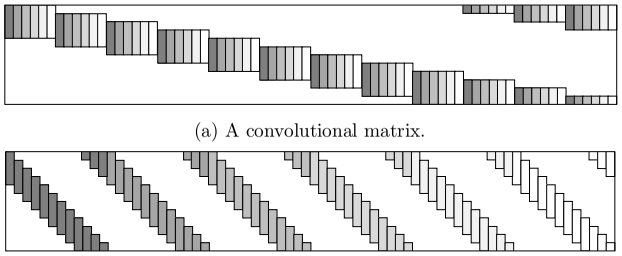
Another prominent paradigm in data processing is the sparse representation concept, being one of the most popular choices for a prior in the signal and image processing communities, and leading to exceptional results in various applications (Elad and Aharon, 2006; Dong et al., 2011; Zhang and Li, 2010; Jiang et al., 2011; Mairal et al., 2014). In this framework, one assumes that a signal can be represented as a linear combination of a few columns (called atoms) from a matrix termed a dictionary. Put differently, the signal is equal to a multiplication of a dictionary by a sparse vector. The task of retrieving the sparsest representation of a signal over a dictionary is called sparse coding or pursuit. Over the years, various algorithms were proposed to tackle this problem, among of which we mention the thresholding algorithm (Elad, 2010) and its iterative variant (Daubechies et al., 2004). When handling natural signals, this model has been commonly used for modeling local patches extracted from the global data mainly due to the computational difficulties related to the task of learning the dictionary (Elad and Aharon, 2006; Dong et al., 2011; Mairal et al., 2014; Romano and Elad, 2015; Sulam and Elad, 2015). However, in recent years an alternative to this patch-based processing has emerged in the form of the Convolutional Sparse Coding (CSC) model (Bristow et al., 2013; Kong and Fowlkes, 2014; Wohlberg, 2014; Gu et al., 2015; Heide et al., 2015; Papyan et al., 2016a,b). This circumvents the afore-

mentioned limitations by imposing a special structure – a union of banded and Circulant matrices – on the dictionary involved. The traditional sparse model has been extensively studied over the past two decades (Elad, 2010; Foucart and Rauhut, 2013). More recently, the convolutional extension was extensively analyzed in (Papyan et al., 2016a,b), shedding light on its theoretical aspects and prospects of success.

In this work, by leveraging the recent study of CSC, we aim to provide a new perspective on CNN, leading to a clear and profound theoretical understanding of this scheme, along with new insights. Embarking from the classic CSC, our approach builds upon the observation that similar to the original signal, the representation vector itself also admits a convolutional sparse representation. As such, it can be modeled as a superposition of atoms, taken from a different convolutional dictionary. This rationale can be extended to several layers, leading to the definition of our proposed ML-CSC model. Building on the recent analysis of the CSC, we provide a theoretical study of this novel model and its associated pursuits, namely the layered thresholding algorithm and the layered basis pursuit (BP).

Our analysis reveals the relation between the CNN and the ML-CSC model, showing that *the forward pass of the CNN is in fact identical to our proposed pursuit – the layered thresholding algorithm*. This connection is of significant importance since it gives a clear mathematical meaning, objective and model to the CNN architecture, which in turn can be accompanied by guarantees for the success of the forward pass, studied via the layered thresholding algorithm. Specifically, we show that the forward pass is guaranteed to recover an estimate of the underlying representations of an input signal, assuming these are sparse in a local sense. Moreover, considering a setting where a norm-bounded noise is added to the signal, we show that such a mild corruption in the input results in a bounded perturbation in the output – indicating the stability of the CNN in recovering the underlying representations. Lastly, we exploit the answers to the above questions in order to propose an alternative to the commonly used forward pass algorithm, which is tightly connected to both deconvolutional (Zeiler et al., 2010; Pu et al., 2016) and recurrent networks (Bengio et al., 1994). The proposed alternative scheme is accompanied by a thorough theoretical study. Although this and the analysis presented throughout this work focus on CNN, we will show that they also hold for fully connected networks.

This paper is organized as follows. In Section 2 we review the basics of both the CNN and the Sparse-Land model. We then define the proposed ML-CSC model in Section 3, together with its corresponding deep sparse coding problem. In Section 4, we aim to solve this using the layered thresholding algorithm, which is shown to be equivalent to the forward pass of the CNN. Next, having established the relevance of our model to CNN, we proceed to its analysis in Section 5. Standing on these theoretical grounds, we then propose in Section 6 a provably improved pursuit, termed the layered BP, accompanied by its theoretical analysis. We revisit the assumptions of our model in Section 7. First, in Section 7.1 we link the double sparsity model to ours by assuming the dictionaries throughout the layers are sparse. Then, in Section 7.2 we consider an idea typically employed in CNN, termed spatial-stride, showing its benefits from a simple theoretical perspective. Combining our insights from Section 7.1 and 7.2, we move to an experimental phase by constructing a family of signals satisfying the assumptions of our model, which are then used in order to verify our theoretical results. Finally, in Section 9 we conclude the contributions of this paper and present several future directions.



(a) A convolutional matrix.

(b) A concatenation of banded and Circulant matrices.

Figure 1: The two facets of the convolutional structure.

2. Background

This section is divided into two parts: The first is dedicated to providing a simple mathematical formulation of the CNN and the forward pass, while the second reviews the Sparse-Land model and its various extensions. Readers familiar with these two topics can skip directly to Section 3, which moves to serve the main contribution of this work.

2.1 Deep Learning - Convolutional Neural Networks

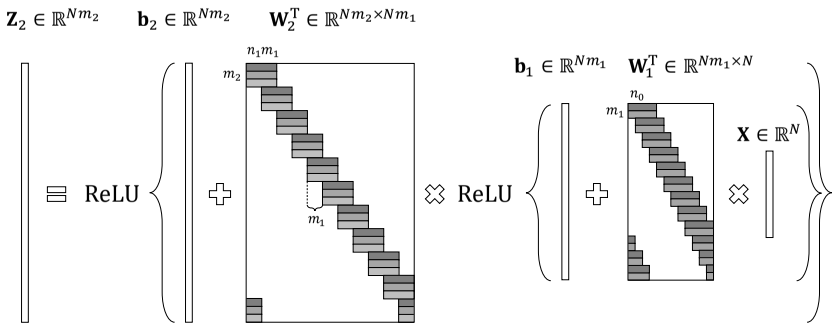
The fundamental algorithm of deep learning is the *forward pass*, employed both in the training and the inference stages. The first step of this algorithm convolves an input (one dimensional) signal $\mathbf{X} \in \mathbb{R}^N$ with a set of m_1 learned filters of length n_0, creating m_1 feature (or kernel) maps. Equally, this convolution can be written as a matrix-vector multiplication, $\mathbf{W}_1^T\mathbf{X} \in \mathbb{R}^{Nm_1}$, where $\mathbf{W}_1 \in \mathbb{R}^{N\times Nm_1}$ is a matrix containing in its columns the m_1 filters with all of their shifts. This structure, also known as a convolutional matrix, is depicted in Figure 1a. A pointwise nonlinear function is then applied on the sum of the obtained feature maps $\mathbf{W}_1^T\mathbf{X}$ and a bias term denoted by $\mathbf{b}_1 \in \mathbb{R}^{Nm_1}$. Many possible functions were proposed over the years, the most popular one being the Rectifier Linear Unit (ReLU) (Glorot et al., 2011; Krizhevsky et al., 2012), formally defined as $\text{ReLU}(z) = \max(z, 0)$. By cascading the basic block of convolutions followed by a nonlinear function, $\mathbf{Z}_1 = \text{ReLU}(\mathbf{W}_1^T\mathbf{X} + \mathbf{b}_1)$, a multi-layer structure of depth K is constructed. Formally, for two layers this is given by

$$f(\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^2, \{\mathbf{b}_i\}_{i=1}^2) = \mathbf{Z}_2 = \text{ReLU}\Big(\mathbf{W}_2^T \ \text{ReLU}\ (\ \mathbf{W}_1^T\mathbf{X} + \mathbf{b}_1\ ) + \mathbf{b}_2 \Big), \qquad (1)$$

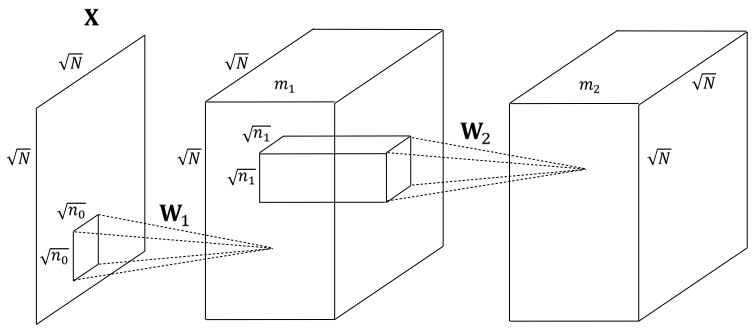
where $\mathbf{W}_2 \in \mathbb{R}^{Nm_1\times Nm_2}$ is a convolutional matrix (up to a small modification discussed below) constructed from m_2 filters of length n_1m_1 and $\mathbf{b}_2 \in \mathbb{R}^{Nm_2}$ is its corresponding bias. Although the two layers considered here can be readily extended to a much deeper configuration, we defer this to a later stage.

By changing the order of the columns in the convolutional matrix, one can observe that it can be equally viewed as a concatenation of banded and Circulant[1] matrices, as depicted

1. We shall assume throughout this paper that boundaries are treated by a periodic continuation, which gives rise to the cyclic structure.



(a) An illustration of Equation (1) for a one dimensional signal $\mathbf{X}$.



(b) The evolution of an image $\mathbf{X}$ throughout the CNN. Notice that the number of channels in $\mathbf{X}$ is equal to one and as such $m_0 = 1$.

Figure 2: The forward pass algorithm for a one dimensional signal (a) and an image (b).

in Figure 1b. Using this observation, the above description for one dimensional signals can be extended to images, with the exception that now every Circulant matrix is replaced by a block Circulant with Circulant blocks one.

An illustration of the forward pass algorithm is presented in Figure 2a and 2b. In Figure 2a one can observe that $\mathbf{W}_2$ is not a regular convolutional matrix but a stride one, since it shifts local filters by skipping m_1 entries at a time. The reason for this becomes apparent once we look at Figure 2b; the convolutions of the second layer are computed by shifting the filters of $\mathbf{W}_2$ that are of size $\sqrt{n_1} \times \sqrt{n_1} \times m_1$ across N places, skipping m_1 indices at a time from the $\sqrt{N} \times \sqrt{N} \times m_1$-sized array. A matrix obeying this structure is called a *stride convolutional matrix*.

Thus far, we have presented the basic structure of CNN. However, oftentimes an additional non-linear function, termed *pooling*, is employed on the resulting feature map obtained from the ReLU operator. In essence, this step summarizes each w_i-dimensional spatial neighborhood from the i-th kernel map $\mathbf{Z}_i$ by replacing it with a single value. If the neighborhoods are non-overlapping, for example, this results in the down-sampling of the feature map by a factor of w_i. The most widely used variant of the above is the max pooling (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014), which picks the maximal value of each neighborhood. In (Springenberg et al., 2014) it was shown that this operator can be replaced by a convolutional layer with increased stride without loss in performance in several image classification tasks. Moreover, the current state-of-the-art in image recognition is obtained by the residual network (He et al., 2015), which does not employ any pooling steps (except for a single layer). As such, we defer the analysis of this operator to a follow-up work.

In the context of classification, for example, the output of the last layer is fed into a simple classifier that attempts to predict the label of the input signal $\mathbf{X}$, denoted by $h(\mathbf{X})$. Given a set of signals $\{\mathbf{X}_j\}_j$, the task of learning the parameters of the CNN – including the filters $\{\mathbf{W}_i\}_{i=1}^K$, the biases $\{\mathbf{b}_i\}_{i=1}^K$ and the parameters of the classifier $\mathbf{U}$ – can be formulated as the following minimization problem

$$\min_{\{\mathbf{W}_i\}_{i=1}^K, \{\mathbf{b}_i\}_{i=1}^K, \mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j), \mathbf{U}, f\left(\mathbf{X}_j, \{\mathbf{W}_i\}_{i=1}^K, \{\mathbf{b}_i\}_{i=1}^K\right)\Big). \tag{2}$$

This optimization task seeks for the set of parameters that minimize the mean of the loss function ℓ, representing the price incurred when classifying the signal $\mathbf{X}$ incorrectly. The input for ℓ is the true label $h(\mathbf{X})$ and the one estimated by employing the classifier defined by $\mathbf{U}$ on the final layer of the CNN given by $f\left(\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^K, \{\mathbf{b}_i\}_{i=1}^K\right)$. Similarly one can tackle various other problems, e.g. regression or prediction.

In the remainder of this work we shall focus on the feature extraction process and assume that the parameters of the CNN model are pre-trained and fixed. These, for example, could have been obtained by minimizing the above objective via the backpropagation algorithm and the stochastic gradient descent, as in the VGG network (Simonyan and Zisserman, 2014).

2.2 Sparse-Land

This section presents an overview of the Sparse-Land model and its many extensions. We start with the traditional sparse representation and the core problem it aims to solve, and then proceed to its nonnegative variant. Next, we continue to the dictionary learning task both in the unsupervised and supervised cases. Finally, we describe the recent CSC model, which will lead us in the next section to the proposal of the ML-CSC model. This, in turn, will naturally connect the realm of sparsity to that of the CNN.

2.2.1 Sparse Representation

In the sparse representation model one assumes a signal $\mathbf{X} \in \mathbb{R}^N$ can be described as a multiplication of a matrix $\mathbf{D} \in \mathbb{R}^{N \times M}$, also called a dictionary, by a sparse vector $\mathbf{\Gamma} \in \mathbb{R}^M$.

Equally, the signal $\mathbf{X}$ can be seen as a linear combination of a few columns from the dictionary $\mathbf{D}$, coined atoms.

For a fixed dictionary, given a signal $\mathbf{X}$, the task of recovering its sparsest representation $\mathbf{\Gamma}$ is called sparse coding, or simply pursuit, and it attempts to solve the following problem (Donoho and Elad, 2003; Tropp, 2004; Elad, 2010):

$$(\mathrm{P}_0): \ \min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_0 \ \text{ s.t. } \ \mathbf{D}\mathbf{\Gamma} = \mathbf{X}, \tag{3}$$

where we have denoted by $\|\mathbf{\Gamma}\|_0$ the number of non-zeros in $\mathbf{\Gamma}$. The above has a convex relaxation in the form of the Basis-Pursuit (BP) problem (Chen et al., 2001; Donoho and Elad, 2003; Tropp, 2006), formally defined as

$$(\mathrm{P}_1): \ \min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_1 \ \text{ s.t. } \ \mathbf{D}\mathbf{\Gamma} = \mathbf{X}. \tag{4}$$

Many questions arise from the above two defined problems. For instance, given a signal $\mathbf{X}$, is its sparsest representation unique? Assuming that such a unique solution exists, can it be recovered using practical algorithms such as the Orthogonal Matching Pursuit (OMP) (Chen et al., 1989; Pati et al., 1993) and the BP (Chen et al., 2001; Daubechies et al., 2004)? The answers to these questions were shown to be positive under the assumption that the number of non-zeros in the underlying representation is not too high and in particular less than $\frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$ (Donoho and Elad, 2003; Tropp, 2004; Donoho et al., 2006). The quantity $\mu(\mathbf{D})$ is the mutual coherence of the dictionary $\mathbf{D}$, being the maximal inner product of two atoms extracted from it[2]. Formally, we can write

$$\mu(\mathbf{D}) = \max_{i \neq j} |\mathbf{d}_i^T \mathbf{d}_j|.$$

Tighter conditions, relying on sharper characterizations of the dictionary, were also suggested in the literature (Candes et al., 2006; Schnass and Vandergheynst, 2007; Candes et al., 2006; Candes and Tao, 2007). However, at this point, we shall not dwell on these.

One of the simplest approaches for tackling the P_0 and P_1 problems is via the hard and soft thresholding algorithms, respectively. These operate by computing the inner products between the signal $\mathbf{X}$ and all the atoms in $\mathbf{D}$ and then choosing the atoms corresponding to the highest responses. This can be described as solving, for some scalar β, the following problems:

$$\min_{\mathbf{\Gamma}} \frac{1}{2}\|\mathbf{\Gamma} - \mathbf{D}^T\mathbf{X}\|_2^2 + \beta\|\mathbf{\Gamma}\|_0$$

for the P_0, or

$$\min_{\mathbf{\Gamma}} \frac{1}{2}\|\mathbf{\Gamma} - \mathbf{D}^T\mathbf{X}\|_2^2 + \beta\|\mathbf{\Gamma}\|_1, \tag{5}$$

for the P_1. The above are simple projection problems that admit a closed-form solution in the form[3] of $\mathcal{H}_\beta(\mathbf{D}^T\mathbf{X})$ or $\mathcal{S}_\beta(\mathbf{D}^T\mathbf{X})$, where we have defined the hard thresholding operator

2. Hereafter, we assume that the atoms are normalized to a unit ℓ_2 norm.
3. The curious reader may identify the relation between the notations used here and the ones in the previous subsection, which starts to reveal the relation between CNN and sparsity-inspired models. This connection will be made clearer as we proceed to CSC.

$\mathcal{H}_\beta(\cdot)$ by

$$\mathcal{H}_\beta(z) = \begin{cases} z, & z < -\beta \\ 0, & -\beta \leq z \leq \beta \\ z, & \beta < z, \end{cases}$$

and the soft thresholding operator $\mathcal{S}_\beta(\cdot)$ by

$$\mathcal{S}_\beta(z) = \begin{cases} z+\beta, & z < -\beta \\ 0, & -\beta \leq z \leq \beta \\ z-\beta, & \beta < z. \end{cases}$$

Both of the above, depicted in Figure 3, nullify small entries and thus promote a sparse solution. However, while the hard thresholding operator does not modify large coefficients (in absolute value), the soft thresholding does, by contracting these to zero. This inherent limitation of the soft version will appear later on in our theoretical analysis.

As for the theoretical guarantees for the success of the simple thresholding algorithms; these depend on the properties of $\mathbf{D}$ and on the ratio between the minimal and maximal coefficients in absolute value in $\mathbf{\Gamma}$, and thus are weaker when compared to those found for OMP and BP (Donoho and Elad, 2003; Tropp, 2004; Donoho et al., 2006). Still, under some conditions, both algorithms are guaranteed to find the true support of $\mathbf{\Gamma}$ along with an approximation of its true coefficients. Moreover, a better estimation of these can be obtained by projecting the input signal onto the atoms corresponding to the found support (indices of the non-zero entries) by solving a Least-Squares problem. This step, termed debiasing (Elad, 2010), results in a more accurate identification of the non-zero values.

2.2.2 Nonnegative Sparse Coding

The nonnegative sparse representation model assumes a signal can be decomposed into a multiplication of a dictionary and a *nonnegative* sparse vector. A natural question arising from this is whether such a modification to the original Sparse-Land model affects its expressiveness. To address this, we hereby provide a simple reduction from the original sparse representation to the nonnegative one.

Consider a signal $\mathbf{X} = \mathbf{D}\mathbf{\Gamma}$, where the signs of the entries in $\mathbf{\Gamma}$ are unrestricted. Notice that this can be equally written as

$$\mathbf{X} = \mathbf{D}\mathbf{\Gamma}_P + (-\mathbf{D})(-\mathbf{\Gamma}_N),$$

where we have split the vector $\mathbf{\Gamma}$ to its positive coefficients, $\mathbf{\Gamma}_P$, and its negative ones, $\mathbf{\Gamma}_N$. Since the coefficients in $\mathbf{\Gamma}_P$ and $-\mathbf{\Gamma}_N$ are all positive, one can thus assume the signal $\mathbf{X}$ admits a non-negative sparse representation over the dictionary $[\mathbf{D}, -\mathbf{D}]$ with the vector $[\mathbf{\Gamma}_P, -\mathbf{\Gamma}_N]^T$. Thus, restricting the coefficients in the sparsity inspired model to be nonnegative does not change its expressiveness.

Similar to the original model, in the nonnegative case, one could solve the associated pursuit problem by employing a soft thresholding algorithm. However, in this case a constraint must be added to the optimization problem in Equation (5), forcing the outcome to be positive, i.e.,

$$\min_{\mathbf{\Gamma}} \frac{1}{2}\|\mathbf{\Gamma} - \mathbf{D}^T\mathbf{X}\|_2^2 + \beta\|\mathbf{\Gamma}\|_1 \quad \text{s.t.} \quad \mathbf{\Gamma} \geq \mathbf{0}.$$

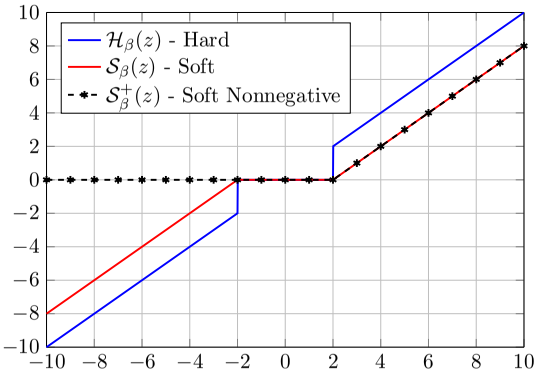


Figure 3: The thresholding operators for a constant $\beta = 2$.

Since the above is a simple projection problem (onto the ℓ_1 ball constrained to positive entries), it admits a closed-form solution $\mathcal{S}_\beta^+(\mathbf{D}^T\mathbf{X})$, where we have defined the soft nonnegative thresholding operator $\mathcal{S}_\beta^+(\cdot)$ as

$$\mathcal{S}_\beta^+(z) = \begin{cases} 0, & z \leq \beta \\ z - \beta, & \beta < z. \end{cases}$$

Remarkably, the above function satisfies

$$\mathcal{S}_\beta^+(z) = \max(z - \beta, 0) = \text{ReLU}(z - \beta).$$

In other words, the ReLU and the soft nonnegative thresholding operator are equal, a fact that will prove to be important later in our work. We should note that a similar conclusion was reached in (Fawzi et al., 2015). To summarize this discussion, we depict in Figure 3 the hard, soft, and nonnegative soft thresholding operators.

2.2.3 Unsupervised and Task Driven Dictionary Learning

At first, the dictionaries employed in conjunction with the sparsity inspired model were analytically defined matrices, such as the Wavelet and the Fourier (Daubechies et al., 1992; Mallat and Zhang, 1993; Elad and Bruckstein, 2002; Mallat, 2008). Although the sparse coding problem under these can be done very efficiently, over the years many have shifted to a data driven approach – adapting the dictionary $\mathbf{D}$ to a set of training signals at hand via some learning procedure. This was empirically shown to lead to sparser representations and better overall performance, at the cost of complicating the involved pursuit, since the dictionary was usually chosen to be redundant (having more columns than rows).

The task of learning a dictionary for representing a set of signals $\{\mathbf{X}_j\}_j$ can be formulated as follows

$$\min_{\mathbf{D},\{\mathbf{\Gamma}^j\}_j} \sum_j \|\mathbf{X}_j - \mathbf{D}\mathbf{\Gamma}^j\|_2^2 + \xi\|\mathbf{\Gamma}^j\|_0.$$

The above formulation is an unsupervised learning procedure, and it was later extended to a supervised setting. In this context, given a set of signals $\{\mathbf{X}_j\}_j$, one attempts to predict their corresponding labels $\{h(\mathbf{X}_j)\}_j$. A common approach for tackling this is first solving a pursuit problem for each signal $\mathbf{X}_j$ over a dictionary $\mathbf{D}$, resulting in

$$\mathbf{\Gamma}^\star(\mathbf{X}_j, \mathbf{D}) = \arg\min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_0 \quad \text{s.t.} \quad \mathbf{D}\mathbf{\Gamma} = \mathbf{X}_j,$$

and then feeding these sparse representations into a simple classifier, defined by the parameters $\mathbf{U}$. The task of learning jointly the dictionary $\mathbf{D}$ and the classifier $\mathbf{U}$ was addressed in (Mairal et al., 2012), where the following optimization problem was proposed

$$\min_{\mathbf{D},\mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j), \mathbf{U}, \mathbf{\Gamma}^\star(\mathbf{X}_j, \mathbf{D})\Big).$$

The loss function ℓ in the above objective penalizes the estimated label if it is different from the true $h(\mathbf{X}_j)$, similar to what we have seen in Section 2.1. The above formulation contains in it the unsupervised option as a special case, in which $\mathbf{U}$ is of no importance, and the loss function is the representation error $\sum_j \|\mathbf{X}_j - \mathbf{D}\mathbf{\Gamma}_j^\star\|_2^2$.

Double sparsity – first proposed in (Rubinstein et al., 2010) and later employed in (Sulam et al., 2016) – attempts to benefit from both the computational efficiency of analytically defined matrices, and the adaptability of data driven dictionaries. In this model, one assumes the dictionary $\mathbf{D}$ can be factorized into a multiplication of two matrices, $\mathbf{D}_1$ and $\mathbf{D}_2$, where $\mathbf{D}_1$ is an analytic dictionary with fast implementation, and $\mathbf{D}_2$ is a trained sparse one. As a result, the signal $\mathbf{X}$ can be represented as

$$\mathbf{X} = \mathbf{D}\mathbf{\Gamma}_2 = \mathbf{D}_1\mathbf{D}_2\mathbf{\Gamma}_2,$$

where $\mathbf{\Gamma}_2$ is sparse.

We propose a different interpretation for the above, which is unrelated to practical aspects. Since both the matrix $\mathbf{D}_2$ and the vector $\mathbf{\Gamma}_2$ are sparse, one would expect their multiplication $\mathbf{\Gamma}_1 = \mathbf{D}_2\mathbf{\Gamma}_2$ to be sparse as well. As such, the double sparsity model implicitly assumes that the signal $\mathbf{X}$ can be decomposed into a multiplication of a dictionary $\mathbf{D}_1$ and sparse vector $\mathbf{\Gamma}_1$, which in turn can also be decomposed similarly via $\mathbf{\Gamma}_1 = \mathbf{D}_2\mathbf{\Gamma}_2$.

2.2.4 Convolutional Sparse Coding Model

Due to the computational constraints entailed when deploying trained dictionaries, this approach seems valid only for treatment of low-dimensional signals. Indeed, the sparse representation model is traditionally used for modeling local patches extracted from a global signal. An alternative, which was recently proposed, is the CSC model that attempts to represent the whole signal $\mathbf{X} \in \mathbb{R}^N$ as a multiplication of a global convolutional dictionary $\mathbf{D} \in \mathbb{R}^{N \times Nm}$ and a sparse vector $\mathbf{\Gamma} \in \mathbb{R}^{Nm}$. Interestingly, the former is constructed by

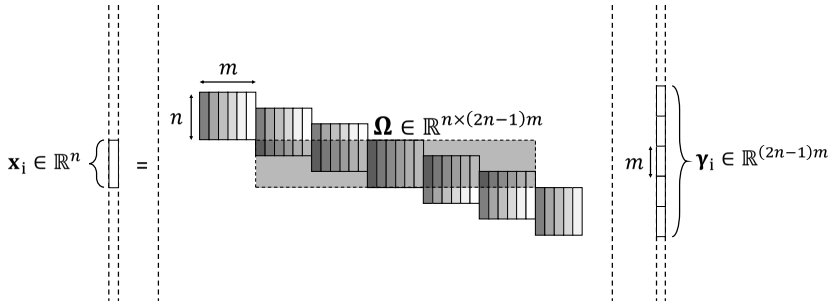


Figure 4: The i-th patch $\mathbf{x}_i$ of the global system $\mathbf{X} = \mathbf{D}\mathbf{\Gamma}$, given by $\mathbf{x}_i = \mathbf{\Omega}\boldsymbol{\gamma}_i$.

shifting a local matrix of size $n \times m$ in all possible positions, resulting in the same structure as the one shown in Figure 1a.

In the convolutional model, the classical theoretical guarantees (we are referring to results reported in (Chen et al., 2001; Donoho and Elad, 2003; Tropp, 2006)) for the P_0 problem, defined in Equation (3), are very pessimistic. In particular, the condition for the uniqueness of the underlying solution and the requirement for the success of the sparse coding algorithms depend on the global number of non-zeros being less than $\frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$. Following the Welch bound (Welch, 1974), this expression was shown in (Papyan et al., 2016a) to be impractical, allowing the global number of non-zeros in $\mathbf{\Gamma}$ to be extremely low.

In order to provide a better theoretical understanding of this model, which exploits the inherent structure of the convolutional dictionary, a recent work (Papyan et al., 2016a) suggested to measure the sparsity of $\mathbf{\Gamma}$ in a localized manner. More concretely, consider the i-th n-dimensional patch of the global system $\mathbf{X} = \mathbf{D}\mathbf{\Gamma}$, given by $\mathbf{x}_i = \mathbf{\Omega}\boldsymbol{\gamma}_i$. The stripe-dictionary $\mathbf{\Omega}$, which is of size $n \times (2n-1)m$, is obtained by extracting the i-th patch from the global dictionary $\mathbf{D}$ and discarding all the zero columns from it. The stripe vector $\boldsymbol{\gamma}_i$ is the corresponding sparse representation of length $(2n-1)m$, containing all coefficients of atoms contributing to $\mathbf{x}_i$. This relation is illustrated in Figure 4. Notably, the choice of a convolutional dictionary results in signals such that every patch of length n extracted from them can be sparsely represented using a single shift-invariant local dictionary $\mathbf{\Omega}$ – a common assumption usually employed in signal and image processing.

Following the above construction, the $\ell_{0,\infty}$ norm of the global sparse vector $\mathbf{\Gamma}$ is defined to be the maximal number of non-zeros in a stripe of length $(2n-1)m$ extracted from it. Formally,

$$\|\mathbf{\Gamma}\|_{0,\infty}^{\mathbf{s}} = \max_i \|\boldsymbol{\gamma}_i\|_0,$$

where the letter $\mathbf{s}$ emphasizes that the $\ell_{0,\infty}$ norm is computed by sweeping over all stripes. Given a signal $\mathbf{X}$, finding its sparest representation $\mathbf{\Gamma}$ in the $\ell_{0,\infty}$ sense is equal to the following optimization problem:

$$(\mathrm{P}_{0,\infty}): \quad \min_{\mathbf{\Gamma}} \quad \|\mathbf{\Gamma}\|_{0,\infty}^{\mathbf{s}} \quad \text{s.t.} \quad \mathbf{D}\mathbf{\Gamma} = \mathbf{X}. \tag{6}$$

Intuitively, this seeks for a global vector $\mathbf{\Gamma}$ that can represent sparsely every patch in the signal $\mathbf{X}$ using the dictionary $\mathbf{\Omega}$. The advantage of the above problem over the traditional P_0 becomes apparent as we move to consider its theoretical aspects. Assuming that the **number of non-zeros per stripe** (and not globally) in $\mathbf{\Gamma}$ is less than $\frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$, in (Papyan et al., 2016a) it was proven that the solution for the $\mathrm{P}_{0,\infty}$ problem is unique. Furthermore, classical pursuit methods, originally tackling the P_0 problem, are guaranteed to find this representation.

When modeling natural signals, due to measurement noise as well as model deviations, one can not impose a perfect reconstruction such as $\mathbf{X} = \mathbf{D}\mathbf{\Gamma}$ on the signal $\mathbf{X}$. Instead, one assumes $\mathbf{Y} = \mathbf{X} + \mathbf{E} = \mathbf{D}\mathbf{\Gamma} + \mathbf{E}$, where $\mathbf{E}$ is, for example, an ℓ_2-bounded error vector. To address this, the work reported in (Papyan et al., 2016b) considered the extension of the $\mathrm{P}_{0,\infty}$ problem to the $\mathrm{P}_{0,\infty}^{\boldsymbol{\mathcal{E}}}$ one, formally defined as

$$(\mathrm{P}_{0,\infty}^{\boldsymbol{\mathcal{E}}}): \quad \min_{\mathbf{\Gamma}} \|\mathbf{\Gamma}\|_{0,\infty}^{\mathbf{s}} \quad \text{s.t.} \quad \|\mathbf{Y} - \mathbf{D}\mathbf{\Gamma}\|_2^2 \leq \boldsymbol{\mathcal{E}}^2.$$

Similar to the $\mathrm{P}_{0,\infty}$ problem, this was also analyzed theoretically, shedding light on the theoretical aspects of the convolutional model in the presence of noise. In particular, a stability claim for the $\mathrm{P}_{0,\infty}^{\boldsymbol{\mathcal{E}}}$ problem and guarantees for the success of both the OMP and the BP were provided. Similar to the noiseless case, these assumed that the number of non-zeros per stripe is low.

3. From Atoms to Molecules: Multi-Layer Convolutional Sparse Model

Convolutional sparsity assumes an inherent structure for natural signals. Similarly, the representations themselves could also be assumed to have such a structure. In what follows, we propose a novel layered model that relies on this rationale.

The convolutional sparse model assumes a global signal $\mathbf{X} \in \mathbb{R}^N$ can be decomposed into a multiplication of a convolutional dictionary $\mathbf{D}_1 \in \mathbb{R}^{N\times Nm_1}$, composed of m_1 local filters of length n_0, and a sparse vector $\mathbf{\Gamma}_1 \in \mathbb{R}^{Nm_1}$. Herein, we extend this by proposing a similar factorization of the vector $\mathbf{\Gamma}_1$, which can be perceived as an N-dimensional global signal with m_1 channels. In particular, we assume $\mathbf{\Gamma}_1 = \mathbf{D}_2\mathbf{\Gamma}_2$, where $\mathbf{D}_2 \in \mathbb{R}^{Nm_1\times Nm_2}$ is a stride convolutional dictionary (skipping m_1 entries at a time) and $\mathbf{\Gamma}_2 \in \mathbb{R}^{Nm_2}$ is a sparse representation. We denote the number of unique filters constructing $\mathbf{D}_2$ by m_2 and their corresponding length by n_1m_1. Due to the multi-layer nature of this model and the imposed convolutional structure, we name this the ML-CSC model.

Intuitively, $\mathbf{X} = \mathbf{D}_1\mathbf{\Gamma}_1$ assumes that the signal $\mathbf{X}$ is a superposition of **atoms** taken from $\mathbf{D}_1$. While equation $\mathbf{X} = \mathbf{D}_1\mathbf{D}_2\mathbf{\Gamma}_2$ views the signal as a superposition of more complex entities taken from the dictionary $\mathbf{D}_1\mathbf{D}_2$, which we coin **molecules**.

While this proposal can be interpreted as a straightforward fusion between the double sparsity model (Rubinstein et al., 2010) and the convolutional one, it is in fact substantially different. The double sparsity model assumes that $\mathbf{D}_2$ is sparse, and forces **only** the deepest representation $\mathbf{\Gamma}_2$ to be sparse as well. Here, on the other hand, we replace this constraint by forcing $\mathbf{D}_2$ to have a stride convolution structure, putting emphasis on the sparsity of both the representations $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$. In Section 7.1 we will revisit the double sparsity work and its ties to ours by showing the benefits of injecting the assumption on the sparsity of $\mathbf{D}_2$ into our proposed model.

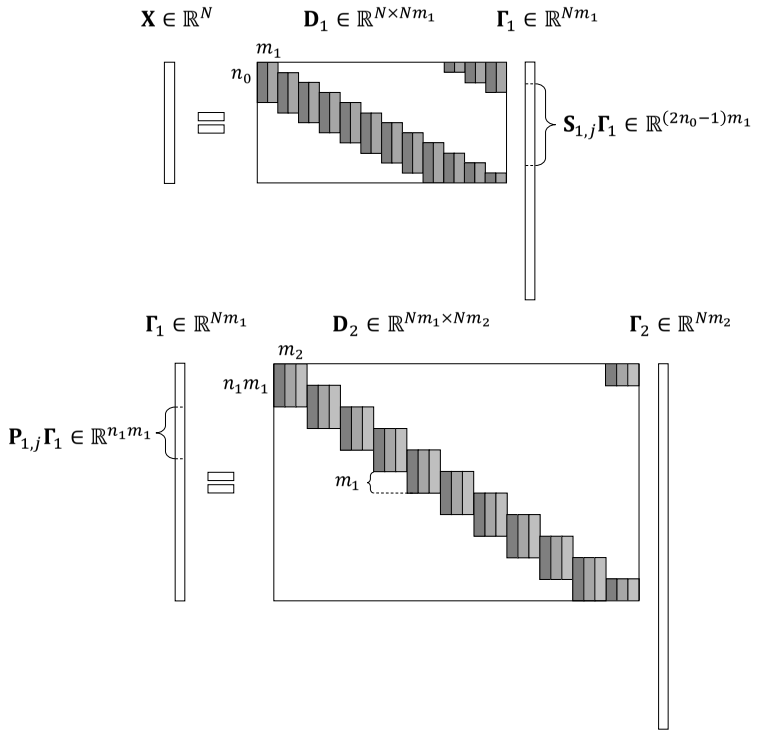


Figure 5: An instance $\mathbf{X} = \mathbf{D}_1\mathbf{\Gamma}_1 = \mathbf{D}_1\mathbf{D}_2\mathbf{\Gamma}_2$ of the ML-CSC model. Notice that $\mathbf{\Gamma}_1$ is built of both stripes $\mathbf{S}_{1,j}\mathbf{\Gamma}_1$ and patches $\mathbf{P}_{1,j}\mathbf{\Gamma}_1$.

Under the above construction the sparse vector $\mathbf{\Gamma}_1$ has two roles. In the context of the system of equations $\mathbf{X} = \mathbf{D}_1\mathbf{\Gamma}_1$, it is the convolutional sparse representation of the signal $\mathbf{X}$ over the dictionary $\mathbf{D}_1$. As such, the vector $\mathbf{\Gamma}_1$ is composed from $(2n_0 - 1)m_1$-dimensional stripes, $\mathbf{S}_{1,j}\mathbf{\Gamma}_1$, where $\mathbf{S}_{i,j}$ is the operator that extracts the j-th stripe from $\mathbf{\Gamma}_i$. From another point of view, $\mathbf{\Gamma}_1$ is in itself a signal that admits a sparse representation $\mathbf{\Gamma}_1 = \mathbf{D}_2\mathbf{\Gamma}_2$. Denoting by $\mathbf{P}_{i,j}$ the operator that extracts the j-th patch from $\mathbf{\Gamma}_i$, the signal $\mathbf{\Gamma}_1$ is composed of patches $\mathbf{P}_{1,j}\mathbf{\Gamma}_1$ of length $n_1 m_1$. The above model is depicted in Figure 5, presenting both roles of $\mathbf{\Gamma}_1$ and their corresponding constituents – stripes and patches. Clearly, the above construction can be extended to more than two layers, leading to the following definition:

Definition 1 *For a global signal $\mathbf{X}$, a set of convolutional dictionaries $\{\mathbf{D}_i\}_{i=1}^K$, and a vector $\boldsymbol{\lambda}$, define the deep coding problem $DCP_{\boldsymbol{\lambda}}$ as:*

$$(DCP_{\boldsymbol{\lambda}}): \quad \text{find} \quad \{\mathbf{\Gamma}_i\}_{i=1}^K \quad \text{s.t.} \quad \begin{aligned} \mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1, & \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} &\le \lambda_1 \\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2, & \|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} &\le \lambda_2 \\ &\vdots & &\vdots \\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K, & \|\mathbf{\Gamma}_K\|_{0,\infty}^{\mathbf{s}} &\le \lambda_K, \end{aligned}$$

where the scalar λ_i is the i-th entry of $\boldsymbol{\lambda}$.

Denoting $\mathbf{\Gamma}_0$ to be the signal $\mathbf{X}$, the $\text{DCP}_{\boldsymbol{\lambda}}$ can be rewritten compactly as

$$(\text{DCP}_{\boldsymbol{\lambda}}): \quad \text{find} \quad \{\mathbf{\Gamma}_i\}_{i=1}^K \quad \text{s.t.} \quad \mathbf{\Gamma}_{i-1} = \mathbf{D}_i\mathbf{\Gamma}_i, \quad \|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} \le \lambda_i, \quad \forall 1 \le i \le K.$$

Intuitively, given a signal $\mathbf{X}$, this problem seeks for a set of representations, $\{\mathbf{\Gamma}_i\}_{i=1}^K$, such that each one is locally sparse. As we shall see next, the above can be easily solved using simple algorithms that also enjoy from theoretical justifications. Next, we extend the $\text{DCP}_{\boldsymbol{\lambda}}$ problem to a noisy regime.

Definition 2 *For a global signal $\mathbf{Y}$, a set of convolutional dictionaries $\{\mathbf{D}_i\}_{i=1}^K$, and vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\mathcal{E}}$, define the deep coding problem $DCP_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ as:*

$$(DCP_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}): \quad \text{find} \quad \{\mathbf{\Gamma}_i\}_{i=1}^K \quad \text{s.t.} \quad \begin{aligned} \|\mathbf{Y} - \mathbf{D}_1\mathbf{\Gamma}_1\|_2 &\le \mathcal{E}_0, & \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} &\le \lambda_1 \\ \|\mathbf{\Gamma}_1 - \mathbf{D}_2\mathbf{\Gamma}_2\|_2 &\le \mathcal{E}_1, & \|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} &\le \lambda_2 \\ &\vdots & &\vdots \\ \|\mathbf{\Gamma}_{K-1} - \mathbf{D}_K\mathbf{\Gamma}_K\|_2 &\le \mathcal{E}_{K-1}, & \|\mathbf{\Gamma}_K\|_{0,\infty}^{\mathbf{s}} &\le \lambda_K, \end{aligned}$$

where the scalars λ_i and $\mathcal{E}_i$ are the i-th entry of $\boldsymbol{\lambda}$ and $\boldsymbol{\mathcal{E}}$, respectively.

We now move to the task of learning the model parameters. Denote by $\text{DCP}_{\boldsymbol{\lambda}}^{\star}(\mathbf{X}, \{\mathbf{D}_i\}_{i=1}^K)$ the representation $\mathbf{\Gamma}_K$ obtained by solving the DCP problem (Definition 1, i.e., noiseless) for the signal $\mathbf{X}$ and the set of dictionaries $\{\mathbf{D}_i\}_{i=1}^K$. Relying on this, we now extend the dictionary learning problem, as presented in Section 2.2.3, to the multi-layer convolutional sparse representation setting.

Definition 3 *For a set of global signals $\{\mathbf{X}_j\}_j$, their corresponding labels $\{h(\mathbf{X}_j)\}_j$, a loss function ℓ, and a vector $\boldsymbol{\lambda}$, define the deep learning problem $DLP_{\boldsymbol{\lambda}}$ as:*

$$(DLP_{\boldsymbol{\lambda}}): \quad \min_{\{\mathbf{D}_i\}_{i=1}^K, \mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j), \mathbf{U}, DCP_{\boldsymbol{\lambda}}^{\star}(\mathbf{X}_j, \{\mathbf{D}_i\}_{i=1}^K)\Big).$$

A clarification for the chosen name, deep learning problem, will be provided shortly. The solution for the above results in an end-to-end mapping, from a set of input signals to their corresponding labels. Similarly, we can define the $\text{DLP}_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problem. However, this is omitted for the sake of brevity. We conclude this section by summarizing, for the convenience of the reader, all notations used throughout this work in Table 1.

$\mathbf{X} = \mathbf{\Gamma}_0$	:	a global signal of length N.
$\mathbf{E}$, $\mathbf{Y} = \hat{\mathbf{\Gamma}}_0$	:	a global error vector and its corresponding noisy signal, where generally $\mathbf{Y} = \mathbf{X} + \mathbf{E}$.
K	:	the number of layers.
m_i	:	the number of local filters in $\mathbf{D}_i$, and also the number of channels in $\mathbf{\Gamma}_i$. Notice that $m_0 = 1$.
n_0	:	the size of a local patch in $\mathbf{X} = \mathbf{\Gamma}_0$.
$n_i,\ i \geq 1$	:	the size of a local patch (not including channels) in $\mathbf{\Gamma}_i$.
$n_i m_i$	:	the size of a local patch (including channels) in $\mathbf{\Gamma}_i$.
$\mathbf{D}_1$	:	a (full) convolutional dictionary of size $N \times Nm_1$ with filters of length n_0.
$\mathbf{D}_i,\ i \geq 2$	:	a convolutional dictionary of size $Nm_{i-1} \times Nm_i$ with filters of length $n_{i-1}m_{i-1}$ and a stride equal to m_{i-1}.
$\mathbf{\Gamma}_i$	:	a sparse vector of length Nm_i that is the representation of $\mathbf{\Gamma}_{i-1}$ over the dictionary $\mathbf{D}_i$, i.e. $\mathbf{\Gamma}_{i-1} = \mathbf{D}_i\mathbf{\Gamma}_i$.
$\mathbf{S}_{i,j}$	:	an operator that extracts the j-th stripe of length $(2n_{i-1}-1)m_i$ from $\mathbf{\Gamma}_i$.
$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{S}}$	:	the maximal number of non-zeros in a stripe from $\mathbf{\Gamma}_i$.
$\mathbf{P}_{i,j}$	:	an operator that extracts the j-th $n_i m_i$-dimensional patch from $\mathbf{\Gamma}_i$.
$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{P}}$	:	the maximal number of non-zeros in a patch from $\mathbf{\Gamma}_i$ (Definition 6).
$\mathbf{R}_{i,j}$	:	an operator that extracts the filter of length $n_{i-1}m_{i-1}$ from the j-th atom in $\mathbf{D}_i$.
$\|\mathbf{V}\|_{2,\infty}^{\mathbf{P}}$	:	the maximal ℓ_2 norm of a patch extracted from a vector $\mathbf{V}$ (Definition 6).

Table 1: Summary of notations used throughout the paper.

4. Layered Thresholding: The Crux of the Matter

Consider the ML-CSC model defined by the set of dictionaries $\{\mathbf{D}_i\}_{i=1}^K$. Assume we are given a signal

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1 \\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2 \\ &\vdots \\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K,\end{aligned}$$

and our goal is to find its underlying representations, $\{\mathbf{\Gamma}_i\}_{i=1}^K$. Tackling this problem by recovering all the vectors at once might be computationally and conceptually challenging; therefore, we propose the *layered thresholding algorithm* that gradually computes the sparse vectors one at a time across the different layers. Denoting by $\mathcal{P}_\beta(\cdot)$ a sparsifying operator that is equal to $\mathcal{H}_\beta(\cdot)$ in the hard thresholding case and $\mathcal{S}_\beta(\cdot)$ in the soft one; we commence by computing $\hat{\mathbf{\Gamma}}_1 = \mathcal{P}_{\beta_1}(\mathbf{D}_1^T\mathbf{X})$, which is an approximation of $\mathbf{\Gamma}_1$. Next, by applying another thresholding algorithm, however this time on $\hat{\mathbf{\Gamma}}_1$, an approximation of $\mathbf{\Gamma}_2$ is obtained, $\hat{\mathbf{\Gamma}}_2 = \mathcal{P}_{\beta_2}(\mathbf{D}_2^T\hat{\mathbf{\Gamma}}_1)$. This process, which is iterated until the last representation $\hat{\mathbf{\Gamma}}_K$ is acquired, is summarized in Algorithm 1.

One might ponder as to why does the application of the thresholding algorithm on the signal $\mathbf{X}$ not result in the true representation $\mathbf{\Gamma}_1$, but instead an approximation of it. As

Algorithm 1 The layered thresholding algorithm.

Input:

$\mathbf{X}$ – a signal.
$\{\mathbf{D}_i\}_{i=1}^K$ – convolutional dictionaries.
$\mathcal{P} \in \{\mathcal{H}, \mathcal{S}, \mathcal{S}^+\}$ – a thresholding operator.
$\{\beta_i\}_{i=1}^K$ – thresholds.

Output:

A set of representations $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$.

Process:

$\hat{\mathbf{\Gamma}}_0 \leftarrow \mathbf{X}$
for $i = 1 : K$ **do**
  $\hat{\mathbf{\Gamma}}_i \leftarrow \mathcal{P}_{\beta_i}(\mathbf{D}_i^T\hat{\mathbf{\Gamma}}_{i-1})$
end for

previously described in Section 2.2.1, assuming some conditions are met, the result of the thresholding algorithm, $\hat{\mathbf{\Gamma}}_1$, is guaranteed to have the correct support. In order to obtain the vector $\mathbf{\Gamma}_1$ itself, one should project the signal $\mathbf{X}$ onto this obtained support, by solving a Least-Squares problem. For reasons that will become clear shortly, we choose not to employ this step in the layered thresholding algorithm. Despite this algorithm failing in recovering the exact representations in the noiseless setting, as we shall see in Section 5, the estimated sparse vectors and the true ones are close – indicating the stability of this simple algorithm.

Thus far, we have assumed a noiseless setting. However, the same layered thresholding algorithm could be employed for the recovery of the representations of a noisy signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$, with the exception that the threshold constants, $\{\beta_i\}_{i=1}^K$, would be different and proportional to the noise level.

Assuming two layers for simplicity, Algorithm 1 can be summarized in the following equation

$$\hat{\mathbf{\Gamma}}_2 = \mathcal{P}_{\beta_2}\Big(\ \mathbf{D}_2^T\mathcal{P}_{\beta_1}\left(\ \mathbf{D}_1^T\mathbf{X}\ \right)\ \Big).$$

Comparing the above with Equation (1), given by

$$f(\mathbf{X}, \{\mathbf{W}_i\}_{i=1}^2, \{\mathbf{b}_i\}_{i=1}^2) = \text{ReLU}\Big(\ \mathbf{W}_2^T\ \text{ReLU}\left(\ \mathbf{W}_1^T\mathbf{X} + \mathbf{b}_1\ \right) + \mathbf{b}_2\ \Big),$$

one can notice a striking similarity between the two. Moreover, by replacing $\mathcal{P}_\beta(\cdot)$ with the soft nonnegative thresholding, $\mathcal{S}_\beta^+(\cdot)$, we obtain that *the aforementioned pursuit and the forward pass of the CNN are equal*! Notice that we are relying here on the discussion of Section 2.2.2, where we have shown that the ReLU and the soft nonnegative thresholding are equal[4].

4. A slight difference does exist between the soft nonnegative layered thresholding algorithm and the forward pass of the CNN. While in the former a constant threshold β is employed for all entries, the latter uses a bias vector, $\mathbf{b}$, that might not be constant in all of its entries. This is of little significance, however, since a similar approach of an entry-based constant could be used in the layered thresholding algorithm as well.

Recall the optimization problem of the training stage of the CNN as shown in Equation (2), given by

$$\min_{\{\mathbf{W}_i\}_{i=1}^K,\{\mathbf{b}_i\}_{i=1}^K,\mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j),\mathbf{U},f\left(\mathbf{X}_j,\{\mathbf{W}_i\}_{i=1}^K,\{\mathbf{b}_i\}_{i=1}^K\right)\Big),$$

and its parallel in the ML-CSC model, the DLP$_{\boldsymbol{\lambda}}$ problem, defined by

$$\min_{\{\mathbf{D}_i\}_{i=1}^K,\mathbf{U}} \sum_j \ell\Big(h(\mathbf{X}_j),\mathbf{U},\mathrm{DCP}^\star_{\boldsymbol{\lambda}}(\mathbf{X}_j,\{\mathbf{D}_i\}_{i=1}^K)\Big).$$

Notice the remarkable similarity between both objectives, the only difference being in the feature vector on which the classification is done; in the CNN this is the output of the forward pass algorithm, given by $f\left(\mathbf{X}_j,\{\mathbf{W}_i\}_{i=1}^K,\{\mathbf{b}_i\}_{i=1}^K\right)$, while in the sparsity case this is the result of the DCP$_{\boldsymbol{\lambda}}$ problem. In light of the discussion above, the solution for the DCP$_{\boldsymbol{\lambda}}$ problem can be approximated using the layered thresholding algorithm, which is in turn equal to the forward pass of the CNN. We can therefore conclude that the problems solved by the training stage of the CNN and the DLP$_{\boldsymbol{\lambda}}$ are tightly connected, and in fact are equal once the solution for the DLP$_{\boldsymbol{\lambda}}$ is approximated via the layered thresholding algorithm (hence the name DLP$_{\boldsymbol{\lambda}}$).

5. Theoretical Study

Thus far, we have defined the ML-CSC model and its corresponding pursuits – the DCP$_{\boldsymbol{\lambda}}$ and DCP$^{\boldsymbol{\mathcal{E}}}_{\boldsymbol{\lambda}}$ problems. We have proposed a method to tackle them, coined the layered thresholding algorithm, which was shown to be equivalent to the forward pass of the CNN. Relying on this, we conclude that the proposed ML-CSC is the global Bayesian model implicitly imposed on the signal $\mathbf{X}$ when deploying the forward pass algorithm. Put differently, the ML-CSC answers the question of who are the signals belonging to the model behind the CNN. Having established the importance of our model, we now proceed to its theoretical analysis.

We should emphasize that the following study does not assume any specific form on the network's parameters, apart from a broad coherence property (as will be shown hereafter). This is in contrast to the work of (Bruna and Mallat, 2013) that assumes that the filters are Wavelets, or the analysis in (Giryes et al., 2015) that considers random weights.

5.1 Uniqueness of the DCP$_{\boldsymbol{\lambda}}$ Problem

Consider a signal $\mathbf{X}$ admitting a multi-layer convolutional sparse representation defined by the sets $\{\mathbf{D}_i\}_{i=1}^K$ and $\{\lambda_i\}_{i=1}^K$. Can another set of sparse vectors represent the signal $\mathbf{X}$? In other words, can we guarantee that, under some conditions, the set $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$ is a unique solution to the DCP$_{\boldsymbol{\lambda}}$ problem? In the following theorem we provide an answer to this question.

Theorem 4 *(Uniqueness via the mutual coherence): Consider a signal* $\mathbf{X}$ *satisfying the DCP*$_{\boldsymbol{\lambda}}$ *model,*

$$\begin{aligned}\mathbf{X}&=\mathbf{D}_1\boldsymbol{\Gamma}_1\\ \boldsymbol{\Gamma}_1&=\mathbf{D}_2\boldsymbol{\Gamma}_2\\ &\vdots\\ \boldsymbol{\Gamma}_{K-1}&=\mathbf{D}_K\boldsymbol{\Gamma}_K,\end{aligned}$$

where $\{\mathbf{D}_i\}_{i=1}^K$ *is a set of convolutional dictionaries and* $\{\mu(\mathbf{D}_i)\}_{i=1}^K$ *are their corresponding mutual coherences. If*

$$\forall\ 1\le i\le K \qquad \|\boldsymbol{\Gamma}_i\|^{\mathbf{s}}_{0,\infty}<\frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right),$$

then the set $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$ *is the unique solution to the DCP*$_{\boldsymbol{\lambda}}$ *problem, assuming that the thresholds* $\{\lambda_i\}_{i=1}^K$ *are chosen to satisfy*

$$\forall\ 1\le i\le K \qquad \|\boldsymbol{\Gamma}_i\|^{\mathbf{s}}_{0,\infty}\le\lambda_i<\frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right).$$

The proof for the above theorem is given in Appendix A. In what follows, we present its importance in the context of CNN. Assume a signal $\mathbf{X}$ is fed into a network, resulting in a set of activation values across the different layers. These, in the realm of sparsity, correspond to the set of sparse representations $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$, which according to the above theorem are in fact unique representations of the signal $\mathbf{X}$.

One might ponder at this point whether there exists an algorithm for obtaining the unique solution guaranteed in this subsection for the DCP$_{\boldsymbol{\lambda}}$ problem. As previously mentioned, the layered thresholding algorithm is incapable of finding the exact representations, $\{\boldsymbol{\Gamma}_i\}_{i=1}^K$, due to the lack of a Least-Squares step after each layer. One should not despair, however, as we shall see in a following section an alternative algorithm, which manages to overcome this hurdle.

5.2 Global Stability of the DCP$^{\boldsymbol{\mathcal{E}}}_{\boldsymbol{\lambda}}$ Problem

Consider an instance signal $\mathbf{X}$ belonging to the ML-CSC model, defined by the sets $\{\mathbf{D}_i\}_{i=1}^K$ and $\{\lambda_i\}_{i=1}^K$. Assume $\mathbf{X}$ is contaminated by a noise vector $\mathbf{E}$, generating the perturbed signal $\mathbf{Y}=\mathbf{X}+\mathbf{E}$. Suppose we solve the DCP$^{\boldsymbol{\mathcal{E}}}_{\boldsymbol{\lambda}}$ problem and obtain a set of solutions $\{\hat{\boldsymbol{\Gamma}}_i\}_{i=1}^K$. How close is every solution in this set, $\hat{\boldsymbol{\Gamma}}_i$, to its corresponding true representation, $\boldsymbol{\Gamma}_i$? In what follows, we provide a theorem addressing this question of stability, the proof of which is deferred to Appendix B.

Theorem 5 *(Stability of the solution to the DCP*$^{\boldsymbol{\mathcal{E}}}_{\boldsymbol{\lambda}}$ *problem): Suppose a signal* $\mathbf{X}$ *that has a decomposition*

$$\begin{aligned}\mathbf{X}&=\mathbf{D}_1\boldsymbol{\Gamma}_1\\ \boldsymbol{\Gamma}_1&=\mathbf{D}_2\boldsymbol{\Gamma}_2\\ &\vdots\\ \boldsymbol{\Gamma}_{K-1}&=\mathbf{D}_K\boldsymbol{\Gamma}_K\end{aligned}$$

is contaminated with noise $\mathbf{E}$*, resulting in* $\mathbf{Y}=\mathbf{X}+\mathbf{E}$*. Assume we solve the DCP*$^{\boldsymbol{\mathcal{E}}}_{\boldsymbol{\lambda}}$ *problem for* $\mathcal{E}_0=\|\mathbf{E}\|_2$ *and* $\mathcal{E}_i=0 \quad \forall 1\leq i\leq K$*, obtaining a set of solutions* $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$*. If for all* $1\leq i\leq K$

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} \leq \lambda_i < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right),$$

then

$$\|\mathbf{\Gamma}_i-\hat{\mathbf{\Gamma}}_i\|_2^2 \leq 4\|\mathbf{E}\|_2^2 \prod_{j=1}^{i}\frac{1}{1-(2\lambda_j-1)\mu(\mathbf{D}_j)}.$$

Intuitively, the above claims that as long as all the feature vectors $\{\mathbf{\Gamma}_i\}_{i=1}^K$ are $\ell_{0,\infty}$-sparse, then the representations obtained by solving the DCP$^{\boldsymbol{\mathcal{E}}}_{\boldsymbol{\lambda}}$ problem must be close to the true ones. Interestingly, the obtained bound increases as a function of the depth of the layer.

Is this necessarily the true behavior of a deep network? Perhaps the answer to this resides in the choice we made above of considering the noise as adversary. A similar, yet somewhat more involved, analysis with a random noise assumption should be done, with the hope to see a better controlled noise propagation in this system. We leave this for our future work.

Another important remark is that the above bounds the *absolute error* between the estimated and the true representation. In practice, however, the *relative error* is of more importance. This is measured in terms of the signal to noise ratio (SNR), which we shall define in Section 8.

Having established the stability of the DCP$^{\boldsymbol{\mathcal{E}}}_{\boldsymbol{\lambda}}$ problem, we now turn to the stability of the algorithms attempting to solve it, the chief one being the forward pass of CNN.

5.3 Stability of the Layered Hard Thresholding

Consider a signal $\mathbf{X}$ that admits a multi-layer convolutional sparse representation, which is defined by the sets $\{\mathbf{D}_i\}_{i=1}^K$ and $\{\lambda_i\}_{i=1}^K$. Assume we run the layered hard thresholding algorithm on $\mathbf{X}$, obtaining the sparse vectors $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$. Under certain conditions, can we guarantee that the estimate $\hat{\mathbf{\Gamma}}_i$ recovers the true support of $\mathbf{\Gamma}_i$? or that the norm of the difference between the two is bounded? Assume $\mathbf{X}$ is contaminated with a noise vector $\mathbf{E}$, resulting in the measurement $\mathbf{Y}=\mathbf{X}+\mathbf{E}$. Assume further that this signal is then fed to the layered thresholding algorithm, resulting in another set of representations. How do the answers to the above questions change? To tackle these, we commence by presenting a stability claim for the simple hard thresholding algorithm, relying on the $\ell_{0,\infty}$ norm. We should note that the analysis conducted in this subsection is for the noisy scenario, and the results for the noiseless case are simply obtained by setting the noise level to zero.

Next, we present a localized ℓ_2 and ℓ_0 measure of a global vector that will prove to be useful in the following analysis.

Definition 6 *Define the* $\|\cdot\|_{2,\infty}^{\mathbf{p}}$ *and* $\|\cdot\|_{0,\infty}^{\mathbf{p}}$ *norm of* $\mathbf{\Gamma}_i$ *to be*

$$\|\mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{p}} = \max_j \|\mathbf{P}_{i,j}\mathbf{\Gamma}_i\|_2$$

and

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{p}} = \max_j \|\mathbf{P}_{i,j}\mathbf{\Gamma}_i\|_0,$$

respectively. The operator $\mathbf{P}_{i,j}$ *extracts the* j*-th patch of length* $n_i m_i$ *from the* i*-th sparse vector* $\mathbf{\Gamma}_i$*.*

In the above definition, the letter $\mathbf{p}$ emphasizes that the norms are computed by sweeping over all patches, rather than stripes. Recall that we have defined $m_0=1$, since the number of channels in the input signal $\mathbf{X}=\mathbf{\Gamma}_0$ is equal to one.

Given $\mathbf{Y}=\mathbf{X}+\mathbf{E}=\mathbf{D}_1\mathbf{\Gamma}_1+\mathbf{E}$, the first stage of the layered hard thresholding algorithm attempts to recover the representation $\mathbf{\Gamma}_1$. Intuitively, assuming that the underlying representation $\mathbf{\Gamma}_1$ is $\ell_{0,\infty}$-sparse, and that the energy of the noise $\mathbf{E}$ is $\ell_{2,\infty}$-bounded; we would expect that the simple hard thresholding algorithm would succeed in recovering a solution $\hat{\mathbf{\Gamma}}_1$, which is both close to $\mathbf{\Gamma}_1$ and has its support. We now present such a claim, the proof of which is found in Appendix C.

Lemma 7 *(Stable recovery of hard thresholding in the presence of noise): Suppose a clean signal* $\mathbf{X}$ *has a convolutional sparse representation* $\mathbf{D}_1\mathbf{\Gamma}_1$*, and that it is contaminated with noise* $\mathbf{E}$ *to create the signal* $\mathbf{Y}=\mathbf{X}+\mathbf{E}$*, such that* $\|\mathbf{E}\|_{2,\infty}^{\mathbf{p}}\leq\epsilon_0$*. Denote by* $|\Gamma_1^{min}|$ *and* $|\Gamma_1^{max}|$ *the lowest and highest entries in absolute value in* $\mathbf{\Gamma}_1$*, respectively. Denote further by* $\hat{\mathbf{\Gamma}}_1$ *the solution obtained by running the hard thresholding algorithm on* $\mathbf{Y}$ *with a constant* β_1*, i.e.* $\hat{\mathbf{\Gamma}}_1=\mathcal{H}_{\beta_1}(\mathbf{D}_1^T\mathbf{Y})$*. Assuming that*

a) $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_1)}\frac{|\Gamma_1^{min}|}{|\Gamma_1^{max}|}\right)-\frac{1}{\mu(\mathbf{D}_1)}\frac{\epsilon_0}{|\Gamma_1^{max}|}$*; and*

b) The threshold β_1 *is chosen according to Equation* (14) *(see below),*

then the following must hold:

1. *The support of the solution* $\hat{\mathbf{\Gamma}}_1$ *is equal to that of* $\mathbf{\Gamma}_1$*; and*
2. $\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{p}} \leq \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{p}}}\Big(\epsilon_0+\mu(\mathbf{D}_1)\left(\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1\right)|\Gamma_1^{max}|\Big)$.

Notice that by plugging $\epsilon_0=0$ the above theorem covers the noiseless scenario. Notably, even in such a case, we obtain a deviation from the true representation due to the lack of a Least-Squares step.

We suspect that, both in the noiseless and the noisy case, the obtained bound might be improved, based on the following observation. Given an $\ell_{2,\infty}$-norm bounded noise, the above proof first quantifies the deviation between the true representation and the estimated one in terms of the ℓ_∞ norm, and only then translates the latter into the $\ell_{2,\infty}$ sense. A direct analysis going from an $\ell_{2,\infty}$ input error to an $\ell_{2,\infty}$ output deviation (bypassing the ℓ_∞ norm) might lead to smaller deviations. We leave this for future work.

We now proceed to the next layer. Given $\hat{\mathbf{\Gamma}}_1$, which can be considered as a perturbed version of $\mathbf{\Gamma}_1$, the second stage of the layered hard thresholding algorithm attempts to recover the representation $\mathbf{\Gamma}_2$. Using the stability of the first layer – guaranteeing that $\mathbf{\Gamma}_1$ and $\hat{\mathbf{\Gamma}}_1$ are close in terms of the $\ell_{2,\infty}$ norm – and relying on the $\ell_{0,\infty}$-sparsity of $\mathbf{\Gamma}_2$, we show next that the second stage of the layered hard thresholding algorithm is stable as well. Applying the same rationale to all the remaining layers, we obtain the theorem below guaranteeing the stability of the complete layered hard thresholding algorithm.

Theorem 8 *(Stability of layered hard thresholding in the presence of noise): Suppose a clean signal* $\mathbf{X}$ *has a decomposition*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1\\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2\\ &\vdots\\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K,\end{aligned}$$

and that it is contaminated with noise $\mathbf{E}$ *to create the signal* $\mathbf{Y} = \mathbf{X}+\mathbf{E}$*, such that* $\|\mathbf{E}\|_{2,\infty}^{\mathbf{p}} \leq \epsilon_0$*. Denote by* $|\Gamma_i^{min}|$ *and* $|\Gamma_i^{max}|$ *the lowest and highest entries in absolute value in the vector* $\mathbf{\Gamma}_i$*, respectively. Let* $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ *be the set of solutions obtained by running the layered hard thresholding algorithm with thresholds* $\{\beta_i\}_{i=1}^K$*, i.e.* $\hat{\mathbf{\Gamma}}_i = \mathcal{H}_{\beta_i}(\mathbf{D}_i^T\hat{\mathbf{\Gamma}}_{i-1})$ *where* $\hat{\mathbf{\Gamma}}_0 = \mathbf{Y}$*. Assuming that* $\forall\ 1 \leq i \leq K$

a) $\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\frac{|\Gamma_i^{min}|}{|\Gamma_i^{max}|}\right) - \frac{1}{\mu(\mathbf{D}_i)}\frac{\epsilon_{i-1}}{|\Gamma_i^{max}|}$*; and*

b) The threshold β_i *is chosen according to Equation* (19)*,*

then[5]

1. *The support of the solution* $\hat{\mathbf{\Gamma}}_i$ *is equal to that of* $\mathbf{\Gamma}_i$*; and*
2. $\|\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_i\|_{2,\infty}^{\mathbf{p}} \leq \epsilon_i$*,*

where $\epsilon_i = \sqrt{\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{p}}}\left(\epsilon_{i-1} + \mu(\mathbf{D}_i)\left(\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} - 1\right)|\Gamma_i^{max}|\right)$*.*

The proof for the above is given in Appendix D. We now turn to an analogous theorem for the forward pass of the CNN, prior to discussing the surprising implications of these theorems.

5.4 Stability of the Forward Pass (Layered Soft Thresholding)

In light of the discussion in Section 4, the equivalence between the layered thresholding algorithm and the forward pass of the CNN is achieved assuming that the operator employed is the nonnegative soft thresholding $\mathcal{S}_\beta^+(\cdot)$. However, thus far, we have analyzed the closely related hard version $\mathcal{H}_\beta(\cdot)$ instead. In what follows, we show how the stability theorem presented in the previous subsection can be modified to the soft version, $\mathcal{S}_\beta(\cdot)$. For simplicity, and in order to stay in line with the vast sparse representation theory, herein we choose not to assume the nonnegative assumption. This implies that we are proposing a slightly different CNN architecture in which the ReLU function is two sided (Kavukcuoglu et al., 2010). We now move to the stable recovery of the soft thresholding algorithm.

5. Recall that $\|\mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{p}}$ is defined to be the maximal ℓ_2 norm of a patch extract from $\mathbf{\Gamma}_i$. The size of this patch is defined according to the dictionary $\mathbf{D}_{i+1}$. However, the last sparse vector $\mathbf{\Gamma}_K$ does not have a corresponding dictionary $\mathbf{D}_{K+1}$. As such, the size of a patch in $\mathbf{\Gamma}_K$ can be chosen arbitrarily. Where the choice of the size directly affects the bound on the difference, ϵ_i, due to the term $\sqrt{\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{p}}}$.

Lemma 9 *(Stable recovery of soft thresholding in the presence of noise): Suppose a clean signal* $\mathbf{X}$ *has a convolutional sparse representation* $\mathbf{D}_1\mathbf{\Gamma}_1$*, and that it is contaminated with noise* $\mathbf{E}$ *to create the signal* $\mathbf{Y} = \mathbf{X} + \mathbf{E}$*, such that* $\|\mathbf{E}\|_{2,\infty}^{\mathbf{p}} \leq \epsilon_0$*. Denote by* $|\Gamma_1^{min}|$ *and* $|\Gamma_1^{max}|$ *the lowest and highest entries in absolute value in* $\mathbf{\Gamma}_1$*, respectively. Denote further by* $\hat{\mathbf{\Gamma}}_1$ *the solution obtained by running the soft thresholding algorithm on* $\mathbf{Y}$ *with a constant* β_1*, i.e.* $\hat{\mathbf{\Gamma}}_1 = \mathcal{S}_{\beta_1}(\mathbf{D}_1^T\mathbf{Y})$*. Assuming that*

a) $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_1)}\frac{|\Gamma_1^{min}|}{|\Gamma_1^{max}|}\right) - \frac{1}{\mu(\mathbf{D}_1)}\frac{\epsilon_0}{|\Gamma_1^{max}|}$*; and*

b) The threshold β_1 *is chosen according to Equation* (14)*,*

then the following must hold:

1. *The support of the solution* $\hat{\mathbf{\Gamma}}_1$ *is equal to that of* $\mathbf{\Gamma}_1$*; and*
2. $\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{p}} \leq \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{p}}}\left(\epsilon_0 + \mu(\mathbf{D}_1)\left(\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} - 1\right)|\Gamma_1^{max}| + \beta_1\right)$*.*

Armed with the above lemma, which is proven in Appendix E, we now proceed to the stability of the forward pass of the CNN.

Theorem 10 *(Stability of the forward pass (layered soft thresholding algorithm) in the presence of noise): Suppose a clean signal* $\mathbf{X}$ *has a decomposition*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1\\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2\\ &\vdots\\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K,\end{aligned}$$

and that it is contaminated with noise $\mathbf{E}$ *to create the signal* $\mathbf{Y} = \mathbf{X}+\mathbf{E}$*, such that* $\|\mathbf{E}\|_{2,\infty}^{\mathbf{p}} \leq \epsilon_0$*. Denote by* $|\Gamma_i^{min}|$ *and* $|\Gamma_i^{max}|$ *the lowest and highest entries in absolute value in the vector* $\mathbf{\Gamma}_i$*, respectively. Let* $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ *be the set of solutions obtained by running the layered soft thresholding algorithm with thresholds* $\{\beta_i\}_{i=1}^K$*, i.e.* $\hat{\mathbf{\Gamma}}_i = \mathcal{S}_{\beta_i}(\mathbf{D}_i^T\hat{\mathbf{\Gamma}}_{i-1})$ *where* $\hat{\mathbf{\Gamma}}_0 = \mathbf{Y}$*. Assuming that* $\forall\ 1 \leq i \leq K$

a) $\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D}_i)}\frac{|\Gamma_i^{min}|}{|\Gamma_i^{max}|}\right) - \frac{1}{\mu(\mathbf{D}_i)}\frac{\epsilon_{i-1}}{|\Gamma_i^{max}|}$*; and*

b) The threshold β_i *is chosen according to Equation* (19) *(with the* ϵ_i *defined below),*

then

1. *The support of the solution* $\hat{\mathbf{\Gamma}}_i$ *is equal to that of* $\mathbf{\Gamma}_i$*; and*
2. $\|\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_i\|_{2,\infty}^{\mathbf{p}} \leq \epsilon_i$*,*

where $\epsilon_i = \sqrt{\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{p}}}\left(\epsilon_{i-1} + \mu(\mathbf{D}_i)\left(\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} - 1\right)|\Gamma_i^{max}| + \beta_i\right)$*.*

The proof for the above is omitted since it is tantamount to that of Theorem 8. As one can see, the layered soft thresholding algorithm is in fact inferior to its hard variant due to the added constant of β_i in the local error level, ϵ_i. This results in a more strict assumption on the $\ell_{0,\infty}$ norm of the various representations and also augments the bound on the distance between the true sparse vector and the one recovered. Following this observation, a natural question arises; why does the deep learning community employ the ReLU, which corresponds to a soft nonnegative thresholding operator instead of another nonlinearity that is more similar to its hard counterpart? One possible explanation could be that the filter training stage of the CNN becomes harder when the ReLU is replaced with a non-convex alternative, which also has discontinuities, such as the hard thresholding operator.

The above theorem guarantees that the distances between the original representations and the ones obtained from the CNN are bounded. Even if we set $\epsilon_0 = 0$, the recovered activations deviate from the true ones, simply because the layered thresholding algorithm does not do a perfect job, even on a noiseless signal. When the signal is noisy, these deviations are strengthened, but still in a controlled way.

This, by itself, might not be surprising. After all, the CNN is a deterministic system of linear operations (convolutions), followed by simple non-linearities that are non-expanding. If we feed a slightly perturbed signal to such a system, it is clear that the activations all along the network will be perturbed as well with a bounded effect. However, the above theorem shows far more than that. There are, in fact, two types of stabilities, the trivial one that considers the sensitivity of the whole feed-forward network to perturbations in its input, and the more intricate one that shows that this system enables a rather accurate recovery of the **generating representations**. The second option is the stability we prove here.

5.5 Guarantees for Fully Connected Networks

One should note that the convolutional structure imposed on the dictionaries in our model could be removed, and the theoretical guarantees we have provided above would still hold. The reason being is that the unconstrained dictionary can be regarded as a convolutional one, constructed from a single shift of a local matrix with no circular boundary. In the context of CNN, this is analogous to a fully connected layer. As such, the theoretical analysis provided here sheds light on both convolutional and fully connected networks. A different point of view on the same matter can also be proposed; fully connected layers can be viewed as convolutional ones with filters that cover their entire input (Long et al., 2015).

5.6 Worst-Case Analysis

The proposed analysis takes a worst-case point of view, where the noise is adversary (rather than random), the number of nonzeros in each stripe is maximal and the characteristics of the dictionaries are simple. Specifically, the success and stability guarantees are given in terms of the mutual coherences of the dictionaries. In classic sparse theory it is known that this measure is pessimistic. Still, it is widely used in proving guarantees for sparse representations, perhaps because it is simple and intuitive to grasp.

Sharper bounds, relying on stronger characterizations of the dictionary, result in significantly harder analysis. One example for a different characterization is that of the average mutual coherence – defined to be the average correlation in absolute value between two distinct atoms taken from the dictionary (instead of the highest correlation, as measured by the original mutual coherence). From a theoretical point of view, this measure was shown to lead to better theoretical guarantees in classic sparse theory (Bajwa et al., 2012). In addition, from a practical point of view, it was proven beneficial penalizing over this quantity in compressed sensing applications (Elad, 2007).

Our analysis did not rely on the average mutual coherence, but rather on the maximal coherence. Still, the two are closely related, and following the discussion above we believe that this might predict better the performance of CNN in practice. Interestingly, the work of (Shang, 2015) measured the average mutual coherences of the different layers in the "all-conv" network, which was trained on the ImageNet dataset (Springenberg et al., 2014). The authors found that most layers have a low average mutual coherence.

6. Layered Basis Pursuit

The stability analysis presented above unveils two significant limitations of the forward pass of the CNN. First, this algorithm is incapable of recovering the unique solution for the $\text{DCP}_{\boldsymbol{\lambda}}$ problem, the existence of which is guaranteed from Theorem 4. This acts against our expectations, since in the traditional sparsity inspired model it is a well known fact that such a unique representation can be retrieved, assuming certain conditions are met.

The second issue is with the condition for the successful recovery of the true support. The $\ell_{0,\infty}$ norm of the true solution, $\boldsymbol{\Gamma}_i$, is required to be less than an expression that depends on the term $|\Gamma_i^{\min}|/|\Gamma_i^{\max}|$. The dependence on this ratio is a direct consequence of the forward pass algorithm relying on the simple thresholding operator that is known for having such a theoretical limitation[6]. However, alternative pursuits whose success would not depend on this ratio could be proposed, as indeed was done in the Sparse-Land model; resulting in both theoretical and practical benefits.

A solution for the first problem, already presented throughout this work, is a two-stage approach. First, run the thresholding operator in order to recover the correct support. Then, once the atoms are chosen, their corresponding coefficients can be obtained by solving a linear system of equations. In addition to retrieving the true representation in the noiseless case, this step can also be beneficial in the noisy scenario, resulting in a solution closer to the underlying one. However, since no such step exists in current CNN architectures, we refrain from further analyzing its theoretical implications.

Next, we present an alternative to the layered soft thresholding algorithm, which will tackle both of the aforementioned problems. Recall that the result of the soft thresholding is a simple approximation of the solution for the P_1 problem, previously defined in Equation (4). In every layer, instead of applying a simple thresholding operator that estimates the sparse vector by computing $\hat{\boldsymbol{\Gamma}}_i = \mathcal{S}_{\beta_i}(\mathbf{D}_i^T \hat{\boldsymbol{\Gamma}}_{i-1})$; we propose to tackle the full pursuit, i.e. to

6. The dependence on the ratio is also a direct consequence of assuming a worst-case analysis. Perhaps in reality this ratio does not play such a critical role.

minimize

$$\hat{\mathbf{\Gamma}}_i = \arg\min_{\mathbf{\Gamma}_i} \|\mathbf{\Gamma}_i\|_1 \quad \text{s.t.} \quad \hat{\mathbf{\Gamma}}_{i-1} = \mathbf{D}_i\mathbf{\Gamma}_i. \tag{7}$$

Notice that one could readily obtain the nonnegative sparse coding problem by simply adding an extra constraint in the above equation, forcing the coefficients in $\mathbf{\Gamma}_i$ to be nonnegative. More generally, Equation (7) can be written in its Lagrangian formulation

$$\hat{\mathbf{\Gamma}}_i = \arg\min_{\mathbf{\Gamma}_i} \xi_i\|\mathbf{\Gamma}_i\|_1 + \frac{1}{2}\|\mathbf{D}_i\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_{i-1}\|_2^2, \tag{8}$$

where the constant ξ_i is proportional to the noise level and should tend to zero in the noiseless scenario. We name the above the *layered basis pursuit* (BP) algorithm. In practice, one possible method for solving it is the iterative soft thresholding (IST). Formally, this obtains the minimizer of Equation (8) by repeating the following recursive formula

$$\hat{\mathbf{\Gamma}}_i^t = \mathcal{S}_{\xi_i/c_i}\left(\hat{\mathbf{\Gamma}}_i^{t-1} + \frac{1}{c_i}\mathbf{D}_i^T\left(\hat{\mathbf{\Gamma}}_{i-1} - \mathbf{D}_i\hat{\mathbf{\Gamma}}_i^{t-1}\right)\right), \tag{9}$$

where $\hat{\mathbf{\Gamma}}_i^t$ is the estimate of $\mathbf{\Gamma}_i$ at iteration t. The above can be interpreted as a simple projected gradient descent algorithm, where the constant c_i is inversely proportional to its step size. As a result, if c_i is chosen to be large enough[7], the above algorithm is guaranteed to converge to its global minimum that is the solution of (8), as was shown in (Daubechies et al., 2004). The method obtained by gradually computing the set of sparse representations, $\{\mathbf{\Gamma}_i\}_{i=1}^K$, via the IST is summarized in Algorithm 2 and named *layered iterative soft thresholding*. Notice that this algorithm coincides with the simple layered soft thresholding if it is run for a single iteration with $c_i = 1$ and initialized with $\hat{\mathbf{\Gamma}}_i^0 = \mathbf{0}$. This implies that the above algorithm is a natural extension to the forward pass of the CNN. Moreover, the above is similar to the approach taken in the work of deconvolutional networks (Zeiler et al., 2010), where the authors suggested to solve a sequence of BP problems across different layers of abstraction.

With respect to the computational aspects of the IST algorithm, the work of (Gregor and LeCun, 2010) proposed the LISTA method, showing how the number of iterations required by the IST to convergence can be reduced using neural networks. Later, (Giryes et al., 2016) proved the theoretical justification for this algorithm, and (Sprechmann et al., 2015) extended LISTA to other sparse and low-rank problems. Analogously, the work of (Xin et al., 2016) presented a generalization for the iterative hard thresholding (IHT), which was shown to be both theoretically and empirically superior to the original IHT.

The original motivation for the layered IST was its theoretical superiority over the forward pass algorithm – one that will be explored in detail in the next subsection. Yet more can be said about this algorithm and the CNN architecture it induces. In (Gregor and LeCun, 2010) it was shown that the IST algorithm can be formulated as a simple recurrent neural network. As such, the same can be said regarding the layered IST algorithm proposed here, with the exception that the induced recurrent network is much deeper. The reader can

7. The constant c_i should satisfy $c_i > 0.5\lambda_{\max}\left(\mathbf{D}_i^T\mathbf{D}_i\right)$, where $\lambda_{\max}\left(\mathbf{D}_i^T\mathbf{D}_i\right)$ is the maximal eigenvalue of the gram matrix $\mathbf{D}_i^T\mathbf{D}_i$ (Combettes and Wajs, 2005).

Algorithm 2 The layered iterative soft thresholding algorithm.

Input:

$\mathbf{X}$ – a signal.
$\{\mathbf{D}_i\}_{i=1}^K$ – convolutional dictionaries.
$\mathcal{P} \in \{\mathcal{S}, \mathcal{S}^+\}$ – a soft thresholding operator.
$\{\xi_i\}_{i=1}^K$ – Lagrangian parameters.
$\{1/c_i\}_{i=1}^K$ – step sizes.
$\{T_i\}_{i=1}^K$ – number of iterations.

Output:

A set of representations $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$.

Process:

$\hat{\mathbf{\Gamma}}_0 \leftarrow \mathbf{X}$
for $i = 1 : K$ **do**
  $\hat{\mathbf{\Gamma}}_i^0 \leftarrow \mathbf{0}$
  for $t = 1 : T_i$ **do**
    $\hat{\mathbf{\Gamma}}_i^t \leftarrow \mathcal{P}_{\xi_i/c_i}\left(\hat{\mathbf{\Gamma}}_i^{t-1} + \frac{1}{c_i}\mathbf{D}_i^T\left(\hat{\mathbf{\Gamma}}_{i-1} - \mathbf{D}_i\hat{\mathbf{\Gamma}}_i^{t-1}\right)\right)$
  end for
  $\hat{\mathbf{\Gamma}}_i \leftarrow \hat{\mathbf{\Gamma}}_i^{T_i}$
end for

therefore interpret this part of the work as a theoretical study of a special case of recurrent neural networks.

From another perspective, the underlying architecture of the layered IST algorithm is a cascade of K blocks. Each of these corresponds to a fixed number of unfolded iterations, T_i, of a single IST algorithm. These unfolded iterations attempt to estimate better the representation after the initial thresholding operator. Practically, this can be implemented using several convolutional layers with shared weights, as well as skip connections in order to compute the residual, $\hat{\mathbf{\Gamma}}_{i-1} - \mathbf{D}_i\hat{\mathbf{\Gamma}}_i^{t-1}$, as defined in Equation (9). Interestingly, the above description is reminiscent of residual networks (He et al., 2015), which have recently led to state-of-the-art results in image recognition. The authors of (Greff et al., 2016) propose a similar viewpoint of residual networks, based on unrolled iterative estimation. Similar to the above discussion, they claim that a group of successive layers iteratively refine estimates of the same features instead of computing an entirely new representation.

Since the submission of our work, the authors of (Sun et al., 2017) suggested an algorithm similar to the layered IST, showing promising results for the task of image recognition when compared to other conventional architectures.

6.1 Noiseless Case: Success of Layered BP Algorithm

In Section 5.1, we established the uniqueness of the solution for the DCP$_{\boldsymbol{\lambda}}$ problem, assuming that certain conditions on the $\ell_{0,\infty}$ norm of the underlying representations are met. However, as we have seen in the theoretical analysis of the previous section, the forward pass of the CNN is incapable of finding this unique solution; instead, it is guaranteed to

be close to it in terms of the $\ell_{2,\infty}$ norm. Herein, we address the question of whether the layered BP algorithm can prevail in a task where the forward pass did not.

Theorem 11 *(Layered BP recovery guarantee using the $\ell_{0,\infty}$ norm): Consider a signal $\mathbf{X}$,*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1\\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2\\ &\vdots\\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K,\end{aligned}$$

where $\{\mathbf{D}_i\}_{i=1}^K$ is a set of convolutional dictionaries and $\{\mu(\mathbf{D}_i)\}_{i=1}^K$ are their corresponding mutual coherences. Assuming that $\forall\ 1 \le i \le K$

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right)$$

then the layered BP algorithm is guaranteed to recover the set $\{\mathbf{\Gamma}_i\}_{i=1}^K$.

The proof for the above can be directly derived from the recovery condition of the BP using the $\ell_{0,\infty}$ norm, as presented in (Papyan et al., 2016a). The implications of this theorem are that the layered BP algorithm can indeed recover the unique solution to the $\text{DCP}_{\boldsymbol{\lambda}}$ problem.

6.2 Noisy Case: Stability of Layered BP Algorithm

Having established the guarantee for the success of the layered BP algorithm, we now move to its stability analysis. In particular, in a noisy scenario where obtaining the true underlying representations is impossible, does this algorithm remain stable? If so, how do its guarantees compare to those of the layered thresholding algorithm? The following theorem, which we prove in Appendix F, aims to answer these questions.

Theorem 12 *(Stability of the layered BP algorithm in the presence of noise): Suppose a clean signal $\mathbf{X}$ has a decomposition*

$$\begin{aligned}\mathbf{X} &= \mathbf{D}_1\mathbf{\Gamma}_1\\ \mathbf{\Gamma}_1 &= \mathbf{D}_2\mathbf{\Gamma}_2\\ &\vdots\\ \mathbf{\Gamma}_{K-1} &= \mathbf{D}_K\mathbf{\Gamma}_K,\end{aligned}$$

and that it is contaminated with noise $\mathbf{E}$ to create the signal $\mathbf{Y} = \mathbf{X}+\mathbf{E}$, such that $\|\mathbf{E}\|_{2,\infty}^{\mathbf{p}} \le \epsilon_0$. Let $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ be the set of solutions obtained by running the layered BP algorithm with parameters $\{\xi_i\}_{i=1}^K$. Assuming that $\forall\ 1 \le i \le K$

a) $\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} \le \frac{1}{3}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right)$*; and*

b) $\xi_i = 4\epsilon_{i-1}$,

then

1. *The support of the solution $\hat{\mathbf{\Gamma}}_i$ is contained in that of $\mathbf{\Gamma}_i$;*
2. *$\|\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_i\|_{2,\infty}^{\mathbf{p}} \le \epsilon_i$;*
3. *In particular, every entry of $\mathbf{\Gamma}_i$ greater in absolute value than $\frac{\epsilon_i}{\sqrt{\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{p}}}}$ is guaranteed to be recovered; and*
4. *The solution $\hat{\mathbf{\Gamma}}_i$ is the unique minimizer of the Lagrangian BP problem (Equation (8)),*

where $\epsilon_i = \|\mathbf{E}\|_{2,\infty}^{\mathbf{p}}\ 7.5^i\ \prod_{j=1}^i \sqrt{\|\mathbf{\Gamma}_j\|_{0,\infty}^{\mathbf{p}}}$.

Several remarks are due at this point. The condition for the stability of the layered thresholding algorithm, given by

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\frac{|\Gamma_i^{\min}|}{|\Gamma_i^{\max}|}\right) - \frac{1}{\mu(\mathbf{D}_i)}\frac{\epsilon_{i-1}}{|\Gamma_i^{\max}|},$$

is expected to be more strict than that of the theorem presented above, which is

$$\|\mathbf{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} < \frac{1}{3}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right).$$

In the case of the layered BP algorithm, the bound on the $\ell_{0,\infty}$ norm of the underlying sparse vectors does no longer depend on the ratio $|\Gamma_i^{\min}|/|\Gamma_i^{\max}|$ – a term present in all the theoretical results of the thresholding algorithm. Moreover, the $\ell_{0,\infty}$ norm becomes independent of the local noise level of the previous layers, thus allowing more non-zeros per stripe.

In addition, similar to the stability analysis presented in Section 5.2, the above shows the growth (as a function of the depth) of the distance between the recovered representations and the true ones.

7. A Closer Look at the Proposed Model

In this section, we revisit the assumptions of our model by imposing additional constraints on the dictionaries involved and showing their theoretical benefits. These additional assumptions originate from the current common practice of both CNN and sparsity.

7.1 When a Patch Becomes a Stripe

Throughout the analysis presented in this work, we have assumed that the representations in the different layers, $\{\mathbf{\Gamma}_i\}_{i=1}^K$, are $\ell_{0,\infty}$-sparse. Herein, we study the propagation of the $\ell_{0,\infty}$ norm throughout the layers of the network, showing how an assumption on the sparsity of the deepest representation $\mathbf{\Gamma}_K$ reflects on that of the remaining layers. The exact connection between the sparsities will be given in terms of a simple characterization of the dictionaries $\{\mathbf{D}_i\}_{i=1}^K$.

Consider the representation $\mathbf{\Gamma}_{K-1}$, given by

$$\mathbf{\Gamma}_{K-1} = \mathbf{D}_K \mathbf{\Gamma}_K,$$

where $\mathbf{\Gamma}_K$ is $\ell_{0,\infty}$-sparse. Following Figure 4, the i-th patch in $\mathbf{\Gamma}_{K-1}$ can be expressed as

$$\mathbf{P}_{K-1,i}\mathbf{\Gamma}_{K-1} = \mathbf{\Omega}_K \boldsymbol{\gamma}_{K,i},$$

where $\mathbf{\Omega}_K$ is the stripe-dictionary of $\mathbf{D}_K$, the vector $\mathbf{P}_{K-1,i}\mathbf{\Gamma}_{K-1}$ is the i-th patch in $\mathbf{\Gamma}_{K-1}$ and $\boldsymbol{\gamma}_{K,i}$ is its corresponding stripe. Recalling the definition of the $\|\cdot\|^{\mathbf{P}}_{0,\infty}$ norm (Definition 6 in Section 5.3), we have that

$$\|\mathbf{\Gamma}_{K-1}\|^{\mathbf{P}}_{0,\infty} = \max_i \|\mathbf{\Omega}_K \boldsymbol{\gamma}_{K,i}\|_0.$$

Consider the following definition.

Definition 13 *Define the induced ℓ_0 pseudo-norm of a dictionary $\mathbf{D}$, denoted by $\|\mathbf{D}\|_0$, to be the maximal number of non-zeros in any of its atoms*[8].

The multiplication $\mathbf{\Omega}_K \boldsymbol{\gamma}_{K,i}$ can be seen as a linear combination of at most $\|\boldsymbol{\gamma}_{K,i}\|_0$ atoms, each contributing no more than $\|\mathbf{\Omega}_K\|_0$ non-zeros. As such

$$\|\mathbf{\Gamma}_{K-1}\|^{\mathbf{P}}_{0,\infty} \leq \max_i \|\mathbf{\Omega}_K\|_0 \, \|\boldsymbol{\gamma}_{K,i}\|_0.$$

Noticing that $\|\mathbf{\Omega}_K\|_0 = \|\mathbf{D}_K\|_0$ (as can be seen in Figure 4), and using the definition of the $\|\cdot\|^{\mathbf{s}}_{0,\infty}$ norm, we conclude that

$$\|\mathbf{\Gamma}_{K-1}\|^{\mathbf{P}}_{0,\infty} \leq \|\mathbf{D}_K\|_0 \, \|\mathbf{\Gamma}_K\|^{\mathbf{s}}_{0,\infty}. \tag{10}$$

In other words, given $\|\mathbf{\Gamma}_K\|^{\mathbf{s}}_{0,\infty}$ and $\|\mathbf{D}_K\|_0$, we can bound the maximal number of non-zeros in a patch from $\mathbf{\Gamma}_{K-1}$.

The claims in Section 5 and 6 are given in terms of not only $\|\mathbf{\Gamma}_{K-1}\|^{\mathbf{P}}_{0,\infty}$, but also $\|\mathbf{\Gamma}_{K-1}\|^{\mathbf{s}}_{0,\infty}$. According to Table 1, the length of a patch in $\mathbf{\Gamma}_{K-1}$ is $n_{K-1}m_{K-1}$, while the size of a stripe is $(2n_{K-2}-1)m_{K-1}$. As such, we can fit $(2n_{K-2}-1)/n_{K-1}$ patches in a stripe. Assume for simplicity that this ratio is equal to one. As a result, we obtain that a patch in the *signal* $\mathbf{\Gamma}_{K-1}$ extracted from the system

$$\mathbf{\Gamma}_{K-1} = \mathbf{D}_K \mathbf{\Gamma}_K,$$

is also a stripe in the *representation* $\mathbf{\Gamma}_{K-1}$ when considering

$$\mathbf{\Gamma}_{K-2} = \mathbf{D}_{K-1}\mathbf{\Gamma}_{K-1},$$

hence the name of this subsection. Leveraging this assumption, we return to Equation (10) and obtain that

$$\|\mathbf{\Gamma}_{K-1}\|^{\mathbf{s}}_{0,\infty} = \|\mathbf{\Gamma}_{K-1}\|^{\mathbf{P}}_{0,\infty} \leq \|\mathbf{D}_K\|_0 \, \|\mathbf{\Gamma}_K\|^{\mathbf{s}}_{0,\infty}.$$

8. According to the definition of the induced norm $\|\mathbf{D}\|_0 = \max_{\mathbf{v}} \|\mathbf{D}\mathbf{v}\|_0$ s.t. $\|\mathbf{v}\|_0 = 1$. Since $\|\mathbf{v}\|_0 = 1$, the multiplication $\mathbf{D}\mathbf{v}$ is simply equal to one of the atoms in $\mathbf{D}$ times a scalar, and $\|\mathbf{D}\mathbf{v}\|_0$ counts the number of non-zeros in this atom. As a result, $\|\mathbf{D}\|_0$ is equal to the maximal number of non-zeros in any atom from $\mathbf{D}$.

Using the same rationale for the remaining layers, and assuming that once again the patches become stripes, we conclude that

$$\|\mathbf{\Gamma}_i\|^{\mathbf{s}}_{0,\infty} = \|\mathbf{\Gamma}_i\|^{\mathbf{P}}_{0,\infty} \leq \|\mathbf{\Gamma}_K\|^{\mathbf{s}}_{0,\infty} \prod_{j=i+1}^{K} \|\mathbf{D}_j\|_0. \tag{11}$$

We note that our assumption here of having sparse dictionaries is reasonable, since at the training stage of the CNN an ℓ_1 penalty is often imposed on the filters as a regularization, promoting their sparsity. The conclusion thus is that the $\ell_{0,\infty}$ norm is expected to decrease as a function of the depth of the representation. This aligns with the intuition that the higher the depth, the more abstraction one obtains in the filters, and thus the less non-zeros are required to represent the data. Taking this to the extreme, if every input signal could be represented via a single coefficient at the deepest layer, we would obtain that its $\ell_{0,\infty}$ norm is equal to one.

7.2 On the Role of the Spatial-Stride

A common step among practitioners of CNN (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2015) is to convolve the input to each layer with a set of filters, skipping a fixed number of spatial locations in a regular pattern. One of the primary motivations for this is to reduce the dimensions of the kernel maps throughout the layers, leading to **computational benefits**. In this subsection we unveil some **theoretical benefits** of this common practice, which we coin *spatial-stride*.

Following Figure 5, recall that $\mathbf{D}_i$ is a stride convolutional dictionary that skips m_{i-1} shifts at a time, which correspond to the number of channels in $\mathbf{\Gamma}_{i-1}$. Translating the spatial-stride to our language, the above mentioned works do not consider all spatial shifts of the filters in $\mathbf{D}_i$. Instead, a stride of $m_{i-1}s_{i-1}$ is employed, where m_{i-1} corresponds to the *channel-stride*, while s_{i-1} is due to the *spatial-stride*. The addition of the latter implies that instead of assuming that the i-th sparse vector satisfies $\mathbf{\Gamma}_{i-1} = \mathbf{D}_i\mathbf{\Gamma}_i$, we have that $\mathbf{Q}_{i-1}\mathbf{\Gamma}_{i-1} = \mathbf{D}_i\mathbf{Q}_i^T\mathbf{Q}_i\mathbf{\Gamma}_i$. We denote $\mathbf{Q}_i^T \in \mathbb{R}^{Nm_i \times Nm_i/s_{i-1}}$ as a columns' selection operator that chooses the atoms from $\mathbf{D}_i$ that align with the spatial-stride. The coefficients corresponding to these atoms are extracted from $\mathbf{\Gamma}_i$ (resulting in its subsampled version) via the $\mathbf{Q}_i$ matrix. In light of the above discussion, we modify the $\text{DCP}_{\boldsymbol{\lambda}}$ problem, as defined in Definition 1, into the following

$$\begin{aligned}
\text{find} \quad \{\mathbf{\Gamma}_i\}_{i=1}^K \qquad \text{s.t.} \qquad & \mathbf{X} = \mathbf{D}_1\mathbf{Q}_1^T\mathbf{Q}_1\mathbf{\Gamma}_1, && \|\mathbf{Q}_1\mathbf{\Gamma}_1\|^{\mathbf{s}}_{0,\infty} \leq \lambda_1 \\
& \mathbf{Q}_1\mathbf{\Gamma}_1 = \mathbf{D}_2\mathbf{Q}_2^T\mathbf{Q}_2\mathbf{\Gamma}_2, && \|\mathbf{Q}_2\mathbf{\Gamma}_2\|^{\mathbf{s}}_{0,\infty} \leq \lambda_2 \\
& \qquad\vdots && \qquad\vdots \\
& \mathbf{Q}_{K-1}\mathbf{\Gamma}_{K-1} = \mathbf{D}_K\mathbf{Q}_K^T\mathbf{Q}_K\mathbf{\Gamma}_K, && \|\mathbf{Q}_K\mathbf{\Gamma}_K\|^{\mathbf{s}}_{0,\infty} \leq \lambda_K.
\end{aligned}$$

Note that while the original $\|\mathbf{\Gamma}_i\|^{\mathbf{s}}_{0,\infty}$ is equal to the maximal number of non-zeros in a stripe of length $(2n_{i-1}-1)m_i$ in $\mathbf{\Gamma}_i$, the term $\|\mathbf{Q}_i\mathbf{\Gamma}_i\|^{\mathbf{s}}_{0,\infty}$ counts the same quantity but for stripes of length $(2\lceil n_{i-1}/s_{i-1}\rceil - 1)m_i$ in $\mathbf{Q}_i\mathbf{\Gamma}_i$.

According to the study in Section 5 and 6, the theoretical advantage of the spatial-stride is twofold. First, consider the mutual coherence of the stride convolutional dictionary $\mathbf{D}_i$.

Due to the locality of the filters and their restriction to certain spatial shifts, the mutual coherence of $\mathbf{D}_i\mathbf{Q}_i^T$ is expected to be lower than that of $\mathbf{D}_i$, thus leading to more non-zeros allowed per stripe. Second, the length of a stripe in $\mathbf{Q}_i\mathbf{\Gamma}_i$ is equal to $(2\lceil n_{i-1}/s_{i-1}\rceil - 1)m_i$, while that of $\mathbf{\Gamma}_i$ is $(2n_{i-1}-1)m_i$. As such, our analysis allows a larger number of non-zeros per a smaller-sized stripe. From another perspective, notice that imposing a spatial-stride on the dictionary $\mathbf{D}_i$ is equivalent to forcing a portion of the entries in $\mathbf{\Gamma}_i$ to be zero. As such, the spatial-stride encourages sparser solutions.

8. Experiments: The Generator Behind the CNN

Consider the following question: can we synthesize signals obeying the ML-CSC model? Throughout this work we have posited that the answer to this question is positive; we have assumed the existence of a set of signals $\mathbf{X}$, which satisfy $\forall i\ \mathbf{\Gamma}_i = \mathbf{D}_{i+1}\mathbf{\Gamma}_{i+1}$ where $\{\mathbf{\Gamma}_i\}_{i=1}^K$ are all $\ell_{0,\infty}$ bounded. However, a natural question arises as to whether we can give a simple example of set of dictionaries $\{\mathbf{D}_i\}_{i=1}^K$ and their corresponding signals $\mathbf{X}$ that indeed satisfy our model assumptions.

A naïve attempt would be to choose an arbitrary set of dictionaries, $\{\mathbf{D}_i\}_{i=1}^K$, and a random deepest representation, $\mathbf{\Gamma}_K$, and compute the remaining sparse vectors (and the signal itself) using the set of relations $\mathbf{\Gamma}_i = \mathbf{D}_{i+1}\mathbf{\Gamma}_{i+1}$. However, without further restrictions, this would lead to a set of representations $\{\mathbf{\Gamma}_i\}_{i=1}^K$ with growing $\ell_{0,\infty}$ norm as we propagate towards $\mathbf{\Gamma}_0$. A somewhat better approach would be to impose sparsity on the dictionaries involved, as suggested in Section 7.1, thus leading to sparser representations. However, besides the obvious drawback of forcing a limiting structure on the dictionaries, as can be seen in Equation (11), in the worst case this would also lead to growth in the density of the representations, even if it is more controlled. The spatial-stride – at first glance unrelated to this discussion – is another solution that addresses the same problem. In particular, in Section 7.2 this idea was shown to encourage sparser vectors by forcing zeros in a regular pattern in the set of representations $\{\mathbf{\Gamma}_i\}_{i=1}^K$.

In this section we combine the above notions in order to achieve our goal – generate a set of signals that will satisfy the ML-CSC assumptions. These will then serve as a playground for several experiments, which will compare both theoretically and practically the different pursuits presented in this paper.

8.1 Designing the Dictionaries

We commence by describing the design of the dictionaries, and in the next subsection continue to the actual generation of the signals. In our experiments, the signal is one dimensional and therefore $m_0 = 1$. Moreover, for simplicity, the dictionary in every layer contains a single atom with its shifts and thus $m_i = 1\ \forall 1 \le i \le K$. We should note that the choice of a single atom simplifies the involved pursuit problem, but as we will see, even in such a case the suggested layered pursuits (including the forward pass) may fail. This is because the mutual coherence and the amount of non-zeros are still non-trivial.

In the first layer we choose this filter to be the analytically defined discrete Meyer Wavelet of length $n_0 = 29$. In order to obtain sparser representations and improve the coherence of the global dictionary $\mathbf{D}_1$, we employ a stride of $s_0 = 6$, resulting in $\mu(\mathbf{D}_1) = 2.44 \times 10^{-4}$. As a consequence of our choice of $\mathbf{D}_1$, the signals resulting from our model

are a superposition of shifted versions of discrete Meyer Wavelets, multiplied by different coefficients.

Recall that in the context of the layered thresholding algorithm, our theoretical study has shown that the stability of the pursuit depends on the ratio $|\Gamma_i^{\min}|/|\Gamma_i^{\max}|$. As such, in addition to requiring $\mu(\mathbf{D}_i)$ to be small, we would also like the ratio $|\Gamma_i^{\min}|/|\Gamma_i^{\max}|$ to be as close as possible to one. Since the sparse vectors satisfy $\mathbf{\Gamma}_i = \mathbf{D}_{i+1}\mathbf{\Gamma}_{i+1}\ \forall 1 \le i \le K-1$, one can control this ratio by forcing the entries in the dictionaries $\{\mathbf{D}_i\}_{i=2}^K$ to be[9] discrete[10]. Following this observation, and motivated by the benefits of a sparse dictionary, we generate a filter of length 20 with 7 non-zero entries belonging to the set $\{-8,-7,...,7,8\}$ (these are the entries before the atom is normalized to a unit ℓ_2 norm). In practice, this is done by sampling random vectors satisfying these constraints and choosing one resulting in a good mutual coherence. For simplicity, all $\{\mathbf{D}_i\}_{i=2}^K$ are created from the very same local atom, i.e. $n_i = 20\ \forall 1 \le i \le K-1$. Moreover, in all the dictionaries this atom is shifted by a stride of $s_i = 6$, leading to $\mu(\mathbf{D}_i) = 4.33 \times 10^{-3}$. Note that in the above description the specific number of layers K was purposely omitted, as this number will vary in the following experiments.

8.2 Noiseless Experiments

We now move to the task of sampling a signal when the number of layers is $K = 3$. First, we draw a random $\mathbf{\Gamma}_3$ of length 100 with an ℓ_0 norm in the range $[20, 66]$ and set each non-zero coefficient in it to ± 1, with equal probability. Given the dictionaries and the sampled sparse vector $\mathbf{\Gamma}_3$, we then compute the representations $\mathbf{\Gamma}_2$, $\mathbf{\Gamma}_1$ and the signal $\mathbf{X}$, which are of length 600, 3,600 and $N = 21{,}600$, respectively. The obtained sparse vectors satisfy $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} = 8$, $5 \le \|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} \le 6$ and $3 \le \|\mathbf{\Gamma}_3\|_{0,\infty}^{\mathbf{s}} \le 7$.

Given the signals, we attempt to retrieve their underlying representations using the layered pursuits presented in this work. Recall that our analysis in Section 5 and 6 indicates that the layered hard thresholding is superior to its soft counterpart, which is equivalent to the forward pass, and that the layered BP is even better than both of these algorithms. We now turn to asserting this claim empirically. While doing so, we aim to study the gap between the theoretical guarantees presented throughout our paper and the empirical performance obtained in practice.

For every signal $\mathbf{X}$ (termed *realization* below), we employ the layered hard thresholding algorithm. The thresholds are set to be the ones presented in Theorem 8, since the $\ell_{0,\infty}$ norms of the representations of each $\mathbf{X}$ satisfy the assumptions of this theorem. Given the estimated sparse vectors $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^3$, we then compute the errors $\|\hat{\mathbf{\Gamma}}_i - \mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{p}}$ and compare these to the theoretical bounds. While doing so, we also verify that the correct support is indeed retrieved, as our theorem guarantees. Next, the same process is repeated for the

9. Note that we do not force the entries in $\mathbf{D}_1$ to be discrete since $\mathbf{X} = \mathbf{D}_1\mathbf{\Gamma}_1$ and the ratio of the entries in $\mathbf{X}$ is of no significance to the success of the layered thresholding algorithms.

10. In our experiments, the non-zero entries in the deepest representation $\mathbf{\Gamma}_K$ are chosen to be ± 1. As such, the sparse vector $\mathbf{\Gamma}_{K-1} = \mathbf{D}_K\mathbf{\Gamma}_K$ is a superposition of filters (or their negative) taken from the dictionary $\mathbf{D}_K$. If the entries in $\mathbf{D}_K$ are non-discrete then the summation of two filters can result in extremely small values in $\mathbf{\Gamma}_{K-1}$, which in turn would lead to a very small $|\Gamma_{K-1}^{\min}|$ and a bad ratio. On the other hand, if the atoms are chosen to be discrete, this would not happen since the entries would simply cancel each other.

layered soft thresholding algorithm, with the exception that the thresholds and the bound on the distance are computed according to Theorem 10. We note that the assumptions of this hold as well for every signal $\mathbf{X}$. The results for both algorithms are depicted in Figure 6 in terms of the local signal to noise ratio (SNR), defined as $20\log 10\Big(\frac{\|\mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{P}}}{\|\hat{\mathbf{\Gamma}}_i-\mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{P}}}\Big)$. Due to the locality of the analysis, we choose to deviate from the classical definition of the (global) SNR, given by $20\log 10\left(\frac{\|\mathbf{\Gamma}_i\|_2}{\|\hat{\mathbf{\Gamma}}_i-\mathbf{\Gamma}_i\|_2}\right)$.

Several remarks are due here. First and foremost, the theoretical bounds indeed hold, since the blue points are above their corresponding green ones and the correct supports are always recovered. Second, our analysis predicts that the distance between the estimated sparse representation, $\hat{\mathbf{\Gamma}}_i$, and the true ones, $\mathbf{\Gamma}_i$, should increase with the layer. This is evident by the decrease in the values of the green points with the layers. The empirical results presented here (blue dots) corroborate this prognosis, as the error in both algorithms is lowest in the first layer and highest in the last[11]. Third, our analysis suggests that the layered hard thresholding algorithm should be superior to its soft counterpart. Once again, this can be deduced from the figure by comparing the values of the green points in both of the algorithms. The empirical results presented in Figure 6 confirm this behavior, as can be clearly seen by comparing the errors (blue points) obtained by both algorithms in the i-th layer. One should note that the performance gap exhibited here is due to the constant β_i being subtracted from every entry in the soft thresholding algorithm.

The implications of the above discussion might be troubling in the context of CNN, as what this experiment shows is a deterioration of the **empirical** SNR throughout the layers of the network. Is this truly the behavior of CNN? Recall that in practice the biases of the different layers (thresholds) are learned in order to achieve the best possible performance in solving a certain task. As such, it might be possible that the decline in SNR presented here is alleviated when better thresholds are employed in lieu of the theoretical ones used thus far. We demonstrate this by running the layered soft thresholding algorithm with an oracle parameter, chosen to be the minimal threshold that leads to $\|\mathbf{\Gamma}_i\|_0$ non-zeros being chosen in the estimated sparse representation $\hat{\mathbf{\Gamma}}_i$. The results for this are presented in Figure 6 and colored in red. Indeed, we observe that this better choice of parameters improves the empirical performance of the layered soft thresholding algorithm and leads to a slower decline in SNR. Still, the performance of the layered soft thresholding is inferior to that of its hard variant[12], as can be seen by comparing the red points with the blue ones in the subplots below.

Next, we proceed our experiments by running the layered BP algorithm, as defined in Section 6, on the same set of signals. Recall that one of the prime motivations for proposing this algorithm was its ability to retrieve the exact underlying representations, as justified theoretically in Theorem 11. In our experiments, we validate this claim by checking that its conditions hold for each signal and that the underlying representations are indeed retrieved. We omit showing a plot for this and comparing it to the layered thresholding algorithms since the errors obtained are simply zeros.

11. Interestingly, the error in the layered hard thresholding algorithm is approximately equal in the second and third layers.
12. Note that in the layered hard thresholding, as long as the correct support is chosen, the threshold does not affect the error and as such the oracle version for it is meaningless.

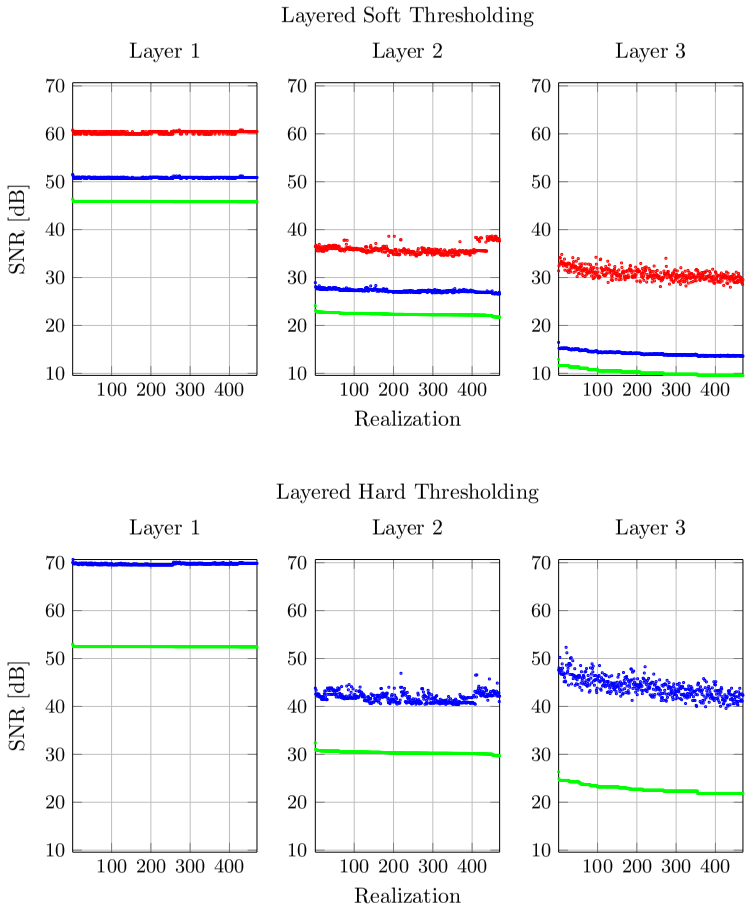


Figure 6: Comparison between the performance of the layered thresholding algorithms in a noiseless setting. All the signals presented here satisfy the assumptions of Theorem 8 and 10 and indeed the correct support of all the representations are recovered. The horizontal axis plots the realization number, while the vertical axis shows the SNR (the higher the better). Blue: layered thresholding algorithm with the theoretically justifiable thresholds. Green: the theoretical bound on the error. Red: layered soft thresholding algorithm with oracle thresholds. Note that the points are sorted according to the theoretical bound.

8.3 Noisy Experiments

Having established the stability of our proposed algorithms in a perfect scenario, where $\epsilon_0 = 0$, we now turn to a noisy setting. Naturally, the estimation task becomes now even more challenging – not only does the SNR drop with each layer, as demonstrated previously, but also the input SNR is no longer infinity. In order to facilitate the success of our algorithms, in this section we demonstrate the empirical performance and theoretical bounds on $K = 2$ layers and a small noise level.

Similar to the previous subsection, we begin by sampling a signal $\mathbf{X}$. To this end, we draw a random $\mathbf{\Gamma}_2$ of length 100 where $20 \leq \|\mathbf{\Gamma}_2\|_0 \leq 66$. Each non-zero coefficient in it is then set to ± 1, with equal probability. Given the dictionaries and the sampled sparse vector $\mathbf{\Gamma}_2$, we then compute the representation $\mathbf{\Gamma}_1$ and the signal $\mathbf{X}$, which are of length 600 and 3,600, respectively. The $\ell_{0,\infty}$ norm of the obtained sparse vectors satisfies $7 \leq \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} \leq 8$ and $3 \leq \|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} \leq 7$.

Next, we contaminate each signal $\mathbf{X}$ with a zero-mean white additive Gaussian noise $\mathbf{E}$, creating a signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. The average SNR of the obtained noisy signals is 68.53 dB. These are then fed into the layered pursuits, resulting in a set of estimated sparse representations, $\hat{\mathbf{\Gamma}}_i$. We note that the $\ell_{0,\infty}$ norms of the representations of each $\mathbf{X}$ satisfy the assumptions of Theorem 8, 10 and 12. As such, the parameters for every algorithm are chosen according to our theoretical study. For each estimated representation we compute the error $\|\hat{\mathbf{\Gamma}}_i - \mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{p}}$ and its corresponding theoretical bound obtained from the aforementioned theorems. Since the underlying representations satisfy the assumptions of the stability theorem for the layered thresholding algorithms, for each signal we verify that indeed the correct support is found. As for the layered BP, our stability analysis guarantees that the support retrieved should be contained in the true one and coefficients that are large enough in $\mathbf{\Gamma}_i$ should be retrieved. In practice, the layered BP always finds the full support.

We present the obtained results in terms of the local SNR in Figure 7, showing the stability of the different algorithms that is in accordance with our theoretical bounds. Similar to the noiseless experiment, we observe that for all the algorithms the error increases both theoretically (green points) and empirically (blue points) with the layer depth. As previously discussed, a performance gap exists between the soft and hard layered thresholding algorithms. To mitigate this, we run the layered soft thresholding with an oracle parameter and compare the obtained errors (red points) to those of the other algorithms. The results, depicted in the same figure, show a clear improvement in the performance.

Interestingly, although theoretically superior, the layered BP leads to similar performance to that of the layered soft thresholding and worse performance than that of the layered hard thresholding (when comparing the blue points). We attribute this phenomenon to the suboptimal choice of the parameter ξ_i, which was chosen thus far according to our theoretical analysis. To validate this suspicion, we run the layered BP with hand-picked ξ_i and plot the obtained SNR in red in Figure 7. Not only are the correct supports retrieved for all the signals, but we can also see a clear improvement in terms of the SNR. In the first layer, the layered BP outperforms the layered soft thresholding and leads to similar results to those of the layered hard thresholding, while in the second, the layered BP significantly outperforms both of the other pursuit algorithms.

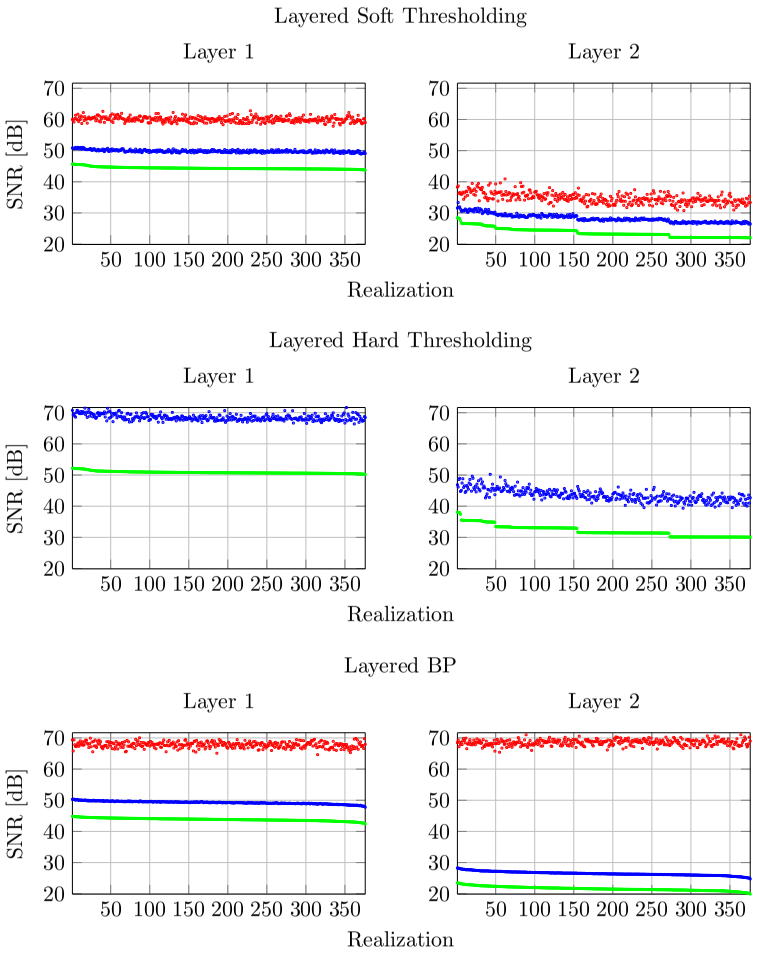


Figure 7: Comparison between the performance of the layered pursuit algorithms in a noisy setting. All the signals presented here satisfy the assumptions of Theorem 8, 10 and 12 and the correct support of all the representations are indeed recovered. The horizontal axis plots the realization number, while the vertical axis shows the SNR (the higher the better). Blue: layered pursuits with the theoretically justifiable thresholds. Green: the theoretical bound on the error. Red: layered soft thresholding with oracle thresholds and layered BP with hand-picked parameters. Note that the points are sorted according to the theoretical bound.

Thus far, our experiments focused on a setting where the ratio of the coefficients in $\mathbf{\Gamma}_i$ is reasonable. One should note, however, that the superiority of the layered BP becomes conspicuous once this ratio is spoiled. In this case, the layered thresholding algorithms will fail, while the layered BP will still succeed. To illustrate this, we create a signal using the dictionaries delineated in subsection 8.1, where the number of layers is $K = 5$. We first draw a random $\mathbf{\Gamma}_5$ of length 100 where its ℓ_0 norm is in the range $20 \leq \|\mathbf{\Gamma}_5\|_0 \leq 66$, and then set the non-zero coefficients in $\mathbf{\Gamma}_5$, similar to how it was done in the previous experiments. Given the dictionaries and the sampled sparse vector $\mathbf{\Gamma}_5$, we compute the representation $\{\mathbf{\Gamma}_i\}_{i=1}^4$ and the signal $\mathbf{X}$, which is of length $N = 777,600$. The $\ell_{0,\infty}$ norms of the obtained sparse vectors are $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} = 8$, $\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} = 6$, $\|\mathbf{\Gamma}_3\|_{0,\infty}^{\mathbf{s}} = 6$, $\|\mathbf{\Gamma}_4\|_{0,\infty}^{\mathbf{s}} = 6$ and $4 \leq \|\mathbf{\Gamma}_5\|_{0,\infty}^{\mathbf{s}} \leq 7$. Besides the depth of the network, the main difference between this experiment and the previous ones is the coefficient ratio. While the ratio of the deepest representation $\mathbf{\Gamma}_5$ is equal to 1, due to the coefficients in it being equal to ± 1, the ratio of $\mathbf{\Gamma}_1$ is equal to 2.44×10^{-4}. As a consequence, the theoretical results we have presented for the layered thresholding algorithms do not hold, while those of the layered BP still do.

Next, each signal $\mathbf{X}$ is contaminated with a zero-mean white additive Gaussian noise $\mathbf{E}$, resulting in a noisy signal $\mathbf{Y} = \mathbf{X} + \mathbf{E}$. The average SNR of the noisy signals obtained is 124.43 dB. Note that this is a weak noise, chosen due to the deterioration of the SNR throughout the layers (one that is worsened when the theoretical parameters are employed). The signals are then fed into the layered BP algorithm, resulting in a set of estimated sparse representations, $\hat{\mathbf{\Gamma}}_i$. The parameters ξ_i employed are the theoretically justified ones, $\xi_i = 4\epsilon_{i-1}$. We should note that in our experiments we attempted to run the layered thresholding algorithms, however, as our theory predicts these failed in recovering the correct supports.

Given the estimated representations, we compute the errors $\|\hat{\mathbf{\Gamma}}_i - \mathbf{\Gamma}_i\|_{2,\infty}^{\mathbf{p}}$ and compare these to their corresponding theoretical bounds, obtained from Theorem 12. In addition, we verify that the retrieved supports are contained in the true one, as the theorem guarantees. In practice, we obtain that the layered BP always finds the full support. The obtained results are depicted in Figure 8 in terms of the local SNR. For comparison, we run the layered BP with hand-picked ξ_i and present the obtained results in the same figure. We conclude that the layered BP remains stable despite the poor coefficient ratio, unlike the layered thresholding algorithms. Moreover, tuning the ξ_i results in a much better performance, similar to what we have seen in the previous experiment.

At this point, one might ponder as to whether the hurdle of poor coefficient ratio is one that the layered soft thresholding (forward pass) can not overcome. We believe that several ideas currently used in CNN, such as Batch Normalization (Ioffe and Szegedy, 2015) or Local Response Normalization (Krizhevsky et al., 2012), are tightly connected to this problem. However, their exact relation to this issue and its theoretical analysis is a matter of future work.

9. Conclusion

Definition: "A guiding question is the fundamental query that directs the search for understanding" (Traver, 1998). In this work our guiding question was who are the signals that the CNN architecture is designed for? To answer this we have defined the ML-CSC model, for which the thresholding pursuit is nothing but the forward pass of the CNN.

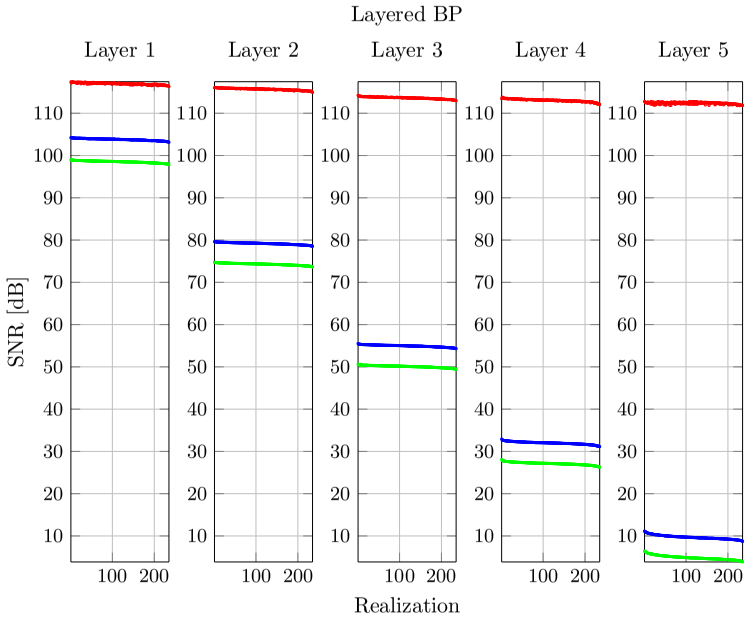


Figure 8: The performance of the layered BP algorithm in a noisy setting. All the signals presented here satisfy the assumptions of Theorem 12 and the correct support of all the representations are recovered. The horizontal axis plots the realization number, while the vertical axis shows the SNR (the higher the better). Blue: layered BP with the theoretically justifiable thresholds. Green: the theoretical bound on the error. Red: layered BP with hand-picked parameters. Note that the points are sorted according to the theoretical bound.

Although nothing promises that the forward pass will lead to the original representation of a signal emerging from the ML-CSC model, we have shown this is indeed the case. Having established the relevance of our model to CNN, we then turned to its theoretical analysis. In particular, we provided guarantees for the uniqueness of the feature maps CNN aims to recover, and the stability of the problem CNN aims to solve.

Inspired by the evolution of the pursuit methods in the theory of Sparse-Land, we continued our work by proposing the layered BP algorithm. In the noiseless case, this was theoretically shown to be capable of finding the unique solution of the deep coding problem, the existence of which has been also guaranteed; while in the noisy setting, we have proved the stability of this algorithm.

We analyzed the theoretical benefits of two popular ideas employed in the CNN community, namely the use of sparse filters and the spatial-stride. Leveraging those, we then

generated signals satisfying the ML-CSC assumptions and demonstrated the performance of the pursuits presented throughout this work.

We conclude this work by presenting our ongoing research directions:

1. Through this paper we have assumed the worst – an adversary noise. Can our theoretical analysis be extended to a setting where the noise is random?

2. Thus far in tackling the deep coding problem, we have restricted ourself to existing methods, such as the forward pass of the CNN or deconvolutional networks (Zeiler et al., 2010). Can we suggest better approximations for the solution of this problem?

3. What is the role of common tricks currently employed in CNN in the context of the ML-CSC model? These include but are not limited to, Batch Normalization (Ioffe and Szegedy, 2015), Local Response Normalization (Krizhevsky et al., 2012), Dropout (Srivastava et al., 2014) and Pooling (LeCun et al., 1990; Krizhevsky et al., 2012; Simonyan and Zisserman, 2014).

Acknowledgments

The research leading to these results has received funding in part from the European Research Council under EU's 7th Framework Program, ERC under Grant 320649, and in part by Israel Science Foundation (ISF) grant no. 1770/14. The authors would like to thank Jeremias Sulam for the inspiring discussions and creative advice.

Appendix A.
Uniqueness via the Mutual Coherence (Proof of Theorem 4)

Proof In (Papyan et al., 2016a) a solution $\mathbf{\Gamma}$ to the $\mathrm{P}_{0,\infty}$ problem, as defined in Equation (6), was shown to be unique assuming that $\|\mathbf{\Gamma}\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D})}\right)$. In other words, if the true representation is sparse enough in the $\ell_{0,\infty}$ sense, no other solution is possible. Herein, we leverage this claim in order to prove the uniqueness of the $\mathrm{DCP}_{\boldsymbol{\lambda}}$ problem.

Let $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ be a set of representations of the signal $\mathbf{X}$, obtained by solving the $\mathrm{DCP}_{\boldsymbol{\lambda}}$ problem. According to our assumptions, $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_1)}\right)$. Moreover, since the set $\{\hat{\mathbf{\Gamma}}_i\}_{i=1}^K$ is a solution of the $\mathrm{DCP}_{\boldsymbol{\lambda}}$ problem, we also have that $\|\hat{\mathbf{\Gamma}}_1\|_{0,\infty}^{\mathbf{s}} \leq \lambda_1 < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_1)}\right)$. As such, in light of the aforementioned uniqueness theorem, both representations are equal. Once we have concluded that $\mathbf{\Gamma}_1 = \hat{\mathbf{\Gamma}}_1$, we would also like to show that the representations $\mathbf{\Gamma}_2$ and $\hat{\mathbf{\Gamma}}_2$ are identical. Similarly, the assumptions $\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\right)$ and $\|\hat{\mathbf{\Gamma}}_2\|_{0,\infty}^{\mathbf{s}} \leq \lambda_2 < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_2)}\right)$ guarantee that $\mathbf{\Gamma}_2 = \hat{\mathbf{\Gamma}}_2$. The same set of steps can be applied for all $1 \leq i \leq K$, leading to the fact that both sets of representations are identical. ■

Appendix B.
Global Stability of the $\mathrm{DCP}_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ Problem (Proof of Theorem 5)

Proof In (Papyan et al., 2016b), for a signal $\mathbf{Y} = \mathbf{X} + \mathbf{E} = \mathbf{D}_1\mathbf{\Gamma}_1 + \mathbf{E}$, it was shown that if the following hold:

1. $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} \leq \lambda_1 < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_1)}\right)$ and $\|\mathbf{E}\|_2 = \|\mathbf{Y} - \mathbf{D}_1\mathbf{\Gamma}_1\|_2$,
2. $\|\hat{\mathbf{\Gamma}}_1\|_{0,\infty}^{\mathbf{s}} \leq \lambda_1 < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_1)}\right)$ and $\|\mathbf{Y} - \mathbf{D}_1\hat{\mathbf{\Gamma}}_1\|_2 \leq \|\mathbf{E}\|_2$,

then

$$\|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_2^2 \leq \frac{4\|\mathbf{E}\|_2^2}{1-(2\lambda_1-1)\mu(\mathbf{D}_1)}.$$

In item 2 we have used the fact that the solution for the $\mathrm{DCP}_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ problem, $\hat{\mathbf{\Gamma}}_1$, must satisfy $\|\mathbf{Y} - \mathbf{D}_1\hat{\mathbf{\Gamma}}_1\|_2 \leq \|\mathbf{E}\|_2$ and $\|\hat{\mathbf{\Gamma}}_1\|_{0,\infty}^{\mathbf{s}} \leq \lambda_1$. Notice that the true solution satisfies $\mathbf{\Gamma}_1 = \mathbf{D}_2\mathbf{\Gamma}_2$ and also the solution of $\mathrm{DCP}_{\boldsymbol{\lambda}}^{\boldsymbol{\mathcal{E}}}$ satisfies $\hat{\mathbf{\Gamma}}_1 = \mathbf{D}_2\hat{\mathbf{\Gamma}}_2$. As such,

$$\|\mathbf{D}_2\mathbf{\Gamma}_2 - \mathbf{D}_2\hat{\mathbf{\Gamma}}_2\|_2^2 = \|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_2^2 \leq \frac{4\|\mathbf{E}\|_2^2}{1-(2\lambda_1-1)\mu(\mathbf{D}_1)}. \tag{12}$$

Using the definition of SRIP from (Papyan et al., 2016b) and the fact that

$$\|\mathbf{\Gamma}_2 - \hat{\mathbf{\Gamma}}_2\|_{0,\infty}^{\mathbf{s}} \leq 2\lambda_2$$

we have

$$(1-\delta_{2\lambda_2})\ \|\mathbf{\Gamma}_2 - \hat{\mathbf{\Gamma}}_2\|_2^2 \leq \|\mathbf{D}_2\mathbf{\Gamma}_2 - \mathbf{D}_2\hat{\mathbf{\Gamma}}_2\|_2^2,$$

where $\delta_{2\lambda_2}$ is the SRIP constant of $\mathbf{D}_2$ for an $\ell_{0,\infty}$ norm equal to $2\lambda_2$. Bounding the SRIP using the mutual coherence, as was done in (Papyan et al., 2016b), the above results in

$$\begin{aligned}(1-(2\lambda_2-1)\,\mu(\mathbf{D}_2))\ \|\mathbf{\Gamma}_2 - \hat{\mathbf{\Gamma}}_2\|_2^2 \leq &(1-\delta_{2\lambda_2})\ \|\mathbf{\Gamma}_2 - \hat{\mathbf{\Gamma}}_2\|_2^2 \\ \leq &\|\mathbf{D}_2\mathbf{\Gamma}_2 - \mathbf{D}_2\hat{\mathbf{\Gamma}}_2\|_2^2.\end{aligned}$$

Combining this together with Equation (12), we obtain

$$\|\mathbf{\Gamma}_2 - \hat{\mathbf{\Gamma}}_2\|_2^2 \leq \frac{4\|\mathbf{E}\|_2^2}{(1-(2\lambda_1-1)\mu(\mathbf{D}_1))\,(1-(2\lambda_2-1)\mu(\mathbf{D}_2))}.$$

Repeating the above set of steps for the deeper layers, we conclude that

$$\forall\ 1 \leq i \leq K \qquad \|\mathbf{\Gamma}_i - \hat{\mathbf{\Gamma}}_i\|_2^2 \leq 4\|\mathbf{E}\|_2^2 \prod_{j=1}^{i} \frac{1}{1-(2\lambda_j-1)\mu(\mathbf{D}_j)},$$

as claimed. ■

Appendix C.
Stable Recovery of Hard Thresholding in the Presence of Noise (Proof of Lemma 7)

Proof Denote by $\mathcal{T}_1$ the support of $\mathbf{\Gamma}_1$. Denote further the i-th atom from $\mathbf{D}_1$ by $\mathbf{d}_{1,i}$. The success of the hard thresholding algorithm with threshold β_1 in recovering the correct support is guaranteed if the following holds

$$\min_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T\mathbf{Y}\right| > \beta_1 > \max_{j\notin\mathcal{T}_1}\left|\mathbf{d}_{1,j}^T\mathbf{Y}\right|.$$

Using the same set of steps as those used in proving Theorem 4 in (Papyan et al., 2016b), we can lower bound the left-hand-side by

$$\min_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T\mathbf{Y}\right| \geq |\mathbf{\Gamma}_1^{\min}| - (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| - \epsilon_0$$

and upper bound the right-hand-side via

$$\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| + \epsilon_0 \geq \max_{j\notin\mathcal{T}_1}\left|\mathbf{d}_{1,j}^T\mathbf{Y}\right|.$$

Next, by requiring

$$\begin{aligned}\min_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T\mathbf{Y}\right| \geq& |\mathbf{\Gamma}_1^{\min}| - (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| - \epsilon_0 \\ >& \beta_1 \\ >& \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| + \epsilon_0 \\ \geq& \max_{j\notin\mathcal{T}_1}\left|\mathbf{d}_{1,j}^T\mathbf{Y}\right|,\end{aligned} \tag{13}$$

we ensure the success of the thresholding algorithm. This condition can be equally written as

$$\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_1)}\frac{|\Gamma_1^{\min}|}{|\Gamma_1^{\max}|}\right) - \frac{1}{\mu(\mathbf{D}_1)}\frac{\epsilon_0}{|\Gamma_1^{\max}|}.$$

Equation (13) also implies that the threshold β_1 that should be employed must satisfy

$$|\mathbf{\Gamma}_1^{\min}| - (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| - \epsilon_0 > \beta_1 > \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| + \epsilon_0. \tag{14}$$

Thus far, we have considered the successful recovery of the support of $\mathbf{\Gamma}_1$. Next, assuming this correct support was recovered, we shall dwell on the deviation of the thresholding result, $\hat{\mathbf{\Gamma}}_1$, from the true $\mathbf{\Gamma}_1$. Denote by $\mathbf{\Gamma}_{1,\mathcal{T}_1}$ and $\hat{\mathbf{\Gamma}}_{1,\mathcal{T}_1}$ the vectors $\mathbf{\Gamma}_1$ and $\hat{\mathbf{\Gamma}}_1$ restricted to the support $\mathcal{T}_1$, respectively. We have that

$$\begin{aligned}\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_\infty &= \|\mathbf{\Gamma}_{1,\mathcal{T}_1}-\hat{\mathbf{\Gamma}}_{1,\mathcal{T}_1}\|_\infty \\ &= \left\|\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{X} - \mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y}\right\|_\infty \\ &= \left\|\left(\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T - \mathbf{D}_{1,\mathcal{T}_1}^T\right)\mathbf{X} - \mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty,\end{aligned}$$

where the Gram $\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}$ is invertible according to Lemma 1 in (Papyan et al., 2016a). Using the triangle inequality of the ℓ_∞ norm and the relation $\mathbf{X}=\mathbf{D}_{1,\mathcal{T}_1}\mathbf{\Gamma}_{1,\mathcal{T}_1}$, we obtain

$$\begin{aligned}\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_\infty &\leq \left\|\left(\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T - \mathbf{D}_{1,\mathcal{T}_1}^T\right)\mathbf{D}_{1,\mathcal{T}_1}\mathbf{\Gamma}_{1,\mathcal{T}_1}\right\|_\infty + \left\|\mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty \\ &= \left\|\left(\mathbf{I}-\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)\mathbf{\Gamma}_{1,\mathcal{T}_1}\right\|_\infty + \left\|\mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty,\end{aligned}$$

where $\mathbf{I}$ is an identity matrix. Relying on the definition of the induced ℓ_∞ norm, the above is equal to

$$\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_\infty \leq \left\|\mathbf{I}-\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right\|_\infty\cdot\|\mathbf{\Gamma}_{1,\mathcal{T}_1}\|_\infty + \left\|\mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty. \tag{15}$$

In what follows, we shall upper bound both of the expressions in the right hand side of the inequality.

Beginning with the first term in the above inequality, $\left\|\mathbf{I}-\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right\|_\infty$, recall that the induced infinity norm of a matrix is equal to its maximum absolute row sum. The diagonal entries of $\mathbf{I}-\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}$ are equal to zero, due to the normalization of the atoms, while the off diagonal entries can be bounded by relying on the locality of the atoms and the definition of the $\ell_{0,\infty}$ norm. As such, each row has at most $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1$ non-zeros, where each is bounded by $\mu(\mathbf{D}_1)$ based on the definition of the mutual coherence. We conclude that the maximum absolute row sum can be bounded by

$$\|\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}-\mathbf{I}\|_\infty \leq (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_1). \tag{16}$$

Next, moving to the second expression, define $\mathbf{R}_{1,i}\in\mathbb{R}^{n_0\times N}$ to be the operator that extracts a filter of length n_0 from $\mathbf{d}_{1,i}$. Consequently, the operator $\mathbf{R}_{1,i}^T$ pads a local filter of length n_0 with zeros, resulting in a global atom of length N. Notice that, due to the locality of the atoms $\mathbf{R}_{1,i}^T\mathbf{R}_{1,i}\mathbf{d}_{1,i}=\mathbf{d}_{1,i}$. Using this together with the Cauchy-Schwarz inequality, the normalization of the atoms, and the local bound on the error $\|\mathbf{Y}-\mathbf{X}\|_{2,\infty}^{\mathbf{p}}\leq\epsilon_0$, we have that

$$\begin{aligned}\left\|\mathbf{D}_{1,\mathcal{T}_1}^T(\mathbf{Y}-\mathbf{X})\right\|_\infty &= \max_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T(\mathbf{Y}-\mathbf{X})\right| \\ &= \max_{i\in\mathcal{T}_1}\left|\left(\mathbf{R}_{1,i}\mathbf{d}_{1,i}\right)^T\mathbf{R}_{1,i}(\mathbf{Y}-\mathbf{X})\right| \\ &\leq \max_{i\in\mathcal{T}_1}\|\mathbf{R}_{1,i}\mathbf{d}_{1,i}\|_2\cdot\|\mathbf{R}_{1,i}(\mathbf{Y}-\mathbf{X})\|_2 \\ &\leq 1\cdot\|\mathbf{Y}-\mathbf{X}\|_{2,\infty}^{\mathbf{p}} \\ &\leq \epsilon_0.\end{aligned} \tag{17}$$

In the second to last inequality we have used Definition 6, denoting the maximal ℓ_2 norm of a *patch* extracted from $\mathbf{Y}-\mathbf{X}$ by $\|\mathbf{Y}-\mathbf{X}\|_{2,\infty}^{\mathbf{p}}$. Plugging (16) and (17) into Equation (15), and using the fact that $\|\mathbf{\Gamma}_{1,\mathcal{T}_1}\|_\infty=|\mathbf{\Gamma}_1^{\max}|$, we obtain that

$$\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_\infty \leq (\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_1)|\mathbf{\Gamma}_1^{\max}| + \epsilon_0. \tag{18}$$

In the remainder of this proof we will localize the above bound into one that is posed in terms of patch-errors. Note that $\|\mathbf{\Gamma}_1-\hat{\mathbf{\Gamma}}_1\|_{2,\infty}^{\mathbf{p}}$ is equal to the maximal energy of an n_1m_1-dimensional patch taken from it, where the i-th patch can be extracted using the operator

$\mathbf{P}_{1,i}$. Relying on this and the relation $\|\mathbf{V}\|_2 \leq \sqrt{\|\mathbf{V}\|_0}\,\|\mathbf{V}\|_\infty$, we have that

$$\begin{aligned}\|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}} &= \max_i \left\|\mathbf{P}_{1,i}\left(\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\right)\right\|_2 \\ &\leq \max_i \sqrt{\left\|\mathbf{P}_{1,i}\left(\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\right)\right\|_0}\ \left\|\mathbf{P}_{1,i}\left(\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\right)\right\|_\infty .\end{aligned}$$

Recalling that, based on Definition 6, $\|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_{0,\infty}^{\mathbf{P}}$ denotes the maximal number of non-zeros in a patch of length n_1m_1 extracted from this vector, we obtain that

$$\begin{aligned}\|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}} &\leq \sqrt{\|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_{0,\infty}^{\mathbf{P}}}\ \|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_\infty \\ &\leq \sqrt{\|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\ \|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_\infty .\end{aligned}$$

In the last inequality we have used the success of the first stage in recovering the correct support, resulting in $\|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_{0,\infty}^{\mathbf{P}} \leq \|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}$. Plugging inequality (18) into the above equation, we conclude that

$$\|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\Big(\epsilon_0+\mu(\mathbf{D}_1)\left(\|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1\right)|\Gamma_1^{\max}|\Big),$$

as claimed. ■

Appendix D. Stability of the Layered Hard Thresholding in the Presence of Noise (Proof of Theorem 8)

Proof The stability of the first stage of the layered hard thresholding algorithm is obtained from Lemma 7. Denoting by $\boldsymbol{\Delta}_1=\hat{\boldsymbol{\Gamma}}_1-\boldsymbol{\Gamma}_1$, notice that $\hat{\boldsymbol{\Gamma}}_1=\boldsymbol{\Gamma}_1+\boldsymbol{\Delta}_1=\mathbf{D}_2\boldsymbol{\Gamma}_2+\boldsymbol{\Delta}_1$. In other words, $\boldsymbol{\Gamma}_1$ is a signal that admits a convolutional sparse representation $\mathbf{D}_2\boldsymbol{\Gamma}_2$, which is contaminated with noise $\boldsymbol{\Delta}_1$, resulting in $\hat{\boldsymbol{\Gamma}}_1$. Next, we would like to employ Lemma 7 for the signal $\hat{\boldsymbol{\Gamma}}_1=\boldsymbol{\Gamma}_1+\boldsymbol{\Delta}_1=\mathbf{D}_2\boldsymbol{\Gamma}_2+\boldsymbol{\Delta}_1$, with the local noise level

$$\|\boldsymbol{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\Big(\epsilon_0+\mu(\mathbf{D}_1)\left(\|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1\right)|\Gamma_1^{\max}|\Big)=\epsilon_1,$$

to obtain the stability of the second stage. To this end, we require its conditions to hold; in particular, the $\ell_{0,\infty}$ norm of $\boldsymbol{\Gamma}_2$ to obey

$$\|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_2)}\frac{|\Gamma_2^{\min}|}{|\Gamma_2^{\max}|}\right)-\frac{1}{\mu(\mathbf{D}_2)}\frac{\epsilon_1}{|\Gamma_2^{\max}|},$$

and the threshold β_2 to satisfy

$$|\boldsymbol{\Gamma}_2^{\min}|-(\|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_2)|\boldsymbol{\Gamma}_2^{\max}|-\epsilon_1 > \ \beta_2 > \|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{s}}\mu(\mathbf{D}_2)|\boldsymbol{\Gamma}_2^{\max}|+\epsilon_1.$$

Assuming the above hold, Lemma 7 guarantees that the support of $\hat{\boldsymbol{\Gamma}}_2$ is equal to that of $\boldsymbol{\Gamma}_2$, and also that

$$\|\boldsymbol{\Gamma}_2-\hat{\boldsymbol{\Gamma}}_2\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{P}}}\Big(\epsilon_1+\mu(\mathbf{D}_2)\left(\|\boldsymbol{\Gamma}_2\|_{0,\infty}^{\mathbf{s}}-1\right)|\Gamma_2^{\max}|\Big)=\epsilon_2.$$

Using the same steps as above, we obtain the desired claim for all the remaining layers, assuming that

$$\|\boldsymbol{\Gamma}_i\|_{0,\infty}^{\mathbf{s}} < \frac{1}{2}\left(1+\frac{1}{\mu(\mathbf{D}_i)}\frac{|\Gamma_i^{\min}|}{|\Gamma_i^{\max}|}\right)-\frac{1}{\mu(\mathbf{D}_i)}\frac{\epsilon_{i-1}}{|\Gamma_i^{\max}|}$$

and that the thresholds β_i are chosen to satisfy

$$|\boldsymbol{\Gamma}_i^{\min}|-(\|\boldsymbol{\Gamma}_i\|_{0,\infty}^{\mathbf{s}}-1)\mu(\mathbf{D}_i)|\boldsymbol{\Gamma}_i^{\max}|-\epsilon_{i-1} > \ \beta_i > \|\boldsymbol{\Gamma}_i\|_{0,\infty}^{\mathbf{s}}\mu(\mathbf{D}_i)|\boldsymbol{\Gamma}_i^{\max}|+\epsilon_{i-1}. \tag{19}$$

This completes our proof. ■

Appendix E. Stable Recovery of Soft Thresholding in the Presence of Noise (Proof of Lemma 9)

Proof The success of the soft thresholding algorithm with threshold β_1 in recovering the correct support is guaranteed if the following holds

$$\min_{i\in\mathcal{T}_1}\left|\mathbf{d}_{1,i}^T\mathbf{Y}\right| > \beta_1 > \max_{j\notin\mathcal{T}_1}\left|\mathbf{d}_{1,j}^T\mathbf{Y}\right|.$$

Since the soft thresholding operator chooses all atoms with correlations greater than β_1, the above implies that the true support $\mathcal{T}_1$ will be chosen. This condition is equal to that of the hard thresholding algorithm, and thus using the same steps as in Lemma 7, we are guaranteed that the correct support will be chosen under Assumptions (a) and (b).

The difference between the hard thresholding algorithm and its soft counterpart becomes apparent once we consider the estimated sparse vector. While the former estimates the non-zero entries in $\hat{\boldsymbol{\Gamma}}_1$ by computing $\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y}$, the latter subtracts or adds a constant β_1 from these, obtaining $\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y}-\beta_1\mathbf{B}$, where $\mathbf{B}$ is a vector of ±1. As a result, the distance between the true sparse vector and the estimated one is given by

$$\begin{aligned}\|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_\infty &= \left\|\boldsymbol{\Gamma}_{1,\mathcal{T}_1}-\hat{\boldsymbol{\Gamma}}_{1,\mathcal{T}_1}\right\|_\infty \\ &= \left\|\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{X}-\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y}-\beta_1\mathbf{B}\right)\right\|_\infty \\ &\leq \left\|\left(\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{D}_{1,\mathcal{T}_1}\right)^{-1}\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{X}-\mathbf{D}_{1,\mathcal{T}_1}^T\mathbf{Y}\right\|_\infty+\|\beta_1\mathbf{B}\|_\infty ,\end{aligned}$$

where in the last step we have used the triangle inequality for the ℓ_∞ norm. Notice that $\|\beta_1\mathbf{B}\|_\infty=\beta_1$, since β_1 must be positive according to Equation (14). Combining this together with the same steps as those used in proving Lemma 7, the above can be bounded by

$$\|\boldsymbol{\Gamma}_1-\hat{\boldsymbol{\Gamma}}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\Big(\epsilon_0+\mu(\mathbf{D}_1)\left(\|\boldsymbol{\Gamma}_1\|_{0,\infty}^{\mathbf{s}}-1\right)|\Gamma_1^{\max}|+\beta_1\Big),$$

resulting in the desired claim. ■

Appendix F.
Stability of the Layered BP Algorithm in the Presence of Noise (Proof of Theorem 12)

Proof In (Papyan et al., 2016b), for a signal $\mathbf{Y} = \mathbf{X} + \mathbf{E} = \mathbf{D}_1\mathbf{\Gamma}_1 + \mathbf{E}$, it was shown that if the following hold:

a) $\|\mathbf{Y} - \mathbf{X}\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_0$; and

b) $\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{S}} \leq \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D}_1)}\right)$,

then the solution $\hat{\mathbf{\Gamma}}_1$ for the Lagrangian formulation of the BP problem with $\beta_1 = 4\epsilon_0$ (see Equation (8)) satisfies that

1. The support of the solution $\hat{\mathbf{\Gamma}}_1$ is contained in that of $\mathbf{\Gamma}_1$;
2. $\|\mathbf{\Delta}_1\|_\infty = \|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_\infty \leq 7.5\ \epsilon_0$;
3. In particular, every entry of $\mathbf{\Gamma}_1$ greater in absolute value than $7.5\ \epsilon_0$ is guaranteed to be recovered; and
4. The solution $\hat{\mathbf{\Gamma}}_1$ is the unique minimizer of the Lagrangian BP problem (Equation (8)).

Using similar steps to those employed in the proof of Theorem 8, we obtain that

$$\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Delta}_1\|_{0,\infty}^{\mathbf{P}}}\ \|\mathbf{\Delta}_1\|_\infty.$$

Plugging above the inequality $\|\mathbf{\Delta}_1\|_\infty \leq 7.5\ \epsilon_0$, we get

$$\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Delta}_1\|_{0,\infty}^{\mathbf{P}}}\ 7.5\ \epsilon_0.$$

Since the support of $\hat{\mathbf{\Gamma}}_1$ is contained in that of $\mathbf{\Gamma}_1$, we have that $\|\mathbf{\Delta}_1\|_{0,\infty}^{\mathbf{P}} = \|\mathbf{\Gamma}_1 - \hat{\mathbf{\Gamma}}_1\|_{0,\infty}^{\mathbf{P}} \leq \|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}$, leading to

$$\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}\ 7.5\ \epsilon_0 = \epsilon_1.$$

We conclude that the first stage of the layered BP is stable and the following must hold

1. The support of the solution $\hat{\mathbf{\Gamma}}_1$ is contained in that of $\mathbf{\Gamma}_1$;
2. $\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq\ \epsilon_1$;
3. In particular, every entry of $\mathbf{\Gamma}_1$ greater in absolute value than $\frac{\epsilon_1}{\sqrt{\|\mathbf{\Gamma}_1\|_{0,\infty}^{\mathbf{P}}}}$ is guaranteed to be recovered; and
4. The solution $\hat{\mathbf{\Gamma}}_1$ is the unique minimizer of the Lagrangian BP problem (Equation (8)).

Next, we turn to the stability of the second stage of the layered BP algorithm. Notice that $\hat{\mathbf{\Gamma}}_1 = \mathbf{\Gamma}_1 + \mathbf{\Delta}_1 = \mathbf{D}_2\mathbf{\Gamma}_2 + \mathbf{\Delta}_1$. Put differently, $\mathbf{\Gamma}_1$ is a signal that admits a convolutional sparse representation $\mathbf{D}_2\mathbf{\Gamma}_2$ that is perturbed by $\mathbf{\Delta}_1$, resulting in $\hat{\mathbf{\Gamma}}_1$. As such, we can invoke once again the same theorem from (Papyan et al., 2016b). Since we have that

a) $\|\mathbf{\Delta}_1\|_{2,\infty}^{\mathbf{P}} \leq \epsilon_1$; and

b) $\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{S}} \leq \frac{1}{3}\left(1 + \frac{1}{\mu(\mathbf{D}_2)}\right)$,

we are guaranteed that the solution $\hat{\mathbf{\Gamma}}_2$ for the Lagrangian formulation of the BP problem with parameter $\beta_2 = 4\epsilon_1$ satisfies

1. The support of the solution $\hat{\mathbf{\Gamma}}_2$ is contained in that of $\mathbf{\Gamma}_2$;
2. $\|\mathbf{\Delta}_2\|_\infty = \|\mathbf{\Gamma}_2 - \hat{\mathbf{\Gamma}}_2\|_\infty \leq 7.5\ \epsilon_1$;
3. In particular, every entry of $\mathbf{\Gamma}_2$ greater in absolute value than $7.5\ \epsilon_1$ is guaranteed to be recovered; and
4. The solution $\hat{\mathbf{\Gamma}}_2$ is the unique minimizer of the Lagrangian BP problem (Equation (8)).

Using similar steps to those used above, the inequality that relies on the ℓ_∞ norm can be translated into another one that depends on the $\ell_{2,\infty}$. This results in

$$\|\mathbf{\Delta}_2\|_{2,\infty}^{\mathbf{P}} \leq \sqrt{\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{P}}}\ 7.5\ \epsilon_1 = \epsilon_2.$$

We conclude that, similar to the first one, the second stage of the layered BP is stable and the following must hold

1. The support of the solution $\hat{\mathbf{\Gamma}}_2$ is contained in that of $\mathbf{\Gamma}_2$;
2. $\|\mathbf{\Delta}_2\|_{2,\infty}^{\mathbf{P}} \leq\ \epsilon_2$;
3. In particular, every entry of $\mathbf{\Gamma}_2$ greater in absolute value than $7.5\ \epsilon_1 = \frac{\epsilon_2}{\sqrt{\|\mathbf{\Gamma}_2\|_{0,\infty}^{\mathbf{P}}}}$ is guaranteed to be recovered; and
4. The solution $\hat{\mathbf{\Gamma}}_2$ is the unique minimizer of the Lagrangian BP problem (Equation (8)).

Using the same set of steps, we obtain similarly the stability of the remaining layers. ■

References

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *ICML*, pages 584–592, 2014.

Waheed U Bajwa, Robert Calderbank, and Dustin G Mixon. Two are better than one: Fundamental parameters of frame coherence. *Applied and Computational Harmonic Analysis*, 33(1):58–78, 2012.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155, 2003.

Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 391–398, 2013.

Joan Bruna and Stéphane Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, pages 2313–2351, 2007.

Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Review*, 43(1):129–159, 2001.

Sheng Chen, Stephen A Billings, and Wan Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50 (5):1873–1896, 1989.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Patrick L Combettes and Valérie R Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

Ingrid Daubechies et al. *Ten lectures on wavelets*, volume 61. SIAM, 1992.

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014.

Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2016.

Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Trans. on Image Process.*, 20(7):1838–1857, 2011.

David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18, 2006.

M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745, December 2006. ISSN 1057-7149.

Michael Elad. Optimized projections for compressed sensing. *IEEE Transactions on Signal Processing*, 55(12):5695–5702, 2007.

Michael Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Publishing Company, Incorporated, 1st edition, 2010. ISBN 144197010X, 9781441970107.

Michael Elad and Alfred M Bruckstein. A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Transactions on Information Theory*, 48(9):2558–2567, 2002.

Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

Alhussein Fawzi, Mike Davies, and Pascal Frossard. Dictionary learning for fast classification based on soft-thresholding. *International Journal of Computer Vision*, 114(2-3):306–321, 2015.

Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Springer, 2013.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Raja Giryes, Guillermo Sapiro, and Alex M Bronstein. Deep neural networks with random gaussian weights: A universal classification strategy? *IEEE Transactions on Signal Processing*, 64(13):3444–3457, 2015.

Raja Giryes, Yonina C Eldar, Alex M Bronstein, and Guillermo Sapiro. Tradeoffs between convergence speed and reconstruction accuracy in inverse problems. *arXiv preprint arXiv:1605.09232*, 2016.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Aistats*, volume 15, page 275, 2011.

Klaus Greff, Rupesh K Srivastava, and Jürgen Schmidhuber. Highway and residual networks learn unrolled iterative estimation. *arXiv preprint arXiv:1612.07771*, 2016.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 399–406, 2010.

Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xiangchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1823–1831, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. Fast and flexible convolutional sparse coding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5135–5143. IEEE, 2015.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Zhuolin Jiang, Zhe Lin, and Larry S Davis. Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1697–1704. IEEE, 2011.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *arXiv preprint arXiv:1603.08155*, 2016.

Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann L Cun. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pages 1090–1098, 2010.

Bailey Kong and Charless C Fowlkes. Fast convolutional sparse coding (fcsc). *Department of Computer Science, University of California, Irvine, Tech. Rep*, 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

B Boser LeCun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012.

Julien Mairal, Francis Bach, and Jean Ponce. Sparse modeling for image and vision processing. *arXiv preprint arXiv:1411.3230*, 2014.

Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

Stphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition, 2008. ISBN 0123743702, 9780123743701.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Vardan Papyan, Jeremias Sulam, and Michael Elad. Working locally thinking globally-part I: Theoretical guarantees for convolutional sparse coding. *arXiv preprint arXiv:1607.02005*, 2016a.

Vardan Papyan, Jeremias Sulam, and Michael Elad. Working locally thinking globally-part II: Stability and algorithms for convolutional sparse coding. *arXiv preprint arXiv:1607.02009*, 2016b.

Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, pages 40–44. IEEE, 1993.

Yunchen Pu, Xin Yuan, Andrew Stevens, Chunyuan Li, and Lawrence Carin. A deep generative deconvolutional image model. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 741–750, 2016.

Yaniv Romano and Michael Elad. Patch-disagreement as a way to improve K-SVD denoising. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1280–1284. IEEE, 2015.

R. Rubinstein, M. Zibulevsky, and M. Elad. Double sparsity : Learning sparse dictionaries for sparse signal approximation. *IEEE Transactions on Signal Processing*, 58(3):1553–1564, 2010.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Karin Schnass and Pierre Vandergheynst. Average performance analysis for thresholding. *IEEE Signal Processing Letters*, 14(11):828–831, 2007.

Wenling Shang. A preliminary study of the norm preservation properties of cnn. `http://www-personal.umich.edu/~shangw/wiml_cnn_2015.pdf`, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Pablo Sprechmann, Alexander M Bronstein, and Guillermo Sapiro. Learning efficient sparse and low rank models. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1821–1833, 2015.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Jeremias Sulam and Michael Elad. Expected patch log likelihood with a sparse prior. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 99–111. Springer, 2015.

Jeremias Sulam, Boaz Ophir, Michael Zibulevsky, and Michael Elad. Trainlets: Dictionary learning in high dimensions. *IEEE Transactions on Signal Processing*, 64(12):3180–3193, 2016.

Xiaoxia Sun, Nasser M Nasrabadi, and Trac D Tran. Supervised multilayer sparse coding networks for image classification. *arXiv preprint arXiv:1701.08349*, 2017.

Rob Traver. What is a good guiding question?. *Educational Leadership*, 55(6):70–73, 1998.

J. A. Tropp. Just Relax : Convex Programming Methods for Identifying Sparse Signals in Noise. *IEEE Transactions on In*, 52(3):1030–1051, 2006.

Joel A Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242, 2004.

Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *arXiv preprint arXiv:1603.03417*, 2016.

Lloyd R Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information Theory*, 20(3):397–399, 1974.

Brendt Wohlberg. Efficient convolutional sparse coding. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7173–7177. IEEE, 2014.

Bo Xin, Yizhou Wang, Wen Gao, and David Wipf. Maximal sparsity with deep networks? *arXiv preprint arXiv:1605.01636*, 2016.

Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535. IEEE, 2010.

Qiang Zhang and Baoxin Li. Discriminative K-SVD for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2691–2698. IEEE, 2010.

Journal of Machine Learning Research 18 (2017) 1-42 Submitted 11/16; Revised 5/17; Published 7/17

Stochastic Primal-Dual Coordinate Method for Regularized Empirical Risk Minimization*

Yuchen Zhang ZHANGYUC@CS.STANFORD.EDU
Department of Computer Science
Stanford University
Stanford, CA 94305, USA

Lin Xiao LIN.XIAO@MICROSOFT.COM
Microsoft Research
Redmond, WA 98052, USA

Editor: Leon Bottou

Abstract

We consider a generic convex optimization problem associated with regularized empirical risk minimization of linear predictors. The problem structure allows us to reformulate it as a convex-concave saddle point problem. We propose a stochastic primal-dual coordinate (SPDC) method, which alternates between maximizing over a randomly chosen dual variable and minimizing over the primal variables. An extrapolation step on the primal variables is performed to obtain accelerated convergence rate. We also develop a mini-batch version of the SPDC method which facilitates parallel computing, and an extension with weighted sampling probabilities on the dual variables, which has a better complexity than uniform sampling on unnormalized data. Both theoretically and empirically, we show that the SPDC method has comparable or better performance than several state-of-the-art optimization methods.

Keywords: empirical risk minimization, randomized algorithms, convex-concave saddle point problems, primal-dual algorithms, computational complexity

1. Introduction

We consider a generic convex optimization problem that arises often in machine learning: regularized empirical risk minimization (ERM) of linear predictors. More specifically, let $a_1, \ldots, a_n \in \mathbb{R}^d$ be the feature vectors of n data samples, $\phi_i : \mathbb{R} \to \mathbb{R}$ be a convex loss function associated with the linear prediction $a_i^T x$, for $i = 1, \ldots, n$, and $g : \mathbb{R}^d \to \mathbb{R}$ be a convex regularization function for the predictor $x \in \mathbb{R}^d$. Our goal is to solve the following optimization problem:

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad \Big\{ P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n \phi_i(a_i^T x) + g(x) \Big\}. \tag{1}$$

Examples of this formulation include many well-known classification and regression problems. For binary classification, each feature vector a_i is associated with a label $b_i \in \{\pm 1\}$. We obtain the linear SVM (support vector machine) by setting $\phi_i(z) = \max\{0, 1 - b_i z\}$ (the hinge loss) and $g(x) = (\lambda/2)\|x\|_2^2$, where $\lambda > 0$ is a regularization parameter. Regularized logistic regression is obtained by setting $\phi_i(z) = \log(1 + \exp(-b_i z))$. For linear regression problems, each feature vector a_i is associated with a dependent variable $b_i \in \mathbb{R}$, and $\phi_i(z) = (1/2)(z - b_i)^2$. Then we get ridge regression with $g(x) = (\lambda/2)\|x\|_2^2$, and the Lasso with $g(x) = \lambda\|x\|_1$. Further background on regularized ERM in machine learning and statistics can be found, e.g., in the book by Hastie et al. (2009).

We are especially interested in developing efficient algorithms for solving the problem (1) when the number of samples n is very large. In this case, evaluating the full gradient or subgradient of the function $P(x)$ is very expensive, thus incremental methods that operate on a single component function ϕ_i at each iteration can be very attractive. There has been extensive research on incremental gradient and subgradient methods (e.g., Tseng, 1998; Blatt et al., 2007; Nedić and Bertsekas, 2001; Bertsekas, 2011, 2012) as well as variants of the stochastic gradient method (e.g., Zhang, 2004; Bottou, 2010; Duchi and Singer, 2009; Langford et al., 2009; Xiao, 2010). While the computational cost per iteration of these methods is only a small fraction, say $1/n$, of that of the batch gradient methods, their iteration complexities are much higher (they need a lot more iterations to reach the same precision). In order to better quantify the complexities of various algorithms and position our contributions, we need to make some concrete assumptions and introduce the notion of *condition number* and *batch complexity*.

1.1 Condition Number and Batch Complexity

Let γ and λ be two positive real parameters. We make the following assumption:

Assumption A *Each ϕ_i is convex and differentiable, and its derivative is $(1/\gamma)$-Lipschitz continuous (same as ϕ_i being $(1/\gamma)$-smooth), i.e.,*

$$|\phi_i'(\alpha) - \phi_i'(\beta)| \le (1/\gamma)|\alpha - \beta|, \quad \forall\, \alpha, \beta \in \mathbb{R}, \quad i = 1, \ldots, n.$$

In addition, the regularization function g is λ-strongly convex, i.e.,

$$g(x) \ge g(y) + g'(y)^T(x - y) + \frac{\lambda}{2}\|x - y\|_2^2, \quad \forall\, g'(y) \in \partial g(y), \quad x, y \in \mathbb{R}^n.$$

For example, the logistic loss $\phi_i(z) = \log(1 + \exp(-b_i z))$ is $(1/4)$-smooth, the squared error $\phi_i(z) = (1/2)(z - b_i)^2$ is 1-smooth, and the squared ℓ_2-norm $g(x) = (\lambda/2)\|x\|_2^2$ is λ-strongly convex. The hinge loss $\phi_i(z) = \max\{0, 1 - b_i z\}$ and the ℓ_1-regularization $g(x) = \lambda\|x\|_1$ do not satisfy Assumption A. Nevertheless, we can treat them using smoothing and strongly convex perturbations, respectively, so that our algorithm and theoretical framework still apply (see Section 3).

Under Assumption A, the gradient of each component function, $\nabla\phi_i(a_i^T x)$, is also Lipschitz continuous, with Lipschitz constant $L_i = \|a_i\|_2^2/\gamma \le R^2/\gamma$, where $R = \max_i \|a_i\|_2$. In other words, each $\phi_i(a_i^T x)$ is (R^2/γ)-smooth. We define the *condition number*

$$\kappa = R^2/(\lambda\gamma), \tag{2}$$

and focus on ill-conditioned problems where $\kappa \gg 1$. In statistical learning, the regularization parameter λ is usually on the order of $1/\sqrt{n}$ or $1/n$ (e.g., Bousquet and Elisseeff, 2002),

*. A short paper based on a previous version of this manuscript (arXiv:1409.3257) appeared in the Proceedings of The 32nd International Conference on Machine Learning (ICML), Lille, France, July 2015.

©2017 Yuchen Zhang and Lin Xiao.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-568.html.

thus the condition number κ is on the order of $\sqrt{n}$ or n. It can be much larger if the strong convexity in g is added purely for numerical regularization purposes (see Section 3). We note that the actual conditioning of problem (1) may be better than κ, if the empirical loss function $(1/n)\sum_{i=1}^n \phi_i(a_i^T x)$ by itself is strongly convex. In those cases, our complexity estimates in terms of κ can be loose (upper bounds), but they are still useful in comparing different algorithms for solving the same problem.

Let $P^\star$ be the optimal value of problem (1), i.e., $P^\star = \min_{x\in\mathbb{R}^d} P(x)$. In order to find an approximate solution $\hat{x}$ satisfying $P(\hat{x}) - P^\star \le \epsilon$, the classical full gradient method and its proximal variants require $\mathcal{O}((1+\kappa)\log(1/\epsilon))$ iterations (e.g., Nesterov, 2004, 2013). Accelerated full gradient methods enjoy the improved iteration complexity $\mathcal{O}((1+\sqrt{\kappa})\log(1/\epsilon))$ (Nesterov, 2004; Tseng, 2008; Beck and Teboulle, 2009; Nesterov, 2013)[1]. However, each iteration of these batch methods requires a full pass over the dataset, computing the gradient of each component function and forming their average, which cost $\mathcal{O}(nd)$ operations (assuming the features vectors $a_i \in \mathbb{R}^d$ are dense). In contrast, the stochastic gradient method and its proximal variants operate on one single component $\phi_i(a_i^T x)$ (chosen randomly) at each iteration, which only costs $\mathcal{O}(d)$. But their iteration complexities are far worse. Under Assumption A, it takes them $\mathcal{O}(\kappa/\epsilon)$ iterations to find an $\hat{x}$ such that $\mathbb{E}[P(\hat{x}) - P^\star] \le \epsilon$, where the expectation is with respect to the random choices made at all the iterations (e.g., Polyak and Juditsky, 1992; Nemirovski et al., 2009; Duchi and Singer, 2009; Langford et al., 2009; Xiao, 2010).

To make fair comparisons with batch methods, we measure the complexity of stochastic or incremental gradient methods in terms of the number of equivalent passes over the dataset required to reach an expected precision ϵ. We call this measure the *batch complexity*, which is usually obtained by dividing their iteration complexities by n. For example, the batch complexity of the stochastic gradient method is $\mathcal{O}(\kappa/(n\epsilon))$. The batch complexities of full gradient methods are the same as their iteration complexities.

By exploiting the finite average structure in (1), several recent work (e.g., Le Roux et al., 2012; Shalev-Shwartz and Zhang, 2013a; Johnson and Zhang, 2013; Xiao and Zhang, 2014; Defazio et al., 2014) proposed new variants of the stochastic gradient and dual coordinate ascent methods which achieve the iteration complexity $\mathcal{O}((n+\kappa)\log(1/\epsilon))$. Since their computational cost per iteration is $\mathcal{O}(d)$, the equivalent batch complexity is $1/n$ of their iteration complexity, i.e., $\mathcal{O}((1+\kappa/n)\log(1/\epsilon))$. This complexity has much weaker dependence on n than the full gradient methods, and also much weaker dependence on ϵ than the stochastic gradient methods.

In this paper, we propose a stochastic primal-dual coordinate (SPDC) method, which has the iteration complexity

$$\mathcal{O}\big((n+\sqrt{\kappa n})\log(1/\epsilon)\big),$$

or equivalently, the batch complexity

$$\mathcal{O}\big((1+\sqrt{\kappa/n})\log(1/\epsilon)\big). \tag{3}$$

When $\kappa > n$, this is lower than the $\mathcal{O}((1+\kappa/n)\log(1/\epsilon))$ batch complexity mentioned above. Indeed, it reaches a lower bound for minimizing finite sums established in Lan and Zhou (2015); see also Agarwal and Bottou (2015) and Woodworth and Srebro (2016). Several other recent work also achieved the same complexity bound, either with a dual or a primal accelerated randomized algorithm (Lin et al., 2015b; Lan and Zhou, 2015; Allen-Zhu, 2017) or through the proximal-point algorithm (Shalev-Shwartz and Zhang, 2015; Frostig et al., 2015; Lin et al., 2015a). We will discuss these related work in Section 5.

1. For the analysis of full gradient methods, we should use $(R^2/\gamma+\lambda)/\lambda = 1+\kappa$ as the condition number of problem (1); see Nesterov (2013, Section 5.1). Here we used the upper bound $\sqrt{1+\kappa} < 1+\sqrt{\kappa}$ for easy comparison. When $\kappa \gg 1$, the additive constant 1 can be dropped.

1.2 Outline of the Paper

Our approach is based on reformulating problem (1) as a convex-concave saddle point problem, and then devising a primal-dual algorithm to approximate the saddle point. More specifically, we replace each component function $\phi_i(a_i^T x)$ through convex conjugation, i.e.,

$$\phi_i(a_i^T x) = \sup_{y_i\in\mathbb{R}} \left\{ y_i \langle a_i, x\rangle - \phi_i^*(y_i) \right\},$$

where $\phi_i^*(y_i) = \sup_{\alpha\in\mathbb{R}}\{\alpha y_i - \phi_i(\alpha)\}$, and $\langle a_i, x\rangle$ denotes the inner product of a_i and x (which is the same as $a_i^T x$, but is more convenient for later presentation). This leads to a convex-concave saddle point problem

$$\min_{x\in\mathbb{R}^d} \max_{y\in\mathbb{R}^n} \left\{ f(x,y) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n \big(y_i\langle a_i, x\rangle - \phi_i^*(y_i)\big) + g(x) \right\}. \tag{4}$$

Under Assumption A, each ϕ_i is $(1/\gamma)$-smooth, which implies that ϕ_i^* is γ-strongly convex (see, e.g., Hiriart-Urruty and Lemaréchal, 2001, Theorem 4.2.2). In addition, the regularization g is λ-strongly convex. As a consequence, the saddle point problem (4) has a unique solution, which we denote by $(x^\star, y^\star)$.

The above saddle-point formulation allows the regularized ERM problem to be solved by the primal-dual first-order algorithms developed in Chambolle and Pock (2011). Under Assumption A, Algorithm 3 in Chambolle and Pock (2011) have the complexity $\mathcal{O}(1+\sqrt{\kappa})\log(1/\epsilon)$, which is that same as that of the accelerated full gradient methods. Our SPDC method can be viewed as a randomized coordinate variant of the primal-dual algorithm in Chambolle and Pock (2011), which achieves a much better complexity by exploiting the finite-sum structure of the regularized ERM problem.

In Section 2, we present the SPDC method as well as its convergence analysis. It alternates between maximizing f over a randomly chosen dual coordinate y_i and minimizing f over the primal variable x. In order to accelerate the convergence, an extrapolation step is applied in updating the primal variable x. We also give a mini-batch SPDC algorithm which is well suited for parallel computing.

In Section 3 and Section 4, we present two extensions of the SPDC method. We first explain how to solve problem (1) when Assumption A does not hold. The idea is to apply small regularizations to the saddle point function so that SPDC can still be applied, which results in accelerated sublinear rates for solving the original problem. The second extension is a SPDC method with non-uniform sampling. The batch complexity of this algorithm has the same form as (3), but with $\kappa = \bar{R}/(\lambda\gamma)$, where $\bar{R} = \frac{1}{n}\sum_{i=1}^n \|a_i\|$, which can be much smaller than $R = \max_i \|a_i\|$ if there is considerable variation in the norms $\|a_i\|$.

In Section 5, we discuss related work. In particular, we explain the connection of SPDC with the primal-dual batch methods in Chambolle and Pock (2011), and several recent work that achieves the same complexity as SPDC.

In Section 6, we discuss efficient implementation of the SPDC method when the feature vectors a_i are sparse. We focus on two popular cases: when g is a squared ℓ_2-norm penalty and when g is an $\ell_1 + \ell_2$ penalty. We show that the computational cost per iteration of SPDC only depends on the number of non-zero elements in the feature vectors.

In Section 7, we present experiment results comparing SPDC with several state-of-the-art optimization methods, including both batch algorithms and randomized incremental and coordinate gradient methods. On all scenarios we tested, SPDC has comparable or better performance.

The SPDC method was first proposed and analyzed in Zhang and Xiao (2015). This manuscript is a significant expansion, which includes several major technical changes, as well as new experiment results. More specifically, the main theoretical result (Theorem 1) has been strengthened to include both iterate convergence and a new result on saddle-point function convergence. This required major changes in the technical proof in Appendix A. Moreover, the convergence rate analysis of the primal-dual gap in Zhang and Xiao (2015) required additional assumptions than those in Theorem 1. In this paper (Section 2.2), we present a new proof that do not require additional assumptions. The SPDC method with non-uniform sampling (Section 4) has been extended with a more general parametrized sampling probability, and we give the optimal parameterization based on the condition number. Additional numerical experiments are presented for both synthetic and real datasets, including new comparisons with the accelerated SDCA algorithm by Shalev-Shwartz and Zhang (2015), and also comparisons between uniform and weighted sampling.

2. The SPDC Method

In this section, we describe and analyze the Stochastic Primal-Dual Coordinate (SPDC) method. The basic idea of SPDC is quite simple: to approach the saddle point of $f(x, y)$ defined in (4), we alternatively maximize f with respect to y, and minimize f with respect to x. Since the dual vector y has n coordinates and each coordinate is associated with a feature vector $a_i \in \mathbb{R}^d$, maximizing f with respect to y takes $\mathcal{O}(nd)$ computation, which can be very expensive if n is large. We reduce the computational cost by randomly picking a single coordinate of y at a time, and maximizing f only with respect to this coordinate. Consequently, the computational cost of each iteration is $\mathcal{O}(d)$.

We give the details of the SPDC method in Algorithm 1. The dual coordinate update and primal vector update are given in equations (5) and (6) respectively. Instead of maximizing f over y_k and minimizing f over x directly, we add two quadratic regularization terms to penalize $y_k^{(t+1)}$ and $x^{(t+1)}$ from deviating from $y_k^{(t)}$ and $x^{(t)}$. The parameters σ and τ control their regularization strength, which we will specify in the convergence analysis (Theorem 1). Moreover, we introduce two auxiliary variables $u^{(t)}$ and $\overline{x}^{(t)}$. Combining the initialization $u^{(0)} = (1/n)\sum_{i=1}^n y_i^{(0)} a_i$ and the update rules (5) and (7), we have

$$u^{(t)} = \frac{1}{n}\sum_{i=1}^n y_i^{(t)} a_i, \qquad t = 0, \ldots, T.$$

Equation (8) obtains $\overline{x}^{(t+1)}$ based on an extrapolation from $x^{(t)}$ and $x^{(t+1)}$. This step is similar to Nesterov's acceleration technique (Nesterov, 2004, Section 2.2), and yields faster convergence rate.

Algorithm 1: The SPDC method

Input: parameters $\tau, \sigma, \theta \in \mathbb{R}_+$, number of iterations T, initial points $x^{(0)}$ and $y^{(0)}$.

Initialize: $\overline{x}^{(0)} = x^{(0)}$, $u^{(0)} = (1/n)\sum_{i=1}^n y_i^{(0)} a_i$.

for $t = 0, 1, 2, \ldots, T-1$ **do**

Pick $k \in \{1, 2, \ldots, n\}$ uniformly at random, and execute the following updates:

$$y_i^{(t+1)} = \begin{cases} \arg\max_{\beta \in \mathbb{R}} \left\{ \beta \langle a_i, \overline{x}^{(t)} \rangle - \phi_i^*(\beta) - \frac{1}{2\sigma}(\beta - y_i^{(t)})^2 \right\} & \text{if } i = k, \\ y_i^{(t)} & \text{if } i \neq k, \end{cases} \tag{5}$$

$$x^{(t+1)} = \arg\min_{x \in \mathbb{R}^d} \left\{ g(x) + \left\langle u^{(t)} + (y_k^{(t+1)} - y_k^{(t)}) a_k,\ x \right\rangle + \frac{\|x - x^{(t)}\|_2^2}{2\tau} \right\}, \tag{6}$$

$$u^{(t+1)} = u^{(t)} + \frac{1}{n}(y_k^{(t+1)} - y_k^{(t)}) a_k, \tag{7}$$

$$\overline{x}^{(t+1)} = x^{(t+1)} + \theta(x^{(t+1)} - x^{(t)}). \tag{8}$$

end

Output: $x^{(T)}$ and $y^{(T)}$

The mini-batch SPDC method in Algorithm 2 is a natural extension of Algorithm 1. The difference between these two algorithms is that, the mini-batch SPDC method may simultaneously select more than one dual coordinates to update. Let m be the mini-batch size. During each iteration, the mini-batch SPDC method randomly picks a subset of indices $K \subset \{1, \ldots, n\}$ of size m, such that the probability of each index being picked is equal to m/n. The following is a simple procedure to achieve this. First, partition the set of indices into m disjoint subsets, so that the cardinality of each subset is equal to n/m (assuming m divides n). Then, during each iteration, randomly select a single index from each subset and add it to K. Other approaches for mini-batch selection are also possible; see the discussions in Richtárik and Takáč (2016).

In Algorithm 2, we also switched the order of updating $x^{(t+1)}$ and $u^{(t+1)}$ (comparing with Algorithm 1), to better illustrate that $x^{(t+1)}$ is obtained based on an extrapolation from $u^{(t)}$ to $u^{(t+1)}$. However, this form is not recommended in implementation, because $u^{(t)}$ is usually a dense vector even if the feature vectors a_k are sparse. Details on efficient implementation of SPDC are given in Section 6. In the following discussion, we do not make sparseness assumptions.

With a single processor, each iteration of Algorithm 2 takes $\mathcal{O}(md)$ time to accomplish. Since the updates of each coordinate y_k are independent of each other, we can use parallel computing to accelerate the mini-batch SPDC method. Concretely, we can use m processors to update the m coordinates in the subset K in parallel, then aggregate them to update $x^{(t+1)}$. In terms of wall-clock time, each iteration takes $\mathcal{O}(d)$ time, which is the same as running one iteration of the basic SPDC algorithm. Not surprisingly, we will show that the mini-batch SPDC algorithm converges faster than SPDC in terms of the iteration complexity, because it processes multiple dual coordinates in a single iteration.

Algorithm 2: The mini-batch SPDC method

Input: mini-batch size m, parameters $\tau, \sigma, \theta \in \mathbb{R}_+$, number of iterations T, and the initial points $x^{(0)}$ and $y^{(0)}$.

Initialize: $\overline{x}^{(0)} = x^{(0)}$, $u^{(0)} = (1/n)\sum_{i=1}^n y_i^{(0)} a_i$.

for $t = 0, 1, 2, \ldots, T-1$ **do**

Randomly pick a subset $K \subset \{1, 2, \ldots, n\}$ of size m, such that the probability of each index being picked is equal to m/n. Execute the following updates:

$$y_i^{(t+1)} = \begin{cases} \arg\max_{\beta \in \mathbb{R}} \left\{ \beta \langle a_i, \overline{x}^{(t)} \rangle - \phi_i^*(\beta) - \frac{1}{2\sigma}(\beta - y_i^{(t)})^2 \right\} & \text{if } i \in K, \\ y_i^{(t)} & \text{if } i \notin K, \end{cases} \tag{9}$$

$$u^{(t+1)} = u^{(t)} + \frac{1}{n} \sum_{k \in K} (y_k^{(t+1)} - y_k^{(t)}) a_k,$$

$$x^{(t+1)} = \arg\min_{x \in \mathbb{R}^d} \left\{ g(x) + \left\langle u^{(t)} + \frac{n}{m}(u^{(t+1)} - u^{(t)}),\ x \right\rangle + \frac{\|x - x^{(t)}\|_2^2}{2\tau} \right\}, \tag{10}$$

$$\overline{x}^{(t+1)} = x^{(t+1)} + \theta(x^{(t+1)} - x^{(t)}).$$

end

Output: $x^{(T)}$ and $y^{(T)}$

2.1 Convergence Analysis

Since the basic SPDC algorithm is a special case of mini-batch SPDC with $m = 1$, we only present a convergence theorem for the mini-batch version. The expectations in the following results are taken with respect to the random variables $\{K^{(0)}, \ldots, K^{(T-1)}\}$, where $K^{(t)}$ denotes the random subset $K \subset \{1, \ldots, n\}$ picked at the t-th iteration of the mini-batch SPDC method.

Theorem 1 *Suppose Assumption A holds. Let $(x^\star, y^\star)$ be the unique saddle point of f defined in (4), $R = \max\{\|a_1\|_2, \ldots, \|a_n\|_2\}$, and define*

$$\Delta^{(t)} = \left(\frac{1}{2\tau} + \frac{\lambda}{2}\right)\|x^{(t)} - x^\star\|_2^2 + \left(\frac{1}{4\sigma} + \frac{\gamma}{2}\right)\frac{\|y^{(t)} - y^\star\|_2^2}{m} + f(x^{(t)}, y^\star) - f(x^\star, y^\star) + \frac{n}{m}\left(f(x^\star, y^\star) - f(x^\star, y^{(t)})\right). \tag{11}$$

If the parameters τ, σ and θ in Algorithm 2 are chosen such that

$$\tau = \frac{1}{2R}\sqrt{\frac{m\gamma}{n\lambda}}, \qquad \sigma = \frac{1}{2R}\sqrt{\frac{n\lambda}{m\gamma}}, \qquad \theta = 1 - \left(\frac{n}{m} + 2R\sqrt{\frac{n}{m\lambda\gamma}}\right)^{-1}, \tag{12}$$

then for each $t \geq 1$, the mini-batch SPDC algorithm achieves

$$\mathbb{E}[\Delta^{(t)}] \leq \theta^t \left(\Delta^{(0)} + \frac{\|y^{(0)} - y^\star\|_2^2}{4m\sigma}\right).$$

Comparing with Theorem 1 in Zhang and Xiao (2015), our definition of $\Delta^{(t)}$ in (11) includes the additional terms $f(x^{(t)}, y^\star) - f(x^\star, y^\star) + \frac{n}{m}\left(f(x^\star, y^\star) - f(x^\star, y^{(t)})\right)$. This is a weighted sum of the primal and dual gaps for the saddle-point problem. It will help us establish the convergence rate of the objective value for the ERM problem in Section 2.2, which is missing in Zhang and Xiao (2015).

The proof of Theorem 1 is given in Appendix A. The following corollary establishes the expected iteration complexity of mini-batch SPDC for obtaining an ϵ-accurate solution.

Corollary 2 *Suppose Assumption A holds and the parameters τ, σ and θ are set as in (12). In order for Algorithm 2 to obtain*

$$\mathbb{E}[\|x^{(T)} - x^\star\|_2^2] \leq \epsilon, \qquad \mathbb{E}[\|y^{(T)} - y^\star\|_2^2] \leq \epsilon, \tag{13}$$

it suffices to have the number of iterations T satisfy

$$T \geq \left(\frac{n}{m} + 2R\sqrt{\frac{n}{m\lambda\gamma}}\right)\log\left(\frac{C}{\epsilon}\right),$$

where

$$C = \frac{\Delta^{(0)} + \left\|y^{(t)} - y^\star\right\|_2^2/(4m\sigma)}{\min\left\{1/(2\tau) + \lambda/2,\ (1/(4\sigma) + \gamma/2)/m\right\}}.$$

Proof By Theorem 1, we have $\mathbb{E}[\|x^{(t)} - x^\star\|_2^2] \leq \theta^t C$ and $\mathbb{E}[\|y^{(t)} - y^\star\|_2^2] \leq \theta^t C$ for each $t > 0$. To obtain (13), it suffices to ensure that $\theta^T C \leq \epsilon$, which is equivalent to

$$T \geq \frac{\log(C/\epsilon)}{-\log(\theta)} = \frac{\log(C/\epsilon)}{-\log\left(1 - \left((n/m) + 2R\sqrt{(n/m)/(\lambda\gamma)}\right)^{-1}\right)}.$$

Applying the inequality $-\log(1-x) \geq x$ to the denominator above completes the proof. ■

Recall the definition of the condition number $\kappa = R^2/(\lambda\gamma)$ in (2). Corollary 2 establishes that the iteration complexity of the mini-batch SPDC method for achieving (13) is

$$\mathcal{O}\left(\left((n/m) + \sqrt{\kappa(n/m)}\right)\log(1/\epsilon)\right).$$

So a larger batch size m leads to less number of iterations. In the extreme case of $n = m$, we obtain a full batch algorithm, which has iteration or batch complexity $\mathcal{O}((1+\sqrt{\kappa})\log(1/\epsilon))$. This complexity is also shared by the accelerated gradient methods (Nesterov, 2004, 2013), as well as the batch primal-dual algorithm of Chambolle and Pock (2011); see discussions in Section 1.1 and related work in Section 5.

Since an equivalent pass over the dataset corresponds to n/m iterations, the batch complexity (the number of equivalent passes over the data) of mini-batch SPDC is

$$\mathcal{O}\left(\left(1 + \sqrt{\kappa(m/n)}\right)\log(1/\epsilon)\right).$$

The above expression implies that a smaller batch size m leads to less number of passes through the data. In this sense, the basic SPDC method with $m = 1$ is the most efficient one. However, if we prefer the least amount of wall-clock time, then the best choice is to choose a mini-batch size m that matches the number of parallel processors available.

2.2 Convergence Rate of Primal-Dual Gap

In the previous subsection, we established iteration complexity of the mini-batch SPDC method in terms of approximating the saddle point of the minimax problem (4), more specifically, to meet the requirement in (13). Next we show that it has the same order of complexity in reducing the primal-dual objective gap $P(x^{(t)}) - D(y^{(t)})$, where $P(x)$ is defined in (1) and

$$D(y) \stackrel{\text{def}}{=} \min_{x\in\mathbb{R}^d} f(x,y) = \frac{1}{n}\sum_{i=1}^n -\phi_i^*(y_i) - g^*\Big(-\frac{1}{n}\sum_{i=1}^n y_i a_i\Big). \tag{14}$$

where $g^*(u) = \sup_{x\in\mathbb{R}^d}\{x^T u - g(x)\}$ is the conjugate function of g.

Under Assumption A, the function $f(x,y)$ defined in (4) has a unique saddle point $(x^\star, y^\star)$, and

$$P(x^\star) = f(x^\star, y^\star) = D(y^\star).$$

However, in general, for any point $(x,y) \in \text{dom}(g)\times\text{dom}(\phi^*)$, we have

$$P(x) = \max_y f(x,y) \geq f(x,y^\star), \qquad D(y) = \min_x f(x,y) \leq f(x^\star, y).$$

Thus the result in Theorem 1 does not translate directly into a convergence bound on the primal-dual gap. We need to bound $P(x)$ and $D(y)$ by $f(x,y^\star)$ and $f(x^\star,y)$, respectively, in the opposite directions. For this purpose, we need the following result extracted from Yu et al. (2015). We provide the proof in Appendix B for completeness.

Lemma 3 *Suppose Assumption A holds. Let $(x^\star, y^\star)$ be the unique saddle-point of $f(x,y)$, and $R = \max_{1\leq i\leq n}\|a_i\|_2$. Then for any point $(x,y)\in\text{dom}(g)\times\text{dom}(\phi^*)$, we have*

$$P(x) \leq f(x,y^\star) + \frac{R^2}{2\gamma}\|x-x^\star\|_2^2, \qquad D(y) \geq f(x^\star,y) - \frac{R^2}{2\lambda n}\|y-y^\star\|_2^2.$$

Corollary 4 *Suppose Assumption A holds and the parameters τ, σ and θ are set as in (12). Let $\widetilde{\Delta}^{(0)} := \Delta^{(0)} + \frac{\|y^{(0)}-y^\star\|_2^2}{4m\sigma}$. Then for any $\epsilon \geq 0$, the iterates of Algorithm 2 satisfy*

$$\mathbb{E}[P(x^{(T)}) - D(y^{(T)})] \leq \epsilon$$

whenever

$$T \geq \left(\frac{n}{m} + 2R\sqrt{\frac{n}{m\lambda\gamma}}\right)\log\left(\left(1+\frac{R^2}{\lambda\gamma}\right)\frac{\widetilde{\Delta}^{(0)}}{\epsilon}\right).$$

Proof The function $f(x,y^\star)$ is strongly convex in x with parameter λ, and $x^\star$ is the minimizer. Similarly, $-f(x^\star,y)$ is strongly convex in y with parameter γ/n, and is minimized by $y^\star$. Therefore,

$$\frac{\lambda}{2}\|x^{(t)}-x^\star\|_2^2 \leq f(x^{(t)},y^\star) - f(x^\star,y^\star), \quad \frac{\gamma}{2n}\|y^{(t)}-y^\star\|_2^2 \leq f(x^\star,y^\star) - f(x^\star,y^{(t)}). \tag{15}$$

We bound the following weighted primal-dual gap

$$\begin{aligned}
& P(x^{(t)}) - P(x^\star) + \frac{n}{m}\Big(D(y^\star) - D(y^{(t)})\Big) \\
\leq\ & f(x^{(t)},y^\star) - f(x^\star,y^\star) + \frac{n}{m}\Big(f(x^\star,y^\star) - f(x^\star,y^{(t)})\Big) + \frac{R^2}{2\gamma}\|x^{(t)}-x^\star\|_2^2 + \frac{n}{m}\frac{R^2}{2n\lambda}\|y^{(t)}-y^\star\|_2^2 \\
\leq\ & \Delta^{(t)} + \frac{R^2}{\lambda\gamma}\left(\frac{\lambda}{2}\|x^{(t)}-x^\star\|_2^2 + \frac{n}{m}\frac{\gamma}{2n}\|y^{(t)}-y^\star\|_2^2\right) \\
\leq\ & \Delta^{(t)} + \frac{R^2}{\lambda\gamma}\left(f(x^{(t)},y^\star) - f(x^\star,y^\star) + \frac{n}{m}\Big(f(x^\star,y^\star) - f(x^\star,y^{(t)})\Big)\right) \\
\leq\ & \left(1+\frac{R^2}{\lambda\gamma}\right)\Delta^{(t)}.
\end{aligned}$$

The first inequality above is due to Lemma 3, the second and fourth inequalities are due to the definition of $\Delta^{(t)}$, and the third inequality is due to (15). Taking expectations on both sides of the above inequality, then applying Theorem 1, we obtain

$$\mathbb{E}\left[P(x^{(t)}) - P(x^\star) + \frac{n}{m}\Big(D(y^\star) - D(y^{(t)})\Big)\right] \ \leq\ \theta^t\left(1+\frac{R^2}{\lambda\gamma}\right)\widetilde{\Delta}^{(0)} = (1+\kappa)\widetilde{\Delta}^{(t)}.$$

Since $n \geq m$ and $D(y^\star) - D(y^{(t)})) \geq 0$, this implies the desired result. ■

3. Extensions to Non-Smooth or Non-Strongly Convex Functions

The complexity bounds established in Section 2 require each ϕ_i be $(1/\gamma)$-smooth, and the function g be λ-strongly convex. For general loss functions where either or both of these conditions fail (e.g., the hinge loss and ℓ_1-regularization), we can slightly perturb the saddle-point function $f(x,y)$ so that the SPDC method can still be applied.

To be concise, we only consider the case where neither ϕ_i is smooth nor g is strongly convex. Instead, we only assume that each ϕ_i and g are convex and Lipschitz continuous, and $f(x,y)$ has a saddle point $(x^\star,y^\star)$. We choose a scalar $\delta > 0$ and consider the modified saddle-point function:

$$f_\delta(x,y) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n\left(y_i\langle a_i,x\rangle - \Big(\phi_i^*(y_i) + \frac{\delta y_i^2}{2}\Big)\right) + g(x) + \frac{\delta}{2}\|x\|_2^2. \tag{16}$$

Denote the saddle-point of f_δ by $(x_\delta^\star, y_\delta^\star)$. We employ the mini-batch SPDC method in Algorithm 2 to approximate $(x_\delta^\star, y_\delta^\star)$, treating $\phi_i^* + \frac{\delta}{2}(\cdot)^2$ as ϕ_i^* and $g + \frac{\delta}{2}\|\cdot\|_2^2$ as g, which are all δ-strongly convex. We note that adding strongly convex perturbation on ϕ_i^* is equivalent to smoothing ϕ_i, which becomes $(1/\delta)$-smooth (see, e.g., Nesterov, 2005). Letting $\gamma = \lambda = \delta$, the parameters τ, σ and θ in (12) become

$$\tau = \frac{1}{2R}\sqrt{\frac{m}{n}}, \qquad \sigma = \frac{1}{2R}\sqrt{\frac{n}{m}}, \qquad \text{and} \qquad \theta = 1 - \left(\frac{n}{m} + \frac{2R}{\delta}\sqrt{\frac{n}{m}}\right)^{-1}.$$

Although $(x_\delta^\star, y_\delta^\star)$ is not exactly the saddle point of f, the following corollary shows that applying SPDC to the perturbed function f_δ effectively minimizes the original loss function P. Similar results for the convergence of the primal-dual gap can also be established.

Corollary 5 *Assume that each ϕ_i is convex and G_ϕ-Lipschitz continuous, and g is convex and G_g-Lipschitz continuous. In addition, assume that f has a saddle point $(x^\star, y^\star)$ and let the unique saddle point of f_δ be $(x_\delta^\star, y_\delta^\star)$. Define two constants:*

$$C_1 = (\|x^\star\|_2^2 + G_\phi^2), \qquad C_2 = (G_\phi R + G_g)^2\Big(\frac{\Delta_\delta^{(0)} + \|y^{(0)} - y_\delta^\star\|_2^2 R/(4\sqrt{mn})}{1/(2\tau) + \lambda/2}\Big),$$

where $\Delta_\delta^{(0)}$ is evaluated as in (11) but in terms of the perturbed function f_δ. If we choose $\delta \le \epsilon/C_1$, then we have $\mathbb{E}[P(x^{(T)}) - P(x^\star)] \le \epsilon$ whenever

$$T \ge \left(\frac{n}{m} + \frac{2R}{\delta}\sqrt{\frac{n}{m}}\right)\log\left(\frac{4C_2}{\epsilon^2}\right).$$

Proof Let $\widetilde{y} = \arg\max_y f(x_\delta^\star, y)$ be a shorthand notation. We have

$$\begin{aligned} P(x_\delta^\star) &\overset{(i)}{=} f(x_\delta^\star, \widetilde{y}) \overset{(ii)}{\le} f_\delta(x_\delta^\star, \widetilde{y}) + \frac{\delta\|\widetilde{y}\|_2^2}{2n} \overset{(iii)}{\le} f_\delta(x_\delta^\star, y_\delta^\star) + \frac{\delta\|\widetilde{y}\|_2^2}{2n} \overset{(iv)}{\le} f_\delta(x^\star, y_\delta^\star) + \frac{\delta\|\widetilde{y}\|_2^2}{2n} \\ &\overset{(v)}{\le} f(x^\star, y_\delta^\star) + \frac{\delta\|x^\star\|_2^2}{2} + \frac{\delta\|\widetilde{y}\|_2^2}{2n} \overset{(vi)}{\le} f(x^\star, y^\star) + \frac{\delta\|x^\star\|_2^2}{2} + \frac{\delta\|\widetilde{y}\|_2^2}{2n} \\ &\overset{(vii)}{=} P(x^\star) + \frac{\delta\|x^\star\|_2^2}{2} + \frac{\delta\|\widetilde{y}\|_2^2}{2n}. \end{aligned}$$

Here, equations (i) and (vii) use the definition of the function f, inequalities (ii) and (v) use the definition of the function f_δ, inequalities (iii) and (iv) use the fact that $(x_\delta^\star, y_\delta^\star)$ is the saddle point of f_δ, and inequality (vi) is due to the fact that $(x^\star, y^\star)$ is the saddle point of f.

Since ϕ_i is G_ϕ-Lipschitz continuous, the domain of ϕ_i^* is in the interval $[-G_\phi, G_\phi]$, which implies $\|\widetilde{y}\|_2^2 \le nG_\phi^2$ (see, e.g., (Shalev-Shwartz and Zhang, 2015, Lemma 1)). Thus, we have

$$P(x_\delta^\star) - P(x^\star) \le \frac{\delta}{2}(\|x^\star\|_2^2 + G_\phi^2) = \frac{\delta}{2}C_1. \tag{17}$$

On the other hand, since P is $(G_\phi R + G_g)$-Lipschitz continuous, Theorem 1 implies

$$\begin{aligned} \mathbb{E}[P(x^{(T)}) - P(x_\delta^\star)] &\le (G_\phi R + G_g)\mathbb{E}[\|x^{(T)} - x_\delta^\star\|_2] \\ &\le \sqrt{C_2}\left(1 - \left(\frac{n}{m} + \frac{2R}{\delta}\sqrt{\frac{n}{m}}\right)^{-1}\right)^{T/2}. \end{aligned} \tag{18}$$

Combining (17) and (18), in order to obtain $\mathbb{E}[P(x^{(T)}) - P(x^\star)] \le \epsilon$, it suffices to have $C_1\delta \le \epsilon$ and

$$\sqrt{C_2}\left(1 - \left(\frac{n}{m} + \frac{2R}{\delta}\sqrt{\frac{n}{m}}\right)^{-1}\right)^{T/2} \le \frac{\epsilon}{2}. \tag{19}$$

ϕ_i	g	iteration complexity $\widetilde{\mathcal{O}}(\cdot)$
$(1/\gamma)$-smooth	λ-strongly convex	$n/m + \sqrt{(n/m)/(\lambda\gamma)}$
$(1/\gamma)$-smooth	non-strongly convex	$n/m + \sqrt{(n/m)/(\epsilon\gamma)}$
non-smooth	λ-strongly convex	$n/m + \sqrt{(n/m)/(\epsilon\lambda)}$
non-smooth	non-strongly convex	$n/m + \sqrt{n/m}/\epsilon$

Table 1: Iteration complexities of the SPDC method under different assumptions on the functions ϕ_i and g. For the last three cases, we solve the perturbed saddle-point problem with $\delta = \epsilon/C_1$.

The corollary is established by finding the smallest T that satisfies inequality (19). ■

There are two other cases that can be considered: when ϕ_i is not smooth but g is strongly convex, and when ϕ_i is smooth but g is not strongly convex. They can be handled with the same technique described above, and we omit the details here. Alternatively, it is possible to use the techniques described in Chambolle and Pock (2011, Section 5.1) to obtain accelerated sublinear convergence rates without using strongly convex perturbations. In Table 1, we list the complexities of the mini-batch SPDC method for finding an ϵ-optimal solution of problem (1) under various assumptions. Similar results are also obtained in Shalev-Shwartz and Zhang (2015).

4. SPDC with Non-Uniform Sampling

One potential drawback of the SPDC algorithm is that, its convergence rate depends on a problem-specific constant R, which is the largest ℓ_2-norm of the feature vectors a_i. As a consequence, the algorithm may perform badly on unnormalized data, especially if the ℓ_2-norms of some feature vectors are substantially larger than others. In this section, we propose an extension of the SPDC method to mitigate this problem, which is given in Algorithm 3.

The basic idea is to use non-uniform sampling in picking the dual coordinate to update at each iteration. In Algorithm 3, we pick coordinate k with the probability

$$p_k = (1-\alpha)\frac{1}{n} + \alpha\frac{\|a_k\|_2}{\sum_{i=1}^n \|a_i\|_2}, \qquad k = 1, \ldots, n, \tag{20}$$

where $\alpha \in (0,1)$ is a parameter. In other words, this distribution is a (strict) convex combination of the uniform distribution and the distribution that is proportional to the feature norms. Therefore, instances with large feature norms are sampled more frequently, controlled by α. Simultaneously, we adopt an adaptive regularization in step (21), imposing stronger regularization on such instances. In addition, we adjust the weight of a_k in (23) for updating the primal variable. As a consequence, the convergence rate of Algorithm 3 depends on the average norm of feature vectors, as well as the parameter α. This is summarized in the following theorem, whose proof is given in Appendix C.

Algorithm 3: SPDC method with weighted sampling

Input: parameters $\tau, \sigma, \theta \in \mathbb{R}_+$, number of iterations T, initial points $x^{(0)}$ and $y^{(0)}$.

Initialize: $\overline{x}^{(0)} = x^{(0)}$, $u^{(0)} = (1/n)\sum_{i=1}^n y_i^{(0)} a_i$.

for $t = 0, 1, 2, \ldots, T-1$ **do**

Randomly pick $k \in \{1, 2, \ldots, n\}$, with probability p_k given in (20).

Execute the following updates:

$$y_i^{(t+1)} = \begin{cases} \arg\max_{\beta\in\mathbb{R}} \left\{ \beta\langle a_i, \overline{x}^{(t)}\rangle - \phi_i^*(\beta) - \frac{p_i n}{2\sigma}(\beta - y_i^{(t)})^2 \right\} & i = k, \\ y_i^{(t)} & i \neq k, \end{cases} \tag{21}$$

$$u^{(t+1)} = u^{(t)} + \frac{1}{n}(y_k^{(t+1)} - y_k^{(t)}) a_k, \tag{22}$$

$$x^{(t+1)} = \arg\min_{x\in\mathbb{R}^d} \left\{ g(x) + \left\langle u^{(t)} + \frac{1}{p_k}(u^{(t+1)} - u^{(t)}),\ x \right\rangle + \frac{\|x - x^{(t)}\|_2^2}{2\tau} \right\}, \tag{23}$$

$$\overline{x}^{(t+1)} = x^{(t+1)} + \theta(x^{(t+1)} - x^{(t)}).$$

end

Output: $x^{(T)}$ and $y^{(T)}$.

Theorem 6 *Suppose Assumption A holds. Let* $R := \max_i \|a_i\|_2$, $\overline{R} := \frac{1}{n}\sum_{i=1}^n \|a_i\|_2$ *and* $R_\alpha := \big((1-\alpha)/R + \alpha/\overline{R}\big)^{-1}$. *If the parameters* τ, σ, θ *in Algorithm 3 are chosen such that*

$$\tau = \frac{1}{2R_\alpha}\sqrt{\frac{\gamma}{n\lambda}}, \qquad \sigma = \frac{1}{2R_\alpha}\sqrt{\frac{n\lambda}{\gamma}}, \qquad \theta = 1 - \left(\frac{n}{1-\alpha} + R_\alpha\sqrt{\frac{n}{\lambda\gamma}}\right)^{-1}, \tag{24}$$

then for each $t \geq 1$, *we have*

$$\begin{aligned} &\Big(\frac{1}{2\tau} + \lambda\Big)\mathbb{E}\big[\|x^{(t)} - x^\star\|_2^2\big] + \Big(\frac{1}{4\sigma} + \frac{\gamma}{n}\Big)\mathbb{E}\big[\|y^{(t)} - y^\star\|_2^2\big] \\ \leq\ & \theta^t\left(\Big(\frac{1}{2\tau} + \lambda\Big)\|x^{(0)} - x^\star\|_2^2 + \Big(\frac{1}{2\sigma} + \frac{\gamma}{1-\alpha}\Big)\|y^{(0)} - y^\star\|_2^2\right). \end{aligned}$$

Note that $R_\alpha \leq \overline{R}/\alpha$ always holds. If we choose $\alpha = 1/2$, then the contraction ratio θ is bounded by $1 - \left(\frac{n}{1-\alpha} + 2\overline{R}\sqrt{\frac{n}{\lambda\gamma}}\right)^{-1}$. Comparing this bound with Theorem 1 with $m = 1$, the convergence rate of Theorem 6 is determined by the average norm of the features, $\overline{R} = \frac{1}{n}\sum_{i=1}^n \|a_i\|_2$, instead of the largest one $R = \max_i \|a_i\|_2$. This difference makes Algorithm 3 more robust to unnormalized feature vectors. For example, if the a_i's are sampled i.i.d. from a multivariate normal distribution, then $\max_i\{\|a_i\|_2\}$ almost surely goes to infinity as $n \to \infty$, but the average norm $\frac{1}{n}\sum_{i=1}^n \|a_i\|_2$ converges to $\mathbb{E}[\|a_i\|_2]$.

Since θ is a bound on the convergence factor, we would like to make it as small as possible. Let $\rho := R/\overline{R} - 1$. The expression of θ in (24) can be minimized by choosing

$$\alpha^\star = \begin{cases} 0 & \text{if } \rho \leq \sqrt{n/\kappa}, \\ \frac{\rho^{1/2}(\kappa/n)^{1/4} - 1}{\rho^{1/2}(\kappa/n)^{1/4} + \rho} & \text{if } \rho > \sqrt{n/\kappa}. \end{cases} \tag{25}$$

where $\kappa = R^2/(\lambda\gamma)$ is the condition number. The value of $\alpha^\star$ will be equal to zero if the condition number is large enough, and increases slowly to one as the condition number increases. Thus, we choose a (more conservative) uniform distribution for ill-conditioned problems, but a more aggressively weighted distribution for well-conditioned problems.

For simplicity of presentation, we described in Algorithm 3 a weighted sampling SPDC method with single dual coordinate update, i.e., the case of $m = 1$. In fact, the non-uniform sampling scheme can also be extended to mini-batch SPDC. For mini-batch size $m > 1$, we randomly pick a subset of indices $K \subset \{1, 2, \ldots, n\}$ of size m. The probability of $i \in K$ is denoted by p_i and should satisfy the constraint:

$$\min\Big\{1, m\Big(\frac{1-\alpha}{n} + \frac{\alpha\|a_i\|_2}{n\overline{R}}\Big)\Big\} \ \leq\ p_i \ \leq\ 1 \tag{26}$$

for $i = 1, \ldots, n$, and

$$\sum_{i=1}^n p_i = m.$$

This constraint can be satisfied by first adding all indices $\{i : p_i = 1\}$ to the set K, then sampling without replacement from the remaining indices in order to make $|K| = m$. More concretely, there is an efficient sampling-without-replacement algorithm (Chao, 1982) which adds each remaining index i to the set K with probability proportional to $\frac{1-\alpha}{n} + \frac{\alpha\|a_i\|_2}{n\overline{R}}$. It can be verified that the lower bound in (26) holds with such a procedure.

For the mini-batch extension, we replace the updates (21)-(23) by the following updates:

$$y_i^{(t+1)} = \begin{cases} \arg\max_{\beta\in\mathbb{R}} \left\{ \beta\langle a_i, \overline{x}^{(t)}\rangle - \phi_i^*(\beta) - \frac{p_i n}{2\sigma m}(\beta - y_i^{(t)})^2 \right\} & i \in K, \\ y_i^{(t)} & i \notin K, \end{cases}$$

$$u^{(t+1)} = u^{(t)} + \frac{1}{n}\sum_{k\in K}(y_k^{(t+1)} - y_k^{(t)}) a_k,$$

$$x^{(t+1)} = \arg\min_{x\in\mathbb{R}^d} \left\{ g(x) + \left\langle u^{(t)} + \frac{1}{n}\sum_{k\in K}\frac{y_k^{(t+1)} - y_k^{(t)}}{p_k} a_k,\ x \right\rangle + \frac{\|x - x^{(t)}\|_2^2}{2\tau} \right\},$$

which resembles the updates of Algorithm 2. Similar to the mini-batch SPDC, we are able to show that by increasing the batch size m, the convergence rate of the algorithm will be improved. On the other hand, the lower bound on p_i given by constraint (26) implies that with a proper choice of α (e.g. $\alpha = 1/2$), the convergence rate will depend on

$$\max\left\{\overline{R},\ \frac{m}{n}R\right\},$$

instead of the maximum norm R. For $m = 1$, we have $\max\{\overline{R}, \frac{m}{n}R\} = \overline{R}$, so that it captures the theoretical guarantee for Algorithm 3 as a special case. We omit the proof details.

5. Related Work

Chambolle and Pock (2011) considered a class of convex optimization problems with the following saddle-point structure:

$$\min_{x\in\mathbb{R}^d}\max_{y\in\mathbb{R}^n} \big\{\langle Kx, y\rangle + G(x) - F^*(y)\big\}, \tag{27}$$

algorithm	τ	σ	θ	batch complexity
Chambolle-Pock	$\frac{\sqrt{n}}{\|A\|_2}\sqrt{\frac{\gamma}{\lambda}}$	$\frac{n\sqrt{n}}{\|A\|_2}\sqrt{\frac{\lambda}{\gamma}}$	$1-\frac{1}{1+\|A\|_2/(2\sqrt{n\lambda\gamma})}$	$\left(1+\frac{\|A\|_2}{2\sqrt{n\lambda\gamma}}\right)\log(1/\epsilon)$
SPDC with $m=n$	$\frac{1}{R}\sqrt{\frac{\gamma}{\lambda}}$	$\frac{1}{R}\sqrt{\frac{\lambda}{\gamma}}$	$1-\frac{1}{1+R/\sqrt{\lambda\gamma}}$	$\left(1+\frac{R}{\sqrt{\lambda\gamma}}\right)\log(1/\epsilon)$
SPDC with $m=1$	$\frac{1}{R}\sqrt{\frac{\gamma}{n\lambda}}$	$\frac{1}{R}\sqrt{\frac{n\lambda}{\gamma}}$	$1-\frac{1}{n+R\sqrt{n/\lambda\gamma}}$	$\left(1+\frac{R}{\sqrt{n\lambda\gamma}}\right)\log(1/\epsilon)$

Table 2: Comparing step sizes and complexity of SPDC with Chambolle and Pock (2011, Algorithm 3, Theorem 3). Here $A\in\mathbb{R}^{n\times d}$ and its spectral norm $\|A\|_2$ usually grows with n, but always bounded by $\sqrt{n}R$.

where $K\in\mathbb{R}^{m\times d}$, G and F^* are proper closed convex functions, with F^* itself being the conjugate of a convex function F. They developed the following first-order primal-dual algorithm:

$$y^{(t+1)}=\arg\max_{y\in\mathbb{R}^n}\left\{\langle K\overline{x}^{(t)},y\rangle-F^*(y)-\frac{1}{2\sigma}\|y-y^{(t)}\|_2^2\right\}, \tag{28}$$

$$x^{(t+1)}=\arg\min_{x\in\mathbb{R}^d}\left\{\langle K^Ty^{(t+1)},x\rangle+G(x)+\frac{1}{2\tau}\|x-x^{(t)}\|_2^2\right\}, \tag{29}$$

$$\overline{x}^{(t+1)}=x^{(t+1)}+\theta(x^{(t+1)}-x^{(t)}). \tag{30}$$

When both F^* and G are strongly convex and the parameters τ, σ and θ are chosen appropriately, this algorithm obtains accelerated linear convergence rate (Chambolle and Pock, 2011, Theorem 3).

We can map the saddle-point problem (4) into the form of (27) by letting $A=[a_1,\ldots,a_n]^T$ and

$$K=\frac{1}{n}A,\qquad G(x)=g(x),\qquad F^*(y)=\frac{1}{n}\sum_{i=1}^n\phi_i^*(y_i). \tag{31}$$

The SPDC method developed in this paper can be viewed as an extension of the batch method (28)-(30), where the dual update step (28) is replaced by a single coordinate update (5) or a mini-batch update (9). However, in order to obtain accelerated convergence rate, more subtle changes are necessary in the primal update step. More specifically, we introduced the auxiliary variable $u^{(t)}=\frac{1}{n}\sum_{i=1}^n y_i^{(t)}a_i=K^Ty^{(t)}$, and replaced the primal update step (29) by (6) and (10). The primal extrapolation step (30) stays the same.

To compare the batch complexity of SPDC with that of (28)-(30), we use the following facts implied by Assumption A and the relations in (31):

$$\|K\|_2=\frac{1}{n}\|A\|_2,\quad G(x)\text{ is }\lambda\text{-strongly convex},\quad\text{and }F^*(y)\text{ is }(\gamma/n)\text{-strongly convex}.$$

Based on these conditions, we list in Table 2 the equivalent parameters used by Algorithm 3 in Chambolle and Pock (2011) and the batch complexity obtained in Theorem 3 of that paper, and compare them with SPDC.

The batch complexity of the Chambolle-Pock algorithm is $\widetilde{\mathcal{O}}(1+\|A\|_2/(2\sqrt{n\lambda\gamma}))$, where the $\widetilde{\mathcal{O}}(\cdot)$ notation hides the $\log(1/\epsilon)$ factor. We can bound the spectral norm $\|A\|_2$ by the Frobenius norm $\|A\|_F$ and obtain

$$\|A\|_2\le\|A\|_F\le\sqrt{n}\max_i\{\|a_i\|_2\}=\sqrt{n}R.$$

(Note that the second inequality above would be an equality if the columns of A are normalized.) So in the worst case, the batch complexity of the Chambolle-Pock algorithm becomes

$$\widetilde{\mathcal{O}}\left(1+R/\sqrt{\lambda\gamma}\right)=\widetilde{\mathcal{O}}\left(1+\sqrt{\kappa}\right),\qquad\text{where }\kappa=R^2/(\lambda\gamma),$$

which matches the worst-case complexity of the accelerated gradient methods (Nesterov, 2004, 2013); see Section 1.1 and also the discussions in Lin et al. (2015b, Section 5). This is also of the same order as the complexity of SPDC with $m=n$ (see Section 2.1). When the condition number $\kappa\gg1$, they can be $\sqrt{n}$ worse than the batch complexity of SPDC with $m=1$, which is $\widetilde{\mathcal{O}}(1+\sqrt{\kappa/n})$.

If either $G(x)$ or $F^*(y)$ in (27) is not strongly convex, Chambolle and Pock (2011, Section 5.1) proposed variants of the primal-dual batch algorithm to achieve accelerated sublinear convergence rates. It is also possible to extend them to coordinate update methods for solving problem (1) when either ϕ_i^* or g is not strongly convex. Their complexities would be similar to those in Table 1.

Our algorithms and theory can be readily generalized to solve the problem of

$$\underset{x\in\mathbb{R}^d}{\text{minimize}}\quad\frac{1}{n}\sum_{i=1}^n\phi_i(A_i^Tx)+g(x),$$

where each A_i is an $d_i\times d$ matrix, and $\phi_i:\mathbb{R}^{d_i}\to\mathbb{R}$ is a smooth convex function. This more general formulation is used, e.g., in Shalev-Shwartz and Zhang (2015). Most recently, Lan and Zhou (2015) considered the case with $d_i=d$ and $A_i=I_d$, which corresponding to a general class of problems with the finite-sum (or finite-average) structure. He extended the primal-dual algorithm by replacing the quadratic penalty terms in (5) and (21) with the Bregman divergence associated with the loss functions themselves. This led to an algorithm that does not rely on computing the proximal mapping of the conjugate $\phi_i^\star$, but only requires computing the primal gradient $\nabla\phi_i$ at a particular sequence of the primal variables. As a result, the algorithm in Lan and Zhou (2015) can be considered as a (primal-only or dual-free) randomized incremental gradient algorithm, which share the same order of iteration complexity as SPDC.

5.1 Dual Coordinate Ascent Methods

We can also solve the primal problem (1) via its dual:

$$\underset{y\in\mathbb{R}^n}{\text{maximize}}\ \left\{D(y)\ \stackrel{\text{def}}{=}\ \frac{1}{n}\sum_{i=1}^n-\phi_i^*(y_i)-g^*\Big(-\frac{1}{n}\sum_{i=1}^n y_ia_i\Big)\right\}, \tag{32}$$

where $g^*(u)=\sup_{x\in\mathbb{R}^d}\{x^Tu-g(x)\}$ is the convex conjugate of g. Due to the problem structure, it is well-known that coordinate ascent methods can be more efficient than full

gradient methods for solving this problem (e.g., Platt, 1999; Chang et al., 2008; Hsieh et al., 2008; Shalev-Shwartz and Zhang, 2013a). In the stochastic dual coordinate ascent (SDCA) method a dual coordinate y_i is picked at random during each iteration and updated to increase the dual objective value. Shalev-Shwartz and Zhang (2013a) showed that the iteration complexity of SDCA is $O\left((n+\kappa)\log(1/\epsilon)\right)$, which corresponds to the batch complexity $\mathcal{O}\left((1+\kappa/n)\log(1/\epsilon)\right)$.

For more general convex optimization problems, there is a vast literature on coordinate descent methods; see, e.g., the recent overview by Wright (2015). In particular, the work of Nesterov (2012) on randomized coordinate descent sparked a lot of recent activities on this topic. Richtárik and Takáč (2014) extended the algorithm and analysis to composite convex optimization. When applied to the dual problem (32), it becomes one variant of the SDCA algorithm studied in Shalev-Shwartz and Zhang (2013a). Mini-batch and distributed versions of SDCA have been proposed and analyzed in Takáč et al. (2013) and Yang (2013) respectively. Non-uniform sampling schemes similar to the one used in Algorithm 3 have been studied for both stochastic gradient and dual coordinate ascent methods (e.g., Needell et al., 2016; Xiao and Zhang, 2014; Zhao and Zhang, 2015; Qu et al., 2015).

Shalev-Shwartz and Zhang (2013b) proposed an accelerated mini-batch SDCA method which incorporates additional primal updates than SDCA, and bears some similarity to our mini-batch SPDC method. They showed that its complexity interpolates between that of SDCA and accelerated gradient methods by varying the mini-batch size m. In particular, for $m = n$, it matches that of the accelerated gradient methods (as SPDC does). But for $m = 1$, the complexity of their method is the same as SDCA, which is worse than SPDC for ill-conditioned problems.

In addition, Shalev-Shwartz and Zhang (2015) developed an accelerated proximal SDCA method which achieves the same batch complexity $\widetilde{\mathcal{O}}\big(1+\sqrt{\kappa/n}\big)$ as SPDC. Their method is an inner-outer iteration procedure, where the outer loop is a full-dimensional accelerated gradient method in the primal space $x \in \mathbb{R}^d$. At each iteration of the outer loop, the SDCA method (Shalev-Shwartz and Zhang, 2013a) is called to solve the dual problem (32) with customized regularization parameter and precision. In contrast, SPDC is a straightforward single-loop coordinate optimization methods. Two recent works extended the inner-outer iteration method to derive more general accelerated proximal-point algorithms: Frostig et al. (2015) and Lin et al. (2015a). Basically, one can replace the inner-loop SDCA algorithm by other efficient algorithms such as Prox-SVRG (Xiao and Zhang, 2014) or SAGA (Defazio et al., 2014) to obtain the same overall complexity.

More recently, Lin et al. (2015b) developed an accelerated proximal coordinate gradient (APCG) method for solving a more general class of composite convex optimization problems. When applied to solve the dual problem (32), APCG enjoys the same batch complexity $\widetilde{\mathcal{O}}\big(1+\sqrt{\kappa/n}\big)$ for reducing the dual objective gap. In order to obtain the same complexity for the primal-dual gap, One needs an extra primal proximal-gradient step at the end after applying the APCG algorithm. The computational cost of this additional step is equivalent to one pass of the dataset, thus it does not affect the overall complexity.

5.2 Other Related Work

Another way to approach problem (1) is to reformulate it as a constrained optimization problem

$$\begin{aligned} \text{minimize} \quad & \frac{1}{n}\sum_{i=1}^{n}\phi_i(z_i) + g(x) \\ \text{subject to} \quad & a_i^T x = z_i, \quad i = 1,\dots,n, \end{aligned} \tag{33}$$

and solve it by ADMM type of operator-splitting methods (e.g., Lions and Mercier, 1979; Boyd et al., 2010). In fact, as shown in Chambolle and Pock (2011), the batch primal-dual algorithm (28)-(30) is equivalent to a pre-conditioned ADMM or an inexact Uzawa method (see, e.g., Zhang et al., 2011). Several authors (Wang and Banerjee, 2012; Ouyang et al., 2013; Suzuki, 2013; Zhong and Kwok, 2014) have considered a more general formulation than (33), where each ϕ_i is a function of the whole vector $z \in \mathbb{R}^n$. They proposed online or stochastic versions of ADMM which operate on only one ϕ_i in each iteration, and obtained sublinear convergence rates. However, their cost per iteration is $\mathcal{O}(nd)$ instead of $\mathcal{O}(d)$.

Suzuki (2014) considered a problem similar to (1), but with more complex regularization function g, meaning that g does not have a simple proximal mapping. Thus primal updates such as step (6) or (10) in SPDC and similar steps in SDCA cannot be computed efficiently. He proposed an algorithm that combines SDCA (Shalev-Shwartz and Zhang, 2013a) and ADMM, and showed that it has linear rate of convergence under similar conditions as Assumption A. It would be interesting to see if the SPDC method can be extended to their setting to obtain accelerated linear convergence rate.

6. Efficient Implementation with Sparse Data

During each iteration of the SPDC method, the update of primal variables (i.e., computing $x^{(t+1)}$) requires full d-dimensional vector operations; see the step (6) of Algorithm 1, the step (10) of Algorithm 2 and the step (23) of Algorithm 3. So the computational cost per iteration is $\mathcal{O}(d)$, and this can be too expensive if the dimension d is very high. In this section, we show how to exploit problem structure to avoid high-dimensional vector operations when the feature vectors a_i are sparse. We illustrate the efficient implementation for two popular cases: when g is an squared-ℓ_2 penalty and when g is an $\ell_1+\ell_2$ penalty. For both cases, we show that the computation cost per iteration only depends on the number of non-zero components of the feature vector.

6.1 Squared ℓ_2-Norm Penalty

Suppose that $g(x) = \frac{\lambda}{2}\|x\|_2^2$. For this case, the updates for each coordinate of x are independent of each other. More specifically, $x^{(t+1)}$ can be computed coordinate-wise in closed form:

$$x_j^{(t+1)} = \frac{1}{1+\lambda\tau}(x_j^{(t)} - \tau u_j^{(t)} - \tau\Delta u_j), \quad j = 1,\dots,n, \tag{34}$$

where Δu denotes $(y_k^{(t+1)} - y_k^{(t)})a_k$ in Algorithm 1, or $\frac{1}{m}\sum_{k\in K}(y_k^{(t+1)} - y_k^{(t)})a_k$ in Algorithm 2, or $(y_k^{(t+1)} - y_k^{(t)})a_k/(p_k n)$ in Algorithm 3, and Δu_j represents the j-th coordinate of Δu.

Although the dimension d can be very large, we assume that each feature vector a_k is sparse. We denote by $J^{(t)}$ the set of non-zero coordinates at iteration t, that is, if for some index $k \in K$ picked at iteration t we have $a_{kj} \neq 0$, then $j \in J^{(t)}$. If $j \notin J^{(t)}$, then the SPDC algorithm (and its variants) updates $y^{(t+1)}$ without using the value of $x_j^{(t)}$ or $\overline{x}_j^{(t)}$. This can be seen from the updates in (5), (9) and (21), where the value of the inner product $\langle a_k, \overline{x}^{(t)} \rangle$ does not depend on the value of $\overline{x}_j^{(t)}$. As a consequence, we can delay the updates on x_j and $\overline{x}_j$ whenever $j \notin J^{(t)}$ without affecting the updates on $y^{(t)}$, and process all the missing updates at the next time when $j \in J^{(t)}$.

Such a delayed update can be carried out very efficiently. We assume that t_0 is the last time when $j \in J^{(t)}$, and t_1 is the current iteration where we want to update x_j and $\overline{x}_j$. Since $j \notin J^{(t)}$ implies $\Delta u_j = 0$, we have

$$x_j^{t+1} = \frac{1}{1+\lambda\tau}(x_j^{(t)} - \tau u_j^{(t)}), \qquad t = t_0+1, t_0+2, \ldots, t_1-1. \tag{35}$$

Notice that $u_j^{(t)}$ is updated only at iterations where $j \in J^{(t)}$. The value of $u_j^{(t)}$ doesn't change during iterations $[t_0+1, t_1]$, so we have $u_j^{(t)} \equiv u_j^{(t_0+1)}$ for $t \in [t_0+1, t_1]$. Substituting this equation into the recursive formula (35), we obtain

$$x_j^{(t_1)} = \frac{1}{(1+\lambda\tau)^{t_1-t_0-1}} \left(x_j^{(t_0+1)} + \frac{u_j^{(t_0+1)}}{\lambda} \right) - \frac{u_j^{(t_0+1)}}{\lambda}. \tag{36}$$

The update (36) takes $\mathcal{O}(1)$ time to compute. Using the same formula, we can compute $x_j^{(t_1-1)}$ and subsequently compute $\overline{x}_j^{(t_1)} = x_j^{(t_1)} + \theta(x_j^{(t_1)} - x_j^{(t_1-1)})$. Thus, the computational complexity of a single iteration in SPDC is proportional to $|J^{(t)}|$, independent of the dimension d.

We note that similar tricks of delayed updates, or "just-in-time" updates, have been derived and used for the SAG algorithm (Schmidt et al., 2013). For SPDC, the delayed updates become more complex due to the full vector extrapolation required for Nesterov-type acceleration.

6.2 $(\ell_1 + \ell_2)$-Norm Penalty

Suppose that $g(x) = \lambda_1 \|x\|_1 + \frac{\lambda_2}{2}\|x\|_2^2$. Since both the ℓ_1-norm and the squared ℓ_2-norm are decomposable, the updates for each coordinate of $x^{(t+1)}$ are independent. More specifically,

$$x_j^{(t+1)} = \arg\min_{\alpha \in \mathbb{R}} \left\{ \lambda_1|\alpha| + \frac{\lambda_2\alpha^2}{2} + (u_j^{(t)} + \Delta u_j)\alpha + \frac{(\alpha - x_j^{(t)})^2}{2\tau} \right\}, \tag{37}$$

where Δu_j follows the definition in Section 6.1. If $j \notin J^{(t)}$, then $\Delta u_j = 0$ and equation (37) can be simplified as

$$x_j^{(t+1)} = \begin{cases} \frac{1}{1+\lambda_2\tau}(x_j^{(t)} - \tau u_j^{(t)} - \tau\lambda_1) & \text{if } x_j^{(t)} - \tau u_j^{(t)} > \tau\lambda_1, \\ \frac{1}{1+\lambda_2\tau}(x_j^{(t)} - \tau u_j^{(t)} + \tau\lambda_1) & \text{if } x_j^{(t)} - \tau u_j^{(t)} < -\tau\lambda_1, \\ 0 & \text{otherwise,} \end{cases} \qquad t \in [t_0+1, t_1]. \tag{38}$$

Similar to the approach of Section 6.1, we delay the update of x_j until $j \in J^{(t)}$. We assume t_0 to be the last iteration when $j \in J^{(t)}$, and let t_1 be the current iteration when we want to update x_j. During iterations $[t_0+1, t_1]$, the value of $u_j^{(t)}$ doesn't change, so we have $u_j^{(t)} \equiv u_j^{(t_0+1)}$ for $t \in [t_0+1, t_1]$. Using equation (38) and the invariance of $u_j^{(t)}$ for $t \in [t_0+1, t_1]$, we have an $\mathcal{O}(1)$ time algorithm to calculate $x_j^{(t_1)}$. More specifically, given $x_j^{(t_0+1)}$ at iteration t_0, we present an efficient algorithm for calculating $x_j^{(t_1)}$. We begin by examining the sign of $x_j^{(t_0+1)}$.

Case I ($x_j^{(t_0+1)} = 0$): If $-u_j^{(t_0+1)} > \lambda_1$, then equation (38) implies $x_j^{(t)} > 0$ for all $t > t_0+1$. Consequently, we have a closed-form formula for $x_j^{(t_1)}$:

$$x_j^{(t_1)} = \frac{1}{(1+\lambda_2\tau)^{t_1-t_0-1}} \left(x_j^{(t_0+1)} + \frac{u_j^{(t_0+1)} + \lambda_1}{\lambda_2} \right) - \frac{u_j^{(t_0+1)} + \lambda_1}{\lambda_2}. \tag{39}$$

If $-u_j^{(t_0+1)} < -\lambda_1$, then equation (38) implies $x_j^{(t)} < 0$ for all $t > t_0+1$. Therefore, we have the closed-form formula:

$$x_j^{(t_1)} = \frac{1}{(1+\lambda_2\tau)^{t_1-t_0-1}} \left(x_j^{(t_0+1)} + \frac{u_j^{(t_0+1)} - \lambda_1}{\lambda_2} \right) - \frac{u_j^{(t_0+1)} - \lambda_1}{\lambda_2}. \tag{40}$$

Finally, if $-u_j^{(t_0+1)} \in [-\lambda_1, \lambda_1]$, then equation (38) implies $x_j^{(t_1)} = 0$.

Case II ($x_j^{(t_0+1)} > 0$): If $-u_j^{(t_0+1)} \geq \lambda_1$, then it is easy to verify that $x_j^{(t_1)}$ is obtained by equation (39). Otherwise, We use the recursive formula (38) to derive the latest time $t^+ \in [t_0+1, t_1]$ such that $x_j^{t^+} > 0$ is true. Indeed, since $x_j^{(t)} > 0$ for all $t \in [t_0+1, t^+]$, we have a closed-form formula for $x_j^{t^+}$:

$$x_j^{t^+} = \frac{1}{(1+\lambda_2\tau)^{t^+-t_0-1}} \left(x_j^{(t_0+1)} + \frac{u_j^{(t_0+1)} + \lambda_1}{\lambda_2} \right) - \frac{u_j^{(t_0+1)} + \lambda_1}{\lambda_2}. \tag{41}$$

We look for the largest t^+ such that the right-hand side of equation (41) is positive, which is equivalent of

$$t^+ - t_0 - 1 < \log\left(1 + \frac{\lambda_2 x_j^{(t_0+1)}}{u_j^{(t_0+1)} + \lambda_1}\right) / \log(1+\lambda_2\tau). \tag{42}$$

Thus, t^+ is the largest integer in $[t_0+1, t_1]$ such that inequality (42) holds. If $t^+ = t_1$, then $x_j^{(t_1)}$ is obtained by (41). Otherwise, we can calculate $x_j^{t^++1}$ by formula (38), then resort to Case I or Case III, treating t^+ as t_0.

Case III ($x_j^{(t_0+1)} < 0$): If $-u_j^{(t_0+1)} \leq -\lambda_1$, then $x_j^{(t_1)}$ is obtained by equation (40). Otherwise, we calculate the largest integer $t^- \in [t_0+1, t_1]$ such that $x_j^{t^-} < 0$ is true. Using the same argument as for Case II, we have the closed-form expression

$$x_j^{t^-} = \frac{1}{(1+\lambda_2\tau)^{t^--t_0-1}} \left(x_j^{(t_0+1)} + \frac{u_j^{(t_0+1)} - \lambda_1}{\lambda_2} \right) - \frac{u_j^{(t_0+1)} - \lambda_1}{\lambda_2}. \tag{43}$$

where t^- is the largest integer in $[t_0+1, t_1]$ such that the following inequality holds:

$$t^- - t_0 - 1 < \log\left(1 + \frac{\lambda_2 x_j^{(t_0+1)}}{u_j^{(t_0+1)} - \lambda_1}\right) / \log(1+\lambda_2\tau). \tag{44}$$

If $t^- = t_1$, then $x_j^{(t_1)}$ is obtained by (43). Otherwise, we can calculate $x_j^{t^-+1}$ by formula (38), then resort to Case I or Case II, treating t^- as t_0.

Finally, we note that formula (38) implies the monotonicity of $x_j^{(t)}$ $(t = t_0+1, t_0+2, \dots)$. As a consequence, the procedure of either Case I, Case II or Case III is executed for at most once. Hence, the algorithm for calculating $x_j^{(t_1)}$ has $\mathcal{O}(1)$ time complexity.

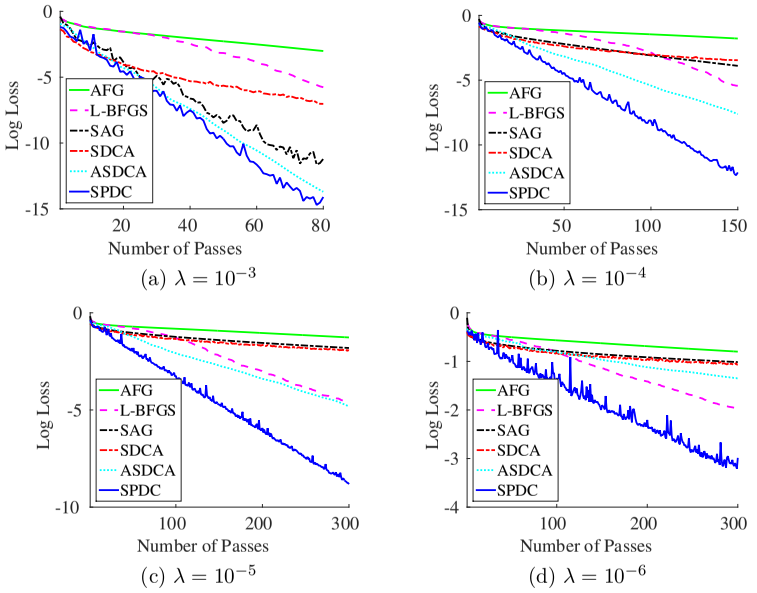
The vector $\overline{x}_j^{(t_1)}$ can be updated by the same algorithm since it is a linear combination of $x_j^{(t_1)}$ and $x_j^{(t_1-1)}$. As a consequence, the computational complexity of each iteration in SPDC is proportional to $|J^{(t)}|$, independent of the dimension d.

7. Experiments

In this section, we compare the basic SPDC method (Algorithm 1) with several state-of-the-art algorithms for solving problem (1). They include two batch-update algorithms: the accelerated full gradient (AFG) method (Nesterov, 2004, Section 2.2), and the limited-memory quasi-Newton method L-BFGS (Nocedal and Wright, 2006, Section 7.2)). For the AFG method, we adopt an adaptive line search scheme (Nesterov, 2013) to improve its efficiency. For the L-BFGS method, we use the memory size 30 as suggested by Nocedal and Wright (2006, Section 7.2).

We also compare SPDC with three stochastic algorithms: the stochastic average gradient (SAG) method (Le Roux et al., 2012; Schmidt et al., 2013), the stochastic dual coordinate descent (SDCA) method (Shalev-Shwartz and Zhang, 2013a) and the accelerated stochastic dual coordinate descent (ASDCA) method(Shalev-Shwartz and Zhang, 2015). We conduct experiments on a synthetic dataset and three real datasets. The hyper-parameters τ, σ, θ of the SPDC algorithm are chosen by their theoretical values given in (12). For SDCA, we use their default parameter settings given in Shalev-Shwartz and Zhang (2013a), which can be determined from the Lipschitz constant $1/\gamma$ of the loss functions and the strongly convex regularization parameter λ. For SAG, we choose the learning rate $\alpha = \gamma/R$ as recommended by Schmidt et al. (2013).

We caution the readers that our numerical experiments in this section focus on comparing the *optimization* performance of various algorithms, i.e., how fast they can reduce the objective function to its minimum value. In order to illustrate their differences for very ill-conditioned problems, we often run the algorithms with hundreds of passes over the datasets in order to reach a small optimization error. Such large numbers of passes over datasets and the small optimization errors may not be appropriate for machine learning problems, especially from the point of view of *generalization* (reducing testing error). See Bottou and Bousquet (2008) for discussions on the fundamental tradeoffs of large scale learning problems.



(a) $\lambda = 10^{-3}$ (b) $\lambda = 10^{-4}$ (c) $\lambda = 10^{-5}$ (d) $\lambda = 10^{-6}$

Figure 1: Comparing SPDC with other methods for ridge regression on synthetic data, with the regularization coefficient $\lambda \in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. The horizontal axis is the number of passes through the entire dataset, and the vertical axis is the logarithmic gap $\log(P(x^{(T)}) - P(x^\star))$.

7.1 Ridge Regression with Synthetic Data

We first compare SPDC with other algorithms on a simple quadratic problem using synthetic data. We generate $n = 500$ i.i.d. training examples $\{a_i, b_i\}_{i=1}^n$ according to the model

$$b = \langle a, x^* \rangle + \varepsilon, \quad a \sim \mathcal{N}(0, \Sigma), \quad \varepsilon \sim \mathcal{N}(0,1),$$

where $a \in \mathbb{R}^d$ and $d = 500$, and x^* is the all-ones vector. To make the problem ill-conditioned, the covariance matrix Σ is set to be diagonal with $\Sigma_{jj} = j^{-2}$, for $j = 1, \dots, d$. Given the set of examples $\{a_i, b_i\}_{i=1}^n$, we then solved a standard ridge regression problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \left\{ P(x) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^n \frac{1}{2}(a_i^T x - b_i)^2 + \frac{\lambda}{2}\|x\|_2^2 \right\}.$$

In the form of problem (1), we have $\phi_i(z) = z^2/2$ and $g(x) = (1/2)\|x\|_2^2$. As a consequence, the derivative of ϕ_i is 1-Lipschitz continuous and g is λ-strongly convex.

Dataset name	# of samples n	# of features d	sparsity	size
Covtype	581,012	54	22%	71 MB
RCV1	20,242	47,236	0.16%	37 MB
News20	19,996	1,355,191	0.04%	140 MB
RCV1-test	677,399	47,236	0.15%	1.2 GB
URL	2,396,130	3,231,961	0.004%	2.2 GB

Table 3: Characteristics of real datasets from LIBSVM data (Fan and Lin, 2011).

We evaluate the algorithms by the logarithmic optimality gap $\log(P(x^{(t)}) - P(x^\star))$, where $x^{(t)}$ is the output of the algorithms after t passes over the entire dataset, and $x^\star$ is the global minimum. When the regularization coefficient is relatively large, e.g., $\lambda = 10^{-1}$ or 10^{-2}, the problem is well-conditioned and we observe fast convergence of the stochastic algorithms SAG, SDCA, ASDCA and SPDC, which are substantially faster than the two batch methods AFG and L-BFGS.

Figure 1 shows the convergence of the five different algorithms when we varied λ from 10^{-3} to 10^{-6}. As the plot shows, when the condition number is greater than n, the SPDC algorithm also converges substantially faster than the other two stochastic methods SAG and SDCA. It is also notably faster than L-BFGS. These results support our theory that SPDC enjoys a faster convergence rate on ill-conditioned problems. In terms of their batch complexities, SPDC is up to $\sqrt{n}$ times faster than AFG, and $(\lambda n)^{-1/2}$ times faster than SAG and SDCA.

Theoretically, ASDCA enjoys the same batch complexity as SPDC up to a multiplicative constant factor. Figure 1 shows that the empirical performance of SPDC is substantially faster that of ASDCA for small λ. This may due to the fact that ASDCA follows an inner-outer iteration procedure, which requires careful selection of the regularization parameter and accuracy to reach for each call of SDCA. SPDC is a single-loop algorithm that needs less parameters to set up, thus it can be empirically more efficient.

7.2 Binary Classification with Real Data

Finally we show the results of solving the binary classification problem on real datasets. The datasets are obtained from the LIBSVM data collection (Fan and Lin, 2011) and summarized in Table 3. The first three datasets are selected to reflect different relations between the sample size n and the feature dimensionality d, which cover $n \gg d$ (Covtype), $n \approx d$ (RCV1) and $n \ll d$ (News20). The remaining two are relatively larger datasets (RCV1-test and URL) that we did not test in previous experiments conducted in Zhang and Xiao (2015). For all tasks, the data points take the form of (a_i, b_i), where $a_i \in \mathbb{R}^d$ is the feature vector, and $b_i \in \{-1, 1\}$ is the binary class label. As a preprocessing step, the feature vectors are normalized to the unit ℓ_2-norm (meaning $R = 1$).

Our goal is to minimize the regularized empirical risk:

$$P(x) = \frac{1}{n}\sum_{i=1}^{n} \phi_i(a_i^T x) + \frac{\lambda}{2}\|x\|_2^2, \quad \text{where} \quad \phi_i(z) = \begin{cases} 0 & \text{if } b_i z \geq 1 \\ \frac{1}{2} - b_i z & \text{if } b_i z \leq 0 \\ \frac{1}{2}(1 - b_i z)^2 & \text{otherwise.} \end{cases}$$

Here, ϕ_i is the smoothed hinge loss (see, e.g., Shalev-Shwartz and Zhang, 2013a). It is easy to verify that the conjugate function of ϕ_i is $\phi_i^*(\beta) = b_i\beta + \frac{1}{2}\beta^2$ for $b_i\beta \in [-1, 0]$ and ∞ otherwise.

The performance of the five algorithms on the three smaller datasets are plotted in Figure 2 and Figure 3. In Figure 2, we compare SPDC with the two batch methods: AFG and L-BFGS. The results show that SPDC is substantially faster than AFG and L-BFGS for relatively large λ, illustrating the advantage of stochastic methods over batch methods on well-conditioned problems. As λ decreases to 10^{-8}, the batch methods (especially L-BFGS) become comparable to SPDC.

In Figure 3, we compare SPDC with the three stochastic methods: SAG, SDCA and ASDCA. Note that the specification of ASDCA (Shalev-Shwartz and Zhang, 2015) requires the regularization coefficient λ satisfies $\lambda \leq \frac{R^2}{10n}$ where R is the maximum ℓ_2-norm of feature vectors. To satisfy this constraint, we run ASDCA with $\lambda \in \{10^{-6}, 10^{-7}, 10^{-8}\}$. In Figure 3, the observations are just the opposite to that of Figure 2. All stochastic algorithms have comparable performances on relatively large λ, but SPDC and ASDCA becomes substantially faster when λ gets closer to zero. In particular, ASDCA converges faster than SPDC on the Covtype dataset, but SPDC is faster on the remaining two datasets. In addition, due to the outer-inner loop structure of the ASDCA algorithm, its error rate oscillates and might be bad at early iterations. In contrast, the curve of SPDC is almost linear and it is more stable than ASDCA.

Figure 4 plots the convergence results on the last two datasets, where both the sample size n and the dimension d are big. If the regularization parameter λ is also relatively large, then the stochastic algorithms (SPDC, SAG and SDCA) will guarantee to converge very quickly, making it an easy optimization problem. We report experiments on small regularization values $\lambda = 10^{-6}$ and $\lambda = 10^{-8}$. The ASDCA algorithm is reported on $\lambda = 10^{-8}$ because it satisfies the constraint $\lambda \leq \frac{R^2}{10n}$. Comparing results on the RCV1 and RCV1-test datasets, the SPDC algorithm has a more significant advantage over the batch methods (AFG and L-BFGS) on the bigger dataset, because the stochastic algorithm converges faster with a larger sample size. On the other hand, the performance gaps between SPDC and the two other stochastic methods (SAG and SDCA) are less significant on the bigger dataset. We observe the same phenomenon on the URL dataset.

Summarizing Figure 2, Figure 3 and Figure 4, the performance of the SPDC algorithm are always comparable or better than the other methods, for various of relations between the sample size n and the dimension d, and on both small and large datasets.

7.3 Uniform Sampling versus Non-Uniform Sampling

In this subsection, we compare the uniform sampling strategy (Algorithm 1) and the non-uniform sampling strategy (Algorithm 3) for SPDC. We repeat the binary classification experiments on the Covtype, RCV1 and News20 datasets, but this time without performing feature normalization. More precisely, for each data point taking the form of (a_i, b_i), we don't normalize the feature vector a_i to the unit ℓ_2-norm. But instead, we multiply a constant number to every feature vector so that the average ℓ_2-norm $\overline{R}$ is equal to one. It ensures that the overall loss function is 1-smooth.

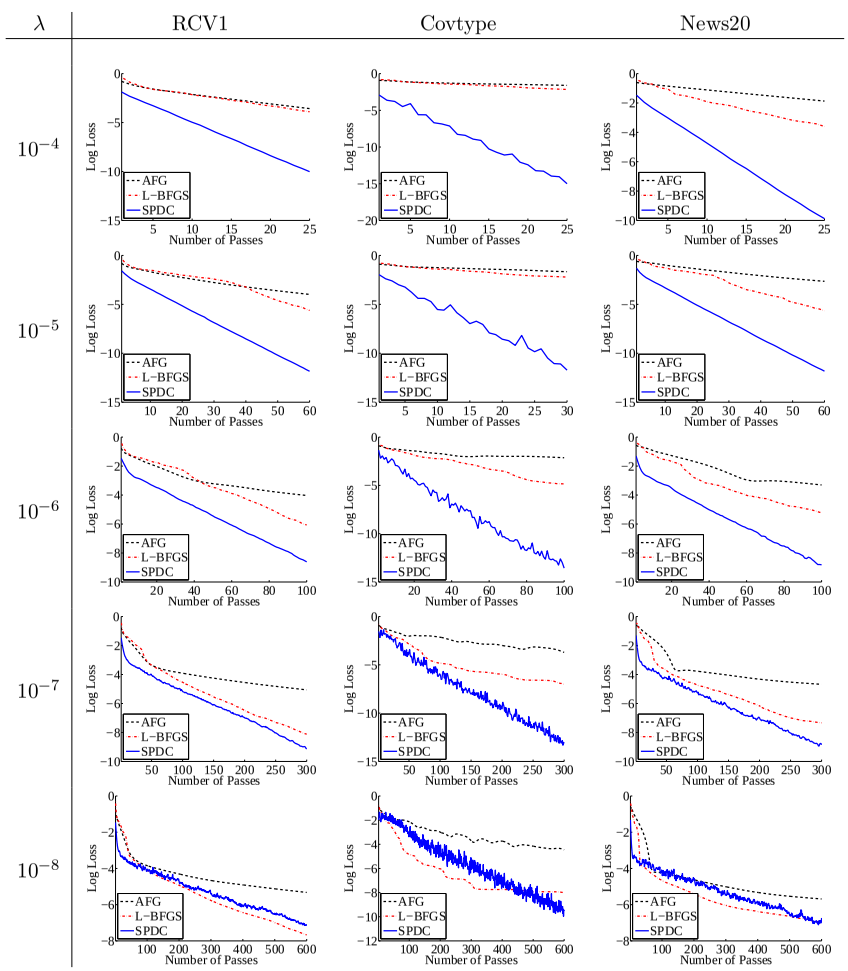


Figure 2: Comparing SPDC with AFG and L-BFGS on three real datasets with smoothed hinge loss. The horizontal axis is the number of passes through the entire dataset, and the vertical axis is the logarithmic optimality gap $\log(P(x^{(t)}) - P(x^\star))$. The SPDC algorithm is faster than the two batch methods when λ is relatively large.

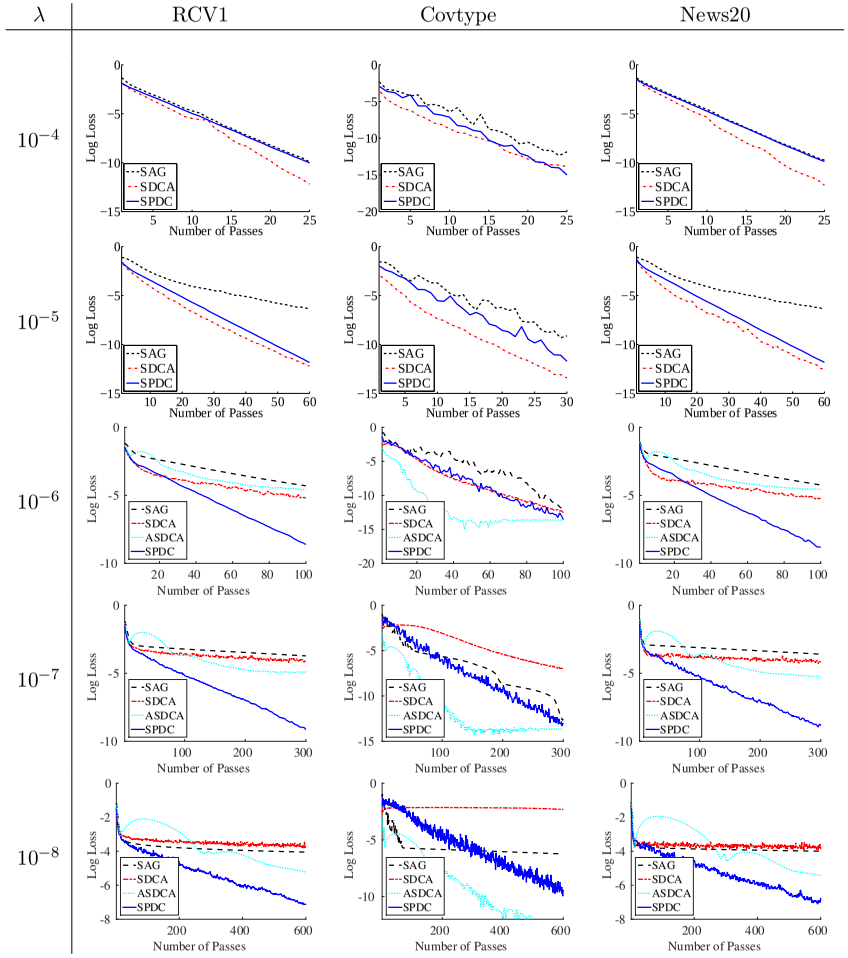
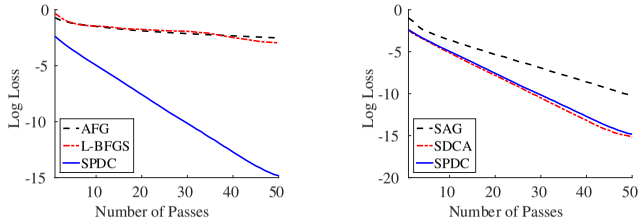
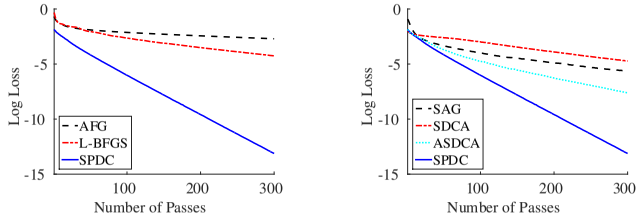


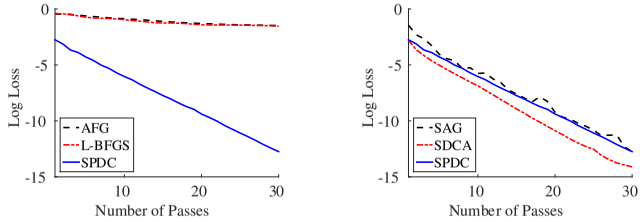
Figure 3: Comparing SPDC with SAG, SDCA and ASDCA on three real datasets with smoothed hinge loss. The horizontal axis is the number of passes through the entire dataset, and the vertical axis is the logarithmic optimality gap $\log(P(x^{(T)}) - P(x^\star))$. The SPDC algorithm is faster than SAG and SDCA when λ is small. It is faster than ASDCA on datasets RCV1 and News20.



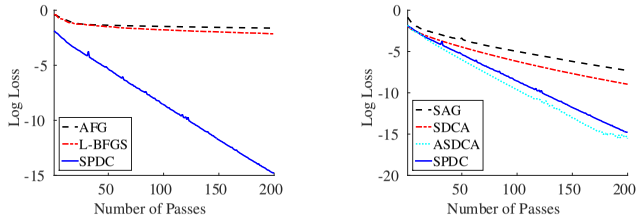
(a) RCV1-test ($\lambda = 10^{-6}$)



(b) RCV1-test ($\lambda = 10^{-8}$)

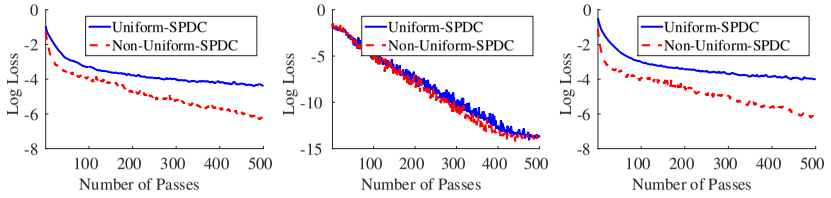


(c) URL ($\lambda = 10^{-6}$)



(d) URL ($\lambda = 10^{-8}$)

Figure 4: Comparing SPDC with other methods on the two larger datasets. The vertical axis is the logarithmic optimality gap $\log(P(x^{(t)}) - P(x^\star))$.



(a) RCV1 ($R/\overline{R} = 3.87$) (b) Covtype ($R/\overline{R} = 1.25$) (c) News20 ($R/\overline{R} = 6.71$)

Figure 5: Comparing the uniform sampling and non-uniform sampling strategies for SPDC, where the vertical axis is the logarithmic optimality gap $\log(P(x^{(t)}) - P(x^\star))$. The quantity $R/\overline{R}$ represents the ratio between the largest feature norm and the average feature norm. When this quantity is large, the non-uniform sampling algorithm converges significantly faster.

For this experiment, we use a small regularization parameter $\lambda = 10^{-8}$, so that the problem has a large condition number. The hyper-parameters τ, σ, θ are chosen by their theoretical values in (12) and (24), respectively for the two sampling strategies. For non-uniform sampling, the hyper-parameter α is chosen by the optimal value given in (25). Figure 5 plots their convergence profiles. On all of the three datasets, the non-uniform sampling algorithm converges faster. The margin is quite significant when the ratio between the largest feature norm R and the average feature norm $\overline{R}$ is relatively large. This is consistent with our analysis in Theorem 1 and Theorem 6, which state that the convergence rate of the uniform sampling algorithm depends R, while that of the non-uniform sampling algorithm depends on $\overline{R}$.

Acknowledgments

We are grateful to Qihang Lin for helpful discussions, especially on the proof of Lemma 3.

Appendix A. Proof of Theorem 1

We focus on characterizing the values of x and y after the t-th update in Algorithm 2. For any $i \in \{1, \ldots, n\}$, let $\widetilde{y}_i$ be the value of $y_i^{(t+1)}$ if $i \in K$, i.e.,

$$\widetilde{y}_i = \arg\max_{\beta \in \mathbb{R}} \left\{ \beta \langle a_i, \overline{x}^{(t)} \rangle - \phi_i^*(\beta) - \frac{(\beta - y_i^{(t)})^2}{2\sigma} \right\}.$$

Since by assumption ϕ_i is $(1/\gamma)$-smooth, its convex conjugate ϕ_i^* is γ-strongly convex (see, e.g., Hiriart-Urruty and Lemaréchal, 2001, Theorem 4.2.2). Thus the function being maxi-

mized above is $(1/\sigma+\gamma)$-strongly concave. Therefore,

$$-y_i^\star\langle a_i,\overline{x}^{(t)}\rangle+\phi_i^*(y_i^\star)+\frac{(y_i^\star-y_i^{(t)})^2}{2\sigma}\geq-\widetilde{y}_i\langle a_i,\overline{x}^{(t)}\rangle+\phi_i^*(\widetilde{y}_i)+\frac{(\widetilde{y}_i-y_i^{(t)})^2}{2\sigma}+\Big(\frac{1}{\sigma}+\gamma\Big)\frac{(\widetilde{y}_i-y_i^\star)^2}{2}.$$

Multiplying both sides of the above inequality by m/n and re-arrange terms, we have

$$\frac{m}{2\sigma n}(y_i^{(t)}-y_i^\star)^2\geq\Big(\frac{1}{\sigma}+\gamma\Big)\frac{m}{2n}(\widetilde{y}_i-y_i^\star)^2+\frac{m}{2\sigma n}(\widetilde{y}_i-y_i^{(t)})^2-\frac{m}{n}(\widetilde{y}_i-y_i^\star)\langle a_i,\overline{x}^{(t)}\rangle+\frac{m}{n}\big(\phi_i^*(\widetilde{y}_i)-\phi_i^*(y_i^\star)\big).\tag{45}$$

According to Algorithm 2, the set K of indices to be updated are chosen randomly. For every specific index i, the event $i\in K$ happens with probability m/n. If $i\in K$, then $y_i^{(t+1)}$ is updated to the value $\widetilde{y}_i$, which satisfies inequality (45). Otherwise, $y_i^{(t+1)}$ is assigned by its old value $y_i^{(t)}$. Let $\mathcal{F}_t$ be the sigma field generated by all random variables defined before round t, and taking expectation conditioned on $\mathcal{F}_t$, we have

$$\begin{aligned}\mathbb{E}[(y_i^{(t+1)}-y_i^\star)^2|\mathcal{F}_t]&=\frac{m(\widetilde{y}_i-y_i^\star)^2}{n}+\frac{(n-m)(y_i^{(t)}-y_i^\star)^2}{n},\\ \mathbb{E}[(y_i^{(t+1)}-y_i^{(t)})^2|\mathcal{F}_t]&=\frac{m(\widetilde{y}_i-y_i^{(t)})^2}{n},\\ \mathbb{E}[y_i^{(t+1)}|\mathcal{F}_t]&=\frac{m\widetilde{y}_i}{n}+\frac{(n-m)y_i^{(t)}}{n}\\ \mathbb{E}[\phi_i^*(y_i^{(t+1)})|\mathcal{F}_t]&=\frac{m}{n}\phi_i^*(\widetilde{y}_i)+\frac{n-m}{n}\phi_i^*(y_i^{(t)}).\end{aligned}$$

As a result, we can represent $(\widetilde{y}_i-y_i^\star)^2$, $(\widetilde{y}_i-y_i^{(t)})^2$, $\widetilde{y}_i$ and $\phi_i^*(\widetilde{y}_i)$ in terms of the conditional expectations on $(y_i^{(t+1)}-y_i^\star)^2$, $(y_i^{(t+1)}-y_i^{(t)})^2$, $y_i^{(t+1)}$ and $\phi_i^*(y_i^{(t+1)})$, respectively. Plugging these representations into inequality (45) and re-arranging terms, we obtain

$$\begin{aligned}\Big(\frac{1}{2\sigma}+\frac{(n-m)\gamma}{2n}\Big)(y_i^{(t)}-y_i^\star)^2\geq&\Big(\frac{1}{2\sigma}+\frac{\gamma}{2}\Big)\mathbb{E}[(y_i^{(t+1)}-y_i^\star)^2|\mathcal{F}_t]+\frac{1}{2\sigma}\mathbb{E}[(y_i^{(t+1)}-y_i^{(t)})^2|\mathcal{F}_t]\\&-\Big(\frac{m}{n}(y_i^{(t)}-y_i^\star)+\mathbb{E}[y_i^{(t+1)}-y_i^{(t)}|\mathcal{F}_t]\Big)\langle a_i,\overline{x}^{(t)}\rangle\\&+\mathbb{E}[\phi_i^*(y_i^{(t+1)})|\mathcal{F}_t]-\phi_i^*(y_i^{(t)})+\frac{m}{n}\big(\phi_i^*(y_i^{(t)})-\phi_i^*(y_i^\star)\big).\end{aligned}\tag{46}$$

Then summing over all indices $i=1,2,\ldots,n$ and dividing both sides of the resulting inequality by m, we have

$$\begin{aligned}\Big(\frac{1}{2\sigma}+\frac{(n-m)\gamma}{2n}\Big)\frac{\|y^{(t)}-y^\star\|_2^2}{m}\geq&\Big(\frac{1}{2\sigma}+\frac{\gamma}{2}\Big)\frac{\mathbb{E}[\|y^{(t+1)}-y^\star\|_2^2|\mathcal{F}_t]}{m}+\frac{1}{2\sigma}\frac{\mathbb{E}[\|y^{(t+1)}-y^{(t)}\|_2^2|\mathcal{F}_t]}{m}\\&+\mathbb{E}\Big[\frac{1}{m}\sum_{k\in K}\big(\phi_k^*(y_k^{(t+1)})-\phi_k^*(y_k^{(t)})\big)\Big|\mathcal{F}_t\Big]+\frac{1}{n}\sum_{i=1}^n\big(\phi_i^*(y_i^{(t)})-\phi_i^*(y_i^\star)\big)\\&-\mathbb{E}\Big[\Big\langle u^{(t)}-u^\star+\frac{n}{m}(u^{(t+1)}-u^{(t)}),\,\overline{x}^{(t)}\Big\rangle\Big|\mathcal{F}_t\Big],\end{aligned}\tag{47}$$

where we used the shorthand notations (appeared in Algorithm 2)

$$u^{(t)}=\frac{1}{n}\sum_{i=1}^n y_i^{(t)}a_i,\qquad u^{(t+1)}=\frac{1}{n}\sum_{i=1}^n y_i^{(t+1)}a_i,\qquad\text{and}\qquad u^\star=\frac{1}{n}\sum_{i=1}^n y_i^\star a_i.\tag{48}$$

Since only the dual coordinates with indices in K are updated, we have

$$\frac{n}{m}(u^{(t+1)}-u^{(t)})=\frac{1}{m}\sum_{i=1}^n(y_i^{(t+1)}-y_i^{(t)})a_i=\frac{1}{m}\sum_{k\in K}(y_k^{(t+1)}-y_k^{(t)})a_k.$$

We also derive an inequality characterizing the relation between $x^{(t+1)}$ and $x^{(t)}$. Since the function being minimized on the right-hand side of (10) has strong convexity parameter $1/\tau+\lambda$ and $x^{(t+1)}$ is the minimizer, we have

$$\begin{aligned}&g(x^\star)+\Big\langle u^{(t)}+\frac{n}{m}(u^{(t+1)}-u^{(t)}),\;x^\star\Big\rangle+\frac{\|x^{(t)}-x^\star\|_2^2}{2\tau}\\\geq\;&g(x^{(t+1)})+\Big\langle u^{(t)}+\frac{n}{m}(u^{(t+1)}-u^{(t)}),\;x^{(t+1)}\Big\rangle+\Big(\frac{1}{2\tau}+\frac{\lambda}{2}\Big)\|x^{(t+1)}-x^\star\|_2^2\\&+\frac{\|x^{(t+1)}-x^{(t)}\|_2^2}{2\tau}.\end{aligned}\tag{49}$$

Rearranging terms and taking expectation conditioned on $\mathcal{F}_t$, we have

$$\begin{aligned}\frac{\|x^{(t)}-x^\star\|_2^2}{2\tau}\geq&\Big(\frac{1}{2\tau}+\frac{\lambda}{2}\Big)\mathbb{E}[\|x^{(t+1)}-x^\star\|_2^2|\mathcal{F}_t]+\frac{\mathbb{E}[\|x^{(t+1)}-x^{(t)}\|_2^2|\mathcal{F}_t]}{2\tau}\\&+\mathbb{E}\Big[g(x^{(t+1)})-g(x^\star)|\mathcal{F}_t\Big]\\&+\mathbb{E}\Big[\Big\langle u^{(t)}+\frac{n}{m}(u^{(t+1)}-u^{(t)}),\;x^{(t+1)}-x^\star\Big\rangle\Big|\mathcal{F}_t\Big].\end{aligned}\tag{50}$$

In addition, we consider a particular combination of the saddle-point function values at different points. By the definition of $f(x,y)$ in (4) and the notations in (48), we have

$$\begin{aligned}&f(x^{(t+1)},y^\star)-f(x^\star,y^\star)+\frac{n}{m}\Big(f(x^\star,y^\star)-f(x^\star,y^{(t+1)})\Big)-\frac{n-m}{m}\Big(f(x^\star,y^\star)-f(x^\star,y^{(t)})\Big)\\=\;&f(x^{(t+1)},y^\star)-f(x^\star,y^{(t)})+\frac{n}{m}\Big(f(x^\star,y^{(t)})-f(x^\star,y^{(t+1)})\Big)\\=\;&\langle u^\star,x^{(t+1)}\rangle-\frac{1}{n}\sum_{i=1}^n\phi_i^*(y_i^\star)+g(x^{(t+1)})-\langle u^{(t)},x^\star\rangle+\frac{1}{n}\sum_{i=1}^n\phi_i^*(y_i^{(t)})-g(x^\star)\\&+\frac{n}{m}\Big(\langle u^{(t)},x^\star\rangle-\frac{1}{n}\sum_{i=1}^n\phi_i^*(y_i^{(t)})+g(x^\star)-\langle u^{(t+1)},x^\star\rangle+\frac{1}{n}\sum_{i=1}^n\phi_i^*(y_i^{(t+1)})-g(x^\star)\Big)\\=\;&\frac{1}{n}\sum_{i=1}^n\Big(\phi_i^*(y_i^{(t)})-\phi_i^*(y_i^\star)\Big)+\frac{1}{m}\sum_{k\in K}\Big(\phi_i^*(y_k^{(t+1)})-\phi_i^*(y_k^{(t)})\Big)+g(x^{(t+1)})-g(x^\star)\\&+\langle u^\star,x^{(t+1)}\rangle-\langle u^{(t)},x^\star\rangle+\frac{n}{m}\langle u^{(t)}-u^{(t+1)},x^\star\rangle.\end{aligned}\tag{51}$$

Next we add both sides of the inequalities (47) and (50) together, and then subtract equality (51) after taking expectation with respect to $\mathcal{F}_t$. This leads to the following inequality:

$$\begin{aligned}
&\frac{\|x^{(t)}-x^\star\|_2^2}{2\tau}+\left(\frac{1}{2\sigma}+\frac{(n-m)\gamma}{2n}\right)\frac{\|y^{(t)}-y^\star\|_2^2}{m}+\frac{n-m}{m}\left(f(x^\star,y^\star)-f(x^\star,y^{(t)})\right)\\
&\geq\left(\frac{1}{2\tau}+\frac{\lambda}{2}\right)\mathbb{E}[\|x^{(t+1)}-x^\star\|_2^2|\mathcal{F}_t]+\left(\frac{1}{2\sigma}+\frac{\gamma}{2}\right)\frac{\mathbb{E}[\|y^{(t+1)}-y^\star\|_2^2|\mathcal{F}_t]}{m}+\frac{\mathbb{E}[\|x^{(t+1)}-x^{(t)}\|_2^2|\mathcal{F}_t]}{2\tau}\\
&\quad+\frac{\mathbb{E}[\|y^{(t+1)}-y^{(t)}\|_2^2|\mathcal{F}_t]}{2\sigma m}+\mathbb{E}\left[f(x^{(t+1)},y^\star)-f(x^\star,y^\star)+\frac{n}{m}\left(f(x^\star,y^\star)-f(x^\star,y^{(t+1)})\right)\Big|\mathcal{F}_t\right]\\
&\quad+\mathbb{E}\left[\left\langle u^{(t)}-u^\star+\frac{n}{m}(u^{(t+1)}-u^{(t)}),\ x^{(t+1)}-\overline{x}^{(t)}\right\rangle\Big|\mathcal{F}_t\right]. \qquad (52)
\end{aligned}$$

We need to lower bound the last term on the right-hand-side of the above inequality. To this end, we have

$$\begin{aligned}
&\left\langle u^{(t)}-u^\star+\frac{n}{m}(u^{(t+1)}-u^{(t)}),\ x^{(t+1)}-\overline{x}^{(t)}\right\rangle\\
=&\left(\frac{y^{(t)}-y^\star}{n}+\frac{y^{(t+1)}-y^{(t)}}{m}\right)^T A(x^{(t+1)}-x^{(t)}-\theta(x^{(t)}-x^{(t-1)}))\\
=&\frac{(y^{(t+1)}-y^\star)^TA(x^{(t+1)}-x^{(t)})}{n}-\frac{\theta(y^{(t)}-y^\star)^TA(x^{(t)}-x^{(t-1)})}{n}\\
&+\frac{n-m}{mn}(y^{(t+1)}-y^{(t)})^TA(x^{(t+1)}-x^{(t)})-\frac{\theta}{m}(y^{(t+1)}-y^{(t)})^TA(x^{(t)}-x^{(t-1)}). \qquad (53)
\end{aligned}$$

Recall that $\|a_k\|_2\leq R$ and, according to (12), $1/\tau=4\sigma R^2$. Therefore,

$$\begin{aligned}
|(y^{(t+1)}-y^{(t)})^TA(x^{(t+1)}-x^{(t)})|&\leq\frac{\|x^{(t+1)}-x^{(t)}\|_2^2}{4\tau/m}+\frac{\|(y^{(t+1)}-y^{(t)})^TA\|_2^2}{m/\tau}\\
&\leq\frac{\|x^{(t+1)}-x^{(t)}\|_2^2}{4\tau/m}+\frac{(\sum_{k\in K}|y_k^{(t+1)}-y_k^{(t)}|\cdot\|a_k\|_2)^2}{4m\sigma R^2}\\
&\leq\frac{m\|x^{(t+1)}-x^{(t)}\|_2^2}{4\tau}+\frac{\|y^{(t+1)}-y^{(t)}\|_2^2}{4\sigma},
\end{aligned}$$

Similarly, we have

$$|(y^{(t+1)}-y^{(t)})^TA(x^{(t)}-x^{(t-1)})|\leq\frac{m\|x^{(t)}-x^{(t-1)}\|_2^2}{4\tau}+\frac{\|y^{(t+1)}-y^{(t)}\|_2^2}{4\sigma}.$$

The above upper bounds on the absolute values imply

$$\begin{aligned}
(y^{(t+1)}-y^{(t)})^TA(x^{(t+1)}-x^{(t)})&\geq-\frac{m\|x^{(t+1)}-x^{(t)}\|_2^2}{4\tau}-\frac{\|y^{(t+1)}-y^{(t)}\|_2^2}{4\sigma},\\
(y^{(t+1)}-y^{(t)})^TA(x^{(t)}-x^{(t-1)})&\geq-\frac{m\|x^{(t)}-x^{(t-1)}\|_2^2}{4\tau}-\frac{\|y^{(t+1)}-y^{(t)}\|_2^2}{4\sigma}.
\end{aligned}$$

Combining the above two inequalities with (52) and (53), we obtain

$$\begin{aligned}
&\frac{\|x^{(t)}-x^\star\|_2^2}{2\tau}+\left(\frac{1}{2\sigma}+\frac{(n-m)\gamma}{2n}\right)\frac{\|y^{(t)}-y^\star\|_2^2}{m}\\
&+\theta\big(f(x^{(t)},y^\star)-f(x^\star,y^\star)\big)+\frac{n-m}{m}\left(f(x^\star,y^\star)-f(x^\star,y^{(t)})\right)\\
&+\theta\frac{\|x^{(t)}-x^{(t-1)}\|_2^2}{4\tau}+\theta\frac{(y^{(t)}-y^\star)^TA(x^{(t)}-x^{(t-1)})}{n}\\
\geq&\left(\frac{1}{2\tau}+\frac{\lambda}{2}\right)\mathbb{E}[\|x^{(t+1)}-x^\star\|_2^2|\mathcal{F}_t]+\left(\frac{1}{2\sigma}+\frac{\gamma}{2}\right)\frac{\mathbb{E}[\|y^{(t+1)}-y^\star\|_2^2|\mathcal{F}_t]}{m}\\
&+\mathbb{E}\left[f(x^{(t+1)},y^\star)-f(x^\star,y^\star)+\frac{n}{m}\left(f(x^\star,y^\star)-f(x^\star,y^{(t+1)})\right)\Big|\mathcal{F}_t\right]\\
&+\frac{\mathbb{E}[\|x^{(t+1)}-x^{(t)}\|_2^2|\mathcal{F}_t]}{4\tau}+\frac{\mathbb{E}[(y^{(t+1)}-y^\star)^TA(x^{(t+1)}-x^{(t)})|\mathcal{F}_t]}{n}. \qquad (54)
\end{aligned}$$

Note that we have added the nonnegative term $\theta\big(f(x^{(t)},y^\star)-f(x^\star,y^\star)\big)$ to the left-hand side in (54) to ensure that each term on one side of the inequality has a corresponding term on the other side.

If the parameters τ, σ, and θ are chosen as in (12), that is,

$$\tau=\frac{1}{2R}\sqrt{\frac{m\gamma}{n\lambda}},\quad\sigma=\frac{1}{2R}\sqrt{\frac{n\lambda}{m\gamma}},\quad\text{and}\quad\theta=1-\frac{1}{(n/m)+2R\sqrt{(n/m)/(\lambda\gamma)}},$$

Then the ratios between the coefficients of the corresponding terms on both sides of the inequality (54) are either equal to θ or bounded by θ. More specifically,

$$\begin{aligned}
\frac{n-m}{m}\Big/\frac{n}{m}=1-\frac{m}{n}&\ \leq\ \theta,\\
\frac{1}{2\tau}\Big/\left(\frac{1}{2\tau}+\frac{\lambda}{2}\right)=1-\frac{1}{1+2R\sqrt{(n/m)/(\lambda\gamma)}}&\ \leq\ \theta,\\
\left(\frac{1}{2\sigma}+\frac{(n-m)\gamma}{2n}\right)\Big/\left(\frac{1}{2\sigma}+\frac{\gamma}{2}\right)=1-\frac{1}{n/m+2R\sqrt{(n/m)/(\lambda\gamma)}}&\ =\ \theta.
\end{aligned}$$

Therefore, if we define the following sequence,

$$\begin{aligned}
\widetilde{\Delta}^{(t)}=&\left(\frac{1}{2\tau}+\frac{\lambda}{2}\right)\|x^{(t)}-x^\star\|_2^2+\left(\frac{1}{2\sigma}+\frac{\gamma}{2}\right)\frac{\|y^{(t)}-y^\star\|_2^2}{m}\\
&+f(x^{(t)},y^\star)-f(x^\star,y^\star)+\frac{n}{m}\left(f(x^\star,y^\star)-f(x^\star,y^{(t)})\right)\\
&+\frac{\|x^{(t)}-x^{(t-1)}\|_2^2}{4\tau}+\frac{(y^{(t)}-y^\star)^TA(x^{(t)}-x^{(t-1)})}{n},
\end{aligned}$$

then inequality (54) implies $\mathbb{E}\big[\widetilde{\Delta}^{(t+1)}|\mathcal{F}_t\big]\leq\theta\,\widetilde{\Delta}^{(t)}$. Apply this relation recursively and taking expectation with respect to all random variables up to time t, we have

$$\mathbb{E}\big[\widetilde{\Delta}^{(t)}\big]\ \leq\ \theta^t\,\widetilde{\Delta}^{(0)}. \qquad (55)$$

Comparing the definition of $\Delta^{(t)}$ in (11), we have

$$\widetilde{\Delta}^{(t)} = \Delta^{(t)} + \frac{\|y^{(t)} - y^\star\|_2^2}{4\sigma m} + \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{(y^{(t)} - y^\star)^T A(x^{(t)} - x^{(t-1)})}{n}. \tag{56}$$

For $t = 0$, by letting $x^{(-1)} = x^{(0)}$, the last two terms in (56) for $\widetilde{\Delta}^{(0)}$ disappears. Moreover, we can show that the sum of the last three terms in (56) are nonnegative, and therefore we can replace $\widetilde{\Delta}^{(t)}$ with $\Delta^{(t)}$ on the left-hand side of (55). To see this, we bound the absolute value of the last term:

$$\begin{aligned}
\frac{\left|(y^{(t)} - y^\star)^T A(x^{(t)} - x^{(t-1)})\right|}{n} &\le \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{\|A\|_2^2 \, \|y^{(t)} - y^\star\|_2^2}{n^2/\tau} \\
&\le \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{nR^2\|y^{(t)} - y^\star\|_2^2}{n^2/\tau} \\
&= \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{\|y^{(t)} - y^\star\|_2^2}{4n\sigma} \\
&\le \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{\|y^{(t)} - y^\star\|_2^2}{4m\sigma},
\end{aligned}$$

where in the second inequality we used $\|A\|_2^2 \le \|A\|_F^2 \le nR^2$, in the equality we used $\tau\sigma = 1/(4R^2)$, and in the last inequality we used $m \le n$. The above upper bound on absolute value implies

$$\frac{(y^{(t)} - y^\star)^T A(x^{(t)} - x^{(t-1)})}{n} \ge -\frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} - \frac{\|y^{(t)} - y^\star\|_2^2}{4m\sigma}.$$

To summarize, we have proved

$$\mathbb{E}\left[\Delta^{(t)}\right] \le \theta^t \left(\Delta^{(0)} + \frac{\|y^{(0)} - y^\star\|_2^2}{4m\sigma}\right),$$

which is the desired result.

Appendix B. Proof of Lemma 3

We can write $P(x) = F(x) + g(x)$ where

$$F(x) = \frac{1}{n}\sum_{i=1}^n \phi_i(a_i^T x) = \max_{y\in\mathbb{R}^n}\left\{\frac{1}{n}y^T Ax - \frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i)\right\}.$$

Assumption A implies that $F(x)$ is smooth and $\nabla F(x)$ is Lipschitz continuous with constant $\|A\|_2^2/(n\gamma)$. We can bound the spectral norm with the Frobenius norm, i.e., $\|A\|_2^2 \le \|A\|_F^2 \le nR^2$, which results in $\|A\|_2^2/(n\gamma) \le nR^2/(n\gamma) = R^2/\gamma$. By definition of the saddle point,

the gradient of F at $x^\star$ is $\nabla F(x^\star) = (1/n)A^T y^\star$. Therefore, we have

$$\begin{aligned}
F(x) &\le F(x^\star) + \langle \nabla F(x^\star), x - x^\star\rangle + \frac{R^2}{2\gamma}\|x - x^\star\|_2^2 \\
&= \max_{y\in\mathbb{R}^n}\left\{\frac{1}{n}y^T Ax^\star - \frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i)\right\} + \frac{1}{n}(y^\star)^T A(x - x^\star) + \frac{R^2}{2\gamma}\|x - x^\star\|_2^2 \\
&= \left\{\frac{1}{n}(y^\star)^T Ax^\star - \frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i^\star)\right\} + \frac{1}{n}(y^\star)^T A(x - x^\star) + \frac{R^2}{2\gamma}\|x - x^\star\|_2^2 \\
&= \frac{1}{n}(y^\star)^T Ax - \frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i^\star) + \frac{R^2}{2\gamma}\|x - x^\star\|_2^2.
\end{aligned}$$

Combining the above inequality with $P(x) = F(x) + g(x)$, we have

$$P(x) \le \frac{1}{n}(y^\star)^T Ax - \frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i^\star) + \frac{R^2}{2\gamma}\|x - x^\star\|_2^2 + g(x) = f(x, y^\star) + \frac{R^2}{2\gamma}\|x - x^\star\|_2^2,$$

which is the first desired inequality.

Similarly, the second inequality can be shown by first writing

$$D(y) = -\frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i) - G^*(y),$$

where

$$G^*(y) = g^*\left(-\frac{1}{n}A^T y\right) = \max_{x\in\mathbb{R}^d}\left\{-\frac{1}{n}x^T A^T y - g(x)\right\}.$$

In this case, $\nabla G^*(y)$ is Lipschitz continuous with constant

$$\|A\|_2^2/(n^2\lambda) \le nR^2/(n^2\lambda) = R^2/(n\lambda).$$

Again by definition of the saddle-point, we have $\nabla G^*(y^\star) = -(1/n)Ax^\star$. Therefore,

$$\begin{aligned}
G^*(y) &\le G^*(y^\star) + \langle \nabla G^*(y^\star), y - y^\star\rangle + \frac{R^2}{2n\lambda}\|y - y^\star\|_2^2 \\
&= \max_{x\in\mathbb{R}^d}\left\{-\frac{1}{n}x^T A^T y^\star - g(x)\right\} - \frac{1}{n}(y - y^\star)^T Ax^\star + \frac{R^2}{2n\lambda}\|y - y^\star\|_2^2 \\
&= \left\{-\frac{1}{n}(x^\star)^T A^T y^\star - g(x^\star)\right\} - \frac{1}{n}(y - y^\star)^T Ax^\star + \frac{R^2}{2n\lambda}\|y - y^\star\|_2^2 \\
&= -\frac{1}{n}y^T Ax^\star - g(x^\star) + \frac{R^2}{2n\lambda}\|y - y^\star\|_2^2.
\end{aligned}$$

Recalling that $D(y) = -\frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i) - G^*(y)$, we conclude with

$$D(y) \ge -\frac{1}{n}\phi_i^*(y_i) + \frac{1}{n}y^T Ax^\star + g(x^\star) - \frac{R^2}{2n\lambda}\|y - y^\star\|_2^2 = f(x^\star, y) - \frac{R^2}{2n\lambda}\|y - y^\star\|_2^2.$$

This finishes the proof.

Appendix C. Proof of Theorem 6

The proof of Theorem 6 follows similar steps for proving Theorem 1. We start by establishing relation between $(y^{(t)}, y^{(t+1)})$ and between $(x^{(t)}, x^{(t+1)})$. Suppose that the quantity $\widetilde{y}_i$ minimizes the function $\phi_i^*(\beta) - \beta\langle a_i, \overline{x}^{(t)}\rangle + \frac{p_i n}{2\sigma}(\beta - y_i^{(t)})^2$. Also notice that $\phi_i^*(\beta) - \beta\langle a_i, x^\star\rangle$ is a γ-strongly convex function minimized by y_i^*, which implies

$$\phi_i^*(\widetilde{y}_i) - \widetilde{y}_i\langle a_i, x^\star\rangle \geq \phi_i^*(y_i^*) - y_i^*\langle a_i, x^\star\rangle + \frac{\gamma}{2}(\widetilde{y}_i - y_i^*)^2. \tag{57}$$

Then, following the same argument for establishing inequality (45) and plugging in inequality (57), we obtain

$$\frac{p_i n}{2\sigma}(y_i^{(t)} - y_i^\star)^2 \geq \Big(\frac{p_i n}{2\sigma} + \gamma\Big)(\widetilde{y}_i - y_i^\star)^2 + \frac{p_i n(\widetilde{y}_i - y_i^{(t)})^2}{2\sigma} + \langle a_i, x^\star - \overline{x}^{(t)}\rangle(\widetilde{y}_i - y_i^\star). \tag{58}$$

Note that $i = k$ with probability p_i. Therefore, we have

$$\begin{aligned}
(\widetilde{y}_i - y_i^\star)^2 &= \frac{1}{p_i}\mathbb{E}[(y_i^{(t+1)} - y_i^\star)^2|\mathcal{F}_t] - \frac{1-p_i}{p_i}(y_i^{(t)} - y_i^\star)^2,\\
(\widetilde{y}_i - y_i^{(t)})^2 &= \frac{1}{p_i}\mathbb{E}[(y_i^{(t+1)} - y_i^{(t)})^2|\mathcal{F}_t],\\
\widetilde{y}_i &= \frac{1}{p_i}\mathbb{E}[y_i^{(t+1)}|\mathcal{F}_t] - \frac{1-p_i}{p_i}y_i^{(t)},
\end{aligned}$$

where $\mathcal{F}_t$ represents the sigma field generated by all random variables defined before iteration t. Substituting the above equations into inequality (58), and averaging over $i = 1, 2, \ldots, n$, we have

$$\begin{aligned}
&\sum_{i=1}^n \left(\frac{1}{2\sigma} + \frac{(1-p_i)\gamma}{p_i n}\right)(y_i^{(t)} - y_i^\star)^2\\
\geq\ &\sum_{i=1}^n \left(\frac{1}{2\sigma} + \frac{\gamma}{p_i n}\right)\mathbb{E}[(y_i^{(t+1)} - y_i^\star)^2|\mathcal{F}_t] + \frac{\mathbb{E}[(y_k^{(t+1)} - y_k^{(t)})^2|\mathcal{F}_t]}{2\sigma}\\
&+ \mathbb{E}\Big[\Big\langle (u^{(t)} - u^\star) + \frac{1}{p_k}(u^{(t+1)} - u^{(t)}),\ x^\star - \overline{x}^{(t)}\Big\rangle\Big|\mathcal{F}_t\Big],
\end{aligned} \tag{59}$$

where $u^\star = \frac{1}{n}\sum_{i=1}^n y_i^\star a_i$ and $u^{(t)} = \frac{1}{n}\sum_{i=1}^n y_i^{(t)} a_i$ have the same definition as in the proof of Theorem 1.

For the relation between $x^{(t)}$ and $x^{(t+1)}$, we first notice that $\langle u^*, x\rangle + g(x)$ is a λ-strongly convex function minimized by x^*, which implies

$$\langle u^*, x^{(t+1)}\rangle + g(x^{(t+1)}) \geq \langle u^*, x^*\rangle + g(x^*) + \frac{\lambda}{2}(x^{(t+1)} - x^*)^2. \tag{60}$$

Following the same argument for establishing inequality (49) and plugging in inequality (60), we obtain

$$\begin{aligned}
\frac{\|x^{(t)} - x^\star\|_2^2}{2\tau} \geq &\left(\frac{1}{2\tau} + \lambda\right)\|x^{(t+1)} - x^\star\|_2^2 + \frac{\|x^{(t+1)} - x^{(t)}\|_2^2}{2\tau}\\
&+ \Big\langle (u^{(t)} - u^\star) + \frac{1}{p_k}(u^{(t+1)} - u^{(t)}),\ x^{(t+1)} - x^\star\Big\rangle.
\end{aligned} \tag{61}$$

Taking expectation over both sides of inequality (61) and adding it to inequality (59) yields

$$\begin{aligned}
\frac{\|x^{(t)} - x^\star\|_2^2}{2\tau} + &\sum_{i=1}^n \left(\frac{1}{2\sigma} + \frac{(1-p_i)\gamma}{p_i n}\right)(y_i^{(t)} - y_i^\star)^2 \geq \left(\frac{1}{2\tau} + \lambda\right)\mathbb{E}[\|x^{(t+1)} - x^\star\|_2^2|\mathcal{F}_t]\\
&+ \sum_{i=1}^n \left(\frac{1}{2\sigma} + \frac{\gamma}{p_i n}\right)\mathbb{E}[(y_i^{(t+1)} - y_i^\star)^2|\mathcal{F}_t] + \frac{\|x^{(t+1)} - x^{(t)}\|_2^2}{2\tau} + \frac{\mathbb{E}[(y_k^{(t+1)} - y_k^{(t)})^2|\mathcal{F}_t]}{2\sigma}\\
&+ \mathbb{E}\Big[\underbrace{\Big(\frac{(y^{(t)} - y^\star)^T A}{n} + \frac{(y_k^{(t+1)} - y_k^{(t)})a_k^T}{p_k n}\Big)((x^{(t+1)} - x^{(t)}) - \theta(x^{(t)} - x^{(t-1)}))}_{v}\Big|\mathcal{F}_t\Big],
\end{aligned} \tag{62}$$

where the matrix A is a n-by-d matrix, whose i-th row is equal to the vector a_i^T.

Next, we lower bound the last term on the right-hand side of inequality (62). Indeed, it can be expanded as

$$\begin{aligned}
v = &\frac{(y^{(t+1)} - y^\star)^T A(x^{(t+1)} - x^{(t)})}{n} - \frac{\theta(y^{(t)} - y^\star)^T A(x^{(t)} - x^{(t-1)})}{n}\\
&+ \frac{1-p_k}{p_k n}(y_k^{(t+1)} - y_k^{(t)})a_k^T(x^{(t+1)} - x^{(t)}) - \frac{\theta}{p_k n}(y_k^{(t+1)} - y_k^{(t)})a_k^T(x^{(t)} - x^{(t-1)}).
\end{aligned} \tag{63}$$

Note that the probability p_k given in (20) satisfies

$$p_k = \frac{1-\alpha}{n} + \frac{\alpha\|a_k\|_2}{\sum_{i=1}^n \|a_i\|_2} \geq \frac{(1-\alpha)\|a_k\|_2}{nR} + \frac{\alpha\|a_k\|_2}{\sum_{i=1}^n \|a_i\|_2} = \frac{\|a_k\|_2}{nR_\alpha}, \qquad k = 1, \ldots, n.$$

Since the parameters τ and σ satisfies $\sigma\tau R_\alpha^2 = 1/4$, we have $p_k^2 n^2/\tau \geq 4\sigma\|a_k\|_2^2$ and consequently

$$\begin{aligned}
\frac{|(y_k^{(t+1)} - y_k^{(t)})a_k^T(x^{(t+1)} - x^{(t)})|}{p_k n} &\leq \frac{\|x^{(t+1)} - x^{(t)}\|_2^2}{4\tau} + \frac{\|(y_k^{(t+1)} - y_k^{(t)})a_k\|_2^2}{p_k^2 n^2/\tau}\\
&\leq \frac{\|x^{(t+1)} - x^{(t)}\|_2^2}{4\tau} + \frac{(y_k^{(t+1)} - y_k^{(t)})^2}{4\sigma}.
\end{aligned}$$

Similarly, we have

$$\frac{|(y_k^{(t+1)} - y_k^{(t)})a_k^T(x^{(t)} - x^{(t-1)})|}{p_k n} \leq \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{(y_k^{(t+1)} - y_k^{(t)})^2}{4\sigma}.$$

Combining the above two inequalities with lower bounds (62) and (63), we obtain

$$\begin{aligned}
\frac{\|x^{(t)} - x^\star\|_2^2}{2\tau} + &\sum_{i=1}^n \left(\frac{1}{2\sigma} + \frac{(1-p_i)\gamma}{p_i n}\right)(y_i^{(t)} - y_i^\star)^2 \geq \left(\frac{1}{2\tau} + \lambda\right)\mathbb{E}[\|x^{(t+1)} - x^\star\|_2^2|\mathcal{F}_t]\\
&+ \sum_{i=1}^n \left(\frac{1}{2\sigma} + \frac{\gamma}{p_i n}\right)\mathbb{E}[(y_i^{(t+1)} - y_i^\star)^2|\mathcal{F}_t] + \frac{\mathbb{E}[\|x^{(t+1)} - x^{(t)}\|_2^2|\mathcal{F}_t] - \theta\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau}\\
&+ \frac{\mathbb{E}[(y^{(t+1)} - y^\star)^T A(x^{(t+1)} - x^{(t)})|\mathcal{F}_t] - \theta(y^{(t)} - y^\star)A(x^{(t)} - x^{(t-1)})}{n}.
\end{aligned} \tag{64}$$

Recall that the parameters τ, σ, and θ are chosen to be

$$\tau = \frac{1}{2R_\alpha}\sqrt{\frac{\gamma}{n\lambda}}, \quad \sigma = \frac{1}{2R_\alpha}\sqrt{\frac{n\lambda}{\gamma}}, \quad \text{and} \quad \theta = 1 - \left(\frac{n}{1-\alpha} + R_\alpha\sqrt{\frac{n}{\lambda\gamma}}\right)^{-1}.$$

Plugging in these assignments and using the fact that $p_i \geq \frac{1-\alpha}{n}$, we find that

$$\frac{1/(2\tau)}{1/(2\tau)+\lambda} = 1 - \left(1 + \frac{1}{2\tau\lambda}\right)^{-1} = 1 - \left(1 + R_\alpha\sqrt{\frac{n}{\lambda\gamma}}\right)^{-1} \leq \theta,$$

and for $i = 1, 2, \ldots, n$,

$$\frac{1/(2\sigma) + (1-p_i)\gamma/(p_i n)}{1/(2\sigma)+\gamma/(p_i n)} = 1 - \left(\frac{1}{p_i} + \frac{n}{2\sigma}\right)^{-1} \leq 1 - \left(\frac{n}{1-\alpha} + \frac{n}{2\sigma\gamma}\right)^{-1} = \theta.$$

Therefore, if we define a sequence $\Delta^{(t)}$ such that

$$\Delta^{(t)} = \left(\frac{1}{2\tau}+\lambda\right)\mathbb{E}[\|x^{(t)} - x^\star\|_2^2] + \sum_{i=1}^n \left(\frac{1}{2\sigma} + \frac{\gamma}{p_i n}\right)\mathbb{E}[(y_i^{(t)} - y_i^\star)^2] + \frac{\mathbb{E}[\|x^{(t)} - x^{(t-1)}\|_2^2]}{4\tau} + \frac{\mathbb{E}[(y^{(t)} - y^\star)^T A(x^{(t)} - x^{(t-1)})]}{n},$$

then inequality (64) implies the recursive relation $\Delta^{(t+1)} \leq \theta \cdot \Delta^{(t)}$, which implies

$$\left(\frac{1}{2\tau}+\lambda\right)\mathbb{E}[\|x^{(t)} - x^\star\|_2^2] + \left(\frac{1}{2\sigma} + \frac{\gamma}{n}\right)\mathbb{E}[\|y^{(t)} - y^\star\|_2^2] + \frac{\mathbb{E}[\|x^{(t)} - x^{(t-1)}\|_2^2]}{4\tau} + \frac{\mathbb{E}[(y^{(t)} - y^\star)^T A(x^{(t)} - x^{(t-1)})]}{n} \leq \theta^t \Delta^{(0)}, \tag{65}$$

where

$$\begin{aligned}\Delta^{(0)} &= \left(\frac{1}{2\tau}+\lambda\right)\|x^{(0)} - x^\star\|_2^2 + \sum_{i=1}^n \left(\frac{1}{2\sigma} + \frac{\gamma}{p_i n}\right)(y_i^{(0)} - y_i^\star)^2 \\ &\leq \left(\frac{1}{2\tau}+\lambda\right)\|x^{(0)} - x^\star\|_2^2 + \left(\frac{1}{2\sigma} + \frac{\gamma}{1-\alpha}\right)\|y^{(0)} - y^\star\|_2^2.\end{aligned}$$

To eliminate the last two terms on the left-hand side of inequality (65), we notice that

$$\begin{aligned}\frac{|(y^{(t)} - y^\star)^T A(x^{(t)} - x^{(t-1)})|}{n} &\leq \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{\|y^{(t)} - y^\star\|_2^2 \|A\|_2^2}{n^2/\tau} \\ &\leq \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{\|y^{(t)} - y^\star\|_2^2 \|A\|_F^2}{n^2/\tau} \\ &= \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{\|y^{(t)} - y^\star\|_2^2 \sum_{i=1}^n \|a_i\|_2^2}{4\sigma(\sum_{i=1}^n \|a_i\|_2)^2} \\ &\leq \frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} + \frac{\|y^{(t)} - y^\star\|_2^2}{4\sigma},\end{aligned}$$

where in the equality we used $n^2/\tau = 4\sigma n^2 R_\alpha^2 \geq 4\sigma n^2 \overline{R} = 4\sigma\left(\sum_{i=1}^n \|a_i\|_2\right)^2$. This implies

$$\frac{(y^{(t)} - y^\star)^T A(x^{(t)} - x^{(t-1)})}{n} \geq -\frac{\|x^{(t)} - x^{(t-1)}\|_2^2}{4\tau} - \frac{\|y^{(t)} - y^\star\|_2^2}{4\sigma}.$$

Substituting the above inequality into inequality (65) completes the proof.

References

Alekh Agarwal and Léon Bottou. A lower bound for the optimization of finite sums. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 78–86, Lille, France, 2015.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1200–1205, Montreal, Canada, June 2017.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-threshold algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Dimitri P. Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical Programming, Ser. B*, 129:163–195, 2011.

Dimitri P. Bertsekas. Incremental gradient, subgradient, and proximal methods for convex optimization: a survey. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*, chapter 4. The MIT Press, 2012.

Doron Blatt, Alfred Hero, and Hillel Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2007.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, pages 177–187, Paris, France, August 2010. Springer.

Léon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 161–168. MIT Press, Cambridge, MA, 2008.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40 (1):120–145, 2011.

Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. Coordinate descent method for large-scale l_2-loss linear support vector machines. *Journal of Machine Learning Research*, 9: 1369–1398, 2008.

Min-Te Chao. A general purpose unequal probability sampling plan. *Biometrika*, 69(3): 653–656, 1982.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27*, pages 1646–1654. 2014.

John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2873–2898, 2009.

Rong-En Fan and Chih-Jen Lin. LIBSVM data: Classification, regression and multi-label. URL: http://www.csie.ntu.edu.tw/˜cjlin/libsvmtools/datasets, 2011.

Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of The 32nd International Conference on Machine Learning (ICML)*, pages 2540–2548. 2015.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2nd edition, 2009.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Springer, 2001.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 408–415, 2008.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323. 2013.

Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. Technical report, Department of Industrial and System Engineering, University of Florida, July 2015.

John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, 10:777–801, 2009.

Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 25*, pages 2672–2680. 2012.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems 28*, pages 3384–3392. 2015a.

Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015b.

P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, December 1979.

Angelia Nedić and Dimitri P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138, 2001.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Mathematical Programming*, 155 (1-2):549–573, 2016.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, 2004.

Yurii Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103:127–152, 2005.

Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming, Ser. B*, 140:125–161, 2013.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.

Hua Ouyang, Niao He, Long Tran, and Alexander Gray. Stochastic alternating direction method of multipliers. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, Atlanta, GA, USA, 2013.

John Platt. Fast training of support vector machine using sequential minimal optimization. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

Boris T. Polyak and Anatoli Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30:838–855, 1992.

Zheng Qu, Peter Richtárik, and Tong Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in Neural Information Processing Systems 28*, pages 865–873. 2015.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144 (1):1–38, 2014.

Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1):433–484, 2016.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. Technical Report HAL 00860051, INRIA, Paris, France, 2013.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013a.

Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 378–385. 2013b.

Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1):105–145, 2015.

Taiji Suzuki. Dual averaging and proximal gradient descent for online alternating direction multiplier method. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 392–400, Atlanta, GA, USA, 2013.

Taiji Suzuki. Stochastic dual coordinate ascent with alternating direction method of multipliers. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 736–744, Beijing, 2014.

Martin Takáč, Avleen Bijral, Peter Richtárik, and Nathan Srebro. Mini-batch primal and dual methods for SVMs. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

Paul Tseng. An incremental gradient(-projection) method with momentum term and adaptive stepsiz rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.

Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. Unpublished manuscript, 2008.

Huahua Wang and Arindam Banerjee. Online alternating direction method. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1119–1126, Edinburgh, Scotland, UK, 2012.

Blake Woodworth and Nathan Srebro. Tight complexity bounds for optimizing composite objectives. arXiv:1605.08003, 2016.

Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming, Series B*, 151(1):3–34, 2015.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2534–2596, 2010.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Tianbao Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems 26*, pages 629–637. 2013.

Adams Wei Yu, Qihang Lin, and Tianbao Yang. Double stochastic primal-dual coordinate method for regularized empirical risk minimization with factorized data. arXiv:1508.03390, 2015.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the 21st International Conference on Machine Learning (ICML)*, pages 116–123, Banff, Alberta, Canada, 2004.

Xiaoqun Zhang, Martin Burger, and Stanley Osher. A unifoed primal-dual algorithm framework based on Bregman iteration. *Journal of Scientific Computing*, 46(1):20–46, January 2011.

Yuchen Zhang and Lin Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proceedings of The 32nd International Conference on Machine Learning (ICML)*, pages 353–361. 2015.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *JMLR Proceedings*, pages 1–9. JMLR.org, 2015.

Leon Wenliang Zhong and James T. Kwok. Fast stochastic alternating direction method of multipliers. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 46–54, Beijing, China, 2014.

Journal of Machine Learning Research 18 (2017) 1-21 Submitted 1/17; Revised 5/17; Published 7/17

Angle-based Multicategory Distance-weighted SVM

Hui Sun SUN400@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47906, USA

Bruce A. Craig BACRAIG@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47906, USA

Lingsong Zhang LINGSONG@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47906, USA

Editor: John Shawe-Taylor

Abstract

Classification is an important supervised learning technique with numerous applications. We develop an angle-based multicategory distance-weighted support vector machine (MDWSVM) classification method that is motivated from the binary distance-weighted support vector machine (DWSVM) classification method. The new method has the merits of both support vector machine (SVM) and distance-weighted discrimination (DWD) but also alleviates both the *data piling* issue of SVM and the imbalanced data issue of DWD. Theoretical and numerical studies demonstrate the advantages of MDWSVM method over existing angle-based methods.

Keywords: Discriminant analysis, Imbalanced data, High dimension, Support vector machine, Distance-weighted discrimination

1. Introduction

Classification is important in both statistics and machine learning. The goal of classification is to build a classifier such that it can predict the category of a new observation. Popular classification methods include Fisher's linear discriminant analysis, logistic regression, support vector machine (SVM), and boosting. See Hastie et al. (2001) for an introduction to various classification methods.

SVM (Schölkopf and Burges, 1999; Cristianini and Shawe-Taylor, 2000) has been shown to be a very popular and powerful method. It is well known that the binary SVM searches for a hyperplane in the feature space (that is a type of projection space from the original data) that maximizes the margin (a gap between the two groups). SVM has numerous applications, such as image classification (Chapelle et al., 1999; Foody and Mathur, 2006) and cancer diagnostics (Duan et al., 2005; Wang and Huang, 2011).

©2017 Sun, Craig and Zhang.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-003.html.

In a high-dimensional, low sample size (HDLSS) setting, Marron et al. (2007) and Ahn and Marron (2010) observed a *data-piling* phenomenon with the binary SVM and other classification methods. A SVM-type linear classifier is a margin-based classifier. It has a separating hyperplane, and its normal vector is essentially the discriminant direction. Data-piling is the phenomenon when projecting the data points to the discriminant direction that many of these projections are identical. This phenomenon indicates that the resulting separating hyperplane might be affected by noise artifacts in the data. Noise artifacts result in a discrimination direction far away from the Bayes direction. See more discussion in Ahn and Marron (2010).

To alleviate the *data-piling* issue, Marron et al. (2007) proposed the binary distance-weighted discrimination (DWD) classifier. The idea of DWD is to minimize the total inverse margin of all the data points. This method works quite well in the HDLSS setting. However, because the DWD method uses all the observations to estimate the decision boundary, it is very sensitive to imbalanced sample sizes (Qiao et al., 2010). In particular, when the sample size of one class is much larger than the other, the classification boundary will be pushed towards the minority class and all future data will be assigned to the majority class. See more discussion in Qiao and Zhang (2015a,b) for the HDLSS overfitting issue of SVM and imbalanced data issue of DWD. To deal with both problems, Qiao and Zhang (2015a) proposed a binary distance-weighted support vector machine (DWSVM) method, which can be viewed as a combination of the binary SVM and DWD. The new method inherits both the merits of SVM and DWD yet outperforms both SVM and DWD in the HDLSS and imbalanced context.

In practice, many classification problems have more than two classes. A natural way to deal with multiclass classification problems is to take a one-versus-one or one-versus-rest approach, see examples in Hastie et al. (1998) and Allwein et al. (2000). Though both approaches are intuitive, the one-versus-one approach can lead to a tie-in-vote problem and the one-versus-rest approach suffers from inconsistency when there is no dominant class (Lee et al., 2004).

It is more desirable to consider all classes simultaneously. In a multiclass setting, the observed data are $(\boldsymbol{x}_i, y_i), i = 1, \ldots, n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is a multivariate predictor, the scalar $y_i \in \{1, \ldots, K\}$ is the corresponding class label, with K as the number of classes. Many classification approaches map x to $\boldsymbol{f}(x) \in \mathbb{R}^K$, and the corresponding prediction rule is $\hat{y} = \arg\max_i f_j(\boldsymbol{x})$, where f_j is the jth element of $\boldsymbol{f}$. In this type of approach, a constraint such as $\sum_{j=1}^K \boldsymbol{f}_j = 0$ is usually imposed to remove redundancy and reduce the dimension of the problem. See Zhu and Hastie (2001); Lee et al. (2004); Liu and Shen (2006); Liu (2007); Liu and Yuan (2011) for more discussion. Fisher consistency for several existing multicategory hinge loss functions are also provided in Liu (2007).

It is straightforward to see that the sum-to-zero constraint can be removed if we redefine $\boldsymbol{f}$ in $\mathbb{R}^K$ to be in $\mathbb{R}^{K-1}$, as the degrees of freedom of $\boldsymbol{f}$ is essentially $K-1$. Several classifiers have been proposed using this fact. For example, Lange and Wu (2008) proposed a vertex based model that maps the data $\boldsymbol{x}$ in $\mathbb{R}^d$ to a $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^{K-1}$, and predicts the label based on the distance of $\boldsymbol{f}(\boldsymbol{x})$ to K predefined vertices. Zhang and Liu (2014) proposed a similar idea where they define the same K vertices in the $\mathbb{R}^{k-1}$ space, and use the angle between $\boldsymbol{f}$ to these vertex vectors to predict labels. More details are given in Section 2 of this paper.

The angle-based method can be viewed as a natural extension of the binary large margin classifier to the multiclass context. Zhang and Liu (2014) replace the usual functional margin by the angle (or inner product) between the projection $\boldsymbol{f}$ and the vertices. Their simulation results show that these angle-based classifiers have good prediction performance. The Fisher consistency of a family of large margin classifiers is also proved. However, as a specific case in large margin classifiers, the angle-based SVM (MSVM) method is not Fisher consistent because its loss function is not a strictly monotone decreasing function. In Zhang and Liu (2014), Fisher consistency of a proximal SVM was proposed and proven instead.

In our experiment in Section 2, under HDLSS and imbalanced data setting, we observed that MSVM suffers from *data piling* issue. Binary DWD, based on the idea of Zhang and Liu (2014), can be extended to multicategory angle-based DWD (MDWD). Though free from the *data piling* concern, MDWD suffers from the imbalanced issue. Both these issues were previously observed in the binary case (Marron et al., 2007; Qiao and Zhang, 2015a).

Note that Huang et al. (2013) extended the binary DWD to a version of multiclass DWD (MDWDH). It adopts the idea of pairwise comparisons from one-versus-one idea. MDWDH considers a data point with label i being misclassified if the difference of projections on ith member and jth member is negative ($i \neq j$). Even though it has nice theoretical properties and empirical performance as shown in their paper, the angle-based methods have better geometric interpretation. In addition, the pair-wise natural of the method will have more expensive computational cost, compared to angle-based approaches. We also observed that the performance is slightly better than angled-based MDWD method. If the angle-based MDWD approach also incorporates the pairwise idea, the two approaches will have similar empirical performance.

In this paper, we adapt the idea in Qiao and Zhang (2015a) to develop a hybrid of MSVM and MDWD. The work can also be viewed as an extension of the binary DWSVM to the multiclass context. We prove its Fisher consistency and use extensive simulation studies to show the usefulness of our approach. For many cases, the novel approach outperforms both MDWD and MSVM, especially under HDLSS and the imbalanced case.

The rest of this article is organized as follows. In Section 2, we briefly review the existing multicategory classifiers, and introduce our angle-based distance-weighted support vector machine (MDWSVM) model. In Section 3, we prove the Fisher consistency and show some imbalance properties of our new approach. In Section 4, we perform simulation studies to compare our model with MSVM and MDWD. The sensitivity of the prediction performance in terms of the tuning parameters is also explored in this section. Section 5 involves a real application and Section 6 discusses some future work for this model. The proofs of all theorems and lemmas are given in the appendix.

2. Methodology

In this section, we give a general introduction to classification, including the angle-based multicategory classifier. We then show some drawbacks of this angle-based classification, which motivates our MDWSVM. We conclude with a detailed introduction of our approach, along with its implementation.

2.1 Classification and Loss Function

Consider a binary classification problem with observed data $(\boldsymbol{x}_i, y_i), i = 1, \ldots, n$. The $\boldsymbol{x}_i \in \mathbb{R}^d$ is a multivariate predictor and the scalar $y_i \in \{1, -1\}$ is the corresponding class label. The goal is to find a decision function f along with its prediction $\hat{y}(\boldsymbol{x}) = \text{sign}(f(\boldsymbol{x}))$ to minimize the misclassification error $E(\hat{Y} \neq Y)$. Note that when $yf(\boldsymbol{x}) > 0$, $f(\boldsymbol{x})$ gives a correct prediction; otherwise $f(\boldsymbol{x})$ gives a misclassification. A natural way to estimate the misclassification error is to use the empirical error $1/n \sum \mathbb{I}(\hat{y}_i(\boldsymbol{x}) \neq y_i) = 1/n \sum \mathbb{I}(y_i f(\boldsymbol{x}_i) < 0)$, where $\mathbb{I}(.)$ is the indicator function. However, due to the discontinuity and nonconvexity of $\mathbb{I}(y_i f(\boldsymbol{x}_i) < 0)$, it is hard to conduct a direct minimization.

A common surrogate is a convex loss function $\ell(.)$, which is commonly used in large margin classifiers (Hastie et al., 2001). A large margin classifier can be viewed as minimizing the loss function given a constraint

$$\min_{f \in \mathbb{F}} \quad \sum_{i=1}^{n} \ell(f(\boldsymbol{x}_i), y_i) + \frac{\lambda}{2} J(f),$$

where $\mathbb{F}$ denotes the function space and $J(.)$ is a type of norm, which is used to control the complexity of the model. The function $\ell(f, y)$ is a loss function surrogate for the 0-1 loss. The tuning parameter λ balances the loss and the norm. For example, the popular linear SVM uses the hinge loss function $\ell_S(u) = (1-u)_+$ where $u = yf(\boldsymbol{x})$, and the L_2 norm.

The SVM method can also be viewed as maximizing the smallest distances of all observations to the separating hyperplane. As discussed in Section 1, SVM suffers from the *data piling* problem in HDLSS setting. Marron et al. (2007) proposed the DWD method, which improves the performance of SVM in the HDLSS setting. Essentially, DWD minimizes the mean of inverse distance of all data vectors to the separating hyperplane. As is discussed in Bartlett et al. (2006); Liu et al. (2011); Qiao and Zhang (2015a), DWD is also a large margin classifier, and its loss function is

$$\ell_D(u) = \begin{cases} 2 - u & u \leq 1 \\ 1/u & \text{otherwise.} \end{cases} \tag{1}$$

In practice, lots of applications deal with multicategory rather than binary classification. For multiclass problems, $y_i \in \{1, 2, \ldots, K\}, i = 1, \ldots, n$, with K the number of classes. The common simultaneous procedure is to map $\boldsymbol{x}$ to $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^K$, and the corresponding prediction rule is $\hat{y} = \arg\max_j f_j(\boldsymbol{x})$, where f_j is the jth element of $\boldsymbol{f}$. Commonly a sum to zero constraint on $\boldsymbol{f}$ is used as discussed in Section 1 to overcome identifiability issues, see more discussion in Vapnik and Vapnik (1998); Lee et al. (2004); Liu and Yuan (2011).

Many multicategory classification methods can be viewed as the following constrained optimization problem,

$$\begin{aligned} \min_{f \in \mathbb{F}} \quad & \sum_{i=1}^{n} \ell(\boldsymbol{f}(\boldsymbol{x}_i), y_i) + \frac{\lambda}{2} \sum_{j=1}^{K} J(f_j), \\ \text{s.t.} \quad & \sum_{j=1}^{K} f_j(\boldsymbol{x}) = 0. \end{aligned}$$

For example, a multicategory SVM with hinge loss (Vapnik and Vapnik, 1998) uses the loss function $\ell(\boldsymbol{f}(\boldsymbol{x}), y) = (1 - f_y(\boldsymbol{x}))_+$. However, unlike binary classification, multicategory classification with a sum to zero constraint does not have a clear geometric explanation. It also suffers from expensive computation (Zhang and Liu, 2014). To overcome these limitations, Lange and Wu (2008) proposed the vertex idea where they define $\boldsymbol{f}$ as a $K-1$ dimensional function instead of a K dimensional function. This removes the need for a sum-to-zero constraint. A similar idea is used later by Zhang and Liu (2014) to conduct angle-based classification, which will be discussed next.

2.2 Angle-based Classification Framework

The idea of angle-based classification is to map $\boldsymbol{x}$ to $\boldsymbol{f}(\boldsymbol{x})$, where $\boldsymbol{f} = (f_1, \ldots, f_{K-1})$, with a set of K predefined vertices in $\mathbb{R}^{K-1}$. We then assess which vertex has the smallest angle to the projection $\boldsymbol{f}$, and the corresponding label is the prediction. In Zhang and Liu (2014), the vertices $W = (W_1, W_2, \ldots, W_K)$ are defined as a collection of K vectors in $\mathbb{R}^{K-1}$ with elements

$$W_j = \begin{cases} (K-1)^{-1/2}\zeta, & j = 1, \\ -(1+K^{1/2})/\{(K-1)^{3/2}\}\zeta + \{K/(K-1)\}^{1/2} e_{j-1}, & 2 \le j \le K. \end{cases}$$

The unit vector ζ is of length $K-1$, and e_j is a vector in $\mathbb{R}^{K-1}$ such that all of its element are 0, except the jth is 1.

In this setting, W form a simplex with K vertices in a $(K-1)$ dimensional space. The center of W is at the origin, and each of the $W_j, j = 1, \ldots, K$ has Euclidean norm of 1. Further, it is easy to check that the angle between each pair of vertices W_i and W_j, $i \neq j$ is the same. Instead of y_i, W_{y_i} is used to represent the observed class. The prediction function is $\hat{y} = \arg\max_j \langle W_j, \hat{\boldsymbol{f}} \rangle$, where the inner product $\langle ., . \rangle$ between the two vectors denotes the projection of $\hat{\boldsymbol{f}}$ to W_j. The larger the inner product, the smaller the angle between $\hat{\boldsymbol{f}}$ and W_j.

With this prediction rule, Zhang and Liu (2014) proposed the optimization model for the angle-based classification

$$\min_{\boldsymbol{f} \in \mathbb{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(\langle \boldsymbol{f}(\boldsymbol{x}_i), W_{y_i} \rangle) + \frac{\lambda}{2} J(\boldsymbol{f}). \tag{2}$$

The product $\langle \boldsymbol{f}(\boldsymbol{x}_i), W_{y_i} \rangle$ can be viewed as a new functional margin of $(\boldsymbol{x}, y)$. Defining $u = \langle \boldsymbol{f}(\boldsymbol{x}_i), W_{y_i} \rangle$, one of the examples given in Zhang and Liu (2014) is the multicategory angle-based SVM (MSVM) where the loss function is replaced by hinge loss $\ell_S(u) = (1-u)_+$ and with L_2 norm. DWD loss can be applied to this framework as well, along with more generalizations of binary large margin classifiers. See Zhang and Liu (2014) for more details.

2.3 From Binary DWSVM to MDWSVM

Through the exploration of the angle-based classification method, we found that MSVM has similar *data piling* issues; while MDWD does not have *data piling* problems, it suffers from imbalanced issues. To demonstrate the *data piling* and imbalanced issues, we show projection plots of a simulated example. In this example, we randomly simulate three

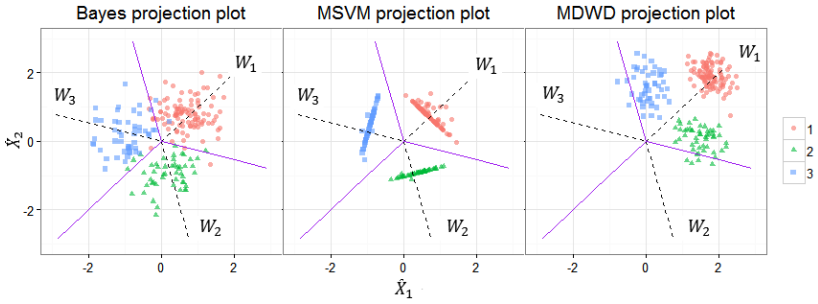


Figure 1: Plots of projections and W_j's in $\mathbb{R}^2$ space. Dashed lines are the W_j's, $j = 1, \ldots, 3$; Dots, triangles and squares represent the points from the three different classes. The left panel shows the projection plot for the Bayes classification, the middle one is for angle-based MSVM, and the right one is for the angle-based MDWD. The middle panel shows the MSVM has severe *data piling* issues (the middle panel), and MDWD in the right panel suffers from imbalanced issues (the right panel).

classes of observations with 500 covariates, the sample sizes for each class are 100, 50, 50 respectively. For each group, the first two covariates are distributed $N(\mu_j, \sigma^2 I_2)$, where μ_j's are three fixed points equally spaced on the unit circle, and $\sigma = 0.5$. All other covariates are independently and identically distributed $N(0, \sigma^2)$. Note that the data are HDLSS and imbalanced.

One representation of the projection plots is given in Figure 1. This plot is used to visualize the n projections $\boldsymbol{f}(x_i)$'s, $i = 1, \ldots, n$ and vectors W_j's, $j = 1, \ldots, 3$ (dashed lines) in the $\mathbb{R}^2$ plane. The different colors and shapes correspond to different groups. The solid purple lines are the Bayesian decision boundaries. $\tilde{X}_1$ and $\tilde{X}_2$ are the two axes in this $\mathbb{R}^2$ plane.

From Figure 1 it is clear that MSVM has severe *data piling* issues since almost all the points project to a single point on the W direction. Furthermore, MDWD suffers from the imbalanced issue as the angle-based classification assigns almost all the points to the dominant Class 1. These findings agree with those in Qiao and Zhang (2015a) for the binary case.

To alleviate both data piling and imbalanced issues, Qiao and Zhang (2015a) proposed binary DWSVM, a combination of SVM and DWD. The method has the form

$$\min_{f \in \mathbb{F}} \frac{1}{n} \sum_{i=1}^{n} \alpha \ell_D(y_i f_0(\boldsymbol{x}_i)) + (1-\alpha)\ell_S(y_i f(\boldsymbol{x}_i)) + \frac{\lambda}{2} J(f), \tag{3}$$

where $f_0(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{\omega} + \beta_0$ and $f(\boldsymbol{x}) = \boldsymbol{x}\boldsymbol{\omega} + \beta$, $\boldsymbol{\omega} \in \mathbb{R}^d$ is the coefficient direction vector and scalar $\beta, \beta_0 \in \mathbb{R}$. The loss function ℓ_D is from formula (1) and ℓ_S is the hinge loss. Notice that β is the SVM intercept and β_0 is the DWD intercept, which is called auxillary intercept in the DWSVM paper. The prediction function is $\hat{y} = \text{sign}(f(\boldsymbol{x})) = \text{sign}(\boldsymbol{x}_i^T \boldsymbol{\omega} + \beta)$. The tuning parameter $0 < \alpha < 1$ is used to balance SVM and DWD losses.

Qiao and Zhang (2015a) show that in binary classification, by choosing the appropriate α, the DWSVM method will result in a smaller misclassification error compared to both the DWD method and SVM method in HDLSS and imbalanced data context. Furthermore, in terms of the similarity to the Bayes classifier, the DWSVM are similar to the DWD but better than the SVM.

The DWSVM method motivated us to build a multicategory DWSVM within the angle-based framework. Applying DWSVM to the angle-based framework, we propose a multicategory angle-based DWSVM (MDWSVM)

$$\min_{\boldsymbol{f}\in\mathbb{F}} \frac{1}{n}\sum_{i=1}^{n} \alpha\ell_D(\langle \boldsymbol{f_0}(\boldsymbol{x}_i), W_{y_i}\rangle) + (1-\alpha)\ell_S(\langle \boldsymbol{f}(\boldsymbol{x}_i), W_{y_i}\rangle) + \frac{\lambda}{2}J(\boldsymbol{f}). \tag{4}$$

In this model $\boldsymbol{f}(\boldsymbol{x}_i) = \boldsymbol{x}_i B + \beta_0$ and $\boldsymbol{f_0}(\boldsymbol{x}_i) = \boldsymbol{x}_i B + \beta_0^d$, where $B = (B_1, B_2, \ldots, B_{K-1})$, each of the $B_j, j = 1, \ldots, K-1$ is a vector of length d, which does not include the intercept.The parameters $\beta_0, \beta_0^d \in \mathbb{R}^{K-1}$ are intercept vectors. Note that $\boldsymbol{f_0}$ and $\boldsymbol{f}$ are only different in terms of the intercept β_0 and β_0^d respectively. In this model, the interest is to find B, β_0 and β_0^d to minimize the loss function. For prediction $\hat{y} = \arg\max_j \langle W_j, \hat{\boldsymbol{f}}\rangle = \arg\max_j \langle W_j, \boldsymbol{x}_i B + \beta_0\rangle$, however, only β_0 and B are used. This avoids the imbalanced issue cost by β_0^d. Note that the prediction $\arg\max_j \langle W_j, \hat{\boldsymbol{f}}\rangle$ is equivalent to $\arg\min_j \angle(W_j, \hat{\boldsymbol{f}})$ where $\angle(a, b)$ represents the angle between vector a and b . We predict $\boldsymbol{x}$ with the label j such that vertex W_j and $\boldsymbol{f}(\boldsymbol{x})$ has the smallest angle among all $\angle(W_j, \hat{\boldsymbol{f}}), j = 1, \ldots, K$. Observe that $\sum_{j=1}^{K} \langle W_j, \hat{\boldsymbol{f}}\rangle = 0$ for all $\boldsymbol{x}$, which means the angle-based classification framework automatically includes sum-to-zero constraints.

2.4 Implementation of MDWSVM

In Qiao and Zhang (2015a), the implementation of the binary DWSVM (3) was through second-order cone programming. Mathematically, the DWSVM model can be written as

$$\begin{aligned} \min_{\boldsymbol{\omega},\beta,\beta_0,\eta_i,\xi_i} \quad & \sum_{i=1}^{n}\{\alpha(\frac{1}{r_i} + \eta_i) + (1-\alpha)\xi_i\}, \\ \text{s.t.} \quad & r_i = y_i(x_i^T\boldsymbol{\omega} + \beta) + \eta_i, \quad r_i \geq 0 \quad \text{and} \quad \eta_i \geq 0, \\ & y_i(x_i^T\boldsymbol{\omega} + \beta) + \xi_i \geq 1, \quad \xi_i \geq 0, \\ & \|\boldsymbol{\omega}\|^2 \leq C. \end{aligned} \tag{5}$$

The first constraint $r_i = y_i(x_i^T\boldsymbol{\omega} + \beta) + \eta_i$ is the distance from each data vector i to its separating hyperplane (adding slackness to allow misclassification), which corresponds to the DWD optimization. The second constraint $y_i(x_i^T\boldsymbol{\omega} + \beta) + \xi_i \geq 1$ is a standard constraint in SVM optimization. Both η_i and ξ_i control the misclassification rate, but with different decision boundaries. The third constraint $\|\boldsymbol{\omega}\|^2 \leq C$ is equivalent to the second term in (2), the Euclidean norm.

To extend (5) to multiclass, we replace the distances (functional margins) to the inner product as introduced earlier in (2). Thus our MDWSVM will have the following mathematical form:

$$\begin{aligned} \min_{\boldsymbol{f},\boldsymbol{f_0}} \quad & \sum_{i=1}^{n}\{\alpha(\frac{1}{r_i} + \eta_i) + (1-\alpha)\xi_i\}, \\ \text{s.t.} \quad & r_i = \langle \boldsymbol{f_0}(x_i), W_{y_i}\rangle + \eta_i, \quad r_i \geq 0 \quad \text{and} \quad \eta_i \geq 0, \\ & \langle \boldsymbol{f}(x_i), W_{y_i}\rangle + \xi_i \geq 1, \quad \xi_i \geq 0, \\ & \sum_{j=1}^{k-1} B_j^T B_j \leq C. \end{aligned} \tag{6}$$

In this form $\boldsymbol{f_0}$ and $\boldsymbol{f}$ are the same as in (4). It is verified in Zhang and Liu (2014) that the first term in the objective function $\sum_{i=1}^{n}(\frac{1}{r_i} + \eta_i)$ along with its constraint is equivalent to the objective of the MDWD method, and the second term in the objective function $\sum_{i=1}^{n}\xi_i$ along with its constraint is equivalent to the objective in the MSVM method. In this case, MDWSVM can be viewed as a convex combination of MDWD and MSVM losses where the parameter α balances the two.

Model (6) can be easily implemented in Matlab using the CVX package (Grant et al., 2008). Notice the only difference between $\boldsymbol{f}(\boldsymbol{x})$ and $\boldsymbol{f_0}(\boldsymbol{x})$ is their location vectors β_0 and β_0^d. For prediction we only adopt the location vector from MSVM, which shows insensitivity to the imbalanced issue from Figure 1. Moreover, by combining the discriminant direction of MDWD and MSVM, our new model will have a better discriminant direction (closer to the Bayes direction) than the MSVM method alone. Both improvements will be shown in Sections 4 and 5 using simulations and real examples.

3. Theoretical Properties

Fisher consistency is a fundamental requirement for a classification method. Fisher consistency implies that when the sample size approaches infinity, the classifier becomes closer and closer to the Bayes classification rule, which corresponds to the minimum misclassification rate. Qiao and Zhang (2015a) explored Fisher consistency of the binary DWSVM model. In the multiclass context, Zhang and Liu (2014) extended Fisher consistency to all large margin classification models under the angle-based framework.

Let $P_j = \Pr(Y = j|X = \boldsymbol{x})$ for $j = 1, \ldots, K$. Note that $\hat{y} = \arg\max_j P_j$ is the Bayes rule. Assume that for a given $\boldsymbol{x}$, the vector $\boldsymbol{f^*}(\boldsymbol{x})$ minimizes $E[\ell\{\langle \boldsymbol{f}(X), W_Y\rangle\}|X = \boldsymbol{x}]$, and the corresponding decision boundary will then be $\hat{y} = \arg\max_j \langle \boldsymbol{f^*}(\boldsymbol{x}), W_Y\rangle$. Note that this is essentially the limit minimizer of (2) when sample size diverges to infinity. Fisher consistency assures that these two decision functions are the same ($\arg\max_j P_j = \arg\max_j \langle \boldsymbol{f^*}(\boldsymbol{x}), W_Y\rangle$).

In this section, we will prove that if using the approximate SVM loss function from Zhang and Liu (2014) in replacement of hinge loss, our MDWSM is Fisher consistent.

Theorem 1 *The MDWSVM is Fisher consistent for any* $0 < \alpha < 1$.

Fisher consistency in Theorem 1 ensures that the minimizer of the expected loss function will assign an observation to the same class as what Bayes rule does. Furthermore, in our numerical study in Section 4.2, we notice that for MDWSVM method, as long as C is fixed, different α's will give similar performance in both prediction error and closeness to

the Bayes rule. Thus we will fix α to be 0.5 in this paper and not discuss the choice of α further.

In the next theorem, we want to prove that MDWSVM is insensitive to imbalance. Using a similar paradigm as in Owen (2007), we consider the case that the sample size of one class diverges to infinity. Qiao and Zhang (2015a) showed that, in binary classification, the intercept term of DWD diverges, but the intercept of SVM and DWSVM will not diverge. This shows that SVM is not sensitive to imbalance, but DWD will be severely affected. In our multiclass setting, for simplicity, it is assumed that only one of the classes is the dominant one, and the sample size of all other classes are equally fixed. Without loss of generality, we assume their sample sizes are all 1. Under this setting, we can simply assume observation $1, \ldots, K-1$ belongs to the class $1, \ldots, K-1$ respectively, and observations $K \ldots, n$ belong to class K. As n goes to infinity, the classifier tends to classify all the points to the dominant class K. If this happens, $\langle \beta_0, W_{yK} \rangle$ goes to infinity. In the next proposition, we prove that this will be not be the case for the angle-based SVM. Furthermore, we present in Theorem 3 that the intercept of our MDWSVM model is not sensitive to imbalance either.

Proposition 2 *In MSVM setting, when the size of the majority class goes to infinity,* $\langle \beta_0, W_{yK} \rangle < \sqrt{2C}K \max |x_{ij}| + 1$.

Theorem 3 *In the MDWSVM setting, when the size of the majority class goes to infinity,* $\langle \beta_0, W_{yK} \rangle < \sqrt{2C}K \max |x_{ij}| + 1$.

Note that Theorem 3 does not ensure that the MDWSVM method completely overcomes the imbalanced issue. When the sample size of the majority group goes to infinity, the method still will ignore some observations in minority groups.

4. Simulation

In this section, we use three simulation examples to demonstrate the performance of our MDWSVM method. We compare it to the angle-based SVM (MSVM) described in Section 2 and the angle-based DWD (MDWD) naturally developed using the ideas from Zhang and Liu (2014).

In each example, we simulate a training data set, a tuning data set, and a testing data set. The training data and tuning data have the same sample sizes and are used to estimate the model and to find the optimal tuning parameters. The size of the testing data set is ten times the size of the training data, and is used to evaluate the prediction performance. As we are interested in the misclassification rate in both the dominant class and the minority classes, we will not use the total error rate $1/n \sum_i I(\hat{y}_i \neq y_i)$ in this paper. Instead, we use the average within-group error rate as follows

$$r = \frac{1}{K} \sum_{j=1}^{K} \left\{ \frac{1}{n_j} \sum_{i: x_i \in C_j} I(\hat{y}_i \neq j | x_i \in C_j) \right\}.$$

Here C_j stands for class j and n_j is the sample size for class j. This measure was previously introduced in Qiao and Liu (2009). Note that the term within the bracket is the error rate for each group, so r is the arithmetic average of all these error rates.

We also want to measure the closeness of the estimated classifier to the Bayes rule. For the binary case, we can measure the angle between the two linear decision boundaries. For the multiclass case, we develop a similar measure as follows. Note that B is the projection matrix from $\mathbb{R}^d$ to $\mathbb{R}^{K-1}$ (the projection space). In the binary case, B is the discrimination direction vector, we can use the Euclidean inner product $\langle B, B_{\text{Bayes}} \rangle$ to measure the angle between the estimated and the Bayes rule. For the multiclass case, both B and B_{Bayes} are matrices. In matrix form, we want to measure the angle between the jth columns in both B and B_{Bayes}, and then calculate an average of these angles.

In this paper, we use the Frobenius inner product: $\langle B, B_{\text{Bayes}} \rangle_F = \sum_{i,j} B_{ij} B_{\text{Bayes } ij}$. Essentially, this is the sum of entries of the Hadamard product between B and B_{Bayes}. One can see that this is the same idea as the inner product of the corresponding columns in B and B_{Bayes}, a scaled mean of the inner products. To make this quantity directly linked to angle, we normalize both B and B_{Bayes} to have Frobenius norm of 1, and thus $\angle \langle B, B_{\text{Bayes}} \rangle = \arccos(\langle B, B_{\text{Bayes}} \rangle_F)$ will be the angle used in this paper.

In all examples, α is set as 0.5, the reason for this is described in Section 4.2. We want to choose C in $\mathbb{R}^+$. For convenience, we use the log scale, and set $\log_2 C$ from -3 to 15. For the first two examples, we generate datasets that have signal based on only a few covariates, and then we add pure noise as additional covariates. To better compare the performance for both balanced and imbalanced scenarios, all the examples are conducted under both balance and imbalance cases. Let $p = \Pr(Y = 1)$ and $\Pr(Y = j) = \frac{1}{K-1}(1-p)$ for $j \neq 1$. We will consider $p = 1/K$ for the balanced case and $p = 1/2$ and $p = 1/3$ for the imbalanced case for all examples. The size of the training dataset for each example is 300, 600 and 300 respectively. In each example, five sets of dimensions are considered: 2, 10, 100, 500 and 1000. The noise covariates are identically independent distributed as $N(0, \sigma^2)$. For the third example, all covariates are signal variables. And for all simulation settings, we repeat the experiments 100 times and report the average performance.

4.1 Performance Comparison

Example 1 *We generate a three class dataset, where the first two covariates are distributed* $N(u_j, \sigma^2 I_2)$. *In this setting, the* u_j*'s are three points equally spaced on a unit circle, and* σ *is chosen such that the Bayes error is 0.1. As we can see, this example is similar to the Example 1 in Zhang and Liu (2014) other than that our case considered both the balanced and imbalanced scenarios.*

Example 2 *We generate a five class dataset, Let* $Pr(Y = 1) = p$, *and the first five covariates are distributed* $N(u_j, \sigma^2 I_5)$. *Here* u_j*'s are five points equally spaced on the sphere of unit ball in* $\mathbb{R}^4$, *and* $\sigma = 0.55$. *When dimension is larger than 4, the last* $d - 4$ *covariates are identically independent distributed as* $N(0, \sigma^2)$.

Example 3 *A three groups dataset is generated with dimension d, the centers of the three groups are equally distributed on the sphere of an unit ball in* $\mathbb{R}^d$. *A random noise* $N(0, \sigma^2 = 0.55^2)$ *was added to each dimension.*

We report the average prediction error rate and the average angle to the Bayes rule in Figures 2. Take Figure 2(a), which corresponds to Example 1, as an example. We report

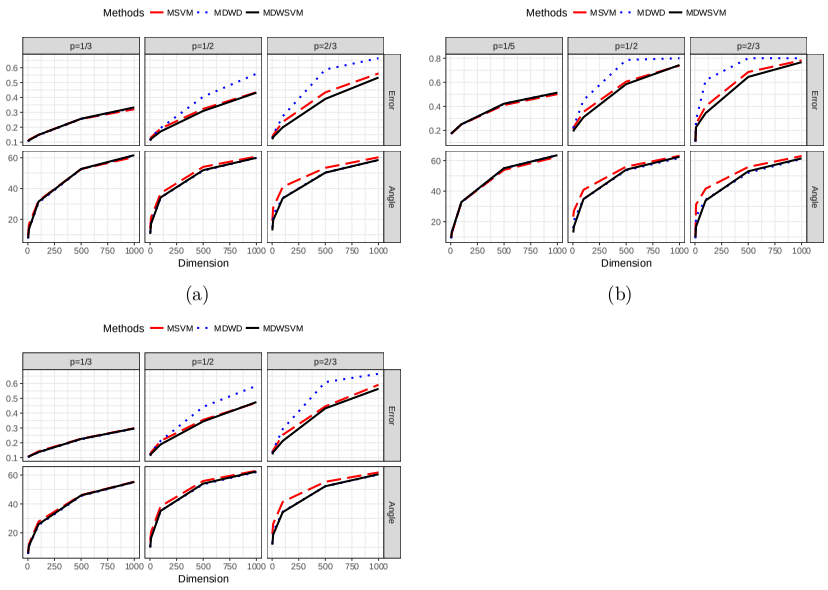


Figure 2: Performance comparison plot between the three methods for Examples 1, 2, and 3. The top row of each graph plots the misclassification rate for different dimensions (the x axis) and different prior probabilities (left, middle and right panels). The bottom row is the angle between the estimated and the Bayes rule. For all measures, smaller implies better result. We can see that our method (the solid black line) performs almost the same as the other two methods (MSVM, the red dashed line, MDWD, the dotted blue line) for balanced case, but outperforms the other two for the imbalanced cases.

both misclassification rate (the top row) and the angle between the estimated classifiers and the Bayes rule. In the plot, we use black solid lines for our MDWSVM method, red dashed lines for MSVM method, and blue dotted lines for MDWD method. The grid points on the x axis represents the different dimensions d. The y axis corresponds to the performance measure. In our plot, the smaller the y axis value, the better the performance. Different imbalance ratios are visualized in different panels from left to right. The left two panels correspond to the balanced case; the middle panels are mild imbalance ($p = 1/2$); and the right two panels are more severe imbalanced case ($p = 2/3$). For the balanced case, our approach performs similar to the MSVM and MDWD methods. However, for the imbalanced cases (the middle and right panels), we can see clear gaps between the performance of our method and the other two methods, demonstrating that our method outperforms the other two. Note that a similar pattern can also be seen in Figures 2(b) and 2(c). All suggests that the novel approach is better than MSVM and MDWD.

It is also shown in these plots that as the dimension of training data changes from small to large (2 to 1000), the classifier's performances become worse and worse. Furthermore, the performance differences of the three methods become more pronounced. Note that MDWD gives the worst prediction error rate compared to the other two methods, and MSVM gives the worst classification direction compared to the other two methods. Our MDWSVM gives comparably the best performance in both aspects.

4.2 Sensitivity to Parameters

There are two parameters C and α in our MDWSVM method. We have conducted many simulations to evaluate the performance of these two parameters. In this section, we will only use Example 1 to show the performance. We set α to be fixed, varied C, and evaluated its performance. Then we fixed an optimal C to evaluate the sensitivity of our approach to different α's. At the beginning, we let $\alpha = 0.5$, and allow C to change from 2^{-3} to 2^{12}. The simulation is conducted under different dimensions $(100, 500, 1000)$. All the simulation results are based on 100 replicates. The left panel of Figure 3 is the prediction error under different values of C with different dimensions of training data. It is clearly shown from the graph that as C increases, the prediction error rate first decreases and then increases. It shows that a minimal prediction error can be reached within this range. The right panel shows the relationship between prediction error rate and different α's. It is also clear that the prediction error rate stays the same as α changes from 0.1 to 0.9, regardless of the slightly increase as α approaches to 1.

Based on the performance from Figure 3, the change of α has little impact on the prediction error rate compared to a change of C. The performance is quite stable for different α's. Since the property of the parameters are not the focus in this paper, this simulation gives us an easy suggestion of choosing parameters. One can simply fix $\alpha = 0.5$ and use cross-validation to choose C. This is why we fix $\alpha = 0.5$ in our simulation.

4.3 Computation Time

In this subsection, we will compare the computational time for these methods (MDWSVM, MSVM and MDWD). To test the computational complexity, we only consider the simulation

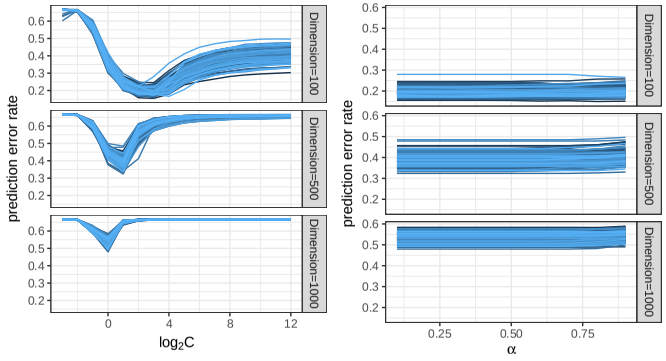


Figure 3: Average within-group error rate change under different parameters. Left panel is the prediction error rate change under different C value for fixed $\alpha = 0.5$; right panel is prediction error rate change for different α when C is fixed at its optimal value

Dimension	MDWSVM	MDWD	MSVM
10	8.31(0.04)	8.22(0.04)	0.88(0.01)
100	14.52(0.07)	12.80(0.05)	2.79(0.01)
1000	41.66(0.20)	27.43(0.12)	9.30(0.07)

Table 1: Computation time comparison for MDWSVM, MDWD and MSVM based on 100 runs of Example 1. The number shows average computing time in seconds, and the number in the parenthesis is the corresponding standard error.

of Example 1. We let the dimension change from 10 to 1000. Table 1 gives the average computation time in seconds for 100 replicates along with their standard error. All numerical experiments were carried out on an Intel Xeon E3-1284L (2.5 GHz) processor.

Table 1 shows that the most efficient method is MSVM and the most time-consuming method is our MDWSVM. Note that MSVM can still be viewed as a quadratic programming problem, while both MDWD and MDWSVM are conic program problems. It is not surprising that MSVM is the most computational efficient one. It is our expectation that MDWSVM would have the longest time to run, since it combines both MSVM and MDWD. From equation 6, we can see that the number of parameters can be viewed as the sum of the ones for MDWD and MSVM. Thus the computation times it takes to solve the problem increases as well. It is good enough that the computational time of MDWSVM is shorter than the sum of the computing time of each individual methods. It is worth mentioning that as dimension increases, the CPU times for the three methods increase.

2002/01/01	New Year	2002/09/02	Labor Day
2002/05/27	Memorial Day	2002/11/28	Thanksgiving
2002/07/04	Independence Day	2002/12/25	Christmas

Table 2: Six national holidays on weekdays that are removed

5. Real Data Application

In this section, we apply our MDWSVM method to a real data used in Shen and Huang (2005). The data were gathered at an inbound call center of a major northeastern U.S financial firm in 2002, and describe the call volume from 7:00am-12:00am. Each day is divided into 408 150-second intervals and the number of phone calls is recorded in each interval. Due to equipment malfunctioning, there are 6 missing weekdays within the whole year. The call volume data form a 360×408 matrix, where each row corresponds to a day and each column is the call volume for one of the 150-second intervals.

Note that the data have been thoroughly analyzed in Shen and Huang (2005). Here we simply add some new insights from the data by using our novel approach. According to their analysis, the pattern for Saturday and Sunday is very different from the weekdays. Thus in this analysis, we only focus on the weekdays. Shen and Huang (2005) show that for weekdays, by using singular value decomposition to analyze the number of phone calls, Monday and Friday are slightly different from all other weekdays, see Section 5.3 and Figure 6 of Shen and Huang (2005) for more details. Tuesday, Wednesday and Thursday are hard to tell apart from each other. In this section, we only focus on classifying Monday, Friday and other weekdays (Tuesday, Wednesday and Thursday). In addition, due to the fact that the center has very low volumes on national holidays, we remove the holidays that fall on weekdays. Table 2 provides the national holidays excluded in our analysis. These days, along with some other more holidays, were also removed in Shen and Huang (2005).

After removing these holidays, we have 48, 50, 51, 50, 52 days for Monday to Friday respectively. The data are divided into three groups, Group 1: Monday (size 48); Group 2: Tuesday to Thursday (size 151); Group 3: Friday (size 52). This dataset is a typical imbalanced HDLSS dataset with Group 2 as the dominant group. The average number of phone calls on each time interval are presented in Figure 4. From Figure 4, we can see that the average number of phone calls on Monday is quite distinct from the other days as it is larger than the other two groups. However, Group 2 and 3 are hard to distinguish from each other by only looking at the average number of phone calls. For this data set, we will compare the performance of three classifiers: our MDWSVM, MSVM, MDWD.

To obtain a good evaluation, all three methods use five-fold cross validation. And the evaluation measures are the total error rate (TER) and the average within-group error rate (AER). We also report the prediction error within each group. Table 3 provides these results for the three different methods.

From Table 3 it is straightforward to conclude that the prediction error of MDWSVM for each of the groups is the smallest, as well as the total error rate and average with-group prediction error rate. Our MDWSVM model works very well for this data set.

Taking another look at the Table 3, the prediction error rate for Monday and Friday are more than 30%, which seems to be large. The result could be explained by the fact

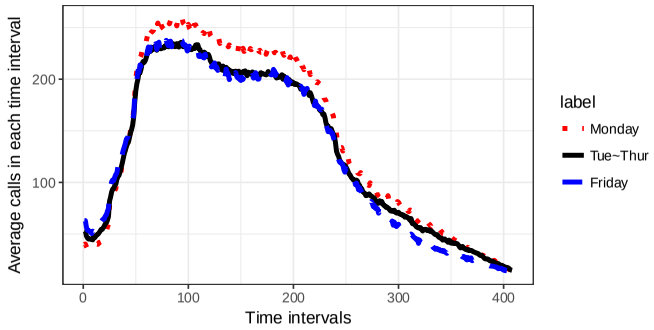


Figure 4: Average number of calls in each time interval for the three classes Monday, Tuesday to Thursday and Friday. In this plot, the red dotted line is the number of phone calls for Mondays, the black solid line is for Tuesday to Thursday, and the blue dashed line is for Friday.

	MDWD	MSVM	MDWSVM
Mon.	0.8156	0.6200	0.4111
Tue. - Thu.	0.0396	0.0594	0.0985
Fri.	0.6145	0.5527	0.3200
TER	0.3098	0.2702	0.2053
AER	0.4899	0.4107	0.2765

Table 3: Prediction error for number of phone calls. The top 3 rows report the cross-validation prediction error for each class, and the bottom two rows are the total error rate and the average within-group error rate.

that the DWSVM used here is a linear classifier, while the nature of the data may not be well classified by a linear classifier. If we incorporate a kernel approach to our classifier, the performance should improve.

6. Discussion

In this paper, we proposed a new angle-based method MDWSVM. We show in this paper that in HDLSS and the imbalanced case, our novel MDWSVM method has smaller misclassification error relative to both MDWD and MSVM. In addition, it is closer to the Bayes discriminant direction compared to that of MSVM. The MDWSVM can be viewed as an extension of the binary DWSVM to the multicategory case. Furthermore, as our model considers both SVM and DWD loss, it can be treated as a weighted hybridization of MSVM and MDWD.

One of the limitations of DWD-type methods is that it suffers from slow computation time compared to SVM-type methods (See Chapter 4.3 for examples). Recently an alternating direction method of multipliers (ADMM) algorithm has been implemented by Lam et al. (2016) for the DWD type methods, which can handle large size data more efficiently. A similar implementation for our MDWSVM method will be explored in the future.

Note that we only consider linear classifiers for simplicity. This method can be extended to a kernel approach (Liu and Yuan, 2011) by allowing $\boldsymbol{f}$ to be a nonlinear mapping. The implementation of such a generalization will be our immediate future work.

Our MDWSVM uses the squared norm as the regularization component so it does not have a variable selection property. To better deal with data of high dimension, variable selection penalties can be added to the model, *e.g.* LASSO (Tibshirani, 1996) or Elastic net (Zou and Hastie, 2005). Work on this type of generalization will follow.

Acknowledgments

We would like to thank the reviewers for the valuable comments and suggestions. Also thank Cheng Li (Purdue University) for the insightful discussion on proof of theorems.

Appendix A.

In this appendix we prove the following theorems and proposition from Section 3:

Lemma 1 *(Zhang and Liu, 2014): Suppose we have an arbitrary $\boldsymbol{f} \in \mathbb{R}^{K-1}$. For any $u, v \in \{1, \ldots, K\}$ such that $u \neq v$, define $T_{u,v} = W_u - W_v$. For any scalar $z \in \mathbb{R}$, $\langle (\boldsymbol{f} + zT_{u,v}), W_w \rangle = \langle \boldsymbol{f}, W_w \rangle$, where $w \in \{1, \ldots, K\}$ and $w \neq u, v$. Furthermore, we have that $\langle (\boldsymbol{f} + zT_{u,v}), W_u \rangle - \langle \boldsymbol{f}, W_u \rangle = -\langle (\boldsymbol{f} + zT_{u,v}), W_v \rangle + \langle \boldsymbol{f}, W_v \rangle$.* ■

The proof of Lemma 1 is given in Zhang and Liu (2014). From Lemma 1, one can see that for a given $\boldsymbol{f}$, if we move it along the direction of $T_{u,v}$, the inner product of $\boldsymbol{f}$ and W_w will stay the same when $w \neq u, v$. Furthermore, the sum of inner product $\langle \boldsymbol{f}, W_u \rangle + \langle \boldsymbol{f}, W_u \rangle, W_u \rangle$ will remain unchanged as well. This lemma will help us to prove the Fisher consistency of the MDWSVM method.

Theorem 1. *The MDWSVM is Fisher consistent for any $0 < \alpha < 1$.*

Proof. Recall the definition of $\boldsymbol{f}^*$ is that

$$(\boldsymbol{f}^*, \boldsymbol{f}_0^*) = \arg\min_{\boldsymbol{f}, \boldsymbol{f}_0} E[(1-\alpha)\ell_s\{\langle \boldsymbol{f}(X), W_Y \rangle\} + \alpha \ell_d\{\langle \boldsymbol{f}_0(X), W_Y \rangle\} | X = x].$$

We need to show that when $P_1 > P_2$, $\langle W_1, \boldsymbol{f}^* \rangle > \langle W_2, \boldsymbol{f}^* \rangle$. This can be easily proved by contradiction.

If $\langle W_1, \boldsymbol{f}^* \rangle = \langle W_2, \boldsymbol{f}^* \rangle$, Let $\boldsymbol{f}_0^* = \boldsymbol{f}^* - \Delta$, here we can see that as Δ is only the difference of intercept, which is independent of X. Let $(\boldsymbol{f}^{**}, \boldsymbol{f}_0^{**}) = (\boldsymbol{f}^{**}, \boldsymbol{f}_0^*)$ be such that $\langle W_j, \boldsymbol{f}^{**} \rangle = \langle W_j, \boldsymbol{f}^* \rangle$ for $j \geq 3$ and $\langle W_1, \boldsymbol{f}^{**} \rangle = \langle W_1, \boldsymbol{f}^* \rangle + \epsilon$, $\langle W_2, \boldsymbol{f}^{**} \rangle = \langle W_2, \boldsymbol{f}^* \rangle - \epsilon$. This $(\boldsymbol{f}^{**}, \boldsymbol{f}_0^{**})$ exists based on Lemma 1 and the fact that inner product is continuous. To get the required $\boldsymbol{f}^{**}$, we only need to move $\boldsymbol{f}^*$ along the direction of $T_{1,2}$.

Then it is easy to get

$$\begin{aligned}
&\sum_{j=1}^{K} P_j[(1-\alpha)\ell_s\{\langle \boldsymbol{f}^{**}, W_j \rangle\} + \alpha \ell_d\{\langle \boldsymbol{f}_0^{**}, W_j \rangle\}] \\
&- \sum_{j=1}^{K} P_j[(1-\alpha)\ell_s\{\langle \boldsymbol{f}^{*}, W_j \rangle\} + \alpha \ell_d\{\langle \boldsymbol{f}_0^{*}, W_j \rangle\}] \\
=&\epsilon(P_1 - P_2)(1-\alpha)\ell_s'\{\langle \boldsymbol{f}^*, W_1 \rangle\} + o(\epsilon)
\end{aligned}$$

Since we are using proximal hinge loss, ℓ_s is differentiable, $P_1 - P_2 > 0$, $\ell_s' < 0$ and $0 < \alpha < 1$. we have $\sum_{j=1}^{K} P_j[(1-\alpha)\ell_s\{\langle \boldsymbol{f}^{**}, W_j \rangle\} + \alpha \ell_d\{\langle \boldsymbol{f}_0^{**}, W_j \rangle\}] < \sum_{j=1}^{K} P_j[(1-\alpha)\ell_s\{\langle \boldsymbol{f}^{*}, W_j \rangle\} + \alpha \ell_d\{\langle \boldsymbol{f}_0^{*}, W_j \rangle\}]$, which is a contradiction.

For $\langle W_1, \boldsymbol{f}^* \rangle < \langle W_2, \boldsymbol{f}^* \rangle$ case, if $P_1\ell_s'\{\langle \boldsymbol{f}^*, W_1 \rangle\} - P_2\ell_s'\{\langle \boldsymbol{f}^*, W_2 \rangle\} < 0$, then choose $(\boldsymbol{f}^{**}, \boldsymbol{f}_0^{**}) = (\boldsymbol{f}^{**}, \boldsymbol{f}_0^*)$ be such that $\langle W_j, \boldsymbol{f}^{**} \rangle = \langle W_j, \boldsymbol{f}^* \rangle$ for $j \geq 3$ and $\langle W_1, \boldsymbol{f}^{**} \rangle = \langle W_1, \boldsymbol{f}^* \rangle + \epsilon$, $\langle W_2, \boldsymbol{f}^{**} \rangle = \langle W_2, \boldsymbol{f}^* \rangle - \epsilon$. Then we have

$$\begin{aligned}
&\sum_{j=1}^{K} P_j[(1-\alpha)\ell_s\{\langle \boldsymbol{f}^{**}, W_j \rangle\} + \alpha \ell_d\{\langle \boldsymbol{f}_0^{**}, W_j \rangle\}] \\
&- \sum_{j=1}^{K} P_j[(1-\alpha)\ell_s\{\langle \boldsymbol{f}^{*}, W_j \rangle\} + \alpha \ell_d\{\langle \boldsymbol{f}_0^{*}, W_j \rangle\}] \\
=&\epsilon(1-\alpha)\{P_1\ell_s'\{\langle \boldsymbol{f}^*, W_1 \rangle\} - P_2\ell_s'\{\langle \boldsymbol{f}^*, W_2 \rangle\}\} + o(\epsilon) < 0
\end{aligned}$$

We can see that If $P_1\ell_s'\{\langle \boldsymbol{f}^*, W_1 \rangle\} - P_2\ell_s'\{\langle \boldsymbol{f}^*, W_2 \rangle\} > 0$, then choose $(\boldsymbol{f}^{**}, \boldsymbol{f}_0^{**}) = (\boldsymbol{f}^{**}, \boldsymbol{f}_0^*)$ be such that $\langle W_j, \boldsymbol{f}^{**} \rangle = \langle W_j, \boldsymbol{f}^* \rangle$ for $j \geq 3$ and $\langle W_1, \boldsymbol{f}^{**} \rangle = \langle W_1, \boldsymbol{f}^* \rangle - \epsilon$, $\langle W_2, \boldsymbol{f}^{**} \rangle = \langle W_2, \boldsymbol{f}^* \rangle + \epsilon$. Then we have

$$\begin{aligned}
&\sum_{j=1}^{K} P_j[(1-\alpha)\ell_s\{\langle \boldsymbol{f}^{**}, W_j \rangle\} + \alpha \ell_d\{\langle \boldsymbol{f}_0^{**}, W_j \rangle\}] \\
&- \sum_{j=1}^{K} P_j[(1-\alpha)\ell_s\{\langle \boldsymbol{f}^{*}, W_j \rangle\} + \alpha \ell_d\{\langle \boldsymbol{f}_0^{*}, W_j \rangle\}] \\
=&\epsilon(1-\alpha)\{-P_1\ell_s'\{\langle \boldsymbol{f}^*, W_1 \rangle\} + P_2\ell_s'\{\langle \boldsymbol{f}^*, W_2 \rangle\}\} + o(\epsilon) < 0
\end{aligned}$$

We can see that this is a contradiction. This completes the proof. ■

Proposition 2. *In MSVM setting, when the size of the majority class goes to infinity, $\langle \beta_0, W_{y_K} \rangle < \sqrt{2C}K \max |x_{ij}| + 1$.*

Proof. Assume observations $1, \ldots, K-1$ belong to the class $1, \ldots, K-1$ respectively, and observations $K, \ldots, n$ belong to class K.

$$\begin{aligned}
\text{Loss} =& \sum_{i=1}^{n} \ell_s\{\langle f(x_i), W_{y_i} \rangle\} \\
=& \sum_{i=1}^{K-1} \ell_s\{\langle f(x_i), W_i \rangle\} + \sum_{i=K}^{n} \ell_s\{\langle f(x_i), W_K \rangle\} \\
=& \sum_{i=1}^{K-1} \ell_s\{\langle x_i^T B, W_i \rangle + \langle \beta_0, W_i \rangle\} + \sum_{i=K}^{n} \ell_s\{\langle x_i^T B, W_K \rangle + \langle \beta_0, W_K \rangle\}
\end{aligned}$$

Now we prove that $\forall B \in \mathbb{R}^{p \times (K-1)}$, we have

$$\langle \beta_0, W_K \rangle < \sup_i |\langle x_i^T B, W_i \rangle| K + 1 < \sqrt{2C}K \max |x_{ij}| + 1.$$

We can use contradiction to prove it, if $\langle \beta_0, W_K \rangle > \sup_i |\langle x_i^T B, W_i \rangle| K + 1$, then

$$\ell_s\{\langle x_i^T B, W_K \rangle + \langle \beta_0, W_K \rangle\} = 0$$

for all $i \in \{K, \ldots, n\}$ as $\langle x_i^T B, W_K \rangle + \langle \beta_0, W_K \rangle > 1$.

$$\text{Loss} = \sum_{i=1}^{n} \ell_s\{\langle f(x_i), W_{y_i}\rangle\} = \sum_{i=1}^{K-1} \ell_s\{\langle x_i^T B, W_i\rangle + \langle \beta_0, W_i\rangle\}.$$

Then $\frac{dL}{d\beta_0} = \sum_{i=1}^{K-1} l_s'\{\langle x_i^T B, W_i\rangle + \langle \beta_0, W_i\rangle\} W_i'$. Since $\langle \beta_0, W_K\rangle > \sup_i |\langle x_i^T B, W_i\rangle| K + 1$, one can get that $u_K = \langle x_K^T B, W_K\rangle + \langle \beta_0, W_K\rangle > 1$. Based on the property of W, we have $\sum_{i=1}^{K} \langle \beta_0, W_i\rangle = 0$. Furthermore, it is easy to deduct that $\min\langle \beta_0, W_i\rangle < -\sup_i |\langle x_i^T B, W_i\rangle|$ for $i \in \{1, \ldots, K-1\}$. Then $\min u_i = \langle x_i^T B, W_i\rangle + \min\langle \beta_0, W_i\rangle < 0$ for $i = 1, \ldots, K-1$. Thus we can choose $K-1$ different values $K_1, K_2, , \ldots, K_{K-1}$ from $1, \ldots, K-1$ such that $u_{K_1} \geq u_{K_2} \geq \ldots \geq 0 \geq \ldots \geq u_{K_{K-1}}$. Assume $i_0 = \max\{i, u_{K_i} < 1\}$, then $\frac{dL}{d\beta_0} = -\sum_{i=K_{K-1}}^{K_{i_0}} W_{K_i}'$. One can simply verify that $\frac{dL}{d\beta_0} \neq 0$ based on the property of vertex W. Thus β_0 cannot be the β_0 that minimize the loss function given B. This step completes the prove. ■

Theorem 3. *In the MDWSVM setting, when the size of the majority class goes to infinity,* $\langle \beta_0, W_{y_K}\rangle < \sqrt{2C} K \max |x_{ij}| + 1$.

Proof. Based on the proof of Proposition 2, for any $\forall B \in \mathbb{R}^{p\times(K-1)}$, we have

$$\langle \beta_0, W_K\rangle < \sup_i |\langle x_i^T B, W_i\rangle| K + 1 < \sqrt{2C} K \max |x_{ij}| + 1.$$

Thus for MDWSVM model, no matter what B we get, the intercept only comes from MSVM part. Therefore, from the conclusion of Proposition 2, Theorem 3 is proved. ■

References

Jeongyoun Ahn and JS Marron. The maximal data piling direction for discrimination. *Biometrika*, 97(1):254–259, 2010.

Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1(Dec): 113–141, 2000.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5): 1055–1064, 1999.

Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

Kai-Bo Duan, Jagath C Rajapakse, Haiying Wang, and Francisco Azuaje. Multiple svm-rfe for gene selection in cancer classification with expression data. *IEEE Transactions on Nanobioscience*, 4(3):228–234, 2005.

Giles M Foody and Ajay Mathur. The use of small training sets containing mixed pixels for accurate hard image classification: Training on mixed spectral responses for classification by a svm. *Remote Sensing of Environment*, 103(2):179–189, 2006.

Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx: Matlab software for disciplined convex programming, 2008.

Trevor Hastie, Robert Tibshirani, et al. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471, 1998.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The elements of statistical learning. 2001. *Springer*, 2001.

Hanwen Huang, Yufeng Liu, Ying Du, Charles M Perou, D Neil Hayes, Michael J Todd, and James Stephen Marron. Multiclass distance-weighted discrimination. *Journal of Computational and Graphical Statistics*, 22(4):953–969, 2013.

Xin Yee Lam, JS Marron, Defeng Sun, and Kim-Chuan Toh. Fast algorithms for large scale generalized distance weighted discrimination. *arXiv preprint arXiv:1604.05473*, 2016.

Kenneth Lange and Tongtong Wu. An mm algorithm for multicategory vertex discriminant analysis. *Journal of Computational and Graphical Statistics*, 17(3):527–544, 2008.

Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

Yufeng Liu. Fisher consistency of multicategory support vector machines. In *AISTATS*, pages 291–298, 2007.

Yufeng Liu and Xiaotong Shen. Multicategory ψ-learning. *Journal of the American Statistical Association*, 101(474):500–509, 2006.

Yufeng Liu and Ming Yuan. Reinforced multicategory support vector machines. *Journal of Computational and Graphical Statistics*, 20(4):901–919, 2011.

Yufeng Liu, Hao Helen Zhang, and Yichao Wu. Hard or soft classification? large-margin unified machines. *Journal of the American Statistical Association*, 106(493):166–177, 2011.

JS Marron, Michael J Todd, and Jeongyoun Ahn. Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271, 2007.

Art B Owen. Infinitely imbalanced logistic regression. *Journal of Machine Learning Research*, 8(Apr):761–773, 2007.

Xingye Qiao and Yufeng Liu. Adaptive weighted learning for unbalanced multicategory classification. *Biometrics*, 65(1):159–168, 2009.

Xingye Qiao and Lingsong Zhang. Distance-weighted support vector machine. *Statistics and Its Interface*, 8(3):331–345, 2015a.

Xingye Qiao and Lingsong Zhang. Flexible high-dimensional classification machines and their asymptotic properties. *Journal of Machine Learning Research*, 16:1547–1572, 2015b.

Xingye Qiao, Hao Helen Zhang, Yufeng Liu, Michael J Todd, and James Stephen Marron. Weighted distance weighted discrimination and its asymptotic properties. *Journal of the American Statistical Association*, 105(489):401–414, 2010.

Bernhard Schölkopf and Christopher JC Burges. *Advances in kernel methods: support vector learning*. MIT press, 1999.

Haipeng Shen and Jianhua Z. Huang. Analysis of call centre arrival data using singular value decomposition. *Applied Stochastic Models in Business and Industry*, 21(3):251–263, 2005.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Vladimir Naumovich Vapnik and Vlamimir Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

Hui Wang and Gang Huang. Application of support vector machine in cancer diagnosis. *Medical Oncology*, 28(1):613–618, 2011.

Chong Zhang and Yufeng Liu. Multicategory angle-based large-margin classification. *Biometrika*, 101(3):625–640, 2014.

Ji Zhu and Trevor Hastie. Kernel logistic regression and the import vector machine. In *Advances in Neural Information Processing Systems*, pages 1081–1088, 2001.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Journal of Machine Learning Research 18 (2017) 1-47 Submitted 1/17; Revised 7/17; Published 7/17

Minimax Estimation of Kernel Mean Embeddings

Ilya Tolstikhin ILYA@TUEBINGEN.MPG.DE
[†]*Department of Empirical Inference*
Max Planck Institute for Intelligent Systems
Spemanstraße 38, Tübingen 72076, Germany

Bharath K. Sriperumbudur BKS18@PSU.EDU
Department of Statistics
Pennsylvania State University
University Park, PA 16802, USA

Krikamol Muandet[†] KRIKAMOL@TUEBINGEN.MPG.DE
Department of Mathematics
Faculty of Science, Mahidol University
272 Rama VI Rd. Rajchathevi, Bangkok 10400, Thailand

Editor: Andreas Christmann

Abstract

In this paper, we study the minimax estimation of the Bochner integral

$$\mu_k(P) := \int_{\mathcal{X}} k(\cdot, x)\, dP(x),$$

also called as the *kernel mean embedding*, based on random samples drawn i.i.d. from P, where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel. Various estimators (including the empirical estimator), $\hat{\theta}_n$ of $\mu_k(P)$ are studied in the literature wherein all of them satisfy $\|\hat{\theta}_n - \mu_k(P)\|_{\mathcal{H}_k} = O_P(n^{-1/2})$ with $\mathcal{H}_k$ being the reproducing kernel Hilbert space induced by k. The main contribution of the paper is in showing that the above mentioned rate of $n^{-1/2}$ is minimax in $\|\cdot\|_{\mathcal{H}_k}$ and $\|\cdot\|_{L^2(\mathbb{R}^d)}$-norms over the class of discrete measures and the class of measures that has an infinitely differentiable density, with k being a continuous translation-invariant kernel on $\mathbb{R}^d$. The interesting aspect of this result is that the minimax rate is independent of the smoothness of the kernel and the density of P (if it exists).

Keywords: Bochner integral, Bochner's theorem, kernel mean embeddings, minimax lower bounds, reproducing kernel Hilbert space, translation invariant kernel

1. Introduction

Over the last few years, kernel embedding of distributions (Smola et al., 2007; Sriperumbudur et al., 2010) has gained a lot of attention in the machine learning community due to the wide variety of applications it has been employed in. Some of these applications include kernel two-sample testing (Gretton et al., 2007, 2012), kernel independence and conditional independence tests (Gretton et al., 2008; Fukumizu et al., 2008), covariate-shift (Smola et al., 2007), density estimation (Sriperumbudur, 2011), feature selection (Song et al., 2012), causal inference (Lopez-Paz et al., 2015), kernel Bayes' rule (Fukumizu et al., 2013) and distribution regression (Szabó et al., 2015).

©2017 Ilya Tolstikhin, Bharath K. Sriperumbudur, and Krikamol Muandet.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-032.html.

Formally, let $\mathcal{H}_k$ be a separable reproducing kernel Hilbert space (RKHS) (Aronszajn, 1950) with a continuous reproducing kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defined on a separable topological space $\mathcal{X}$. Given a Borel probability measure P defined over $\mathcal{X}$ such that $\int_{\mathcal{X}} \sqrt{k(x,x)}\, dP(x) < \infty$, the kernel mean or the mean element is defined as the Bochner integral

$$\mu_P := \int_{\mathcal{X}} k(\cdot, x)\, dP(x) \in \mathcal{H}_k. \tag{1}$$

We refer the reader to Diestel and Uhl (1977, Chapter 2) and Dinculeanu (2000, Chapter 1) for the definition of a Bochner integral. The mean element in (1) can be viewed as an embedding of P in $\mathcal{H}_k$,

$$\mu_k : M_+^1(\mathcal{X}) \to \mathcal{H}_k, \qquad \mu_k(P) = \mu_P,$$

where $M_+^1(\mathcal{X})$ denotes the set of all Borel probability measures on $\mathcal{X}$. Hence, we also refer to μ_k as the *kernel mean embedding* (KME). The mean embedding can be seen as a generalization of the classical kernel feature map that embeds points of an input space $\mathcal{X}$ as elements in $\mathcal{H}_k$. The mean embedding μ_k can also be seen as a generalization of the classical notions of characteristic function, moment generation function (if it exists), and Weierstrass transform of P (all defined on $\mathbb{R}^d$) to an arbitrary topological space $\mathcal{X}$ as the choice of $k(\cdot, x)$ as $(2\pi)^{-d/2} e^{-\sqrt{-1}\langle \cdot, x\rangle}$, $e^{\langle \cdot, x\rangle}$, and $(4\pi)^{-d/2} e^{-\|\cdot - x\|_2^2}$, $x \in \mathbb{R}^d$ respectively reduces μ_k to these notions. The mean embedding μ_k is closely related to the *maximum mean discrepancy* (MMD) (Gretton et al., 2007), which is the RKHS distance between the mean embeddings of two probability measures. We refer the reader to (Sriperumbudur et al., 2010; Sriperumbudur, 2016) for more details on the properties of μ_k and the corresponding MMD.

In all the above mentioned statistical and machine learning applications, since the underlying distribution P is known only through random samples $X_1, \ldots, X_n$ drawn i.i.d. from it, an estimator of μ_P is employed. The goal of this paper is to study the minimax optimal estimation of μ_P. In the literature, various estimators of μ_P have been proposed. The simplest and most popular is the empirical estimator μ_{P_n}, which is constructed by replacing P by its empirical counterpart, $P_n := \frac{1}{n}\sum_{i=1}^n \delta_{X_i}$, where δ_x denotes a Dirac measure at $x \in \mathcal{X}$. In fact, all the above mentioned applications deal with the empirical estimator of μ_P because of its simplicity. Using Bernstein's inequality in separable Hilbert spaces (Yurinsky, 1995, Theorem 3.3.4), it follows that for bounded continuous kernels, $\|\mu_{P_n} - \mu_P\|_{\mathcal{H}_k} = O_P(n^{-1/2})$ for any P, i.e., the empirical estimator is a $\sqrt{n}$-consistent estimator of μ_P in $\mathcal{H}_k$-norm. This result is also proved in Smola et al. (2007, Theorem 2), Gretton et al. (2012), and Lopez-Paz et al. (2015) using McDiarmid's inequality, which we improve in Proposition A.1 (also see Remark A.2 in Appendix A) by providing a better constant. Assuming $\mathcal{X} = \mathbb{R}^d$ and P to have a density p, Sriperumbudur (2016, Theorem 4.1) proposed to estimate $\mu_P = \int_{\mathbb{R}^d} k(\cdot, x)p(x)\, dx$ by replacing p with a kernel density estimator, which is then shown to be $\sqrt{n}$-consistent in $\mathcal{H}_k$-norm if k is a bounded continuous *translation invariant kernel*—see Section 2 for its definition—on $\mathbb{R}^d$. Recently, Muandet et al. (2016, Section 2.4, Theorem 7) proposed a non-parametric shrinkage estimator of μ_P and established its $\sqrt{n}$-consistency in $\mathcal{H}_k$-norm for bounded continuous kernels on $\mathcal{X}$. Muandet et al. (2016, Section 3, Theorem 10) also proposed a penalized M-estimator for μ_P

where the penalization parameter is computed in a completely data-driven manner using leave-one-out cross validation and showed that it is also $\sqrt{n}$-consistent in $\mathcal{H}_k$-norm. In fact, the $\sqrt{n}$-consistency of all these estimators is established by showing that they are all within a $\|\cdot\|_{\mathcal{H}_k}$-ball of size $o_P(n^{-1/2})$ around the empirical estimator μ_{P_n}.

In the above discussion, it is important to note that the convergence rate of μ_{P_n} (and also other estimators) to μ_P in $\mathcal{H}_k$-norm does not depend on the smoothness of k or the density, p (if it exists). Under some mild conditions on the kernel (defined on $\mathbb{R}^d$), it can be shown (see Section 4) that $\mathcal{H}_k$ is continuously included in $L^2(\mathbb{R}^d)$ and $\|f\|_{L^2(\mathbb{R}^d)} \leq c_k \|f\|_{\mathcal{H}_k}$ for all $f \in \mathcal{H}_k$, where c_k is a constant that depends only on the kernel. This means, $\|\cdot\|_{L^2(\mathbb{R}^d)}$ is a weaker norm than $\|\cdot\|_{\mathcal{H}_k}$ and therefore it could be possible that μ_{P_n} converges to μ_P in $L^2(\mathbb{R}^d)$ at a rate faster than $n^{-1/2}$ (depending on the smoothness of k). In Proposition A.1 (also see Remark A.3 in Appendix A) we show that $\|\mu_{P_n} - \mu_P\|_{L^2(\mathbb{R}^d)} = O_P(n^{-1/2})$. Now given these results, it is of interest to understand whether these rates are optimal in a minimax sense, i.e., whether the above mentioned estimators are minimax rate optimal or can they be improved upon? Therefore the goal of this work is to obtain minimax rates for the estimation of μ_P in $\|\cdot\|_{\mathcal{H}_k}$ and $\|\cdot\|_{L^2(\mathbb{R}^d)}$.

Formally, we would like to find the minimax rate $r_{n,k}(\mathcal{F}, \mathcal{P})$ and a positive constant $c_k(\mathcal{F}, \mathcal{P})$ (independent of n) such that

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ r_{n,k}^{-1}(\mathcal{F}, \mathcal{P}) \|\hat{\theta}_n - \mu_P\|_{\mathcal{F}} \geq c_k(\mathcal{F}, \mathcal{P}) \right\} > 0, \tag{2}$$

where $\mathcal{F}$ is either $\mathcal{H}_k$ or $L^2(\mathbb{R}^d)$, $\mathcal{P}$ is a suitable subset of Borel probability measures on $\mathcal{X}$, and the infimum is taken over all estimators $\hat{\theta}_n$ mapping the i.i.d. sample $X_1, \ldots, X_n$ to $\mathcal{F}$. Suppose $k(x, y) = \langle x, y \rangle$, $x, y \in \mathbb{R}^d$. Norms $\|\cdot\|_{\mathcal{H}_k}$ and $\|\cdot\|_{L^2(\mathbb{R}^d)}$ match for this choice of k and the corresponding RKHS is finite dimensional, i.e., $\mathcal{H}_k = \mathbb{R}^d$. For a distribution P on $\mathbb{R}^d$ satisfying $\int_{\mathbb{R}^d} \|x\|_2 \, dP(x) < \infty$, this choice of kernel yields $\mu_P = \int x \, dP(x)$ as the mean embedding of P which simply is the mean of P. It is well-known (Lehmann and Casella, 2008, Chapter 5, Example 1.14) that the minimax rate of estimating $\mu_P \in \mathbb{R}^d$ based on $(X_i)_{i=1}^n$ is $r_{n,k}(\mathcal{F}, \mathcal{P}) = n^{-1/2}$ for the class $\mathcal{P}$ of Gaussian distributions on $\mathbb{R}^d$. In fact, this rate is attained by the empirical estimator $\mu_{P_n} = \frac{1}{n} \sum_{i=1}^n X_i$, which is the sample mean. Based on this observation, while one can intuitively argue that the minimax rate of estimating μ_P is $n^{-1/2}$ even if $\mathcal{H}_k$ is an infinite dimensional RKHS, it is difficult to extend the finite dimensional argument in a rigorous manner to the estimation of the infinite dimensional object, μ_P. In this paper, through a key inequality—see (3)— we rigorously show that it is indeed the case.

The main result of the paper is that if k is translation invariant on $\mathcal{X} = \mathbb{R}^d$ (see Theorems 1 and 9 for precise conditions on the kernel) and $\mathcal{P}$ is the set of all Borel discrete probability measures on $\mathbb{R}^d$, then the minimax rate $r_{n,k}(\mathcal{F}, \mathcal{P})$ is $n^{-1/2}$ for both $\mathcal{F} = \mathcal{H}_k$ and $\mathcal{F} = L^2(\mathbb{R}^d)$. Next, we show in Theorems 6 and 12 that the minimax rate for the estimation of μ_P in both $\|\cdot\|_{\mathcal{H}_k}$ and $\|\cdot\|_{L^2(\mathbb{R}^d)}$ still remains $n^{-1/2}$ even when $\mathcal{P}$ is restricted to the class of Borel probability measures which have densities, p that are continuously infinitely differentiable. The reason for considering such a class of distributions with smooth densities is that μ_P, which is the convolution of k and p, is smoother than k. Therefore one might wonder if it could be possible to estimate μ_P at a rate faster than $n^{-1/2}$ that depends on

the smoothness of k and p. Our result establishes that even for the class of distributions with very smooth densities, the minimax rate is independent of the smoothness of k and the density of P. The key ingredient in the proofs of Theorems 6 and 12 is the non-trivial inequality (see Proposition 3)

$$\|\mu_{G_0} - \mu_{G_1}\|_{\mathcal{F}} \geq c'_{k,\sigma^2} \|\tau_0 - \tau_1\|_2, \tag{3}$$

which relates the $\mathcal{F}$-distance between the mean embeddings of the Gaussian distributions, $G_0 = N(\tau_0, \sigma^2 I)$ and $G_1 = N(\tau_1, \sigma^2 I)$ to the Euclidean distance between the means of these Gaussians, where c'_{k,σ^2} is a constant that depends only on σ^2 and the translation invariant characteristic kernel k. Combining (3) with Le Cam's method (see Appendix B) implies that the estimation of an infinite dimensional object μ_P is as hard as the estimation of finite dimensional mean of a Gaussian distribution, thereby establishing the minimax rate to be $n^{-1/2}$. These results show that the empirical estimator—and other estimators we discussed above—of μ_P is minimax rate optimal.

Ramdas et al. (2015, Corollary 1) derived a special case of (3) for the Gaussian kernel k by ignoring small terms in the Taylor series expansion of $\|\mu_{G_0} - \mu_{G_1}\|_{\mathcal{H}_k}$ (refer to Remark 4). They used this result to show that the MMD between G_0 and G_1 decreases to zero exponentially/polynomially fast in d even when the Kullback-Leibler divergence between the two is kept constant, which in turn sheds some light on the decaying power of MMD-based hypothesis tests in high dimensions. Proposition 3 is more general, as it holds for *any* translation-invariant kernel k and does not require a truncation of small reminder terms.

The paper is organized as follows. Various notations used throughout the paper and definitions are collected in Section 2. The main results on minimax estimation of μ_P in $\|\cdot\|_{\mathcal{H}_k}$ and $\|\cdot\|_{L^2(\mathbb{R}^d)}$ for translation invariant kernels (and also *radial* kernels) on $\mathbb{R}^d$ are presented in Sections 3 and 4 respectively. The proofs of the results are provided in Section 5 while some supplementary results needed in the proofs are collected in appendices.

2. Definitions & Notation

Define $\|a\|_2 := \sqrt{\sum_{i=1}^d a_i^2}$ and $\langle a, b \rangle := \sum_{i=1}^d a_i b_i$, where $a := (a_1, \ldots, a_d) \in \mathbb{R}^d$ and $b := (b_1, \ldots, b_d) \in \mathbb{R}^d$. $C(\mathbb{R}^d)$ (*resp.* $C_b(\mathbb{R}^d)$) denotes the space of all continuous (*resp.* bounded continuous) functions on $\mathbb{R}^d$. $f \in C(\mathbb{R}^d)$ is said to *vanish at infinity* if for every $\epsilon > 0$ the set $\{x : |f(x)| \geq \epsilon\}$ is compact. The class of all continuous f on $\mathbb{R}^d$ which vanish at infinity is denoted as $C_0(\mathbb{R}^d)$. For $f \in C_b(\mathbb{R}^d)$, $\|f\|_\infty := \sup_{x \in \mathbb{R}^d} |f(x)|$ denotes the supremum norm of f. $M_b(\mathbb{R}^d)$ (*resp.* $M_+^b(\mathbb{R}^d)$) denotes the set of all finite (*resp.* finite non-negative) Borel measures on $\mathbb{R}^d$. $\text{supp}(\mu)$ denotes the support of $\mu \in M_b(\mathbb{R}^d)$ which is defined as $\text{supp}(\mu) = \{x \in \mathbb{R}^d \mid \text{for any open set } U \text{ such that } x \in U, |\mu|(U) \neq 0\}$, where $|\mu|$ is the total-variation of μ. $M_+^1(\mathbb{R}^d)$ denotes the set of Borel probability measures on $\mathbb{R}^d$. For $\mu \in M_+^b(\mathbb{R}^d)$, $L^r(\mathbb{R}^d, \mu)$ denotes the Banach space of r-power ($r \geq 1$) μ-integrable functions and we will use $L^r(\mathbb{R}^d)$ for $L^r(\mathbb{R}^d, \mu)$ if μ is a Lebesgue measure on $\mathbb{R}^d$. For $f \in L^r(\mathbb{R}^d, \mu)$, $\|f\|_{L^r(\mathbb{R}^d, \mu)} := \left(\int_{\mathbb{R}^d} |f|^r \, d\mu \right)^{1/r}$ denotes the L^r-norm of f for $1 \leq r < \infty$ and we denote it as $\|\cdot\|_{L^r(\mathbb{R}^d)}$ if μ is the Lebesgue measure. The convolution $f * g$ of two measurable functions

f and g on $\mathbb{R}^d$ is defined as

$$(f * g)(x) := \int_{\mathbb{R}^d} f(y)g(x-y)\,dy,$$

provided the integral exists for all $x \in \mathbb{R}^d$. The Fourier transforms of $f \in L^1(\mathbb{R}^d)$ and $\mu \in M_b(\mathbb{R}^d)$ are defined as

$$f^\wedge(y) := \mathscr{F}[f](y) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} f(x)\, e^{-i\langle y,x\rangle}\,dx, \qquad y \in \mathbb{R}^d$$

and

$$\mu^\wedge(y) := \mathscr{F}[\mu](y) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} e^{-i\langle y,x\rangle}\,d\mu(x), \qquad y \in \mathbb{R}^d$$

respectively, where i denotes the imaginary unit $\sqrt{-1}$.

A kernel $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is called *translation invariant* if there exists a symmetric positive definite function, ψ such that $k(x,y) = \psi(x-y)$ for all $x, y \in \mathbb{R}^d$. Bochner's theorem (see Wendland, 2005, Theorem 6.6) provides a complete characterization for a positive definite function ψ: A continuous function $\psi\colon \mathbb{R}^d \to \mathbb{R}$ is positive definite if and only if it is the Fourier transform of $\Lambda_\psi \in M_+^b(\mathbb{R}^d)$, i.e.,

$$\psi(x) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-i\langle x,w\rangle} d\Lambda_\psi(w), \quad x \in \mathbb{R}^d. \tag{4}$$

A kernel k is called *radial* if there exists $\phi : \mathbb{R}_+ \to \mathbb{R}$ such that $k(x,y) = \phi(\|x-y\|_2^2)$ for all $x, y \in \mathbb{R}^d$. From Schönberg's representation (Schoenberg, 1938; Wendland, 2005, Theorems 7.13 & 7.14) it is known that a kernel k is radial on every $\mathbb{R}^d$ if and only if there exists $\nu \in M_+^b([0,\infty))$ such that the following holds for all $x, y \in \mathbb{R}^d$:

$$k(x,y) = \phi(\|x-y\|^2) = \int_0^\infty e^{-t\|x-y\|^2} d\nu(t). \tag{5}$$

Some examples of reproducing kernels on $\mathbb{R}^d$ (in fact all these are radial) that appear throughout the paper are:

1. *Gaussian:* $k(x,y) = \exp\left(-\frac{\|x-y\|_2^2}{2\eta^2}\right)$, $\eta > 0$;
2. *Mixture of Gaussians:* $k(x,y) = \sum_{i=1}^M \beta_i \exp\left(-\frac{\|x-y\|_2^2}{2\eta_i^2}\right)$, where $M \geq 2$, $\eta_1^2 \geq \eta_2^2 \geq \cdots \geq \eta_M^2 > 0$, and positive constants $\beta_1, \ldots, \beta_M$ such that $\sum_{i=1}^M \beta_i = C_M < \infty$;
3. *Inverse Multiquadrics:* $k(x,y) = (c^2 + \|x-y\|_2^2)^{-\gamma}$, $c, \gamma > 0$;
4. *Matérn:* $k(x,y) = \frac{c^{2\tau-d}}{\Gamma(\tau-\frac{d}{2})2^{\tau-1-d/2}} \left(\frac{\|x-y\|_2}{c}\right)^{\tau-\frac{d}{2}} \mathcal{K}_{\frac{d}{2}-\tau}(c\|x-y\|_2)$, $\tau > d/2$, $c > 0$, where $\mathcal{K}_\alpha$ is the *modified Bessel function of the third kind* of order α and Γ is the Gamma function.

A kernel k is said to be *characteristic* if the mean embedding, $\mu_k : P \to \mu_P$ is injective, where μ_P is defined in (1). Various characterizations for the injectivity of μ_k (or k being characteristic) are known in literature (for details, see Sriperumbudur et al., 2011 and references therein). If k is a bounded continuous translation invariant positive definite kernel on $\mathbb{R}^d$, a simple characterization can be obtained for it to be characteristic (Sriperumbudur et al., 2010, Theorem 9): k is characteristic if and only if $\operatorname{supp}(\Lambda_\psi) = \mathbb{R}^d$ where Λ_ψ is defined in (4). This characterization implies that the above mentioned examples are characteristic kernels. Examples of non-characteristic kernels of translation invariant type include $k(x,y) = \frac{\sin(x-y)}{x-y}$, $x, y \in \mathbb{R}$ and $k(x,y) = \cos(x-y)$, $x, y \in \mathbb{R}$. More generally, polynomial kernels of any finite order are non-characteristic.

3. Minimax Estimation of μ_P in the RKHS Norm

In this section, we present our main results related to the minimax estimation of kernel mean embeddings (KMEs) in the RKHS norm. As discussed in Section 1, various estimators of $\mu_k(P)$ are known in literature and all these have a convergence rate of $n^{-1/2}$ if the kernel is bounded. The main goal of this section is to show that the rate $n^{-1/2}$ is actually minimax optimal for different choices of $\mathcal{P}$ (see (2)) under some mild conditions on k.

First, choosing $\mathcal{P}$ to be the set of all discrete probability measures on $\mathbb{R}^d$, in Section 3.1 (see Theorem 1 and Corollary 2), we present the minimax lower bounds of order $\Omega(n^{-1/2})$ with constant factors depending only on the properties of the kernel for translation invariant and radial kernels respectively. Next we will show in Section 3.2 that the rate $n^{-1/2}$ remains minimax optimal for translation invariant and radial kernels even if we choose the class $\mathcal{P}$ to contain only probability distributions with infinitely continuously differentiable densities. For translation invariant kernels the result (see Theorem 6) is based on a key inequality, which relates the RKHS distance between embeddings of Gaussian distributions to the Euclidean distance between the mean vectors of these distributions (see Proposition 3). The minimax lower bound for radial kernels (see Theorem 8) is derived using a slightly different argument. Instead of applying the bound of Theorem 6 to the particular case of radial kernels, we will present a direct analysis based on the special properties of radial kernels. This will lead us to the lower bound with almost optimal constant factors, depending only on the shape of Borel measure ν corresponding to the kernel.

Our analysis is based on the following simple idea: if a kernel k is characteristic, there is a one-to-one correspondence between any given set of Borel probability measures $\mathcal{P}$ defined over $\mathbb{R}^d$ and a set $\mu_k(\mathcal{P})$ of their embeddings into the RKHS $\mathcal{H}_k$. This means that distributions in $\mathcal{P}$ are indexed by their embeddings $\Theta := \mu_k(\mathcal{P})$ and so (2) can be equivalently written as

$$\inf_{\hat{\theta}_n} \sup_{\theta \in \Theta} \mathbb{P}_\theta \left\{ r_{n,k}^{-1}(\mathcal{H}_k, \mathcal{P}) \|\hat{\theta}_n - \theta\|_{\mathcal{H}_k} \geq c_k(\mathcal{H}_k, \mathcal{P}) \right\} > 0, \tag{6}$$

where the goal is to find the minimax rate $r_{n,k}(\mathcal{H}_k, \mathcal{P})$ and a positive constant $c_k(\mathcal{H}_k, \mathcal{P})$ (independent of n) such that (6) holds and $\mathbb{P}_\theta = P^n$ when $\theta = \mu_k(P)$. Using this equivalence, we obtain the minimax rates by employing Le Cam's method (Tsybakov, 2008)—see Theorems B.1 and B.2 for a reference.

3.1 Lower Bounds for Discrete Probability Measures

The following result (proved in Section 5.1) presents a minimax rate of $n^{-1/2}$ for estimating $\mu_k(P)$, where k is assumed to be translation invariant on $\mathbb{R}^d$.

Theorem 1 (Translation invariant kernels) *Let $\mathcal{P}$ be the set of all Borel discrete probability measures on $\mathbb{R}^d$. Suppose $k(x,y) = \psi(x-y)$, where $\psi \in C_b(\mathbb{R}^d)$ is positive definite and k is characteristic. Assume there exists $z \in \mathbb{R}^d$ and $\beta > 0$, such that $\psi(0) - \psi(z) \geq \beta$. Then the following holds:*

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ \|\hat{\theta}_n - \mu_k(P)\|_{\mathcal{H}_k} \geq \frac{1}{6}\sqrt{\frac{2\beta}{n}} \right\} \geq \frac{1}{4}.$$

The result is based on Le Cam's method involving two hypotheses (see Theorem B.1), where we choose them to be KMEs of discrete measures, both supported on the same pair of points separated by z in $\mathbb{R}^d$.

Remark (Choosing z and β) *As discussed in Sriperumbudur et al. (2010, Section 3.4), if k is translation invariant and characteristic on $\mathbb{R}^d$, then it is also strictly positive definite. This means that $\psi(0) > 0$. Moreover, the following hold: (a) Since ψ is positive definite, we have $|\psi(x)| \leq \psi(0)$ for all $x \in \mathbb{R}^d$ and (b) since ψ is characteristic, it cannot be a constant function. Together these facts show that there always exist $z \in \mathbb{R}^d$ and $\beta > 0$ satisfying the assumptions of Theorem 1. For instance, a Gaussian kernel $k(x,v) = \exp\left(-\|x-v\|_2^2/(2\eta^2)\right)$ satisfies $\psi(0) - \psi(z) \geq \|z\|_2^2/(4\eta^2)$ if $\|z\|_2^2 \leq 2\eta^2$, where we used a simple fact that $1 - e^{-x} \geq x/2$ for $0 \leq x \leq 1$.*

While Theorem 1 dealt with general translation invariant kernels, the following result (proved in Section 5.2) specializes it to radial kernels, i.e., kernels of the form in (5), by providing a simple condition on ν under which Theorem 1 holds.

Corollary 2 (Radial kernels) *Let $\mathcal{P}$ be the set of all Borel discrete probability measures on $\mathbb{R}^d$ and k be radial on $\mathbb{R}^d$, i.e., $k(x,y) = \psi_\nu(x-y) := \int_0^\infty e^{-t\|x-y\|_2^2} d\nu(t)$, where $\nu \in M_+^b([0,\infty))$ such that $\operatorname{supp}(\nu) \neq \{0\}$. Assume there exist $0 < t_1 < \infty$ and $\alpha > 0$ satisfying $\nu([t_1,\infty)) \geq \alpha$. Then the following holds:*

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ \|\hat{\theta}_n - \mu_k(P)\|_{\mathcal{H}_k} \geq \frac{1}{6}\sqrt{\frac{\alpha}{n}} \right\} \geq \frac{1}{4}.$$

Remark (Choosing t_1 and α) *Since $\operatorname{supp}(\nu) \neq \{0\}$ the assumption of $\nu[t_1,\infty) \geq \alpha$ is always satisfied. For instance, if ν is a probability measure with positive median η then we can set $t_1 = \eta$ and $\alpha = \frac{1}{2}$. Based on this, it is easy to verify (see Appendix D.1) that $\alpha = 1$ for Gaussian, $\alpha = C_M$ for mixture of Gaussian kernels, $\alpha = \frac{c^{-2\gamma}}{2}$ for inverse multiquadrics and $\alpha = \frac{1}{2}$ for Matérn kernels.*

3.2 Lower Bounds for Probability Measures with Smooth Densities

So far, we have shown that the rate $n^{-1/2}$ is minimax optimal for the problem of KME estimation (both for translation invariant and radial kernels). As discussed in Section 1,

since this rate is independent of the smoothness of the estimand (which is determined by the smoothness of the kernel), one might wonder whether the minimax rate can be improved by restricting $\mathcal{P}$ to distributions with smooth densities. We show in this section (see Theorems 6 and 8) that this is not the case by restricting $\mathcal{P}$ to contain only distributions with infinitely continuously differentiable densities and proving the minimax lower bound of order $n^{-1/2}$.

We will start the analysis with translation invariant kernels and present a corresponding lower bound in Theorem 6. The proof of this result is again based on an application of Le Cam's method involving two hypotheses (see Theorem B.1), where this time these hypotheses are chosen to be embeddings of the d-dimensional Gaussian distributions. One of the main steps, when applying Theorem B.1, is to lower bound the distance between these embeddings. This is done in the following result (proved in Section 5.3), which essentially shows that if we take two Gaussian distributions $G(\mu_0, \sigma^2 I)$ and $G(\mu_1, \sigma^2 I)$ with the mean vectors $\mu_0, \mu_1 \in \mathbb{R}^d$ which are close enough to each other, then the RKHS distance between the corresponding embeddings can be lower bounded by the Euclidean distance $\|\mu_0 - \mu_1\|_2$.

Proposition 3 *Let $\sigma > 0$. Suppose $k(x,y) = \psi(x-y)$, where $\psi \in C_b(\mathbb{R}^d)$ is positive definite and k is characteristic. Then there exist constants $\epsilon_{\psi,\sigma^2}, c_{\psi,\sigma^2} > 0$ depending only on ψ and σ^2, such that the following condition holds for any $a \in \mathbb{R}^d$ with $\|a\|_2^2 \leq \epsilon_{\psi,\sigma^2}$:*

$$c_{\psi,\sigma^2} \leq \min_{e_z \in S^{d-1}} \frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} \langle e_z, w\rangle^2 \cos\left(\langle a, w\rangle\right) d\Lambda_\psi(w) < \infty, \tag{7}$$

where S^{d-1} is a unit sphere in $\mathbb{R}^d$ and $\Lambda_\psi \in M_+^b(\mathbb{R}^d)$ is defined in (4). Moreover, for all vectors $\mu_0, \mu_1 \in \mathbb{R}^d$ satisfying $\|\mu_0 - \mu_1\|_2^2 \leq \epsilon_{\psi,\sigma^2}$, the following holds:

$$\|\theta_0 - \theta_1\|_{\mathcal{H}_k} \geq \sqrt{\frac{c_{\psi,\sigma^2}}{2}} \|\mu_0 - \mu_1\|_2, \tag{8}$$

where θ_0 and θ_1 are KMEs of Gaussian measures $G(\mu_0, \sigma^2 I)$ and $G(\mu_1, \sigma^2 I)$ respectively.

Remark 4 (KME expands small distances) *For a Gaussian kernel, it is possible to show (Sriperumbudur et al., 2012, Example 3; Ramdas et al., 2015, Proposition 1) that $\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 = C_1\left(1 - \exp(-C_2\|\mu_0 - \mu_1\|_2^2)\right)$, where C_1 and C_2 are positive constants that depend only on σ^2 and η^2. This shows that (8) holds for $\|\mu_0 - \mu_1\|_2 \in [0, D]$, where D satisfies $C_1\left(1 - \exp(-C_2 D^2)\right) = \frac{1}{2} D^2 c_{\psi,\sigma^2}$. In other words, Proposition 3 states that the mapping $f_{\sigma^2} \colon \mathbb{R}^d \to \mathcal{H}_k$ defined by $f_{\sigma^2}(x) := \mu_k\left(G(x, \sigma^2 I)\right)$ expands small distances.*

Remark 5 (Computing c_{ψ,σ^2} and ϵ_{ψ,σ^2}) *Generally it may be very hard to compute (or bound) the constants c_{ψ,σ^2} and ϵ_{ψ,σ^2} appearing in the statement of Proposition 3. However, in some cases this may be still possible. In Appendix E we will provide an extensive analysis for the case of radial kernels.*

Based on Proposition 3, the following result shows that the rate of $n^{-1/2}$ remains minimax optimal for the problem of KME estimation with translation invariant kernels, even if we restrict the class of distributions $\mathcal{P}$ to contain only measures with smooth densities.

Theorem 6 (Translation invariant kernels) *Let $\mathcal{P}$ be the set of distributions over $\mathbb{R}^d$ whose densities are continuously infinitely differentiable. Suppose $k(x,y) = \psi(x-y)$, where $\psi \in C_b(\mathbb{R}^d)$ is positive definite and k is characteristic. Define $c_\psi := c_{\psi,1}$ and $\epsilon_\psi := \epsilon_{\psi,1}$ where $c_{\psi,1}$ and $\epsilon_{\psi,1}$ are positive constants that satisfy (7) in Proposition 3. Then for any $n \geq \frac{1}{\epsilon_\psi}$, the following holds:*

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ \|\hat{\theta}_n - \mu_k(P)\|_{\mathcal{H}_k} \geq \frac{1}{2}\sqrt{\frac{c_\psi}{2n}} \right\} \geq \frac{1}{4}.$$

Proof The proof will be based on Theorem B.1. For this we need to find two probability measures P_0 and P_1 on $\mathbb{R}^d$ and corresponding KMEs θ_0 and θ_1, such that $\|\theta_0 - \theta_1\|_{\mathcal{H}_k}$ is of the order $\Omega(n^{-1/2})$, while $\mathrm{KL}(P_0^n \| P_1^n)$ is upper bounded by a constant independent of n. Here $\mathrm{KL}(P_0\|P_1)$ denotes the Kullback-Leibler divergence between P_0 and P_1, which is defined as $\mathrm{KL}(P_0\|P_1) = \int \log \frac{dP_0}{dP_1}\, dP_0$ where P_0 is absolutely continuous w.r.t. P_1.

Pick two Gaussian distributions $G_0 := G(\mu_0, \sigma^2 I)$ and $G_1 := G(\mu_1, \sigma^2 I)$ for $\mu_0, \mu_1 \in \mathbb{R}^d$, and $\sigma^2 > 0$. It is known that (Tsybakov, 2008, Section 2.4)

$$\mathrm{KL}(G_0^n \| G_1^n) = n \cdot \frac{\|\mu_0 - \mu_1\|_2^2}{2\sigma^2}, \tag{9}$$

where G_0^n and G_1^n are n-fold product distributions. Choose μ_0 and μ_1 such that

$$\|\mu_0 - \mu_1\|_2^2 = \frac{1}{n}.$$

Denote KMEs of G_0 and G_1 using θ_0 and θ_1 respectively. Next we will take $\sigma^2 = 1$ and apply Proposition 3. Since c_ψ and ϵ_ψ satisfy (7) in Proposition 3, it follows from Proposition 3 that for $1/n \leq \epsilon_\psi$,

$$\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 \geq \frac{c_\psi}{2}\|\mu_0 - \mu_1\|_2^2 = \frac{c_\psi}{2n}.$$

This shows that the first condition of Theorem B.1 is satisfied for θ_0 and θ_1 with $s := \frac{1}{2}\sqrt{c_\psi/(2n)}$. Moreover, using (9) we can show that the second condition of Theorem B.1 is satisfied with $\alpha = \frac{1}{2}$. We conclude the proof with an application of Theorem B.1. ■

Remark 7 (Lower bound on the sample size n) *Note that Theorem 6 holds only for large enough sample size n (i.e., $n \geq 1/\epsilon_\psi$). This assumption on n can be dropped if we set $\|\mu_0 - \mu_1\|_2^2 = \epsilon_\psi/n$ in the proof. In this case, the lower bound $\frac{1}{2}\sqrt{c_\psi/(2n)}$ will be replaced with $\frac{1}{2}\sqrt{c_\psi \epsilon_\psi/(2n)}$, while the lower bound on the minimax probability $1/4$ will be replaced with*

$$\max\left(\frac{1}{4}e^{-\frac{\epsilon_\psi}{2}}, \frac{1-\sqrt{\epsilon_\psi/4}}{2}\right).$$

The latter is generally undesirable, especially if ϵ_ψ grows with $d \to \infty$, since we want the minimax probability to be lower bounded by some universal non-zero constant that does not depend on the properties of the problem at hand.

Since radial kernels are particular instances of translation invariant kernels, Theorem 6 can be specialized by explicitly computing the constants c_ψ and ϵ_ψ to derive a minimax lower bound of order $\Omega(n^{-1/2})$. Unfortunately, the resulting lower bound will depend on the dimensionality d in a rather bad way and, as a consequence, is suboptimal in some situations. For instance, if we consider a Gaussian kernel $k(x,y) = \exp\big(-\frac{1}{2\eta^2}\|x-y\|_2^2\big)$, then a straightforward computation of c_ψ shows that the lower bound in Theorem 6 has the form $\sqrt{(1+2/\eta^2)^{-d/2}/n}$ which shrinks to zero as $d \to \infty$, while Proposition A.1 (also see Remark A.2) provides a dimension independent upper bound of the order $O_p(n^{-1/2})$. Therefore, instead of specializing Theorem 6 to radial kernels, we obtain the following result for radial kernels by using a refined analysis which yields a minimax rate of $\Omega(n^{-1/2})$ that matches the upper bound of Proposition A.1 up to constant factors that depend only on the shape of Borel measure ν. In particular, when specialized to the Gaussian kernel, the result matches the upper bound up to a constant factor independent of d.

Theorem 8 (Radial kernels) *Let k be radial on $\mathbb{R}^d$, i.e., $k(x,y) = \int_0^\infty e^{-t\|x-y\|_2^2}\, d\nu(t)$, where $\nu \in M_+^b([0,\infty))$ and $\mathcal{P}$ be the set of distributions over $\mathbb{R}^d$ whose densities are continuously infinitely differentiable. Assume that $\mathrm{supp}(\nu) \neq \{0\}$ and there exist $0 < t_0 \leq t_1 < \infty$, $0 < \beta < \infty$ such that $\nu([t_0, t_1]) \geq \beta$. Then the following holds:*

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ \|\hat{\theta}_n - \mu_k(P)\|_{\mathcal{H}_k} \geq \frac{1}{50}\sqrt{\frac{1}{n} \cdot \frac{\beta t_0}{t_1 e}\left(1 - \frac{2}{2+d}\right)} \right\} \geq \frac{1}{5}.$$

Proof The proof, which is presented in Section 5.4, is based on an application of Le Cam's method involving multiple hypotheses (see Theorem B.2), where we use exponential (in d) number of Gaussian distributions with variances decaying as $\frac{1}{d}$. ■

Remark (Non-trivial lower bound as $d \to \infty$) *The proof of Theorem 8 is based on Gaussian distributions with variances decaying as $1/d$. As $d \to \infty$, it is obvious the densities of these distributions do not have uniformly bounded Lipschitz constants, i.e., they are arbitrarily "peaky". Hence, if we choose $\mathcal{P}$ to be class of distributions with infinitely differentiable densities that have uniformly bounded Lipschitz constants, then as $d \to \infty$, the densities considered in the proof of Theorem 8 do not belong to $\mathcal{P}$. On the other hand, the densities considered in the proof of Theorem 6 still belong to $\mathcal{P}$ but yielding an uninteresting result since $c_\psi \to 0$ when $d \to \infty$. Therefore, it is an open question whether a non-trivial lower bound can be obtained for radial kernels (or any other translation invariant kernels) if we choose $\mathcal{P}$ to contain only distributions with densities having uniformly bounded Lipschitz constants.*

Remark (Alternative Proof) *For completeness, we also present an alternative proof of Theorem 8 in Appendix E. It is based on Proposition 3, which holds for any translation invariant kernel. As a result, this proof leads to slightly worse constants compared to Theorem 8 (where we used an analysis specific to radial kernels), as well as a superfluous condition on the minimal sample size n.*

In Appendix D.2, we compute the positive constant $B_k := \frac{\beta t_0}{t_1}$ that appears in the lower bound in Theorem 8 in a closed form for Gaussian, mixture of Gaussian, inverse multiquadric and Matérn kernels.

4. Minimax Estimation of μ_P in the $L^2(\mathbb{R}^d)$ Norm

So far, we have discussed the minimax estimation of the kernel mean embedding (KME) in the RKHS norm. In this section, we investigate the minimax estimation of KME in $L^2(\mathbb{R}^d)$ norm. The reason for this investigation is as follows. Let $k(x,y) = \psi(x-y)$, $x, y \in \mathbb{R}^d$, where $\psi \in L^1(\mathbb{R}^d) \cap C(\mathbb{R}^d)$ is strictly positive definite. The corresponding RKHS is given by (see Wendland, 2005, Theorem 10.12)

$$\mathcal{H}_k = \left\{ f \in L^2(\mathbb{R}^d) \cap C(\mathbb{R}^d) \, : \, \int_{\mathbb{R}^d} \frac{|f^\wedge(\omega)|^2}{\psi^\wedge(\omega)} \, d\omega < \infty \right\}, \tag{10}$$

which is endowed with the inner product $\langle f, g \rangle_{\mathcal{H}_k} = \int_{\mathbb{R}^d} \frac{f^\wedge(\omega)\overline{g^\wedge(\omega)}}{\psi^\wedge(\omega)} d\omega$ with $f^\wedge$ being the Fourier transform of f in the L^2-sense. It follows from (10) that for any $f \in \mathcal{H}_k$,

$$\|f\|^2_{L^2(\mathbb{R}^d)} \overset{(\star)}{=} \|f^\wedge\|^2_{L^2(\mathbb{R}^d)} = \int_{\mathbb{R}^d} \left|f^\wedge(\omega)\right|^2 d\omega = \int_{\mathbb{R}^d} \frac{|f^\wedge(\omega)|^2}{\psi^\wedge(\omega)} \psi^\wedge(\omega) \, d\omega \overset{(\dagger)}{\leq} \|\psi^\wedge\|_\infty \|f\|^2_{\mathcal{H}_k} \overset{(\ddagger)}{<} \infty, \tag{11}$$

where $(\star)$ follows from Plancherel theorem (Wendland, 2005, Corollary 5.25), $\|f\|_{\mathcal{H}_k}$ is defined in (10), $(\dagger)$ follows from Hölder's inequality, and $(\ddagger)$ holds since $\psi^\wedge \in C_0(\mathbb{R}^d)$ (by Riemann-Lebesgue lemma, Folland, 1999, Theorem 8.22). Note that $\psi^\wedge$ is non-negative (Wendland, 2005, Theorem 6.11) and so the inequality in $(\dagger)$ is valid. It therefore follows from (11) that $\mathcal{H}_k$ is continuously included in $L^2(\mathbb{R}^d)$ and $\|\cdot\|_{L^2(\mathbb{R}^d)}$ is a weaker norm than $\|\cdot\|_{\mathcal{H}_k}$.[1] This means it is possible that the minimax rate of estimating μ_P in $\|\cdot\|_{L^2(\mathbb{R}^d)}$ could be faster than its RKHS counterpart with the rate possibly depending on the smoothness of k. Hence, it is of interest to analyze the minimax rates of estimating μ_P in $\|\cdot\|_{L^2(\mathbb{R}^d)}$. Interestingly, we show in this section that the minimax rate in the L^2 setting is still $n^{-1/2}$.

The analysis in the L^2 setting follows ideas similar to those of the RKHS setting wherein, first, in Section 4.1, we consider the minimax rate of estimating μ_P for translation invariant and radial kernels when $\mathcal{P}$ is the set of all Borel discrete probability measures on $\mathbb{R}^d$ (see Theorem 9 and Corollary 10). Next, in Section 4.2, we choose $\mathcal{P}$ to be the set of all probability distributions that have infinitely continuously differentiable densities and study the question of minimax rates for translation invariant (see Theorem 12) and radial kernels (see Theorem 13). For both these choices of $\mathcal{P}$, we show that the rate is $n^{-1/2}$ irrespective of the smoothness of k. Exploiting the injectivity of mean embedding for characteristic kernels (see the paragraph below and the paragraph around (6)), these results are derived using Le Cam's method (see Theorems B.1 and B.2). Combined with Proposition A.1 (also see Remark A.3), these results show that the empirical estimator, μ_{P_n} is minimax optimal. Finally, in Section 4.3 we discuss the relation between our results and some classical results of nonparametric density estimation, particularly, those of the kernel density estimator.

1. The continuous inclusion of $\mathcal{H}_k$ in $L^2(\mathbb{R}^d)$ is known for Gaussian kernels on $\mathbb{R}^d$ (e.g., see Vert and Vert, 2006, Lemma 11). Similar result is classical for Sobolev spaces in general (e.g., see Folland, 1999, Section 9.3, p.302) and particularly for those induced by Matérn kernels. Steinwart and Christmann (2008, Theorem 4.26) provides a general result for continuous inclusion of $\mathcal{H}_k$ in $L^2(\mu)$ assuming $\int_{\mathcal{X}} \sqrt{k(x,x)} \, d\mu(x) < \infty$ where μ is a σ-finite measure. However, the result does not hold for translation invariant kernels on $\mathbb{R}^d$ as the integrability condition is violated.

Before we proceed to the main results of this section, we briefly discuss the difference between estimation in RKHS and $L^2(\mathbb{R}^d)$ norms. Suppose $k(x,y) = \psi(x-y)$, $x, y \in \mathbb{R}^d$ where $\psi \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d) \cap C(\mathbb{R}^d)$ is positive definite and characteristic. It is easy to verify that $\mu_P \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$. Since $\mu_P = \psi * P$, (10) implies

$$\|\mu_P\|^2_{\mathcal{H}_k} = \int_{\mathbb{R}^d} \frac{|(\psi * P)^\wedge|^2}{\psi^\wedge(\omega)} \, d\omega = \int_{\mathbb{R}^d} |\phi_P(\omega)|^2 \psi^\wedge(\omega) \, d\omega = \|\phi_P\|^2_{L^2(\mathbb{R}^d, \psi^\wedge)} \tag{12}$$

whereas

$$\|\mu_P\|^2_{L^2(\mathbb{R}^d)} \overset{(\star)}{=} \int_{\mathbb{R}^d} |\mu_P^\wedge(\omega)|^2 \, d\omega = \int_{\mathbb{R}^d} |\phi_P(\omega)|^2 (\psi^\wedge)^2(\omega) \, d\omega = \|\phi_P\|^2_{L^2(\mathbb{R}^d, (\psi^\wedge)^2)}, \tag{13}$$

where $\phi_P(\omega) := \int e^{-i\omega^T x} dP(x)$ is the characteristic function of P and $(\star)$ follows from Plancherel's theorem. It follows from (12) and (13) that the RKHS norm emphasizes the high frequencies of ϕ_P compared to that of the L^2-norm. Since ψ is characteristic, i.e., $P \mapsto \mu_k(P) \in \mathcal{H}_k$ is injective, which is guaranteed if and only if $\text{supp}(\psi^\wedge) = \mathbb{R}^d$ (Sriperumbudur et al., 2010, Theorem 9), it follows from (13) that $P \mapsto \mu_k(P) \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ is injective. Therefore (2) can be equivalently written as (6) by replacing $\|\cdot\|_{\mathcal{H}_k}$ with $\|\cdot\|_{L^2(\mathbb{R}^d)}$ (see the discussion around (6)) and we obtain minimax rates by employing Le Cam's method as we did in the previous section.

4.1 Lower Bounds for Discrete Probability Measures

The following result (proved in Section 5.5) for translation invariant kernels is based on an application of Le Cam's method involving two hypotheses (see Theorem B.1), where we choose them to be KMEs of discrete measures, both supported on the same pair of points separated by a vector z in $\mathbb{R}^d$.

Theorem 9 (Translation invariant kernels) *Let $\mathcal{P}$ be the set of all Borel discrete probability measures on $\mathbb{R}^d$. Suppose $k(x,y) = \psi(x-y)$, $x, y \in \mathbb{R}^d$ where $\psi \in L^2(\mathbb{R}^d) \cap C(\mathbb{R}^d)$ is positive definite and k is characteristic. Define*

$$C_z^\psi := 2 \left(\|\psi\|^2_{L^2(\mathbb{R}^d)} - \int_{\mathbb{R}^d} \psi(y)\psi(y+z) dy \right) \tag{14}$$

for some $z \in \mathbb{R}^d \setminus \{0\}$. Then $C_z^\psi > 0$ and

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ \|\hat{\theta}_n - \mu_k(P)\|_{L^2(\mathbb{R}^d)} \geq \frac{1}{6} \sqrt{\frac{C_z^\psi}{n}} \right\} \geq \frac{1}{4}.$$

Using Cauchy-Schwartz inequality, the constant C_z^ψ in Theorem 9 can be shown (see the proof of Lemma 15 in Section 5.5) to be positive for every $z \in \mathbb{R}^d \setminus \{0\}$ if k is characteristic, i.e., $\text{supp}(\Lambda_\psi) = \mathbb{R}^d$ (see (4) for Λ_ψ). The following result (proved in Section 5.6) specializes Theorem 9 to radial kernels.

Corollary 10 (Radial kernels) *Let $\mathcal{P}$ be the set of all Borel discrete probability measures on $\mathbb{R}^d$ and k be radial on $\mathbb{R}^d$, i.e., $k(x,y) = \psi_\nu(x-y) := \int_0^\infty e^{-t\|x-y\|_2^2} d\nu(t)$, where $\nu \in M_+^b([0,\infty))$ such that $\mathrm{supp}(\nu) \neq \{0\}$ and*

$$\int_0^\infty t^{-d/2} d\nu(t) < \infty. \tag{15}$$

Assume that there exist $0 < \delta_0 \leq \delta_1 < \infty$ and $\beta > 0$ such that $\nu([\delta_0, \delta_1]) \geq \beta$. Then the following holds:

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ \|\hat{\theta}_n - \mu_k(P)\|_{L^2(\mathbb{R}^d)} \geq \frac{\beta}{6} \sqrt{\frac{1}{n} \left(\frac{\pi}{2\delta_1} \right)^{d/2}} \right\} \geq \frac{1}{4}.$$

In Corollary 10, since $\mathrm{supp}(\nu) \neq \{0\}$, the assumption of $\nu([\delta_0, \delta_1]) \geq \beta$ is always satisfied. In addition, the condition (15) on ν is satisfied by Gaussian, mixture of Gaussians, inverse multiquadric (while (15) is satisfied for $\gamma > d/2$, the result in Corollary 10 holds for $\gamma > d/4$) and Matérn kernels—refer to Remark A.3 for more details. Also, for these examples of kernels, the positive constant $A_k := \beta^2 \delta_1^{-d/2}$ in the lower bound in Corollary 10 can be computed in a closed form (see Appendix D.3 for details).

4.2 Lower Bounds for Probability Measures with Smooth Densities

Next, as we did in Section 3.2, we choose $\mathcal{P}$ to be the set of all probability measures that have infinitely continuously differentiable densities and show that the minimax rate of estimating μ_P in L^2-norm for translation invariant (see Theorem 12) and radial kernels (see Theorem 13) is $n^{-1/2}$. The proof of these results are again based on an application of Le Cam's method involving two (see Theorem B.1) and multiple hypotheses (see Theorem B.2), where these hypotheses are chosen to be embeddings of the d-dimensional Gaussian distributions. As in Section 3.2, the results of this section are based on the following result (proved in Section 5.7), which is conceptually similar to that of Proposition 3.

Proposition 11 *Let $\sigma > 0$. Suppose $k(x,y) = \psi(x-y)$, where $\psi \in L^1(\mathbb{R}^d) \cap C_b(\mathbb{R}^d)$ is positive definite and k is characteristic. Then there exist constants $\epsilon_{\psi,\sigma^2}, c_{\psi,\sigma^2} > 0$ depending only on ψ and σ^2, such that the following condition holds for any $a \in \mathbb{R}^d$ with $\|a\|_2^2 \leq \epsilon_{\psi,\sigma^2}$:*

$$c_{\psi,\sigma^2} \leq \min_{e_z \in S^{d-1}} 2 \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} \langle e_z, w \rangle^2 \cos\left(\langle a, w \rangle\right) \left(\psi^\wedge(w)\right)^2 dw < \infty, \tag{16}$$

where S^{d-1} is a unit sphere in $\mathbb{R}^d$. Moreover, for all vectors $\mu_0, \mu_1 \in \mathbb{R}^d$ satisfying $\|\mu_0 - \mu_1\|_2^2 \leq \epsilon_{\psi,\sigma^2}$, the following holds:

$$\|\theta_0 - \theta_1\|_{L^2(\mathbb{R}^d)} \geq \sqrt{\frac{c_{\psi,\sigma^2}}{2}} \|\mu_0 - \mu_1\|_2,$$

where θ_0 and θ_1 are KMEs of the Gaussian measures $G(\mu_0, \sigma^2 I)$ and $G(\mu_1, \sigma^2 I)$ respectively.

The following result for translation invariant kernels is established using the above result wherein the proof is exactly the same as that of Theorem 6 except for an application of Proposition 11 in place of Proposition 3.

Theorem 12 (Translation invariant kernels) *Let $\mathcal{P}$ be the set of distributions over $\mathbb{R}^d$ whose densities are continuously infinitely differentiable. Suppose $k(x,y) = \psi(x-y)$, where $\psi \in L^1(\mathbb{R}^d) \cap C_b(\mathbb{R}^d)$ is positive definite and k is characteristic. Define $c_\psi := c_{\psi,1}$ and $\epsilon_\psi := \epsilon_{\psi,1}$ where $c_{\psi,1}$ and $\epsilon_{\psi,1}$ are positive constants that satisfy (16) in Proposition 11. Then for any $n \geq \frac{1}{\epsilon_\psi}$, the following holds:*

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ \|\hat{\theta}_n - \mu_k(P)\|_{L^2(\mathbb{R}^d)} \geq \frac{1}{2} \sqrt{\frac{c_\psi}{2n}} \right\} \geq \frac{1}{4}.$$

As discussed in Remark 7, it is possible to remove the requirement of minimal sample size in Theorem 12. Also, as discussed in Remark 5 and in the paragraph following Remark 7, the constants c_ψ and ϵ_ψ appearing in the bound in Theorem 12 are not only difficult to compute but also may depend on the dimensionality d in a sup-optimal manner, particularly as $d \to \infty$. Therefore, similar to what was done in Section 3.2, we will not specialize Theorem 12 to radial kernels but instead present the following result (proved in Section 5.8 and the proof closely follows that of Theorem 8), which is based on a direct analysis involving the properties of radial kernels. For the particular case of a Gaussian kernel, this lower bound matches the upper bound of Proposition A.1 (also see Remark A.3) up to a constant factor independent of d.

Theorem 13 (Radial kernels) *Let k be radial on $\mathbb{R}^d$, i.e., $k(x,y) = \int_0^\infty e^{-t\|x-y\|_2^2} d\nu(t)$, where $\nu \in M_+^b([0,\infty))$ and $\mathcal{P}$ be the set of distributions over $\mathbb{R}^d$ whose densities are continuously infinitely differentiable. Assume that (15) holds, $\mathrm{supp}(\nu) \neq \{0\}$ and there exist $0 < \delta_0 \leq \delta_1 < \infty$, $0 < \beta < \infty$ such that $\nu([\delta_0, \delta_1]) \geq \beta$. Then the following holds:*

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ \|\hat{\theta}_n - \mu_k(P)\|_{L^2(\mathbb{R}^d)} \geq \frac{1}{50} \sqrt{\frac{1}{n} \left(\frac{\pi}{2\delta_1} \right)^{d/2} \frac{\beta^2 \delta_0}{\delta_1 e} \left(1 - \frac{2}{2+d} \right)} \right\} \geq \frac{1}{5}.$$

The constant $B_k := \beta^2 \delta_0 \delta_1^{-\frac{d+2}{2}}$ in the lower bound in the above result can be computed in a closed form for Gaussian, mixture of Gaussian, inverse multiquadric, and Matérn kernels (see Appendix D.4 for details). The factor $(\pi/2)^{d/4}$ can be eliminated from the lower bound by considering a rescaled kernel $(\pi/2)^{-d/4}\psi(x-y)$. Nevertheless, the bound will still depend on d exponentially as captured by the constant B_k. This can be further overcome by using the normalized kernel $k(x,y)/\|\psi\|_{L^2(\mathbb{R}^d)}$. In the particular case of normalized Gaussian kernels $(\pi\eta^2)^{-d/2} \exp\left(-\frac{1}{2\eta^2}\|x-y\|_2^2\right)$ this will lead to dimension-free lower bounds.

4.3 Relation to Kernel Density Estimation

In this section, we discuss the relation between the estimation of μ_P and density estimation. The problem of density estimation deals with estimating an unknown density, p based on

random samples $(X_i)_{i=1}^n$ drawn i.i.d. from it. One of the popular non-parametric methods for density estimation is kernel density estimation (KDE), where the estimator is of the form (Tsybakov, 2008, Section 1.2)

$$\hat{p}_n(x_1,\dots,x_d) = \frac{1}{n\prod_{i=1}^d h_i}\sum_{i=1}^n K\left(\frac{X_{i,1}-x_1}{h_1},\cdots,\frac{X_{i,d}-x_d}{h_d}\right).$$

Here $K\colon \mathbb{R}^d \to \mathbb{R}$ is the *smoothing kernel* (this kernel should not be confused with the reproducing kernel k which we used throughout the paper), $h_1,\dots,h_d > 0$ are bandwidths, and $X_{i,j}$ is the j-th coordinate of the i-th sample point. Assuming $p \in L^2(\mathbb{R}^d)$, the consistency of $\hat{p}_n$ is usually studied in the sense of *mean integrated squared error* (MISE) $\mathbb{E}\|\hat{p}_n - p\|^2_{L^2(\mathbb{R}^d)}$, which can be decomposed into variance and bias terms as:

$$\mathbb{E}\|\hat{p}_n - p\|^2_{L^2(\mathbb{R}^d)} = \mathbb{E}\big\|\hat{p}_n - \mathbb{E}[\hat{p}_n]\big\|^2_{L^2(\mathbb{R}^d)} + \big\|p - \mathbb{E}[\hat{p}_n]\big\|^2_{L^2(\mathbb{R}^d)}. \tag{17}$$

Assume K to be bounded and $h_1 = \cdots = h_d = h$. Define $K_h := h^{-d}K(\cdot/h)$. Then for any fixed $x \in \mathbb{R}^d$,

$$\hat{p}_n(x) = \frac{1}{nh^d}\sum_{i=1}^n K\left(\frac{X_i - x}{h}\right) = \frac{1}{n}\sum_{i=1}^n K_h(X_i - x) = \int_{\mathbb{R}^d} K_h(z-x)\,dP_n(z)$$

and

$$\mathbb{E}[\hat{p}_n(x)] = \frac{1}{h^d}\int_{\mathbb{R}^d} K\left(\frac{z-x}{h}\right)p(z)dz = (K_h * p)(x).$$

This shows that $\hat{p}_n = \mu_{K_h}(P_n)$ and $\mathbb{E}[\hat{p}_n] = \mu_{K_h}(P)$ where P is the distribution with p as its density w.r.t. the Lebesgue measure and P_n is the empirical measure constructed based on samples $(X_i)_{i=1}^n$ drawn from p. Therefore the results of Section 4 (and more generally of this paper) are about the minimax rates for $\mathbb{E}[\hat{p}_n]$. However, note that K_h need not be positive definite (and therefore need not be the reproducing kernel of some RKHS). On the other hand, K has to be positive, i.e., $K(x) \geq 0$, $\forall\, x \in \mathbb{R}^d$ and normalized, i.e., $\int_{\mathbb{R}^d} K(x)\,dx = 1$ to yield an estimator that is a valid density, unlike in kernel mean estimation where k need not be positive nor normalized. The minimax rate of $n^{-1/2}$ for estimating $\mathbb{E}[\hat{p}_n]$ is achieved by the kernel density estimator $\hat{p}_n$ (which is nothing but the empirical estimator of $\mu_{K_h}(P)$) as it is known (based on a straightforward generalization of Tsybakov, 2008, Proposition 1.4 for multiple dimensions) that

$$\mathbb{E}\big\|\hat{p}_n - \mathbb{E}[\hat{p}_n]\big\|^2_{L^2(\mathbb{R}^d)} \leq \frac{\|K\|^2_{L^2(\mathbb{R}^d)}}{nh^d},$$

where we assume $K \in L^2(\mathbb{R}^d)$. The bandwidth parameter h is immaterial in the estimation of $\mu_{K_h}(P)$ and can be treated as a constant (independent of n) unlike in the problem of estimating p where h should decay to zero at an appropriate rate for the bias $\big\|p - \mathbb{E}[\hat{p}_n]\big\|_{L^2(\mathbb{R}^d)}$ to converge to zero as $n \to \infty$. In particular, if p lies in a Sobolev space of smoothness index s, then the bias-squared term in (17) behaves as h^{2s}, which combined with the above bound on the variance yields a rate of $n^{-\frac{2s}{2s+1}}$ for $h = n^{-\frac{1}{2s+1}}$. This rate is known to be minimax optimal for the problem of estimating p while our rates are minimax optimal for the problem of smoothed density estimation where the smoothing is carried out by the kernel.

5. Proofs

In this section we present all the missing proofs of results of Sections 3 and 4.

5.1 Proof of Theorem 1

Pick two discrete distributions $P_0 = p_0\delta_x + (1-p_0)\delta_v$ and $P_1 = p_1\delta_x + (1-p_1)\delta_v$, where $x, v \in \mathbb{R}^d$, $0 < p_0 < 1$, $0 < p_1 < 1$ and δ_x denotes a Dirac measure supported at x. Define $\theta_0 = \mu_k(P_0)$ and $\theta_1 = \mu_k(P_1)$. Since $\|\theta_0\|^2_{\mathcal{H}_k} = \int\int k(x,y)\,dP_0(x)\,dP_0(y)$, which follows from the reproducing property of k, it is easy to verify that

$$\|\theta_0 - \theta_1\|^2_{\mathcal{H}_k} = \mathbb{E}[k(\xi,\xi')] + \mathbb{E}[k(\eta,\eta')] - 2\,\mathbb{E}[k(\xi,\eta)],$$

where ξ and η are random variables distributed according to P_0 and P_1 respectively, and ξ' and η' are independent copies of ξ and η. Since k is translation invariant, we have $k(v,v) = k(x,x) = \psi(0)$ and $k(x,v) = k(v,x) = \psi(x-v)$, which imply

$$\|\theta_0 - \theta_1\|^2_{\mathcal{H}_k} = 2(p_0-p_1)^2\big(\psi(0) - \psi(x-v)\big). \tag{18}$$

Also note that

$$\begin{aligned}
\mathrm{KL}(P_0\|P_1) &= p_0\log\frac{p_0}{p_1} + (1-p_0)\log\frac{1-p_0}{1-p_1}\\
&= p_0\log\left(1 + \frac{p_0-p_1}{p_1}\right) + (1-p_0)\log\left(1 + \frac{p_1-p_0}{1-p_1}\right)\\
&\overset{(*)}{\leq} \log\left\{p_0\left(1 + \frac{p_0-p_1}{p_1}\right) + (1-p_0)\left(1 + \frac{p_1-p_0}{1-p_1}\right)\right\}\\
&= \log\left(1 + (p_0-p_1)\left(\frac{p_0}{p_1} - \frac{1-p_0}{1-p_1}\right)\right),
\end{aligned}$$

where we used Jensen's inequality in $(*)$ for the logarithmic function, which is concave. Next, using a simple inequality $\log(1+x) \leq x$, which holds for all $x > -1$, we get

$$\mathrm{KL}(P_0\|P_1) \leq (p_0-p_1)\left(\frac{p_0}{p_1} - \frac{1-p_0}{1-p_1}\right) = \frac{(p_0-p_1)^2}{p_1(1-p_1)}.$$

Note that a maximal value of denominator is achieved when $p_1 = \frac{1}{2}$. Setting $p_1 = \frac{1}{2}$ we get the following upper bound: $\mathrm{KL}(P_0\|P_1) \leq 4\left(p_0 - \frac{1}{2}\right)^2$, which when used in the chain rule of KL-divergence yields

$$\mathrm{KL}(P_0^n\|P_1^n) \leq 4n\left(p_0 - \frac{1}{2}\right)^2.$$

Choosing p_0 such that $(p_0 - \frac{1}{2})^2 = \frac{1}{9n}$ yields $\mathrm{KL}(P_0^n\|P_1^n) \leq \frac{4}{9}$ and $\|\theta_0 - \theta_1\|^2_{\mathcal{H}_k} = \frac{2}{9n}\big(\psi(0) - \psi(x-v)\big)$. Choose x and v in such a way that $x - v = z$, where $z \in \mathbb{R}^d$ is a point for which $\psi(0) - \psi(z) \geq \beta$ and $\beta > 0$. This yields

$$\|\theta_0 - \theta_1\|^2_{\mathcal{H}_k} \geq \frac{2\beta}{9n},$$

which shows that the assumptions of Theorem B.1 are satisfied with $s := \frac{1}{6}\sqrt{\frac{2\beta}{n}}$ and $\alpha := \frac{4}{9}$. The result follows from an application of Theorem B.1 by noticing that $\frac{1-\sqrt{\alpha/2}}{2} > 1/4$.

Remark (Measures with bounded support) *It is evident from the above proof that exactly the same lower bound holds if we restrict $\mathcal{P}$ to contain only probability measures with bounded support. We can proceed further and assume that for each $P \in \mathcal{P}$ the radius of* $\mathrm{supp}(P)$ *is upper bounded by some positive constant R. In this case the same reasoning will work as long as ψ is not "flat" on the ball of radius R centered around origin.*

5.2 Proof of Corollary 2

The proof is based on application of Theorem 1. Since $\mathrm{supp}(\nu) \neq \{0\}$, it follows from (Sriperumbudur et al., 2011, Proposition 5) that k is characteristic. We now show that there exist $z \in \mathbb{R}^d$ and $\beta > 0$, such that $\psi_\nu(0) - \psi_\nu(z) \geq \beta$. Note that for any $x \in \mathbb{R}^d$

$$\begin{aligned}\psi_\nu(0) - \psi_\nu(x) &= \int_0^\infty \left(1 - e^{-t\|x\|_2^2}\right) d\nu(t) \geq \int_{t_1}^\infty \left(1 - e^{-t\|x\|_2^2}\right) d\nu(t) \\ &\geq \int_{t_1}^\infty \left(1 - e^{-t_1\|x\|_2^2}\right) d\nu(t) = \nu([t_1,\infty))\left(1 - e^{-t_1\|x\|_2^2}\right) \\ &\geq \alpha\left(1 - e^{-t_1\|x\|_2^2}\right) \geq \frac{\alpha t_1}{2}\|x\|_2^2,\end{aligned}$$

where the last inequality holds whenever $\|x\|_2^2 \leq \frac{1}{t_1}$. Choosing z such that $\|z\|_2^2 = \frac{1}{t_1}$ yields $\psi_\nu(0) - \psi_\nu(z) \geq \frac{\alpha}{2}$. The result therefore follows from Theorem 1 by choosing $\beta = \frac{\alpha}{2}$.

5.3 Proof of Proposition 3

Before we prove Proposition 3, first we will derive a closed form expression for the RKHS distance between KMEs of two d-dimensional Gaussian distributions with the kernel being translation invariant, i.e., $k(x,y) = \psi(x-y)$. Throughout this section Λ_ψ will denote a finite non-negative Borel measure corresponding to the positive-definite function ψ from (4).

Lemma 14 *Let θ_0 and θ_1 be KME of Gaussian measures $G(\mu_0, \sigma^2 I)$ and $G(\mu_1, \sigma^2 I)$ for $\mu_0, \mu_1 \in \mathbb{R}^d$ and $\sigma^2 > 0$. Suppose $k(x,y) = \psi(x-y)$, where $\psi \in C_b(\mathbb{R}^d)$ is positive definite. Then*

$$\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 = \frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} \left(1 - \cos\left(\langle \mu_0 - \mu_1, w\rangle\right)\right) d\Lambda_\psi(w). \tag{19}$$

Proof Note that

$$\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 = \|\theta_0\|_{\mathcal{H}_k}^2 + \|\theta_1\|_{\mathcal{H}_k}^2 - 2\langle \theta_0, \theta_1\rangle_{\mathcal{H}_k}, \tag{20}$$

where $\langle \theta_0, \theta_1\rangle_{\mathcal{H}_k} = \mathbb{E}_X \mathbb{E}_Y[k(X,Y)]$ with $X \sim G(\mu_0, \sigma^2 I)$ and $Y \sim G(\mu_1, \sigma^2 I)$. We will now derive the closed form for the inner product:

$$\begin{aligned}\langle \theta_0, \theta_1\rangle_{\mathcal{H}_k} &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \psi(x-y) \frac{1}{(2\pi\sigma^2)^d} e^{-\frac{1}{2\sigma^2}\|x-\mu_0\|_2^2 - \frac{1}{2\sigma^2}\|y-\mu_1\|_2^2} dx\, dy \\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-i\langle x-y, w\rangle} d\Lambda_\psi(w) \frac{1}{(2\pi\sigma^2)^d} e^{-\frac{1}{2\sigma^2}\|x-\mu_0\|_2^2 - \frac{1}{2\sigma^2}\|y-\mu_1\|_2^2} dx\, dy,\end{aligned}$$

where we used (4). The function appearing under the integral is absolutely integrable and so by Tonelli-Fubini theorem (Dudley, 2002, Theorem 4.4.5) we obtain

$$\begin{aligned}\langle \theta_0, \theta_1\rangle_{\mathcal{H}_k} &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \frac{e^{i\langle y, w\rangle} e^{-\frac{1}{2\sigma^2}\|y-\mu_1\|_2^2}}{(2\pi\sigma^2)^{d/2}(2\pi)^{d/2}} \left\{\int_{\mathbb{R}^d} \frac{e^{-i\langle x, w\rangle}}{(2\pi\sigma^2)^{d/2}} e^{-\frac{1}{2\sigma^2}\|x-\mu_0\|_2^2} dx\right\} dy\, d\Lambda_\psi(w) \\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \frac{1}{(2\pi\sigma^2)^{d/2}} \frac{1}{(2\pi)^{d/2}} e^{i\langle y, w\rangle} e^{-\frac{1}{2\sigma^2}\|y-\mu_1\|_2^2} e^{-i\langle \mu_0, w\rangle - \frac{\sigma^2\|w\|_2^2}{2}} dy\, d\Lambda_\psi(w) \\ &= \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} \left\{\int_{\mathbb{R}^d} \frac{1}{(2\pi\sigma^2)^{d/2}} e^{i\langle y, w\rangle} e^{-\frac{1}{2\sigma^2}\|y-\mu_1\|_2^2} dy\right\} e^{-i\langle \mu_0, w\rangle - \frac{\sigma^2\|w\|_2^2}{2}} d\Lambda_\psi(w) \\ &= \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{i\langle \mu_1, w\rangle - \frac{\sigma^2\|w\|_2^2}{2}} e^{-i\langle \mu_0, w\rangle - \frac{\sigma^2\|w\|_2^2}{2}} d\Lambda_\psi(w),\end{aligned}$$

where we used Lemma C.1 to compute the Fourier transform for a Gaussian density. Using Euler's formula and the fact that Λ_ψ is symmetric according to Lemma C.2, while $\sin(x)$ is an odd function, we get

$$\langle \theta_0, \theta_1\rangle_{\mathcal{H}_k} = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} \cos\left(\langle \mu_0 - \mu_1, w\rangle\right) e^{-\sigma^2\|w\|_2^2} d\Lambda_\psi(w). \tag{21}$$

The result in (19) follows by using (21) in (20). ■

Proof of Proposition 3: Define $a := \mu_0 - \mu_1$ and

$$G(a) := \frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} \left(1 - \cos\langle a, w\rangle\right) d\Lambda_\psi(w).$$

Note that $G(0) = 0$. Next, since for any $i = 1, \ldots, d$

$$\left|\frac{\partial}{\partial a_i} e^{-\sigma^2\|w\|_2^2}(1 - \cos\langle a, w\rangle)\right| = \left|e^{-\sigma^2\|w\|_2^2} w_i \sin\langle a, w\rangle\right| \leq \left|e^{-\sigma^2\|w\|_2^2} w_i\right| \in L_1(\Lambda_\psi),$$

we can differentiate G under the integral sign (Folland, 1999, Theorem 2.27) and get $\nabla G(0) = 0$.

If a function $f\colon \mathbb{R}^d \to \mathbb{R}$ is *strongly convex* with parameter $m > 0$ on some set $A \subseteq \mathbb{R}^d$, then for all $x, y \in A$:

$$f(x) \geq f(y) + \langle \nabla f(y), x - y\rangle + \frac{m}{2}\|x - y\|_2^2.$$

If we can show that G is strongly convex on $B_\epsilon := \{b \in \mathbb{R}^d\colon \|b\|_2^2 \leq \epsilon\}$ for some $\epsilon > 0$, then we can apply previous inequality with $y = 0$ and $x = a$ to obtain

$$G(a) \geq \frac{m}{2}\|a\|_2^2, \ \forall a \in B_\epsilon.$$

It is known that a twice continuously differentiable function f is strongly convex on $A \subseteq \mathbb{R}^d$ with parameter $m > 0$ if the matrix $\nabla^2 f(x) - m \cdot I$ is positive definite for all $x \in A$, where

$\nabla^2 f$ is the Hessian and $I \in \mathbb{R}^{d\times d}$ is an identity matrix. Next we compute the Hessian of G by once again employing differentiation under the integral sign (justified in the similar way as above) to obtain

$$\frac{\partial^2 G(a)}{\partial a_i \partial a_j} = \frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} w_i w_j \cos\left(\langle a, w\rangle\right) d\Lambda_\psi(w), \quad 0 \le i, j \le d.$$

Thus

$$\nabla^2 G(a) = \frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} w w^T \cos\left(\langle a, w\rangle\right) d\Lambda_\psi(w).$$

In order to prove that G is strongly convex on $B_\epsilon \subseteq \mathbb{R}^d$ we need to show that $\nabla^2 G(a) - m \cdot I$ is positive definite for each $a \in B_\epsilon$ and some $m > 0$. In other words, we need to show that there is $m > 0$ such that for each $z \in \mathbb{R}^d \setminus \{0\}$ and $a \in B_\epsilon$ the following holds:

$$\langle z, \nabla^2 G(a) z\rangle \ge m\|z\|_2^2,$$

or, equivalently,

$$\frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} \langle e_z, w\rangle^2 e^{-i\langle a, w\rangle} d\Lambda_\psi(w) \ge m, \tag{22}$$

where $e_z := z/\|z\|_2 \in \mathbb{R}^d$ is a vector of unit length pointed in the direction of z. Note that l.h.s. of (22) is the Fourier transform of a measure $\mathcal{T}_z$ on $\mathbb{R}^d$, which is absolutely continuous with respect to Λ_ψ with Radon-Nikodym derivative $2e^{-\sigma^2\|w\|_2^2}\langle e_z, w\rangle^2$.

Fix any $z \in \mathbb{R}^d$. We will first show that we can apply Bochner's Theorem (see (4)) for the measure $\mathcal{T}_z$. For this we need to check that it is (a) non-negative and (b) finite. Part (a) is apparent from the facts that Λ_ψ is non-negative and $\mathcal{T}_z$ has a non-negative density with respect to Λ_ψ. To check (b) we write

$$\int_{\mathbb{R}^d} d\mathcal{T}_z(x) = \frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} \langle e_z, w\rangle^2 d\Lambda_\psi(w) < \infty,$$

as $e^{-\sigma^2\|w\|_2^2}\langle e_z, w\rangle^2$ is positive and bounded for any $z \in \mathbb{R}^d$, while Λ_ψ is finite. We conclude from Bochner's Theorem, that function $\tilde{\psi}_z(x)$ defined in the following way:

$$\tilde{\psi}_z(x) = \int_{\mathbb{R}^d} e^{-i\langle x, w\rangle} d\mathcal{T}_z(w),$$

is positive-definite. Moreover it is well known (Dudley, 2002, Theorem 9.4.4) that $\tilde{\psi}_z \in C_b(\mathbb{R}^d)$ as $\tilde{\psi}_z$ is the characteristic function of $\mathcal{T}_z$. Finally, it follows from the discussion in (Sriperumbudur et al., 2011, Section 3.3), that if $\mathrm{supp}(\mathcal{T}_z) = \mathbb{R}^d$, then a bounded and continuous function $\tilde{\psi}_z(x)$ is *strictly* positive definite. To check the condition $\mathrm{supp}(\mathcal{T}_z) = \mathbb{R}^d$ we note that $\mathrm{supp}(\Lambda_\psi) = \mathbb{R}^d$ since ψ is characteristic (Sriperumbudur et al., 2010, Theorem 9), and no open sets of $\mathbb{R}^d$ are contained in the region where $e^{-\sigma^2\|w\|_2^2}\langle e_z, w\rangle^2 = 0$.

Summarizing, we have established that the l.h.s. of (22) is equal to $\tilde{\psi}_z(a)$, where $\tilde{\psi}_z \colon \mathbb{R}^d \to \mathbb{R}$ is a bounded, continuous and strictly positive definite function for each

$z \in \mathbb{R}^d \setminus \{0\}$. In particular, we have $\tilde{\psi}_z(0) > 0$ for all $z \in \mathbb{R}^d$. Note that $\tilde{\psi}_z(0)$ depends on z only through its direction. Next we want to show that

$$\inf_{z\in S_d} \tilde{\psi}_z(0) > 0, \tag{23}$$

where the infimum is over the unit sphere $S_d := \{b \in \mathbb{R}^d \colon \|b\|_2^2 = 1\}$. Note that the function $F \colon z \to \tilde{\psi}_z(0)$ defined on S_d is continuous. Since S_d is closed and bounded, we know that F attains its minimum on it. In other words, there is $z^* \in S_d$, such that

$$\inf_{z\in S_d} \tilde{\psi}_z(0) = \tilde{\psi}_{z^*}(0).$$

Thus, if $\inf_{z\in S_d} \tilde{\psi}_z(0) = 0$, we will also get $\tilde{\psi}_{z^*}(0) = 0$, which will contradict the fact that $\tilde{\psi}_z(0) > 0$ for each $z \in \mathbb{R}^d\backslash\{0\}$. This proves (23). Using Lemma C.3 we also conclude that $\inf_{z\in S_d} \tilde{\psi}_z \colon \mathbb{R}^d \to \mathbb{R}$ is a continuous function. Now we may finally conclude that there are constants $c_{\psi,\sigma^2}, \epsilon_{\psi,\sigma^2} > 0$ such that

$$\inf_{z\in S_d} \tilde{\psi}_z(a) \ge c_{\psi,\sigma^2}$$

for all $a \in B_{\epsilon_{\psi,\sigma^2}}$. Finally, we take $m = c_{\psi,\sigma^2}$ and this concludes the proof.

5.4 Proof of Theorem 8

The proof is based on application of Theorem B.2 where we choose $\theta_0, \ldots, \theta_M$ to be KMEs of d-dimensional Gaussian measures with variances decaying to zero as $d \to \infty$.

Let $G(\mu_0, \sigma^2 I)$ and $G(\mu_1, \sigma^2 I)$ be two d-dimensional Gaussian distributions with mean vectors $\mu_0, \mu_1 \in \mathbb{R}^d$ and variance $\sigma^2 > 0$. Define θ_0 and θ_1 to be the embeddings of $G(\mu_0, \sigma^2 I)$ and $G(\mu_1, \sigma^2 I)$ respectively.

(A) Deriving a closed form expression for $\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2$.

Using Lemma 14, presented in Section 5.3, we have

$$\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 = \frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} \left(1 - \cos\left(\langle \mu_0 - \mu_1, w\rangle\right)\right) d\Lambda_\psi(w), \tag{24}$$

where Λ_ψ is a finite non-negative Borel measure from the Bochner's Theorem corresponding to the kernel k. We now show that Λ_ψ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$ and has the following density:

$$\lambda_\psi(w) = \int_0^\infty \frac{1}{(2t)^{d/2}} e^{-\frac{\|w\|_2^2}{4t}} d\nu(t), \quad w \in \mathbb{R}^d.$$

Indeed, by noticing that

$$\int_0^\infty \left(\int_{\mathbb{R}^d} \left| e^{-i\langle w, x\rangle} \frac{1}{(2t)^{d/2}} e^{-\frac{\|w\|_2^2}{4t}} \right| dw \right) d\nu(t) < \infty$$

we may apply Tonelli-Fubini theorem (Dudley, 2002, Theorem 4.4.5) to interchange the order of integration and get

$$\int_{\mathbb{R}^d} \frac{e^{-i\langle w,x\rangle}}{(2\pi)^{d/2}} \left(\int_0^\infty \frac{1}{(2t)^{d/2}} e^{-\frac{\|w\|_2^2}{4t}} d\nu(t) \right) dw = \int_0^\infty \left(\int_{\mathbb{R}^d} \frac{e^{-i\langle w,x\rangle}}{(2\pi)^{d/2}} \frac{1}{(2t)^{d/2}} e^{-\frac{\|w\|_2^2}{4t}} dw \right) d\nu(t)$$
$$= \int_0^\infty e^{-t\|x\|_2^2} d\nu(t).$$

Substituting the form of λ_ψ into (24) we can write

$$\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 = \frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} \int_0^\infty e^{-\sigma^2\|w\|_2^2} \left(1 - \cos\left(\langle \mu_0 - \mu_1, w\rangle\right)\right) \frac{1}{(2t)^{d/2}} e^{-\frac{\|w\|_2^2}{4t}} \, d\nu(t)\, dw.$$

Applying Tonelli-Fubini theorem once again and using Lemma C.1 together with Euler's formula we obtain

$$\begin{aligned}
\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 &= \frac{2}{(2\pi)^{d/2}} \int_0^\infty \frac{1}{(2t)^{d/2}} \int_{\mathbb{R}^d} e^{-\frac{\|w\|_2^2}{2}(2\sigma^2 + \frac{1}{2t})} \left(1 - \cos\left(\langle \mu_0 - \mu_1, w\rangle\right)\right) dw\, d\nu(t) \\
&= \int_0^\infty \frac{2}{(4\sigma^2 t + 1)^{d/2}} d\nu(t) - \int_0^\infty \frac{2}{(4\sigma^2 t + 1)^{d/2}} e^{-\frac{t\|\mu_0 - \mu_1\|_2^2}{4\sigma^2 t + 1}} d\nu(t) \\
&= \int_0^\infty 2 \left(\frac{1}{1 + 4t\sigma^2} \right)^{d/2} \left(1 - \exp\left(-\frac{t\|\mu_0 - \mu_1\|_2^2}{1 + 4t\sigma^2} \right) \right) d\nu(t). \qquad (25)
\end{aligned}$$

(B) Lower bounding $\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2$ in terms of $\|\mu_0 - \mu_1\|_2^2$.

It follows from (25) that

$$\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 \geq \int_{t_0}^{t_1} 2 \left(\frac{1}{1 + 4t\sigma^2} \right)^{d/2} \left(1 - \exp\left(-\frac{t\|\mu_0 - \mu_1\|_2^2}{1 + 4t\sigma^2} \right) \right) d\nu(t),$$

where $0 < t_0 \leq t_1 < \infty$. Note that $1 - e^{-x} \geq \frac{x}{2}$ for $0 \leq x \leq 1$. Using this we get

$$1 - \exp\left(-\frac{t\|\mu_0 - \mu_1\|_2^2}{1 + 4t\sigma^2} \right) \geq \frac{t\|\mu_0 - \mu_1\|_2^2}{2(1 + 4t\sigma^2)}, \quad \forall t \in [t_0, t_1]$$

as long as

$$t_1 \|\mu_0 - \mu_1\|_2^2 \leq 1 + 4t_1\sigma^2. \qquad (26)$$

Thus, as long as (26) holds, we can lower bound the RKHS distance as:

$$\begin{aligned}
\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 &\geq \int_{t_0}^{t_1} \left(\frac{1}{1 + 4t\sigma^2} \right)^{d/2} \frac{t\|\mu_0 - \mu_1\|_2^2}{1 + 4t\sigma^2} \, d\nu(t) \\
&= \|\mu_0 - \mu_1\|_2^2 \int_{t_0}^{t_1} \frac{t}{(1 + 4t\sigma^2)^{(d+2)/2}} \, d\nu(t). \qquad (27)
\end{aligned}$$

Note that the function $t \mapsto \frac{t}{(1+4t\sigma^2)^{(d+2)/2}}$ monotonically increases on $[0, \frac{1}{2d\sigma^2}]$, reaches its global maximum at $t = \frac{1}{2d\sigma^2}$ and then decreases on $[\frac{1}{2d\sigma^2}, \infty)$. Thus we have

$$\int_{t_0}^{t_1} \frac{t}{(1 + 4t\sigma^2)^{(d+2)/2}} \, d\nu(t) \geq \beta \min\left\{ \frac{t_0}{(1 + 4t_0\sigma^2)^{(d+2)/2}}, \frac{t_1}{(1 + 4t_1\sigma^2)^{(d+2)/2}} \right\}.$$

Setting

$$\sigma^2 = \frac{1}{2t_1 d} \qquad (28)$$

yields that $t = t_1$ is the global maximum of the function $t \mapsto \frac{t}{\left(1 + \frac{2t}{t_1 d}\right)^{(d+2)/2}}$, in which case

$$\frac{t_0}{(1 + 4t_0\sigma^2)^{(d+2)/2}} \leq \frac{t_1}{(1 + 4t_1\sigma^2)^{(d+2)/2}}.$$

With this choice of σ^2 we get

$$\begin{aligned}
\int_{t_0}^{t_1} \frac{t}{(1 + 4t\sigma^2)^{(d+2)/2}} \, d\nu(t) &\geq \frac{\beta t_0}{\left(1 + \frac{2t_0}{t_1 d}\right)^{(d+2)/2}} \geq \frac{\beta t_0}{\left(1 + \frac{2}{d}\right)^{(d+2)/2}} \\
&= \beta t_0 \left(1 - \frac{2}{2 + d} \right)^{(d+2)/2} \geq \frac{\beta t_0}{e} \left(1 - \frac{2}{2 + d} \right), \qquad (29)
\end{aligned}$$

where we used the fact that $(1 - \frac{1}{x})^{x-1}$ monotonically decreases to $\frac{1}{e}$. Using (29) in (27), we obtain

$$\|\theta_0 - \theta_1\|_{\mathcal{H}_k}^2 \geq \frac{\beta t_0}{e} \left(1 - \frac{2}{2 + d} \right) \|\mu_0 - \mu_1\|_2^2. \qquad (30)$$

(C.1) Application of Theorem B.2: Choosing $\theta_0, \ldots, \theta_M$.

Now we are going to apply Theorem B.2. First of all, we need to choose $M + 1$ embeddings. Recall that Theorem B.2 requires these embeddings to be sufficiently distant from each other, while the corresponding distributions should be close. We will choose the embeddings $\{\theta_0, \ldots, \theta_M\}$ to be KMEs of Gaussian distributions $G(\mu_i, \sigma^2 I)$ for specific choice of $\sigma^2 > 0$ and $\mu_i \in \mathbb{R}^d$, $i = 0, \ldots, M$. Mean vectors $\{\mu_i\}_{i=0}^M$ will be constrained to live in the ball $B(c_\nu, n) := \left\{ x \in \mathbb{R}^d \colon \|x\|_2^2 \leq c_\nu / n \right\}$, where c_ν is a positive constant to be specified later. This guarantees that KL-divergences between the Gaussian distributions will remain small. At the same time, it was shown in (30) that the RKHS distance between embeddings θ_i and θ_j is lower bounded by the Euclidean distance between μ_i and μ_j. In other words, in order for the embeddings $\theta_0, \ldots, \theta_M$ to be sufficiently separated we need to make sure that the mean vectors $\mu_0, \ldots, \mu_M$ are not too close to each other. Summarizing, we face the problem of choosing a finite collection of pairwise distant points in the Euclidean ball. This question is closely related to the concepts of *packing* and *covering numbers*.

For any set $A \in \mathbb{R}^d$ its *ϵ-packing number* $M(A, \epsilon)$ is the largest number of points in A separated from each other by at least a distance of ϵ. An *ϵ-covering number* $N(A, \epsilon)$ of A

is the minimal number of balls of radius ϵ needed to cover A. Packing numbers are lower bounded by the covering numbers (Dudley, 1999, Theorem 1.2.1):

$$N(A,\epsilon) \le M(A,\epsilon).$$

Also, it is well known that the ϵ-covering number of a unit d-dimensional Euclidean ball is lower bounded by $\lfloor \epsilon^{-d} \rfloor$. Together, these facts state that we can find at least $\lfloor \epsilon^{-d} \rfloor$ points in the d-dimensional unit ball, which are at least ϵ away from each other. Similarly (just by a simple scaling) we can find at least $\lfloor \epsilon^{-d} \rfloor$ points in the d-dimensional ball of radius $R > 0$, which are at least $R \cdot \epsilon$ away from each other. Applying this fact to $B(c_\nu, n)$ we can finally argue that there are at least N^d points in $B(c_\nu, n)$ which are at least $N^{-1}\sqrt{c_\nu/n}$ away from each other, where $N \ge 3$ is an integer to be specified later. Now, take $M = N^d - 1 \ge 2$ (which explains the lower bound $N \ge 3$) and fix $\mu_0, \dots, \mu_M$ to be these $M+1$ points.

(C.2) Application of Theorem B.2: Lower bounding $\|\theta_i - \theta_j\|_{\mathcal{H}_k}$.

With this choice of parameters $\mu_0, \dots, \mu_M$, for any $0 \le i < j \le M$, we have

$$\|\theta_i - \theta_j\|^2_{\mathcal{H}_k} \ge \frac{c_\nu}{N^2 n} \frac{\beta t_0}{e} \left(1 - \frac{2}{2+d}\right),$$

where we used (30) and the lower bound on $\|\mu_i - \mu_j\|_2$. Setting

$$c_\nu = \frac{C}{\beta t_0} \tag{31}$$

for some $C > 0$ we obtain

$$\|\theta_i - \theta_j\|^2_{\mathcal{H}_k} \ge \frac{C}{N^2 e n} \left(1 - \frac{2}{2+d}\right).$$

This satisfies the first assumption of Theorem B.2 with $s := \frac{1}{2N}\sqrt{\frac{C}{en}\left(1 - \frac{2}{2+d}\right)}$.

(C.3) Application of Theorem B.2: Upper bounding $\mathrm{KL}(P_{\theta_i} \| P_{\theta_j})$.

Note that for any $0 \le i < j \le M$ we have

$$\mathrm{KL}\big(G^n(\mu_i, \sigma^2 I) \| G^n(\mu_j, \sigma^2 I)\big) = n \cdot \frac{\|\mu_i - \mu_j\|_2^2}{2\sigma^2} \le \frac{2c_\nu}{\sigma^2} = 4C\frac{t_1 d}{\beta t_0},$$

where the inequality holds since $\mu_i \in B(c_\nu, n)$ and the equality follows from (28) and (31). Here we used the fact that for any points x and y contained in a ball of radius R we obviously have $\|x - y\| \le 2R$. Also note that we chose $M = N^d - 1 \ge 2$ and thus

$$\log(M) = d\log(N) + \log(1 - N^{-d}) \ge d\log(N) + 1 - \frac{N^d}{N^d - 1} \ge d\log(N) - \frac{1}{N-1}$$

for $d \ge 1$, where we used the inequality $\log(x) \ge 1 - \frac{1}{x}$ which holds for $x \ge 0$. Taking

$$C = \frac{\beta t_0}{32 t_1}\left(\log N - \frac{1}{N-1}\right) \tag{32}$$

we get

$$4C\frac{t_1 d}{\beta t_0} = \frac{1}{8}\left(d\log N - \frac{d}{N-1}\right) \le \frac{1}{8}\left(d\log N - \frac{1}{N-1}\right) \le \frac{1}{8}\log(M)$$

for any $d \ge 1$. Concluding, we get

$$\mathrm{KL}\big(G^n(\mu_i, \sigma^2 I) \| G^n(\mu_j \| \sigma^2 I)\big) \le \frac{1}{8}\log(M)$$

and thus the second assumption of Theorem B.2 is satisfied with $\alpha = \frac{1}{8}$. Finally, it is easy to check that if we take $N = 5$ then condition (26) will be satisfied. Indeed,

$$t_1\|\mu_0 - \mu_1\|_2^2 \le \frac{4t_1 c_\nu}{n} = \frac{4t_1}{\beta t_0 n}\frac{\beta t_0}{32 t_1}\left(\log N - \frac{1}{N-1}\right) = \frac{1}{8n}\left(\log N - \frac{1}{N-1}\right),$$

while

$$1 + 4t_1\sigma^2 \ge 1.$$

Thus (26) holds whenever

$$\log N - \frac{1}{N-1} \le 8n,$$

which obviously holds for $N = 5$ and any $n \ge 1$.

To conclude the proof we insert (32) into $s := \frac{1}{2N}\sqrt{\frac{C}{en}\left(1 - \frac{2}{2+d}\right)}$, lower bound this value using $\frac{1}{8}\left(\log N - \frac{1}{N-1}\right) \ge \frac{4}{25}$, and notice that

$$\frac{\sqrt{M}}{1+\sqrt{M}}\left(1 - 2\alpha - \sqrt{\frac{2\alpha}{\log(M)}}\right) \ge \frac{1}{5}.$$

5.5 Proof of Theorem 9

The proof is based on the following result, which gives a closed form expression for the $L^2(\mathbb{R}^d)$ distance between embeddings of two discrete distributions supported on the same pair of points in $\mathbb{R}^d$.

Lemma 15 *Suppose $k(x,y) = \psi(x-y)$, where $\psi \in L^2(\mathbb{R}^d) \cap C_b(\mathbb{R}^d)$ is positive definite and k is characteristic. Define $P_0 = p_0\delta_x + (1-p_0)\delta_v$ and $P_1 = p_1\delta_x + (1-p_1)\delta_v$, where $0 < p_0 < 1$, $0 < p_1 < 1$, $x, v \in \mathbb{R}^d$, and $x \ne v$. Then*

$$\|\mu_k(P_0) - \mu_k(P_1)\|^2_{L^2(\mathbb{R}^d)} = C^\psi_{x,v}(p_0 - p_1)^2,$$

where

$$C^\psi_{x,v} := 2\left(\|\psi\|^2_{L^2(\mathbb{R}^d)} - \int_{\mathbb{R}^d}\psi(y)\psi(y+x-v)dy\right) > 0.$$

Proof We have

$$\begin{aligned}\|\mu_k(P_0) - \mu_k(P_1)\|^2_{L^2(\mathbb{R}^d)} &= (p_0-p_1)^2\,\|k(x,\cdot) - k(v,\cdot)\|^2_{L^2(\mathbb{R}^d)}\\ &= (p_0-p_1)^2\int_{\mathbb{R}^d}\big(\psi(x-y)-\psi(v-y)\big)^2 dy\\ &\overset{(\star)}{=} (p_0-p_1)^2\int_{\mathbb{R}^d}\big(\psi(y)-\psi(y+x-v)\big)^2 dy\\ &= 2(p_0-p_1)^2\int_{\mathbb{R}^d}\psi(y)\big(\psi(y)-\psi(y+x-v)\big)dy\\ &= 2(p_0-p_1)^2\left(\|\psi\|^2_{L^2(\mathbb{R}^d)} - \int_{\mathbb{R}^d}\psi(y)\psi(y+x-v)dy\right),\end{aligned}$$

where we used the symmetry of ψ in $(\star)$. Cauchy-Schwartz inequality states that

$$\int_{\mathbb{R}^d}\psi(y)\psi(y+x-v)dy \overset{(\star)}{\leq} \sqrt{\int_{\mathbb{R}^d}\psi^2(y)dy}\sqrt{\int_{\mathbb{R}^d}\psi^2(y+x-v)dy} = \|\psi\|^2_{L^2(\mathbb{R}^d)},$$

where the equality in $(\star)$ holds if and only if $\psi(y) = \lambda\,\psi(y+x-v)$ for some constant λ and all $y\in\mathbb{R}^d$. If we take $y = v-x$ the above condition implies $\psi(v-x) = \lambda\,\psi(0)$ and with $y=0$ we get $\psi(0) = \lambda\,\psi(x-v)$. Together these identities show that $\psi(0) = \lambda^2\,\psi(0)$. Since the kernel is characteristic and translation invariant, ψ is strictly positive definite, which means $\psi(0) > 0$. We conclude that $\lambda = \pm 1$. Assume that $\lambda = -1$. In this case $\psi(x-v) = -\psi(0) < 0$. Repeating the argument we can show that $\psi\big(2(x-v)\big) = \psi(0)$ and generally $\psi\big(m(x-v)\big) = (-1)^m\psi(0)$ for all $m\in\mathbb{N}$. Since $\psi\in L^2(\mathbb{R}^d)$ we need ψ^2 to be integrable on $\mathbb{R}^d$. Summarizing, we showed that a non-negative, integrable, and continuous function takes the same strictly positive value $\psi^2(0) > 0$ infinitely many times, leading to a contradiction. Arguing similarly for $\lambda = 1$ will result in a contradiction. This means the equality in $(\star)$ is never attained which concludes the proof. ■

The proof of Theorem 9 is carried out by simply repeating the proof of Theorem 1 but replacing (18) with the result in Lemma 15 and using $x - v := z$.

5.6 Proof of Corollary 10

The proof will be based on Theorem 9. The moment condition (15) on ν is sufficient for $\psi_\nu \in L^2(\mathbb{R}^d)$ to hold (see Remark A.3). Thus we only need to compute the expression $\|\psi_\nu\|^2_{L^2(\mathbb{R}^d)} - \int_{\mathbb{R}^d}\psi_\nu(y)\psi_\nu(y+z)dy$ appearing in (14). Note that

$$\int_{\mathbb{R}^d}\psi_\nu(y)\psi_\nu(y+z)dy = \int_{\mathbb{R}^d}\int_0^\infty\int_0^\infty e^{-t_1\|y\|_2^2 - t_2\|y+z\|_2^2}d\nu(t_1)d\nu(t_2)dy. \tag{33}$$

Since

$$\begin{aligned}\int_0^\infty\int_0^\infty\int_{\mathbb{R}^d} e^{-t_1\|y\|_2^2-t_2\|y+z\|_2^2}dy\,d\nu(t_1)d\nu(t_2) &= \int_0^\infty\int_0^\infty\left(\frac{\pi}{t_1+t_2}\right)^{d/2} e^{-\frac{t_1t_2\|z\|_2^2}{t_1+t_2}}d\nu(t_1)d\nu(t_2)\\ &\leq \nu([0,\infty))\int_0^\infty\left(\frac{\pi}{t_1}\right)^{d/2}d\nu(t_1) < \infty,\end{aligned}$$

we may apply Tonelli-Fubini theorem to switch the order of integration in (33) and get

$$\int_{\mathbb{R}^d}\psi_\nu(y)\psi_\nu(y+z)dy = \int_0^\infty\int_0^\infty\left(\frac{\pi}{t_1+t_2}\right)^{d/2} e^{-\frac{t_1t_2\|z\|_2^2}{t_1+t_2}}d\nu(t_1)d\nu(t_2).$$

Using this we get

$$\begin{aligned}\|\psi_\nu\|^2_{L^2(\mathbb{R}^d)} - \int_{\mathbb{R}^d}\psi_\nu(y)\psi_\nu(y+z)dy &= \int_0^\infty\int_0^\infty\left(\frac{\pi}{t_1+t_2}\right)^{d/2}\left(1-e^{-\frac{t_1t_2\|z\|_2^2}{t_1+t_2}}\right)d\nu(t_1)d\nu(t_2)\\ &\geq \left(\frac{\pi}{2\delta_1}\right)^{d/2}\int_{\delta_0}^{\delta_1}\int_{\delta_0}^{\delta_1}\left(1-e^{-\frac{t_1t_2\|z\|_2^2}{t_1+t_2}}\right)d\nu(t_1)d\nu(t_2).\end{aligned}$$

Since t_1 and t_2 are bounded below by $\delta_0 > 0$ we may take $\|z\|_2$ large enough so that the following will hold:

$$\|\psi_\nu\|^2_{L^2(\mathbb{R}^d)} - \int_{\mathbb{R}^d}\psi_\nu(y)\psi_\nu(y+z)dy \geq \frac{\beta^2}{2}\left(\frac{\pi}{2\delta_1}\right)^{d/2}.$$

5.7 Proof of Proposition 11

The proof is based on the following result, which provides a closed form expression for the $L^2(\mathbb{R}^d)$ distance between the embeddings of Gaussian measures.

Lemma 16 *Let θ_0 and θ_1 be KME of Gaussian measures $G(\mu_0,\sigma^2 I)$ and $G(\mu_1,\sigma^2 I)$ for $\mu_0,\mu_1\in\mathbb{R}^d$ and $\sigma^2 > 0$. Suppose $k(x,y) = \psi(x-y)$, where $\psi\in L^1(\mathbb{R}^d)\cap C_b(\mathbb{R}^d)$ is positive definite and and k is characteristic. Then*

$$\|\theta_0-\theta_1\|^2_{L^2(\mathbb{R}^d)} = 2\int_{\mathbb{R}^d}\left(1-e^{-i\langle w,\mu_0-\mu_1\rangle}\right)\big(\psi^\wedge(w)\big)^2 e^{-\sigma^2\|w\|^2}dw. \tag{34}$$

Proof First of all, note that $\psi\in L^2(\mathbb{R}^d)$ since $\psi\in L^1(\mathbb{R}^d)$ and ψ is bounded. This shows that $\theta_0,\theta_1\in L^2(\mathbb{R}^d)$. We will use P_0 and P_1 to denote the corresponding Gaussian distributions $G(\mu_0,\sigma^2 I)$ and $G(\mu_1,\sigma^2 I)$. By definition we have

$$\begin{aligned}\langle\theta_0,\theta_1\rangle_{L^2(\mathbb{R}^d)} &= \int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^d}k(x,y)\,dP_0(x)\right)\left(\int_{\mathbb{R}^d}k(z,y)\,dP_1(z)\right)dy\\ &= \int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}k(x,y)k(z,y)\,dP_0(x)\,dP_1(z)\right)dy.\end{aligned}$$

Using the fact that ψ is bounded (Wendland, 2005, Theorem 6.2) we get

$$\begin{aligned}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\big|k(x,y)k(z,y)\big|\,dy\,dP_0(x)\,dP_1(z) &\leq \psi(0)\iiint_{\mathbb{R}^d}\big|k(x,y)\big|\,dy\,dP_0(x)\,dP_1(z)\\ &= \psi(0)\iiint_{\mathbb{R}^d}\big|\psi(x-y)\big|dy\,dP_0(x)dP_1(z)\\ &= \psi(0)\|\psi\|_{L^1(\mathbb{R}^d)}\iint_{\mathbb{R}^d}dP_0(x)dP_1(z) < \infty.\end{aligned}$$

This allows us to use Tonelli-Fubini theorem (Dudley, 2002, Theorem 4.4.5) and get

$$\begin{aligned}\langle \theta_0, \theta_1 \rangle_{L^2(\mathbb{R}^d)} &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^d} k(x,y)k(z,y)dy\right) dP_0(x)dP_1(z)\\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^d} \psi(y)\psi(y+z-x)dy\right) dP_0(x)dP_1(z)\\ &\overset{(\star)}{=} \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^d}\psi(y)\int_{\mathbb{R}^d} e^{-i\langle y+z-x,w\rangle} d\Lambda_\psi(w)dy\right) dP_0(x)dP_1(z)\\ &= \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\psi(y)e^{-i\langle y+z-x,w\rangle} d\Lambda_\psi(w)dy\right) dP_0(x)dP_1(z),\end{aligned}$$

where we used (4) in $(\star)$. Since $\psi \in L^1(\mathbb{R}^d)$ we have

$$\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left|e^{-i\langle y,w\rangle}\psi(y)e^{-i\langle z-x,w\rangle}\right| dy\, d\Lambda_\psi(w) = \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}|\psi(y)|\, dy\, d\Lambda_\psi(w) < \infty$$

and thus we can use Tonelli-Fubini theorem to switch the order of integration:

$$\begin{aligned}\langle \theta_0, \theta_1 \rangle_{L^2(\mathbb{R}^d)} &= \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\psi(y)e^{-i\langle y+z-x,w\rangle} dy\, d\Lambda_\psi(w)\right) dP_0(x)dP_1(z)\\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^d}\psi^\wedge(w)e^{-i\langle z-x,w\rangle}\, d\Lambda_\psi(w)\right) dP_0(x)dP_1(z). \qquad (35)\end{aligned}$$

Next we are going to argue that if both ψ and $\psi^\wedge$ belong to $L^1(\mathbb{R}^d)$ (the latter is true as it follows from Wendland, 2005, Corollary 6.12) then $\psi^\wedge$ is the Radon-Nikodym derivative of Λ_ψ with respect to the Lebesgue measure. To this end, since $\psi^\wedge \in L^1(\mathbb{R}^d)$, Fourier inversion theorem (Wendland, 2005, Corollary 5.24) yields that for all $x \in \mathbb{R}^d$, the following holds:

$$\psi(x) = \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} e^{i\langle w,x\rangle}\psi^\wedge(w)dw.$$

On the other hand, using (4) and Lemma C.2, we also have

$$\psi(x) = \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} e^{i\langle w,x\rangle} d\Lambda_\psi(w).$$

These two identities show that for all $x \in \mathbb{R}^d$

$$\int_{\mathbb{R}^d} e^{i\langle w,x\rangle}\psi^\wedge(w)dw = \int_{\mathbb{R}^d} e^{i\langle w,x\rangle} d\Lambda_\psi(w). \qquad (36)$$

Note that since k is translation invariant and characteristic, ψ is a strictly positive definite function (Sriperumbudur et al., 2010, Section 3.4) and therefore it follows from (Wendland, 2005, Theorem 6.11) that $\psi^\wedge$ is non-negative (and nonvanishing). Since $\psi^\wedge \in L^1(\mathbb{R}^d)$ we conclude that $\psi^\wedge$ is the Radon-Nikodym derivative of a finite non-negative measure $\mathcal{T}_\psi$ on $\mathbb{R}^d$, which is absolutely continuous with respect to the Lebesgue measure. (36) (after proper normalization) shows that the characteristic functions of measures Λ_ψ and $\mathcal{T}_\psi$ coincide. We finally conclude from (Dudley, 2002, Theorem 9.5.1) that $\Lambda_\psi = \mathcal{T}_\psi$, which means that Λ_ψ is absolutely continuous with respect to the Lebesgue measure and has a density $\psi^\wedge$.

Returning to (35) we can write it as

$$\langle \theta_0, \theta_1 \rangle_{L^2(\mathbb{R}^d)} = \int_{\mathbb{R}^d}\int_{\mathbb{R}^d}\left(\int_{\mathbb{R}^d}\left(\psi^\wedge(w)\right)^2 e^{-i\langle z-x,w\rangle}\, dw\right) dP_0(x)dP_1(z).$$

We already showed that $\psi \in L^2(\mathbb{R}^d)$. From Plancherel's theorem (Wendland, 2005, Corollary 5.25), we have $\|\psi\|_{L^2(\mathbb{R}^d)} = \|\psi^\wedge\|_{L^2(\mathbb{R}^d)}$ and thus $\psi^\wedge \in L^2(\mathbb{R}^d)$. Another application of Tonelli-Fubini theorem yields

$$\begin{aligned}\langle \theta_0, \theta_1 \rangle_{L^2(\mathbb{R}^d)} &= \int_{\mathbb{R}^d}\left(\psi^\wedge(w)\right)^2\left(\int_{\mathbb{R}^d}\int_{\mathbb{R}^d} e^{-i\langle z-x,w\rangle} dP_0(x)dP_1(z)\right) dw\\ &= \int_{\mathbb{R}^d}\left(\psi^\wedge(w)\right)^2 e^{i\langle w,\mu_0-\mu_1\rangle} e^{-\sigma^2\|w\|_2^2} dw.\end{aligned}$$

Noticing that the Fourier transform of a real and even function is also even, we conclude that $\psi^\wedge$ is also even. This finishes the proof since $\|\theta_0-\theta_1\|^2_{L^2(\mathbb{R}^d)} = \|\theta_1\|^2_{L^2(\mathbb{R}^d)} + \|\theta_0\|^2_{L^2(\mathbb{R}^d)} - 2\langle \theta_0, \theta_1 \rangle_{L^2(\mathbb{R}^d)}$. ■

Now we turn to the proof of Theorem 11. We will write $\tilde{\Lambda}_\psi$ to denote a non-negative finite measure, absolutely continuous with respect to the Lebesgue measure with density $(2\pi)^{d/2}(\psi^\wedge)^2$. Then (34) in Lemma 16 can be written as

$$\begin{aligned}\|\theta_0-\theta_1\|^2_{L^2(\mathbb{R}^d)} &= \frac{2}{(2\pi)^{d/2}}\int_{\mathbb{R}^d}\left(1-e^{i\langle w,\mu_0-\mu_1\rangle}\right) e^{-\sigma^2\|w\|_2^2} d\tilde{\Lambda}_\psi(w).\\ &= \frac{2}{(2\pi)^{d/2}}\int_{\mathbb{R}^d}\left(1-\cos(\langle w,\mu_0-\mu_1\rangle)\right) e^{-\sigma^2\|w\|_2^2} d\tilde{\Lambda}_\psi(w),\end{aligned}$$

which is exactly of the form in Lemma 14 but with Λ_ψ replaced by $\tilde{\Lambda}_\psi$. From the proof of Lemma 16, since $\psi^\wedge$ is even and non-vanishing, the corresponding measure $\tilde{\Lambda}_\psi$ is symmetric and $\mathrm{supp}(\tilde{\Lambda}_\psi) = \mathbb{R}^d$. The result therefore follows by carrying out the proof of Proposition 3 verbatim but for replacing Λ_ψ with $\tilde{\Lambda}_\psi$.

5.8 Proof of Theorem 13

The proof will closely follow that of Theorem 8. Let $G(\mu_0, \sigma^2 I)$ and $G(\mu_1, \sigma^2 I)$ be two d-dimensional Gaussian distributions with mean vectors $\mu_0, \mu_1 \in \mathbb{R}^d$ and variance $\sigma^2 > 0$. Let θ_0 and θ_1 denote the kernel mean embeddings of $G(\mu_0, \sigma^2 I)$ and $G(\mu_1, \sigma^2 I)$ respectively.

(A) Deriving a closed form expression for $\|\theta_0-\theta_1\|^2_{L^2(\mathbb{R}^d)}$.

The condition in (15) ensures that $\psi_\nu \in L^2(\mathbb{R}^d)$ (see Remark A.3). In fact, using a similar argument it can be shown that $\psi_\nu \in L^1(\mathbb{R}^d)$. Also it is easy to verify that $\psi_\nu \in C_b(\mathbb{R}^d)$. Next, under this moment condition we may apply Tonelli-Fubini theorem to compute the Fourier transform of ψ_ν:

$$\psi_\nu^\wedge(w) = \frac{1}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} e^{-i\langle w,x\rangle}\int_0^\infty e^{-t\|x\|_2^2} d\nu(t)dx = \int_0^\infty \frac{1}{(2t)^{d/2}} e^{-\frac{\|w\|_2^2}{4t}} d\nu(t). \qquad (37)$$

It is immediate to see that $\psi_\nu^\wedge \in L_1(\mathbb{R}^d)$. Therefore Lemma 16 yields

$$\|\theta_0 - \theta_1\|^2_{L^2(\mathbb{R}^d)} = 2\int_{\mathbb{R}^d} \left(1 - e^{-i\langle w, \mu_0 - \mu_1\rangle}\right) \left(\psi_\nu^\wedge(w)\right)^2 e^{-\sigma^2\|w\|_2^2} dw. \tag{38}$$

Denoting $G(w) := \psi_\nu^\wedge(w) e^{-\frac{\sigma^2\|w\|_2^2}{2}}$ and using a well-known property of the Fourier transform we get

$$(G^2)^\wedge(\tau) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-i\langle w,\tau\rangle} G^2(w) dw = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} G^\wedge(x) G^\wedge(\tau - x) dx. \tag{39}$$

Next we compute the Fourier transform of G using (37):

$$G^\wedge(x) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-i\langle w,x\rangle} \left(\int_0^\infty \frac{1}{(2t)^{d/2}} e^{-\frac{\|w\|_2^2}{4t}} d\nu(t)\right) e^{-\frac{\sigma^2\|w\|_2^2}{2}} dw.$$

Using the moment condition on ν we have

$$\begin{aligned}\int_0^\infty \frac{1}{(4\pi t)^{d/2}} \int_{\mathbb{R}^d} e^{-\frac{\|x\|_2^2}{4t}} e^{-\frac{\sigma^2\|x\|_2^2}{2}} dx d\nu(t) &= \frac{1}{(2\sigma^2)^{d/2}} \int_0^\infty \left(t + \frac{1}{2\sigma^2}\right)^{-d/2} d\nu(t) \\ &\le \frac{1}{(2\sigma^2)^{d/2}} \int_0^\infty t^{-d/2} d\nu(t) < \infty. \end{aligned} \tag{40}$$

This allows us to use Tonelli-Fubini theorem and write

$$\begin{aligned} G^\wedge(x) &= \frac{1}{(2\pi)^{d/2}} \int_0^\infty \frac{1}{(2t)^{d/2}} \int_{\mathbb{R}^d} e^{-i\langle w,x\rangle} e^{-\frac{\|w\|_2^2}{4t}} e^{-\frac{\sigma^2\|w\|_2^2}{2}} dw\, d\nu(t) \\ &= \int_0^\infty \frac{1}{(2t\sigma^2+1)^{d/2}} \exp\left(-\frac{t\|x\|_2^2}{2t\sigma^2+1}\right) d\nu(t). \end{aligned}$$

Returning to (39) and denoting $\Delta_1 := 2t_1\sigma^2 + 1$, $\Delta_2 := 2t_2\sigma^2 + 1$ we obtain

$$(G^2)^\wedge(\tau) = \int_{\mathbb{R}^d} \int_0^\infty \int_0^\infty \frac{1}{(2\pi\Delta_1\Delta_2)^{d/2}} \exp\left(-\frac{t_1\|x\|_2^2}{\Delta_1} - \frac{t_2\|\tau - x\|_2^2}{\Delta_2}\right) d\nu(t_1)\, d\nu(t_2)\, dx.$$

Using a simple identity

$$a\|x\|_2^2 + b\|x - y\|_2^2 = (a+b)\left\|x - \frac{b}{a+b}y\right\|_2^2 + \frac{ab}{a+b}\|y\|_2^2,$$

which holds for any $x, y \in \mathbb{R}^d$ and $a, b \in \mathbb{R}$ with $a + b \neq 0$, we obtain

$$\begin{aligned}(G^2)^\wedge(\tau) = \int_{\mathbb{R}^d} \int_0^\infty \int_0^\infty \frac{1}{(2\pi\Delta_1\Delta_2)^{d/2}} \exp\left(-\left(\frac{t_1}{\Delta_1} + \frac{t_2}{\Delta_2}\right)\left\|x - \frac{t_2\Delta_1\tau}{t_1\Delta_2 + t_2\Delta_1}\right\|_2^2\right) \\ \times \exp\left(-\frac{t_1t_2\|\tau\|_2^2}{t_1\Delta_2 + t_2\Delta_1}\right) d\nu(t_1)\, d\nu(t_2)\, dx.\end{aligned}$$

Using an argument similar to (40) we can show that Tonelli-Fubini theorem is applicable to the r.h.s. of the above equation. Therefore, changing the order of integration we get

$$(G^2)^\wedge(\tau) = \int_0^\infty \int_0^\infty \left(\frac{1}{2(t_1\Delta_2 + t_2\Delta_1)}\right)^{d/2} \exp\left(-\frac{t_1t_2\|\tau\|_2^2}{t_1\Delta_2 + t_2\Delta_1}\right) d\nu(t_1)\, d\nu(t_2).$$

Noticing that $(G^2)^\wedge(0) = \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} G^2(w)\, dw$ and returning to (38) we get

$$\begin{aligned}\|\theta_0 - \theta_1\|^2_{L^2(\mathbb{R}^d)} &= 2(2\pi)^{d/2} \left((G^2)^\wedge(0) - (G^2)^\wedge(\mu_0 - \mu_1)\right) \\ &= 2\int_0^\infty \int_0^\infty \left(\frac{\pi}{t_1\Delta_2 + t_2\Delta_1}\right)^{d/2} \left(1 - \exp\left(-\frac{t_1t_2\|\mu_0 - \mu_1\|_2^2}{t_1\Delta_2 + t_2\Delta_1}\right)\right) d\nu(t_1) d\nu(t_2).\end{aligned}$$

(B) Lower bounding $\|\theta_0 - \theta_1\|^2_{L^2(\mathbb{R}^d)}$ in terms of $\|\mu_0 - \mu_1\|_2^2$.

Consider

$$\begin{aligned}&\|\theta_0 - \theta_1\|^2_{L_2(\mathbb{R}^d)} \\ &\ge 2\int_{\delta_0}^{\delta_1} \int_{\delta_0}^{\delta_1} \left(\frac{\pi}{t_1\Delta_2 + t_2\Delta_1}\right)^{d/2} \left(1 - \exp\left(-\frac{t_1t_2\|\mu_0 - \mu_1\|_2^2}{t_1\Delta_2 + t_2\Delta_1}\right)\right) d\nu(t_1) d\nu(t_2) \\ &= 2\int_{\delta_0}^{\delta_1} \int_{\delta_0}^{\delta_1} \left(\frac{\pi}{4t_1t_2\sigma^2 + t_1 + t_2}\right)^{d/2} \left(1 - \exp\left(-\frac{t_1t_2\|\mu_0 - \mu_1\|_2^2}{4t_1t_2\sigma^2 + t_1 + t_2}\right)\right) d\nu(t_1) d\nu(t_2).\end{aligned}$$

Using the fact that $1 - e^{-x} \ge \frac{x}{2}$ for $0 \le x \le 1$, we obtain

$$\|\theta_0 - \theta_1\|^2_{L^2(\mathbb{R}^d)} \ge \int_{\delta_0}^{\delta_1} \int_{\delta_0}^{\delta_1} \left(\frac{\pi}{4t_1t_2\sigma^2 + t_1 + t_2}\right)^{d/2} \frac{t_1t_2\|\mu_0 - \mu_1\|_2^2}{4t_1t_2\sigma^2 + t_1 + t_2} d\nu(t_1)\, d\nu(t_2) \tag{41}$$

whenever

$$\frac{t_1t_2\|\mu_0 - \mu_1\|_2^2}{4t_1t_2\sigma^2 + t_1 + t_2} \le 1.$$

Note that the expression on the left hand side of the previous inequality is increasing both in t_1 and t_2. This means that for $t_1, t_2 \in [\delta_0, \delta_1]$ we have:

$$\frac{t_1t_2\|\mu_0 - \mu_1\|_2^2}{4t_1t_2\sigma^2 + t_1 + t_2} \le \frac{\delta_1\|\mu_0 - \mu_1\|_2^2}{4\delta_1\sigma^2 + 2}$$

and thus (41) holds whenever

$$\delta_1\|\mu_0 - \mu_1\|_2^2 \le 4\delta_1\sigma^2 + 2$$

which will be satisfied later. (41) can be rewritten as

$$\begin{aligned}\|\theta_0 - \theta_1\|^2_{L^2(\mathbb{R}^d)} &\ge \int_{\delta_0}^{\delta_1} \int_{\delta_0}^{\delta_1} \frac{(\pi)^{d/2} t_1t_2\|\mu_0 - \mu_1\|_2^2}{(4t_1t_2\sigma^2 + t_1 + t_2)^{d/2+1}} d\nu(t_1)\, d\nu(t_2) \\ &= \int_{\delta_0}^{\delta_1} \left(\frac{\pi}{t_1 + t_2}\right)^{d/2} \int_{\delta_0}^{\delta_1} \frac{\frac{t_1t_2}{t_1+t_2}\|\mu_0 - \mu_1\|_2^2}{\left(4\frac{t_1t_2}{t_1+t_2}\sigma^2 + 1\right)^{d/2+1}} d\nu(t_1)\, d\nu(t_2) \\ &\ge \left(\frac{\pi}{2\delta_1}\right)^{d/2} \int_{\delta_0}^{\delta_1} \int_{\delta_0}^{\delta_1} \frac{S(t_1,t_2)\|\mu_0 - \mu_1\|_2^2}{\left(4S(t_1,t_2)\sigma^2 + 1\right)^{d/2+1}} d\nu(t_1)\, d\nu(t_2),\end{aligned} \tag{42}$$

where $S(t_1,t_2) := \frac{t_1 t_2}{t_1+t_2}$. Note that $S(t_1,t_2)$ takes values in $[\frac{\delta_0}{2}, \frac{\delta_1}{2}]$ as t_1 and t_2 varies in $[\delta_0, \delta_1]$. We can now repeat part of the proof of Theorem 8 where we showed that the function $t \mapsto \frac{t}{(1+4t\sigma^2)^{(d+2)/2}}$ monotonically increases on $[0, \frac{1}{2d\sigma^2}]$, reaches its global maximum at $t = \frac{1}{2d\sigma^2}$, and then decreases on $[\frac{1}{2d\sigma^2}, \infty)$. Using this fact we have

$$\int_{\delta_0}^{\delta_1}\int_{\delta_0}^{\delta_1} \frac{S(t_1,t_2)}{\left(4S(t_1,t_2)\sigma^2+1\right)^{\frac{d}{2}+1}} d\nu(t_1)d\nu(t_2) \geq \frac{\beta^2}{2}\min\left\{\frac{\delta_0}{(1+2\delta_0\sigma^2)^{\frac{d}{2}+1}}, \frac{\delta_1}{(1+2\delta_1\sigma^2)^{\frac{d}{2}+1}}\right\}.$$

By setting $\sigma^2 := \frac{1}{\delta_1 d}$ we ensure that $t = \frac{\delta_1}{2}$ is the global maximum of the function $t \mapsto \frac{t}{\left(1+\frac{4t}{\delta_1 d}\right)^{\frac{d}{2}+1}}$ and thus $\frac{\delta_0}{(1+2\delta_0\sigma^2)^{\frac{d}{2}+1}} \leq \frac{\delta_1}{(1+2\delta_1\sigma^2)^{\frac{d}{2}+1}}$. Combining this with (42) we have

$$\begin{aligned}
\|\theta_0-\theta_1\|^2_{L^2(\mathbb{R}^d)} &\geq \frac{\beta^2\delta_0}{2\left(1+\frac{2\delta_0}{\delta_1 d}\right)^{\frac{d}{2}+1}}\left(\frac{\pi}{2\delta_1}\right)^{d/2}\|\mu_0-\mu_1\|_2^2 \\
&\geq \frac{\beta^2\delta_0}{2(1+\frac{2}{d})^{\frac{d}{2}+1}}\left(\frac{\pi}{2\delta_1}\right)^{d/2}\|\mu_0-\mu_1\|_2^2 \geq \frac{\beta^2\delta_0}{2e}\left(1-\frac{2}{2+d}\right)\left(\frac{\pi}{2\delta_1}\right)^{d/2}\|\mu_0-\mu_1\|_2^2,
\end{aligned}$$

where we used an analysis similar to (29).

(C) Application of Theorem B.2.

We finish by repeating all the remaining steps carried out in the proof of Theorem 8 (steps C.1, C.2, and C.3), where we set

$$\sigma^2 := \frac{1}{\delta_1 d} \quad \text{and} \quad c_\nu := \frac{1}{16\delta_1}\left(\log N - \frac{1}{N-1}\right).$$

Acknowledgements

The authors thank the action editor and two anonymous reviewers for their detailed comments, which helped to improve the presentation. The authors would also like to thank David Lopez-Paz, Jonas Peters, Bernhard Schölkopf, and Carl-Johann Simon-Gabriel for useful discussions.

Appendix A. $\sqrt{n}$-consistency of $\mu_k(P_n)$

In the following, we present a general result whose special cases establishes the convergence rate of $n^{-1/2}$ for $\|\mu_k(P_n)-\mu_k(P)\|_{\mathcal{F}}$ when $\mathcal{F}=\mathcal{H}_k$ and $\mathcal{F}=L^2(\mathbb{R}^d)$.

Proposition A.1 *Let $(X_i)_{i=1}^n$ be random samples drawn i.i.d. from P defined on a separable topological space $\mathcal{X}$. Suppose $r:\mathcal{X}\to H$ is continuous and*

$$\sup_{x\in\mathcal{X}}\|r(x)\|_H^2 \leq C_k < \infty, \tag{43}$$

where H is a separable Hilbert space of real-valued functions. Then for any $0<\delta\leq 1$ with probability at least $1-\delta$ we have

$$\left\|\int_{\mathcal{X}} r(x)\,dP_n(x) - \int_{\mathcal{X}} r(x)\,dP(x)\right\|_H \leq \sqrt{\frac{C_k}{n}} + \sqrt{\frac{2C_k\log(1/\delta)}{n}}.$$

Proof Note that $r:\mathcal{X}\to H$ is a H-valued measurable function as r is continuous and H is separable (Steinwart and Christmann, 2008, Lemma A.5.18). The condition in (43) ensures that $\int\|r(x)\|_H dQ(x) \leq \sqrt{C_k} < \infty$ for any $Q\in M_+^1(\mathcal{X})$ and therefore $\int r(x)\,dQ(x)$ is well defined as a Bochner integral for any $Q\in M_+^1(\mathcal{X})$ (Diestel and Uhl, 1977, Theorem 2, p.45). By McDiarmid's inequality, it is easy to verify that with probability at least $1-\delta$,

$$\begin{aligned}
\left\|\int_{\mathcal{X}} r(x)\,dP_n(x) - \int_{\mathcal{X}} r(x)\,dP(x)\right\|_H \leq \mathbb{E}\left\|\int_{\mathcal{X}} r(x)\,dP_n(x) - \int_{\mathcal{X}} r(x)\,dP(x)\right\|_H \\
+\sqrt{\frac{2C_k\log(1/\delta)}{n}},
\end{aligned} \tag{44}$$

where

$$\begin{aligned}
&\mathbb{E}\left\|\int_{\mathcal{X}} r(x)\,dP_n(x) - \int_{\mathcal{X}} r(x)\,dP(x)\right\|_H \leq \sqrt{\mathbb{E}\left\|\int_{\mathcal{X}} r(x)\,dP_n(x) - \int_{\mathcal{X}} r(x)\,dP(x)\right\|_H^2} \\
&= \sqrt{\mathbb{E}\left\|\int_{\mathcal{X}} r(x)\,dP_n(x)\right\|_H^2 + \left\|\int_{\mathcal{X}} r(x)\,dP(x)\right\|_H^2 - 2\mathbb{E}\left\langle\int_{\mathcal{X}} r(x)\,dP_n(x), \int_{\mathcal{X}} r(x)\,dP(x)\right\rangle_H} \\
&= \sqrt{\mathbb{E}\left\|\int_{\mathcal{X}} r(x)\,dP_n(x)\right\|_H^2 + \left\|\int_{\mathcal{X}} r(x)\,dP(x)\right\|_H^2 - \frac{2}{n}\sum_{i=1}^n \mathbb{E}\left\langle r(X_i), \int_{\mathcal{X}} r(x)\,dP(x)\right\rangle_H}.
\end{aligned} \tag{45}$$

To simplify the r.h.s. of (45), we make the following observation. Note that for any $g\in H$, $T_g : H\to\mathbb{R}$, $f\mapsto\langle g,f\rangle_H$ is a bounded linear functional on H. Choose $f=\int_{\mathcal{X}} r(y)\,dP(y)$. It follows from (Diestel and Uhl, 1977, Theorem 6, p.47) that

$$\left\langle g, \int_{\mathcal{X}} r(y)\,dP(y)\right\rangle_H = T_g\left(\int_{\mathcal{X}} r(y)\,dP(y)\right) = \int_{\mathcal{X}} T_g(r(y))\,dP(y) = \int_{\mathcal{X}}\langle g, r(y)\rangle_H\,dP(y). \tag{46}$$

Applying (46) to the third term in the r.h.s. of (45) with $g=\int_{\mathcal{X}} r(x)\,dP(x)$, we obtain

$$\mathbb{E}\left\langle r(X_i), \int_{\mathcal{X}} r(x)\,dP(x)\right\rangle_H = \int_{\mathcal{X}}\langle r(x_i), g\rangle_H\,dP(x_i) = \left\langle\int_{\mathcal{X}} r(x_i)\,dP(x_i), g\right\rangle_H = \|g\|_H^2$$

and so (45) reduces to

$$\mathbb{E}\left\|\int_{\mathcal{X}} r(x)\,dP_n(x) - \int_{\mathcal{X}} r(x)\,dP(x)\right\|_H \leq \sqrt{\mathbb{E}\left\|\int_{\mathcal{X}} r(x)\,dP_n(x)\right\|_H^2 - \left\|\int_{\mathcal{X}} r(x)\,dP(x)\right\|_H^2}. \tag{47}$$

Consider

$$\mathbb{E}\left\|\int_{\mathcal{X}} r(x)\,dP_n(x)\right\|_H^2 = \mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n r(X_i)\right\|_H^2 = \frac{1}{n^2}\sum_{i,j=1}^n \mathbb{E}\langle r(X_i), r(X_j)\rangle_H$$
$$= \frac{1}{n^2}\sum_{i=j}\mathbb{E}\langle r(X_i), r(X_j)\rangle_H + \frac{1}{n^2}\sum_{i\neq j}\mathbb{E}\langle r(X_i), r(X_j)\rangle_H$$
$$= \frac{1}{n}\mathbb{E}_{X\sim P}\|r(X)\|_H^2 + \frac{n-1}{n}\mathbb{E}_{X\sim P, Y\sim P}\langle r(X), r(Y)\rangle_H. \qquad (48)$$

Using (46), the second term in (48) can be equivalently written as

$$\mathbb{E}_{X\sim P, Y\sim P}\langle r(X), r(Y)\rangle_H = \int_{\mathcal{X}}\left(\int_{\mathcal{X}}\langle r(x), r(y)\rangle_H\, dP(y)\right) dP(x)$$
$$\overset{(\star)}{=} \int_{\mathcal{X}}\left\langle r(x), \int_{\mathcal{X}} r(y)\,dP(y)\right\rangle_H dP(x)$$
$$\overset{(\star)}{=} \left\langle \int_{\mathcal{X}} r(x)\,dP(x), \int_{\mathcal{X}} r(y)\,dP(y)\right\rangle_H x = \left\|\int_{\mathcal{X}} r(x)\,dP(x)\right\|_H^2,$$

where we invoked (46) in ($\star$). Combining the above with (48) and using the result in (47) yields

$$\mathbb{E}\left\|\int_{\mathcal{X}} r(x)\,dP_n(x) - \int_{\mathcal{X}} r(x)\,dP(x)\right\|_H \leq \sqrt{\frac{\mathbb{E}_{X\sim P}\|r(X)\|_H^2 - \|\int_{\mathcal{X}} r(x)\,dP(x)\|_H^2}{n}} \leq \sqrt{\frac{C_k}{n}}$$

and the result follows. ■

Remark A.2 *Suppose H is an RKHS with a reproducing kernel k that is continuous and satisfies $\sup_{x\in\mathcal{X}} k(x,x) < \infty$. Choosing $r(x) = k(\cdot, x)$, $x \in \mathcal{X}$ in Proposition A.1 yields a concentration inequality for $\|\mu_k(P_n) - \mu_k(P)\|_H$ with $C_k := \sup_{x\in\mathcal{X}} k(x,x)$, thereby establishing a convergence rate of $n^{-1/2}$ for $\|\mu_k(P_n) - \mu_k(P)\|_H$. While such a result has already appeared in Smola et al. (2007, Theorem 2), Gretton et al. (2012) and Lopez-Paz et al. (2015), the result derived from Proposition A.1 improves upon them by providing better constants. While all these works including Proposition A.1 are based on McDiarmid's inequality (see (44)), the latter obtains better constants by carefully bounding the expectation term in (44). It is easy to verify that $C_k = 1$ for Gaussian and $C_k = C_M$ for mixture of Gaussian kernels, $C_k = c^{-2\gamma}$ for inverse multiquadrics, and $C_k = 1$ for Matérn kernels.*

Remark A.3 *Assuming $\mathcal{X} = \mathbb{R}^d$, $H = L^2(\mathbb{R}^d)$ and $r(x) = k(\cdot, x)$, $x \in \mathbb{R}^d$, where k is a continuous positive definite kernel on $\mathbb{R}^d$, Proposition A.1 establishes a convergence rate of $n^{-1/2}$ for $\|\mu_k(P_n) - \mu_k(P)\|_{L^2(\mathbb{R}^d)}$ under the condition that $\sup_{x\in\mathbb{R}^d}\|k(x,\cdot)\|_{L^2(\mathbb{R}^d)}^2 < \infty$. If k is translation invariant on $\mathbb{R}^d$, i.e., $k(x,y) = \psi(x-y)$, $x, y \in \mathbb{R}^d$ where $\psi \in C(\mathbb{R}^d)$ is positive definite, then $\psi \in L^2(\mathbb{R}^d)$ ensures that $\sup_{x\in\mathbb{R}^d}\|k(x,\cdot)\|_{L^2(\mathbb{R}^d)}^2 = \sup_{x\in\mathbb{R}^d}\|\psi(x-\cdot)\|_{L^2(\mathbb{R}^d)}^2 = \|\psi\|_{L^2(\mathbb{R}^d)}^2$ and therefore Propositions A.1 holds with $C_k := \|\psi\|_{L^2(\mathbb{R}^d)}^2$. On the other hand, for radial kernels on $\mathbb{R}^d$, i.e., kernels of the form in (5), the condition in (43) is ensured if*

$$\int_0^\infty t^{-d/2} d\nu(t) < \infty \qquad (49)$$

since

$$\sup_{x\in\mathbb{R}^d}\|k(x,\cdot)\|_{L^2(\mathbb{R}^d)}^2 = \sup_{x\in\mathbb{R}^d}\int_{\mathbb{R}^d}\left(\int_0^\infty e^{-t\|x-y\|^2} d\nu(t)\right)^2 dy$$
$$\overset{(\dagger)}{\leq} \nu([0,\infty))\sup_{x\in\mathbb{R}^d}\int_{\mathbb{R}^d}\int_0^\infty e^{-2t\|x-y\|^2} d\nu(t) dy$$
$$\overset{(\ddagger)}{=} \nu([0,\infty))\sup_{x\in\mathbb{R}^d}\int_0^\infty\int_{\mathbb{R}^d} e^{-2t\|x-y\|^2} dy\, d\nu(t) = \frac{\nu([0,\infty))}{(2/\pi)^{d/2}}\int_0^\infty \frac{d\nu(t)}{t^{d/2}},$$

where we used Jensen's inequality in (†) and Fubini's theorem in (‡). Therefore the bound in Proposition A.1 holds with $C_k := \nu([0,\infty))\int_0^\infty \left(\frac{\pi}{2t}\right)^{d/2} d\nu(t)$. (49) is satisfied by Gaussian, mixture of Gaussian and Matérn (see Sriperumbudur, 2016, Equation 6.17) kernels. For inverse multiquadrics, while (49) holds for $\gamma > d/2$ since $\nu = c^{-2\gamma}Gamma(\gamma, c^2)$ (see Wendland, 2005, Theorem 7.15), in fact the condition in (43) holds for $\gamma > d/4$ (see Lemma C.4).

Appendix B. Minimax Lower Bounds and Le Cam's Method

Let Θ be a set of parameters (or functions) containing the element θ which we want to estimate. Assume there is a class $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ of probability measures on $\mathbb{R}^d$ indexed by Θ. Suppose $d\colon \Theta\times\Theta \to [0,\infty)$ is a metric on Θ. Le Cam's method provides a lower bound on the minimax probability, $\inf_{\hat{\theta}_n}\sup_{\theta\in\Theta} P_\theta^n(d(\hat{\theta}_n, \theta) \geq s)$ for $s > 0$, where the infimum is taken over all possible estimators $\hat{\theta}_n\colon \mathbb{R}^d \to \Theta$ that are constructed from an i.i.d. sample $(X_i)_{i=1}^n$ drawn from P_θ. The following two results which we used throughout this work are based on Le Cam's method and they provide a lower bound on the minimax probability. The first one follows from Theorem 2.2 and Equation (2.9) of Tsybakov (2008). It requires a construction of two sufficiently distant elements of the set Θ corresponding to the probability distributions similar in the Kullback-Leibler (KL) divergence sense, where the KL divergence between two distributions P and Q with P absolutely continuous w.r.t. Q is defined as $\mathrm{KL}(P\|Q) = \int \log\frac{dP}{dQ}\, dP$.

Theorem B.1 (Lower bound based on two hypotheses) *Assume Θ contains θ_0 and θ_1 such that $d(\theta_0, \theta_1) \geq 2s$ and $\mathrm{KL}(P_{\theta_0}^n\|P_{\theta_1}^n) \leq \alpha$ for some $s > 0$ and $0 < \alpha < \infty$. Then*

$$\inf_{\hat{\theta}_n}\sup_{\theta\in\Theta} P_\theta^n\left\{d(\hat{\theta}_n, \theta) \geq s\right\} \geq \max\left(\frac{1}{4}e^{-\alpha}, \frac{1-\sqrt{\alpha/2}}{2}\right).$$

Note that the second condition of the theorem bounds the distance between the n-fold product distributions by a constant independent of n. Recalling the chain rule of the KL-divergence, which states that $\mathrm{KL}(P_{\theta_0}^n\|P_{\theta_1}^n) = n\cdot\mathrm{KL}(P_{\theta_0}\|P_{\theta_1})$, we can see that this condition is rather restrictive and requires the marginal distributions to satisfy $\mathrm{KL}(P_{\theta_0}\|P_{\theta_1}) = O(n^{-1})$. This condition is slightly relaxed in the following result, which follows from Theorem 2.5 of Tsybakov (2008).

Theorem B.2 (Lower bound based on many hypotheses) *Assume $M \geq 2$ and suppose that there exist $\theta_0, \ldots, \theta_M \in \Theta$ such that (i) $d(\theta_i, \theta_j) \geq 2s > 0$, $\forall\, 0 \leq i < j \leq M$; (ii) P_{θ_j} is absolutely continuous w. r. t. P_{θ_0} for all $j = 1, \ldots, M$, and $\frac{1}{M}\sum_{i=1}^{M} \mathrm{KL}(P_{\theta_j}^n \| P_{\theta_0}^n) \leq \alpha \log M$ with $0 < \alpha < 1/8$. Then*

$$\inf_{\hat{\theta}_n} \sup_{\theta \in \Theta} P_\theta^n \left\{ d(\hat{\theta}_n, \theta) \geq s \right\} \geq \frac{\sqrt{M}}{1+\sqrt{M}} \left(1 - 2\alpha - \sqrt{\frac{2\alpha}{\log M}} \right) > 0.$$

The above result is commonly used with M tending to infinity as $n \to \infty$. In this case the second condition on the KL-divergence indeed becomes less restrictive than the one of Theorem B.1, since the upper bound $\alpha \log M$ may now grow with the sample size n. At the same time, Theorem B.2 still provides a lower bound on the minimax probability independent of n, since $\sqrt{M}/(1+\sqrt{M})$ and $\log M$ can be lower bounded by $1/2$ and $\log 2$ respectively.

Appendix C. Technical Lemmas

The following technical results are used to prove the main results of Sections 3 and 4.

Lemma C.1 (Theorem 5.18, Wendland, 2005) *For any $\mu \in \mathbb{R}^d$ and $\sigma^2 > 0$ the following holds:*

$$\left[\frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\frac{\|x-\mu\|_2^2}{2\sigma^2}} \right]^\wedge (w) = \frac{1}{(2\pi)^{d/2}} \exp\left(-i\langle \mu, w \rangle - \frac{\sigma^2 \|w\|_2^2}{2} \right), \quad w \in \mathbb{R}^d.$$

Lemma C.2 *Let $\psi\colon \mathbb{R}^d \to \mathbb{R}$ be a symmetric and positive definite function. Let Λ_ψ be the corresponding finite non-negative Borel measure from (4). Then Λ_ψ is symmetric, i.e., $\Lambda_\psi(A) = \Lambda_\psi(-A)$ for all $A \subset \mathbb{R}^d$.*

Proof From the definition of Λ_ψ we know that it is finite, non-negative, and

$$\psi(x) = \int_{\mathbb{R}^d} e^{-i\langle w, x\rangle} \Lambda_\psi(dw) = \int_{\mathbb{R}^d} \cos(\langle w, x\rangle) \Lambda_\psi(dw) - i \cdot \int_{\mathbb{R}^d} \sin(\langle w, x\rangle) \Lambda_\psi(dw).$$

Since $\psi(-x) = \psi(x)$ for all $x \in \mathbb{R}^d$, we get $\int_{\mathbb{R}^d} \sin(\langle w, x\rangle)\Lambda_\psi(dw) = 0$. Note that $\psi(-x)$ is by definition a characteristic function of measure Λ_ψ, and we have just proved that it is real-valued. It is known (Bogachev, 2007, Corollary 3.8.7) that in this case the measure Λ_ψ is invariant under the mapping $x \to -x$. ■

Lemma C.3 *Assume $X, Y \subseteq \mathbb{R}^d$. If $f\colon X \times Y \to \mathbb{R}$ is a continuous function and Y is a compact set then $g(x) := \inf_{y \in Y} f(x, y)$ is continuous.*

Proof First, the map $g\colon X \to \mathbb{R}$ is well defined since $f_x(y) := f(x, y)$ is a continuous function for any $x \in X$ and thus f_x achieves its infimum since Y is a compact set. We will prove that the map $g\colon X \to \mathbb{R}$ is continuous by showing that $g^{-1}(-\infty, a)$ and $g^{-1}(a, \infty)$ are open sets for all $a \in \mathbb{R}$ (Dudley, 2002, Corollary 2.2.7 (a)).

Now we will show that $g^{-1}(-\infty, a)$ is open for any $a \in \mathbb{R}$. It suffices to show that for any $x \in g^{-1}(-\infty, a)$ there is an open neighborhood U_x of x which also belongs to $g^{-1}(-\infty, a)$. The set $g^{-1}(-\infty, a)$ consists of elements $x \in X$ for which $g(x) < a$. In other words, it consists of such elements $x \in X$ for which there is corresponding $y_x \in Y$ satisfying $f(x, y_x) < a$. Take any $x \in g^{-1}(-\infty, a)$. Since f is continuous, $f^{-1}(-\infty, a)$ is open and contains (x, y_x). Moreover $f^{-1}(-\infty, a)$ contains $U_x \times V_y$, where U_x and V_y are open sets with $x \in U_x$ and $y_x \in V_y$. Now suppose $x' \in U_x$. Then for any $y \in V_y$ we have $f(x', y) < a$. In particular, $f(x', y_x) < a$, which means that $g(x') < a$ and $x' \in g^{-1}(-\infty, a)$. This shows that $g^{-1}(-\infty, a)$ is open.

Next we will show that $g^{-1}(a, \infty)$ is also an open set for any $a \in \mathbb{R}$. Assume this is not the case. Then there is $x \in g^{-1}(a, \infty)$ such that for any neighborhood U_x of x there is a point $x' \in U_x$ such that $x' \notin g^{-1}(a, \infty)$. This means that for any such x' there is $y_{x'}$ satisfying $f(x', y_{x'}) \leq a$. Using this we can construct a sequence $\{x_n, y_n\}$ from $X \times Y$, such that $x_n \notin g^{-1}(a, \infty)$ for every n, $\lim_{n\to\infty} x_n = x$ and for any n it holds that $f(x_n, y_n) \leq a$. Since Y is compact we conclude that $\{y_n\}$ has a converging subsequence $\{y_{n(k)}\}$ (Dudley, 2002, Theorem 2.3.1) with limit $y^* \in Y$. We just showed that there is a sequence $\{x_{n(k)}, y_{n(k)}\}$ in $X \times Y$, which converges to (x, y^*), such that $\lim_{k\to\infty} f(x_{n(k)}, y_{n(k)}) \leq a$. Since f is continuous, this also means that $\lim_{k\to\infty} f(x_{n(k)}, y_{n(k)}) = f(x, y^*) \leq a$. This means that $\inf_{y\in Y} f(x, y) \leq f(x, y^*) \leq a$. In other words, this shows that $x \notin g^{-1}(a, \infty)$ leading to a contradiction and therefore $g^{-1}(a, \infty)$ is open. ■

Lemma C.4 (L^2 norm of inverse multiquadrics kernels) *For any $c > 0$ and $\gamma > \frac{d}{4}$,*

$$\int_{\mathbb{R}^d} (c^2 + \|x\|_2^2)^{-2\gamma}\, dx = c^{d-4\gamma} \pi^{d/2} \frac{\Gamma(2\gamma - \frac{d}{2})}{\Gamma(2\gamma)}.$$

Proof

$$\begin{aligned}
\int_{\mathbb{R}^d} (c^2 + \|x\|_2^2)^{-2\gamma} dx &= c^{-4\gamma} \int_{\mathbb{R}^d} \left(1 + \left\| \frac{x}{c} \right\|_2^2 \right)^{-2\gamma} dx = c^{d-4\gamma} \int_{\mathbb{R}^d} \left(1 + \|x\|_2^2 \right)^{-2\gamma} dx \\
&= c^{d-4\gamma} \frac{2\pi^{d/2}}{\Gamma(d/2)} \int_0^\infty \left(1 + r^2 \right)^{-2\gamma} r^{d-1} dr \\
&= c^{d-4\gamma} \frac{\pi^{d/2}}{\Gamma(d/2)} \int_0^\infty (1 + x)^{-2\gamma} x^{d/2-1} dx \\
&= c^{d-4\gamma} \frac{\pi^{d/2}}{\Gamma(d/2)} \frac{\Gamma(d/2)\Gamma(2\gamma - d/2)}{\Gamma(2\gamma)},
\end{aligned}$$

where last identity can be found in (Gradshteyn and Ryzhik, 2000, 3.194.3). ■

Appendix D. Bounds on Constants for Various Radial Kernels

In this appendix, we present bounds on the constants that appear in Corollaries 2, 10 and Theorems 8, 13. for various radial kernels.

D.1 α in Corollary 2

In Corollary 2, we assumed that there exist $0 < t_1 < \infty$ and $\alpha > 0$ such that $\nu([t_1, \infty)) \geq \alpha$. In the following, we present the values of t_1 and α for various radial kernels.

(i) Gaussian kernel: $\nu = \delta_{\frac{1}{2\eta^2}}$ and so for any $t_1 < \frac{1}{2\eta^2}$, we obtain $\alpha = 1$.

(ii) Mixture of Gaussians: $\nu = \sum_{i=1}^M \beta_i \delta_{\frac{1}{2\eta_i^2}}$ and so $\alpha = C_M$ for any $t_1 < \frac{1}{2\eta_1^2}$.

(iii) Inverse multiquadric kernel: It follows from (Wendland, 2005, Theorem 7.15) that

$$k(x,y) = \int_0^\infty e^{-t\|x-y\|_2^2} \frac{t^{\gamma-1}e^{-c^2 t}}{\Gamma(\gamma)} dt, \tag{50}$$

and so

$$\nu = c^{-2\gamma}\text{Gamma}(\gamma, c^2) \tag{51}$$

where the density of a Gamma distribution with parameters $a, b > 0$ is defined as

$$\text{Gamma}(t; a, b) = \frac{b^a}{\Gamma(a)} t^{a-1} e^{-tb}, \ t \geq 0.$$

Therefore choosing t_1 to be the median of ν, we obtain $\alpha = \frac{c^{-2\gamma}}{2}$.

(iv) Matérn kernel: We know from (Wendland, 2005, Theorem 6.13) that Matérn kernel is related to the Fourier transform of the inverse multiquadric kernel as

$$\frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-i\langle v, w\rangle} \left(c^2 + \|v\|_2^2\right)^{-\tau} dv = \frac{2^{1-\tau}}{\Gamma(\tau)} K_{d/2-\tau}(c\|w\|_2) \left(\frac{\|w\|_2}{c}\right)^{\tau - d/2},$$

where $c > 0$ and $\tau > d/2$. Using this together with the representation (50) of an inverse multiquadrics kernel we obtain the following identity, which already appeared in (Sriperumbudur, 2016, Equation (72)):

$$\begin{aligned}
k(x,y) &= \frac{\Gamma(\tau) c^{2\tau-d} 2^{d/2}}{\Gamma(\tau - d/2)} \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-i\langle v, x-y\rangle} \left[\int_0^\infty e^{-t\|v\|_2^2} \frac{t^{\tau-1}e^{-c^2 t}}{\Gamma(\tau)} dt\right] dv \\
&\overset{(\star)}{=} \frac{c^{2\tau-d} 2^{d/2}}{\Gamma(\tau - d/2)} \int_0^\infty t^{\tau-1} e^{-c^2 t} \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-i\langle v, x-y\rangle} e^{-t\|v\|_2^2} dv\, dt \\
&= \frac{c^{2\tau-d} 2^{d/2}}{\Gamma(\tau - d/2)} \int_0^\infty t^{\tau-1} e^{-c^2 t} \frac{1}{(2t)^{d/2}} e^{-\frac{\|x-y\|_2^2}{4t}} dt \\
&= \frac{c^{2\tau-d}}{\Gamma(\tau - d/2)} \int_0^\infty t^{\tau-d/2-1} e^{-c^2 t} e^{-\frac{\|x-y\|_2^2}{4t}} dt,
\end{aligned}$$

where we invoked Tonelli-Fubini theorem (Dudley, 2002, Theorem 4.4.5) in $(\star)$ since $\left(c^2 + \|\cdot\|_2^2\right)^{-\tau} \in L_1(\mathbb{R}^d)$ for $\tau > d/2$. After change of variables we finally obtain

$$k(x,y) = \frac{1}{\Gamma\left(\tau - \frac{d}{2}\right)} \left(\frac{c^2}{4}\right)^{\tau - \frac{d}{2}} \int_0^\infty e^{-t\|x-y\|^2} t^{d/2-\tau-1} e^{-\frac{c^2}{4t}} dt,$$

which shows that Matérn kernel is a particular instance of radial kernels with

$$\nu = \text{InvGamma}\left(\tau - \frac{d}{2}, \frac{c^2}{4}\right),$$

where the density of an inverse-Gamma distribution with parameters $a, b > 0$ has the form

$$\text{InvGamma}(t; a, b) = \frac{b^a}{\Gamma(a)} t^{-a-1} e^{-b/t}.$$

Therefore $\alpha = \frac{1}{2}$ for the choice of t_1 to be the median of ν.

D.2 $\frac{\beta t_0}{t_1}$ in Theorem 8

In Theorem 8, we assumed that there exist $0 < t_0 \leq t_1 < \infty$ and $0 < \beta < \infty$ such that $\nu([t_0, t_1]) \geq \beta$. Define $B_k := \frac{\beta t_0}{t_1}$. In the following, we present the values of B_k for various radial kernels.

(i) Gaussian kernel: Choose $t_0 = t_1 = \frac{1}{2\eta^2}$ so that $\beta = 1$ and $B_k = 1$.

(ii) Mixture of Gaussians: Set $t_0 = \frac{1}{2\eta_1^2}$, $t_1 = \frac{1}{2\eta_M^2}$ so that $\beta = C_M$ implying $B_k = \frac{C_M \eta_M^2}{\eta_1^2}$.

(iii) Inverse multiquadric kernel: From (51), we have $\nu = c^{-2\gamma}\text{Gamma}(\gamma, c^2)$. Therefore

$$\begin{aligned}
\nu\left(\left[\frac{\gamma}{2c^2}, \frac{\gamma}{c^2}\right]\right) &= \frac{1}{\Gamma(\gamma)} \int_{\gamma/(2c^2)}^{\gamma/c^2} t^{\gamma-1} e^{-tc^2} dt \\
&\geq \begin{cases} \frac{1}{\Gamma(\gamma)} \left(\frac{\gamma}{2c^2}\right)^{\gamma-1} \exp\left(-\frac{\gamma}{c^2}c^2\right) \frac{\gamma}{2c^2}, & \text{for } \gamma \geq 1; \\ \frac{1}{\Gamma(\gamma)} \left(\frac{\gamma}{c^2}\right)^{\gamma-1} \exp\left(-\frac{\gamma}{c^2}c^2\right) \frac{\gamma}{2c^2}, & \text{for } \gamma \in (0,1). \end{cases}
\end{aligned}$$

Therefore with $t_0 = \frac{\gamma}{2c^2}$, $t_1 = \frac{\gamma}{c^2}$ and $\beta = \begin{cases} \frac{c^{-2\gamma}}{\Gamma(\gamma)} \left(\frac{\gamma}{2e}\right)^\gamma, & \text{for } \gamma \geq 1; \\ \frac{c^{-2\gamma}}{2\Gamma(\gamma)} \left(\frac{\gamma}{e}\right)^\gamma, & \text{for } \gamma \in (0,1) \end{cases}$, we obtain

$$B_k = \begin{cases} \frac{c^{-2\gamma}}{2\Gamma(\gamma)} \left(\frac{\gamma}{2e}\right)^\gamma, & \text{for } \gamma \geq 1; \\ \frac{c^{-2\gamma}}{4\Gamma(\gamma)} \left(\frac{\gamma}{e}\right)^\gamma, & \text{for } \gamma \in (0,1) \end{cases}.$$

(iv) Matérn kernel: It is easy to check that if $X \sim \text{Gamma}(a, b)$ and $Y \sim \text{InvGamma}(a, b)$ for $a, b > 0$ then for any $0 < x \leq y < \infty$ the following holds:

$$\mathbb{P}\{x \leq X \leq y\} = \mathbb{P}\{1/y \leq Y \leq 1/x\}.$$

This means, the above calculations for inverse multiquadrics can be used to obtain the following for the Matérn kernel:

$$B_k = \begin{cases} \frac{1}{2\Gamma(\tau - \frac{d}{2})} \left(\frac{2\tau-d}{4e}\right)^{\tau - \frac{d}{2}}, & \text{for } \tau - \frac{d}{2} \geq 1; \\ \frac{1}{4\Gamma(\tau - \frac{d}{2})} \left(\frac{2\tau-d}{2e}\right)^{\tau - \frac{d}{2}}, & \text{for } \tau - \frac{d}{2} \in (0,1) \end{cases}.$$

D.3 $\beta^2\delta_1^{-d/2}$ in Corollary 10

In Corollary 10, we assumed that there exist $0 < \delta_0 \leq \delta_1 < \infty$ and $0 < \beta < \infty$ such that $\nu([\delta_0, \delta_1]) \geq \beta$. Define $A_k := \beta^2\delta_1^{-d/2}$. Based on the analysis carried out in Appendix D.2, in the following, we present the values of A_k for various radial kernels.

(i) Gaussian kernel: Choose $\delta_0 = \delta_1 = \frac{1}{2\eta^2}$ so that $\beta = 1$ and $A_k = (2\eta^2)^{d/2}$.

(ii) Mixture of Gaussians: Set $\delta_0 = \frac{1}{2\eta_1^2}$, $\delta_1 = \frac{1}{2\eta_M^2}$ so that $\beta = C_M$ implying $A_k = C_M^2(2\eta_M^2)^{d/2}$.

(iii) Inverse multiquadric kernels: Choosing $\delta_0 = t_0$ and $\delta_1 = t_1$ as in Appendix D.2, we obtain

$$A_k = \begin{cases} \frac{c^{d-4\gamma}}{\Gamma^2(\gamma)} \frac{\gamma^{2\gamma-\frac{d}{2}}}{(2e)^{2\gamma}}, & \text{for } \gamma \geq 1; \\ \frac{c^{d-4\gamma}}{4\Gamma^2(\gamma)} \frac{\gamma^{2\gamma-\frac{d}{2}}}{e^{2\gamma}}, & \text{for } \gamma \in (0,1) \end{cases}.$$

(iv) Matérn kernel: Define $\tilde{\gamma} := \tau - \frac{d}{2}$ and $\tilde{c} := \frac{c}{2}$. Choosing $\delta_0 = \frac{\tilde{c}^2}{\tilde{\gamma}}$ and $\delta_1 = \frac{2\tilde{c}^2}{\tilde{\gamma}}$, we obtain

$$\beta = \begin{cases} \frac{1}{\Gamma(\tilde{\gamma})} \left(\frac{\tilde{\gamma}}{2e}\right)^{\tilde{\gamma}}, & \text{for } \tilde{\gamma} \geq 1; \\ \frac{1}{2\Gamma(\tilde{\gamma})} \left(\frac{\tilde{\gamma}}{e}\right)^{\tilde{\gamma}}, & \text{for } \tilde{\gamma} \in (0,1) \end{cases},$$

using the analysis in Appendix D.2. Therefore,

$$A_k = \begin{cases} \frac{c^{-d}e^{-2\tilde{\gamma}}}{\Gamma^2(\tilde{\gamma})} \left(\frac{\tilde{\gamma}}{2}\right)^{2\tilde{\gamma}+\frac{d}{2}}, & \text{for } \tilde{\gamma} \geq 1; \\ \frac{c^{-d}e^{-2\tilde{\gamma}}}{\Gamma^2(\tilde{\gamma})} \frac{\tilde{\gamma}^{2\tilde{\gamma}+\frac{d}{2}}}{2^{2+\frac{d}{2}}}, & \text{for } \tilde{\gamma} \in (0,1) \end{cases}.$$

D.4 $\beta^2\delta_0\delta_1^{-\frac{d+2}{2}}$ in Theorem 13

In Theorem 13, we assumed that there exist $0 < \delta_0 \leq \delta_1 < \infty$ and $0 < \beta < \infty$ such that $\nu([\delta_0, \delta_1]) \geq \beta$. Define $B_k := \beta^2\delta_0\delta_1^{-\frac{d+2}{2}}$. Based on the analysis carried out in Appendix D.2, in the following, we present the values of B_k for various radial kernels.

(i) Gaussian kernel: Choose $\delta_0 = \delta_1 = \frac{1}{2\eta^2}$ so that $\beta = 1$ and $B_k = (2\eta^2)^{d/2}$.

(ii) Mixture of Gaussians: Set $\delta_0 = \frac{1}{2\eta_1^2}$, $\delta_1 = \frac{1}{2\eta_M^2}$ so that $\beta = C_M$ implying $B_k = \frac{C_M^2 2^{d/2}\eta_M^{d+2}}{\eta_1^2}$.

(iii) Inverse multiquadric kernels: Choosing $\delta_0 = t_0$ and $\delta_1 = t_1$ as in Appendix D.2, we obtain

$$B_k = \begin{cases} \frac{c^{d-4\gamma}}{2\Gamma^2(\gamma)} \frac{\gamma^{2\gamma-\frac{d}{2}}}{(2e)^{2\gamma}}, & \text{for } \gamma \geq 1; \\ \frac{c^{d-4\gamma}}{8\Gamma^2(\gamma)} \frac{\gamma^{2\gamma-\frac{d}{2}}}{e^{2\gamma}}, & \text{for } \gamma \in (0,1) \end{cases}.$$

(iv) Matérn kernel: With the choice of δ_0 and δ_1 as in Appendix D.3, we obtain

$$B_k = \begin{cases} \frac{c^{-d}e^{-2\tilde{\gamma}}}{2\Gamma^2(\tilde{\gamma})} \left(\frac{\tilde{\gamma}}{2}\right)^{2\tilde{\gamma}+\frac{d}{2}}, & \text{for } \tilde{\gamma} \geq 1; \\ \frac{c^{-d}e^{-2\tilde{\gamma}}}{\Gamma^2(\tilde{\gamma})} \frac{\tilde{\gamma}^{2\tilde{\gamma}+\frac{d}{2}}}{2^{3+\frac{d}{2}}}, & \text{for } \tilde{\gamma} \in (0,1) \end{cases}.$$

Appendix E. Alternate Proof of Theorem 8

In Theorem 8 we presented a minimax lower bound for radial kernels based on an appropriate construction of d-dimensional Gaussian distributions. By a clever choice of the variance σ^2, which decays to zero as $d \to \infty$, we obtained a lower bound of the order $\Omega(n^{-1/2})$ independent of d. This result was based on the direct analysis and special properties of radial kernels. In this appendix we will show that we can recover almost the same result using only Proposition 3, which holds for any translation invariant kernel. As we will see, this leads to slightly worse constant factors and an additional lower bound on the sample size n in terms of the properties of distribution ν, which specifies the kernel. Essentially we will repeat the main steps of the proof of Theorem 8. However, we will use Proposition 3 instead of direct computations (based on the form of radial kernels) to lower bound the RKHS distance between embeddings of Gaussian distributions with the Euclidean distance between their mean vectors.

Theorem E.1 *Let $\mathcal{P}$ be the set of distributions over $\mathbb{R}^d$ whose densities are continuously infinitely differentiable and k be radial on $\mathbb{R}^d$, i.e.,*

$$k(x,y) = \int_0^\infty e^{-t\|x-y\|_2^2}\, d\nu(t),$$

where $\nu \in M_+^b([0,\infty))$ such that $\operatorname{supp}(\nu) \neq \{0\}$. Assume that there exist $0 < t_0 \leq t_1 < \infty$ and $0 < \beta < \infty$ such that $\nu([t_0, t_1]) \geq \beta$. Suppose $n \geq 24\frac{t_1 Z_\nu}{\beta t_0}$ where $Z_\nu := \nu([0,\infty))$. Then

$$\inf_{\hat{\theta}_n} \sup_{P \in \mathcal{P}} P^n \left\{ \|\hat{\theta}_n - \mu_k(P)\|_{\mathcal{H}_k} \geq \frac{1}{50}\sqrt{\frac{1}{2n} \cdot \frac{\beta t_0}{t_1 e}\left(1 - \frac{2}{2+d}\right)} \right\} \geq \frac{1}{5}.$$

Proof We apply Proposition 3 to the radial kernel k. In order to do so, we need to lower bound the quantity appearing in r.h.s. of Condition (7), which we do as follows. We already saw in the proof of Theorem 8 that in our case Λ_ψ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^d$ and has the following density:

$$\lambda_\psi(w) = \int_0^\infty \frac{1}{(2t)^{d/2}} e^{-\frac{\|w\|_2^2}{4t}}\, d\nu(t), \quad w \in \mathbb{R}^d.$$

Therefore the r.h.s. of (7) reduces to

$$\begin{aligned} &\frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} \langle e_z, w\rangle^2 \cos\left(\langle a, w\rangle\right) d\Lambda_\psi(w) \\ &= \frac{2}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2} \langle e_z, w\rangle^2 \cos\left(\langle a, w\rangle\right) \left(\int_0^\infty \frac{1}{(2t)^{d/2}} e^{-\frac{\|w\|_2^2}{4t}}\, d\nu(t)\right) dw \end{aligned}$$

$$= \frac{2}{(2\pi)^{d/2}} \int_0^\infty \frac{1}{(2t)^{d/2}} \underbrace{\int_{\mathbb{R}^d} e^{-\frac{1}{2}\left(2\sigma^2+\frac{1}{2t}\right)\|w\|_2^2} \langle e_z, w\rangle^2 e^{-i\langle a,w\rangle} dw}_{\clubsuit} \, d\nu(t), \tag{52}$$

where we used Euler's formula and Tonelli-Fubini theorem in the last equality. Denoting $\delta := 2\sigma^2 + \frac{1}{2t}$, we have

$$\clubsuit = \int_{\mathbb{R}^d} \exp\left(-\frac{\delta}{2}\sum_{\ell=1}^d w_\ell^2\right)\left(\sum_{j=1}^d (e_z)_j^2 w_j^2 + \sum_{j\neq\ell}(e_z)_j(e_z)_\ell w_j w_\ell\right)\exp\left(-i\sum_{\ell=1}^d a_\ell w_\ell\right) dw$$
$$= \bigstar + \spadesuit,$$

where

$$\bigstar := \sum_{j=1}^d \int_{\mathbb{R}^d} e^{-\frac{1}{2}\delta\|w\|_2^2}(e_z)_j^2 w_j^2 e^{-i\langle a,w\rangle} dw$$

and

$$\spadesuit := \sum_{j\neq l}^d \int_{\mathbb{R}^d} e^{-\frac{1}{2}\delta\|w\|_2^2}(e_z)_j(e_z)_l w_j w_l e^{-i\langle a,w\rangle} dw.$$

Note that

$$(e_z)_j^2 \int_{\mathbb{R}^d} e^{-\frac{1}{2}\delta\|w\|_2^2} w_j^2 e^{-i\langle a,w\rangle} dw = (e_z)_j^2 \left(\prod_{\ell\neq j}\int_{-\infty}^\infty e^{-\frac{\delta}{2}w_\ell^2} e^{-ia_\ell w_\ell} dw_\ell\right)$$
$$\times \left(\int_{-\infty}^\infty e^{-\frac{\delta}{2}w_j^2} w_j^2 e^{-ia_j w_j} dw_j\right)$$
$$= (e_z)_j^2 \left(\prod_{\ell\neq j}\sqrt{\frac{2\pi}{\delta}} e^{-\frac{a_\ell^2}{2\delta}}\right)\cdot\left(\int_{-\infty}^\infty e^{-\frac{\delta}{2}w_j^2} w_j^2 e^{-ia_j w_j} dw_j\right),$$

where we used Lemma C.1. It follows from (Folland, 1999, Theorem 8.22(d)) that if $g = x^2 f \in L^1(\mathbb{R})$, then $f^\wedge$ is twice differentiable and

$$g^\wedge(y) = -\frac{\partial^2 f^\wedge(y)}{\partial^2 y},$$

which together with Lemma C.1 shows that

$$\int_{-\infty}^\infty e^{-\frac{\delta}{2}w_j^2} w_j^2 e^{-ia_j w_j} dw_j = \frac{1}{\delta}\sqrt{\frac{2\pi}{\delta}} e^{-\frac{a_j^2}{2\delta}}\left(1 - \frac{a_j^2}{\delta}\right).$$

Therefore, we get

$$(e_z)_j^2 \int_{\mathbb{R}^d} e^{-\frac{1}{2}\delta\|w\|_2^2} w_j^2 e^{-i\langle a,w\rangle} dw = (e_z)_j^2 \left(\prod_{\ell\neq j}\sqrt{\frac{2\pi}{\delta}} e^{-\frac{a_\ell^2}{2\delta}}\right)\sqrt{\frac{2\pi}{\delta}}\frac{1}{\delta} e^{-\frac{a_j^2}{2\delta}}\left(1 - \frac{a_j^2}{\delta}\right)$$
$$= \frac{(e_z)_j^2}{\delta}\left(\frac{2\pi}{\delta}\right)^{d/2} e^{-\frac{\|a\|_2^2}{2\delta}}\left(1 - \frac{a_j^2}{\delta}\right).$$

Summing over $j = 1, \ldots, d$ we get

$$\bigstar = \sum_{j=1}^d (e_z)_j^2 \int_{\mathbb{R}^d} e^{-\frac{1}{2}\delta\|w\|_2^2} w_j^2 e^{-i\langle a,w\rangle} dw = \frac{1}{\delta}\left(\frac{2\pi}{\delta}\right)^{d/2} e^{-\frac{\|a\|_2^2}{2\delta}} - \sum_{j=1}^d \frac{(e_z)_j^2 a_j^2}{\delta^2}\left(\frac{2\pi}{\delta}\right)^{d/2} e^{-\frac{\|a\|_2^2}{2\delta}}.$$

Next, for any $j \neq \ell$ we compute

$$\int_{\mathbb{R}^d} \exp\left(-\frac{\delta}{2}\sum_{\ell=1}^d w_\ell^2\right)(e_z)_j(e_z)_\ell w_j w_\ell \exp\left(-i\sum_{\ell=1}^d a_\ell w_\ell\right) dw$$
$$= (e_z)_j(e_z)_\ell \left(\prod_{q\notin\{j,\ell\}}\int_{-\infty}^\infty e^{-\frac{\delta}{2}w_q^2} e^{-ia_q w_q} dw_q\right)\left(\prod_{q\in\{j,\ell\}}\int_{-\infty}^\infty e^{-\frac{\delta}{2}w_q^2} w_q e^{-ia_q w_q} dw_q\right)$$
$$= (e_z)_j(e_z)_\ell \left(\prod_{q\notin\{j,\ell\}}\sqrt{\frac{2\pi}{\delta}} e^{-\frac{a_q^2}{2\delta}}\right)\left(\prod_{q\in\{j,\ell\}}\sqrt{\frac{2\pi}{\delta}}\frac{ia_q}{\delta} e^{-\frac{a_q^2}{2\delta}}\right)$$
$$= (e_z)_j(e_z)_\ell \left(\frac{2\pi}{\delta}\right)^{d/2} e^{-\frac{\|a\|_2^2}{2\delta}}\left(-\frac{a_j a_\ell}{\delta^2}\right).$$

Summing over $j \neq \ell$ we get

$$\spadesuit = -\frac{\langle e_z, a\rangle^2}{\delta^2}\left(\frac{2\pi}{\delta}\right)^{d/2} e^{-\frac{\|a\|_2^2}{2\delta}} + \sum_{j=1}^d \frac{(e_z)_j^2 a_j^2}{\delta^2}\left(\frac{2\pi}{\delta}\right)^{d/2} e^{-\frac{\|a\|_2^2}{2\delta}}.$$

Returning to (52), we get

$$\frac{2}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2}\langle e_z, w\rangle^2 \cos\left(\langle a, w\rangle\right) d\Lambda_\psi(w)$$
$$= \frac{2}{(2\pi)^{d/2}}\int_0^\infty \frac{1}{(2t)^{d/2}} e^{-\frac{\|a\|_2^2}{2\delta}}\left(\frac{2\pi}{\delta}\right)^{d/2}\frac{1}{\delta}\left(1 - \frac{\langle e_z, a\rangle^2}{\delta}\right) d\nu(t)$$
$$= 4\int_0^\infty \exp\left(-\frac{1}{2}\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)\frac{t}{(4\sigma^2 t+1)^{1+d/2}}\left(1 - \frac{2t\langle e_z, a\rangle^2}{4\sigma^2 t+1}\right) d\nu(t).$$

In order to apply Proposition 3 we need to lower bound the following value, appearing in Condition (7):

$$\Delta(a) := \min_{z\in\mathbb{R}^d\setminus\{0\}} \frac{2}{(2\pi)^{d/2}}\int_{\mathbb{R}^d} e^{-\sigma^2\|w\|_2^2}\langle e_z, w\rangle^2 \cos\left(\langle a, w\rangle\right) d\Lambda_\psi(w)$$
$$= 4\int_0^\infty \exp\left(-\frac{1}{2}\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)\frac{t}{(4\sigma^2 t+1)^{1+d/2}}\left(1 - \frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right) d\nu(t).$$

Next we will separately treat two different cases.

Case 1: $d > 2$. Note that the function $\rho(t) = t(4\sigma^2 t+1)^{-(d+2)/2}$ is positive and bounded on $[0, \infty)$ for any $d > 0$. Thus, we can define a non-negative and finite measure

$\tilde{\tau}$, absolutely continuous with respect to ν with density $\rho(t)$. If we denote $Z_\tau := \int_0^\infty 1\, d\tilde{\tau}(t)$ and write τ for the normalized version of $\tilde{\tau}$, then we can rewrite

$$\begin{aligned}\Delta(a) &= 4\int_0^\infty \exp\left(-\frac{1}{2}\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)\left(1-\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)d\tilde{\tau}(t)\\ &= 4\,Z_\tau \mathbb{E}_{t\sim\tau}\left[\exp\left(-\frac{1}{2}\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)\left(1-\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)\right]\\ &= 4\,Z_\tau \mathbb{E}_{t\sim\tau}\left[\exp\left(-\frac{1}{2}\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)\right] - 4\,Z_\tau \mathbb{E}_{t\sim\tau}\left[\exp\left(-\frac{1}{2}\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right].\end{aligned}$$

Note that for $d>2$, $\mathbb{E}_{t\sim\tau}[|t|]$ is finite, since in this case $t\mapsto \frac{t^2}{(4\sigma^2 t+1)^{(d+2)/2}}$ is bounded and ν is a finite measure. Denote $\mu_\tau := \mathbb{E}_{t\sim\tau}[t]$ and note that $t\mapsto \exp\left(-\frac{1}{2}\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)$ is a convex function on $[0,\infty)$. Thus, for $d>2$ we can use Jensen's inequality to get

$$\mathbb{E}_{t\sim\tau}\left[\exp\left(-\frac{1}{2}\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)\right] \geq \exp\left(-\frac{1}{2}\frac{2\mu_\tau\|a\|_2^2}{4\sigma^2\mu_\tau+1}\right).$$

Also note that

$$-4\,Z_\tau\mathbb{E}_{t\sim\tau}\left[\exp\left(-\frac{1}{2}\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right)\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right] \geq 4\,Z_\tau\mathbb{E}_{t\sim\tau}\left[-\frac{2t\|a\|_2^2}{4\sigma^2 t+1}\right] \geq -4\,Z_\tau\frac{2\mu_\tau\|a\|_2^2}{4\sigma^2\mu_\tau+1},$$

where we used inequality $e^{-x}\leq 1$, which holds for $x\geq 0$, together with Jensen's inequality and the fact that $t\mapsto -\frac{2t\|a\|_2^2}{4\sigma^2 t+1}$ is concave on $[0,\infty)$. Summarizing, we have

$$\begin{aligned}\Delta(a) &\geq 4Z_\tau\left(\exp\left(-\frac{1}{2}\frac{2\mu_\tau\|a\|_2^2}{4\sigma^2\mu_\tau+1}\right)-\frac{2\mu_\tau\|a\|_2^2}{4\sigma^2\mu_\tau+1}\right)\\ &\geq 4Z_\tau\left(1-\frac{1}{2}\frac{2\mu_\tau\|a\|_2^2}{4\sigma^2\mu_\tau+1}-\frac{2\mu_\tau\|a\|_2^2}{4\sigma^2\mu_\tau+1}\right)\\ &= 2Z_\tau\left(2-3\frac{2\mu_\tau\|a\|_2^2}{4\sigma^2\mu_\tau+1}\right),\end{aligned}$$

where we used a simple inequality $e^x\geq 1+x$. If the following condition is satisfied:

$$\frac{2\mu_\tau\|a\|_2^2}{4\sigma^2\mu_\tau+1}\leq\frac{1}{3}, \tag{53}$$

then we get

$$\Delta(a)\geq 2Z_\tau = 2\int_0^\infty \frac{t}{(4\sigma^2 t+1)^{1+d/2}}d\nu(t). \tag{54}$$

Together with Proposition 3 this leads to the following lower bound, which holds for any $\mu_0,\mu_1\in\mathbb{R}^d$ and $\sigma^2>0$ satisfying (53) with $a:=\mu_0-\mu_1$:

$$\|\theta_0-\theta_1\|_{\mathcal{H}_k}^2 \geq \int_0^\infty \frac{t\|\mu_0-\mu_1\|_2^2}{(4\sigma^2 t+1)^{1+d/2}}d\nu(t),$$

where θ_0 and θ_1 are KME's of Gaussian measures $G(\mu_0,\sigma^2 I)$ and $G(\mu_1,\sigma^2 I)$ respectively. Note that this lower bound is identical to the one in (27), which we obtained using direct analysis for the radial kernels. However, condition (26) is now replaced with the stronger one in (53). We can now repeat the proof of Theorem 8 starting from inequality (27) and making sure that condition (53) is satisfied when we choose constants appearing in definitions of μ_0,μ_1 and σ^2.

In order to check condition (53) we need to upper bound the expectation μ_τ. It is easily seen that for $d>2$, $t\mapsto\frac{t^2}{(4\sigma^2 t+1)^{(d+2)/2}}$ achieves its maximum on $[0,\infty)$ for $t^*=\frac{1}{\sigma^2(d-2)}$. Using this fact, denoting $Z_\nu=\int_0^\infty 1\,d\nu(t)$, and setting $\sigma^2=\frac{1}{2t_1 d}$ we get

$$\begin{aligned}\mu_\tau &= \frac{1}{Z_\tau}\int_0^\infty\frac{t^2}{(4\sigma^2 t+1)^{1+d/2}}d\nu(t)\leq\frac{Z_\nu}{Z_\tau}\frac{(t^*)^2}{(4\sigma^2 t^*+1)^{1+d/2}}\\ &= \frac{4t_1^2 Z_\nu}{Z_\tau}\frac{d^2}{(d-2)^2}\frac{1}{(\frac{4}{d-2}+1)^{1+d/2}}\\ &= \frac{4t_1^2 Z_\nu}{Z_\tau}\left(1+\frac{2}{d-2}\right)^2\left(\left(1-\frac{4}{d+2}\right)^{(d+2)/4}\right)^2\leq\frac{4t_1^2 Z_\nu}{Z_\tau e^2}\left(1+\frac{2}{d-2}\right)^2.\end{aligned}$$

We may finally use (29) to get

$$Z_\tau\geq\frac{\beta t_0}{e}\left(1-\frac{2}{d+2}\right),$$

which leads to the following upper bound on the expectation μ_τ:

$$\mu_\tau\leq\frac{4t_1^2 Z_\nu}{\beta t_0 e}\frac{d(d+2)}{(d-2)^2}.$$

This upper bound shows that the condition (53) is satisfied if the following holds:

$$t_1\|a\|_2^2\leq\frac{2}{3}t_1\sigma^2+\frac{\beta t_0 e}{24t_1 Z_\nu}\frac{(d-2)^2}{d(d+2)}. \tag{55}$$

We conclude the proof by repeating the remaining steps of the proof of Theorem 8 and replacing condition (26) on the value $\|\mu_0-\mu_1\|_2^2$ with (55) specified to $a=\mu_0-\mu_1$.

Case 2: $d\leq 2$. We can use a simple inequality $e^{-x/2}(1-x)\geq 1-3x/2$ which holds for any x and get the following lower bound:

$$\Delta(a)\geq 4\int_0^\infty\frac{t}{(4\sigma^2 t+1)^{1+d/2}}\left(1-\frac{3t\|a\|_2^2}{4\sigma^2 t+1}\right)d\nu(t).$$

Assuming $\|a\|_2^2\leq\sigma^2$ we further get

$$\left(1-\frac{3t\|a\|_2^2}{4\sigma^2 t+1}\right)\geq\left(1-\frac{3t\sigma^2}{4\sigma^2 t+1}\right)\geq\left(1-\frac{3t\sigma^2}{4\sigma^2 t}\right)=\frac{1}{4}$$

and as a consequence, we also get

$$\Delta(a) \geq \int_0^\infty \frac{t}{(4\sigma^2 t + 1)^{1+d/2}} d\nu(t),$$

which coincides with (54) up to an additional factor of 2. We can now repeat all the steps for the previous case, and it is also easy to check that in this case $\|\mu_0 - \mu_1\|_2^2 \leq \sigma^2$ will be indeed satisfied. This concludes the proof. ■

Remark E.2 *This result should be compared to Theorem 8, which was based on the direct analysis for radial kernels. We see that apart from an extra factor* 2 *appearing under the square root in the lower bound, Theorem E.1 also requires a superfluous condition on the minimal sample size* n*, which depends on properties of* ν*. For instance, for Gaussian kernel with* ν *concentrated on a single point* $\frac{1}{2\eta^2}$ *for some* $\eta^2 > 0$*, the result holds as long as* $n \geq 24$*, because in this case we can take* $t_0 = t_1 = \frac{1}{2\eta^2}$ *and* $\beta = 1$*. However, other choices of* ν *may lead to quite restrictive lower bounds on* n*.*

Remark E.3 *Conceptually, the main difference between the proofs of Theorems 8 and E.1 lies in the way we lower bound the RKHS distance between embeddings of Gaussian measures with the Euclidean distance between their mean vectors. In Theorem 8 we derived a closed-form expression for the RKHS distance in* (25) *and then lower bounded it directly using the properties specific to its form. On the other hand, in Theorem E.1 we resorted to the lower bound of Lemma 3, which holds for any translation invariant kernel and hence is less tight.*

References

N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

V. I. Bogachev. *Measure Theory*, volume 1. Springer, 2007.

J. Diestel and J. J. Uhl. *Vector Measures*. American Mathematical Society, Providence, 1977.

N. Dinculeanu. *Vector Integration and Stochastic Integration in Banach Spaces*. Wiley, 2000.

R. M. Dudley. *Uniform Central Limit Theorems*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1999.

R. M. Dudley. *Real Analysis and Probability*. Cambridge University Press, 2002.

G. B. Folland. *Real Analysis: Modern Techniques and Their Applications*. Wiley, 1999.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 489–496, Cambridge, MA, 2008. MIT Press.

K. Fukumizu, L. Song, and A. Gretton. Kernel Bayes' rule: Bayesian inference with positive definite kernels. *J. Mach. Learn. Res.*, 14:3753–3783, 2013.

I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, San Diego, USA, 2000.

A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola. A kernel method for the two sample problem. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 513–520, Cambridge, MA, 2007. MIT Press.

A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola. A kernel statistical test of independence. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 585–592. MIT Press, 2008.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer-Verlag, New York, 2008.

D. Lopez-Paz, K. Muandet, B. Schölkopf, and I. Tolstikhin. Towards a learning theory of cause-effect inference. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015.

K. Muandet, B. Sriperumbudur, K. Fukumizu, A. Gretton, and B. Schölkopf. Kernel mean shrinkage estimators. *Journal of Machine Learning Research*, 2016. To appear.

A. Ramdas, S. Reddi, B. Poczos, A. Singh, and L. Wasserman. On the decreasing power of kernel and distance based nonparametric hypothesis tests in high dimensions. In *AAAI Conference on Artificial Intelligence*, 2015.

I. J. Schoenberg. Metric spaces and completely monotone functions. *The Annals of Mathematics*, 39(4):811–841, 1938.

A. J. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *Proceedings of the 18th International Conference on Algorithmic Learning Theory (ALT)*, pages 13–31. Springer-Verlag, 2007.

L. Song, A. Smola, A. Gretton, J. Bedo, and K. Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13:1393–1434, 2012.

B. K. Sriperumbudur. Mixture density estimation via Hilbert space embedding of measures. In *Proceedings of International Symposium on Information Theory*, pages 1027–1030, 2011.

B. K. Sriperumbudur. On the optimal estimation of probability measures in weak and strong topologies. *Bernoulli*, 22(3):1839–1893, 2016.

B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R. G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *J. Mach. Learn. Res.*, 11:1517–1561, 2010.

B. K. Sriperumbudur, K. Fukumizu, and G. R. G. Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *J. Mach. Learn. Res.*, 12:2389–2410, 2011.

B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. G. Lanckriet. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, 2008.

Z. Szabó, A. Gretton, B. Póczos, and B. K. Sriperumbudur. Two-stage sampled learning theory on distributions. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38, pages 948–957. JMLR Workshop and Conference Proceedings, 2015.

A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, NY, 2008.

R. Vert and J-P. Vert. Consistency and convergence rates of one-class SVMs and related algorithms. *Journal of Machine Learning Research*, 7:817–854, 2006.

H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2005.

V. Yurinsky. *Sums and Gaussian Vectors*, volume 1617 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1995.

Journal of Machine Learning Research 18 (2017) 1-28 Submitted 1/17; Revised 6/17; Published 7/17

The Impact of Random Models on Clustering Similarity

Alexander J. Gates ajgates@indiana.edu

Yong-Yeol Ahn yyahn@indiana.edu

Department of Informatics and Program in Cognitive Science
Indiana University
919 East 10th Street
Bloomington, IN 47408, USA

Editor: Sebastian Nowozin

Abstract

Clustering is a central approach for unsupervised learning. After clustering is applied, the most fundamental analysis is to quantitatively compare clusterings. Such comparisons are crucial for the evaluation of clustering methods as well as other tasks such as consensus clustering. It is often argued that, in order to establish a baseline, clustering similarity should be assessed in the context of a random ensemble of clusterings. The prevailing assumption for the random clustering ensemble is the permutation model in which the number and sizes of clusters are fixed. However, this assumption does not necessarily hold in practice; for example, multiple runs of K-means clustering returns clusterings with a fixed number of clusters, while the cluster size distribution varies greatly. Here, we derive corrected variants of two clustering similarity measures (the Rand index and Mutual Information) in the context of two random clustering ensembles in which the number and sizes of clusters vary. In addition, we study the impact of one-sided comparisons in the scenario with a reference clustering. The consequences of different random models are illustrated using synthetic examples, handwriting recognition, and gene expression data. We demonstrate that the choice of random model can have a drastic impact on the ranking of similar clustering pairs, and the evaluation of a clustering method with respect to a random baseline; thus, the choice of random clustering model should be carefully justified.

Keywords: clustering comparison, clustering evaluation, adjustment for chance, Rand index, normalized mutual information

1. Introduction

Clustering is one of the most fundamental techniques of unsupervised learning and one of the most common ways to analyze data. Naturally, numerous methods have been developed and studied (Jain, 2010). To interpret clustering results, it is crucial to compare them to each other. For instance, the evaluation of a clustering method is usually carried out by comparing the method's results with a planted reference clustering, assuming that the more similar the method's solution is to the reference clustering, the better the method. This is particularly common in the field of complex networks in which clustering similarity measures are used to justify the performance of community detection methods (Danon et al., 2005; Lancichinetti and Fortunato, 2009). As quantitative comparison is a fundamental operation, it plays a key role in many other tasks. For instance, comparisons of clusterings can facilitate taxonomies for clustering solutions, can be used as a criteria for parameter estimation, and form the basis of consensus clustering methods (Meila, 2005; Vinh et al., 2009; Yeung et al., 2001).

Among the many clustering comparison methods (see, e.g. Meila, 2005; Pfitzner et al., 2009), two of the most prominent measures are the Rand index (Rand, 1971) and the Normalized Mutual Information (NMI, Danon et al., 2005). In both cases, the similarity score exists in the range $[0, 1]$, where 1 corresponds to identical clusterings and 0 implies maximally dissimilar clusterings. However, in practice, both measures do not efficiently use the full range of values in between 0 and 1, with many comparisons concentrating near the extreme values (Vinh et al., 2009; Hubert and Arabie, 1985). This makes it difficult to directly interpret the results of a comparison.

Thus, it is often argued that clustering similarity should be assessed in the context of a random ensemble of clusterings (Vinh et al., 2009; Hubert and Arabie, 1985; DuBien and Warde, 1981; DuBien et al., 2004; Albatineh et al., 2006; Romano et al., 2014; Zhang, 2015; Romano et al., 2016) and rescaled (see Equation 1). Such a correction for chance establishes a baseline by using the expected similarity of all pair-wise comparisons between clusterings specified by a random model; the resulting similarity values have a new interpretation that facilitates comparisons within a set of clusterings. Specifically, once corrected for chance, a similarity value of 1 still corresponds to identical clusterings, but a value of 0 now corresponds to the expected value amongst random clusterings. Positive values of corrected similarity better reflect an intuitive comparison of clusterings (Hubert and Arabie, 1985; Steinley et al., 2016). The correction may also introduce negative values when two clusterings are less similar than expected by chance.

The correction procedure requires two choices: *a model for random clusterings* and *how clusterings are drawn from the random model*. However, even the existence of these choices is usually ignored or relegated to the status of technical trivialities. Here, we demonstrate that these choices may dramatically affect results, and therefore the choice of a particular model for random clusterings should be justified based on the understanding of the clustering scenario. A poor choice of the random model may "not be random enough" and encode crucial features of the clusterings in all of the random clusterings, providing a poor baseline. At the same time, a random model may be "too random" in which crucial features are lost in a sea of random clusterings that are not representative of the particular problem. Characterizing random models is an important topic of research across statistical physics, network science, and combinatorial mathematics (Sethna, 2006; Goldenberg et al., 2010; Mansour, 2012). Yet, despite the importance of random model selection, almost no study that uses clustering comparison provides a justification for their choice of random model.

By far, the most common approach to correct clustering similarity for chance assumes that both clusterings are uniformly and independently sampled from the *permutation model* (M_{perm}). In the permutation model, the number and size of clusters within a clustering are fixed, and all random clusterings are generated by shuffling the elements between the fixed clusters. However, the premises of the permutation model are frequently violated; in many clustering scenarios, either the number of clusters, the size distribution of those clusters, or both vary drastically (Hubert and Arabie, 1985; Wallace, 1983). For example, K-means clustering, probably the most common technique, fixes the number of clusters but not the sizes of those clusters (Jain, 2010). Later, we explore a real example in which K-means

©2017 Alexander J Gates and Yong-Yeol Ahn.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-039.html.

produces clusterings with large variations in the clusterings' cluster size sequences. This suggests that comparing K-means clusterings based on M_{perm} is misleading.

Furthermore, even the assumption that both clusterings were randomly drawn from the same random model (a two-sided comparison) is often problematic. For example, when comparing against a given reference clustering, it is more reasonable to find the expected similarity of the reference clustering with all of the random clusterings from the random model. This *one-sided* comparison accounts for the fixed structure of the reference clustering which is always present in the comparisons, providing a more meaningful baseline.

Here, we present a general framework to adjust measures of clustering similarity for chance by considering a broader class of random clustering models and one-sided comparisons. Specifically, we consider two other random models for clusterings: a uniform distribution over the ensemble of all clusterings of N elements with the same number of clusters (M_{num}), and a uniform distribution over the ensemble of all clusterings of N elements (M_{all}). The resulting expectations for the Rand index under all three random models are summarized in Table 1 and for Mutual Information in Table 2, with the full derivations given in Section 4 and Section 5 respectively. The adjusted similarity measures used throughout this work rescale the Rand and MI measures by these expectations according to Equation 1. We

Adjusted Rand Index

$$\text{ARI}_{\text{model}}(\mathcal{A},\mathcal{B})=\frac{\text{RI}(\mathcal{A},\mathcal{B})-\mathbb{E}_{\text{model}}[\text{RI}(\mathcal{A},\mathcal{B})]}{1.0-\mathbb{E}_{\text{model}}[\text{RI}(\mathcal{A},\mathcal{B})]}$$

Permutation Model

Two-sided = One-sided

$$\mathbb{E}_{\text{perm}}[\text{RI}(\mathcal{A},\mathcal{B})]=\frac{\sum_i\binom{a_i}{2}}{\binom{N}{2}}\frac{\sum_j\binom{b_j}{2}}{\binom{N}{2}}+\left(1-\frac{\sum_i\binom{a_i}{2}}{\binom{N}{2}}\right)\left(1-\frac{\sum_j\binom{b_j}{2}}{\binom{N}{2}}\right)$$

Fixed Number of Clusters

Two-sided

$$\mathbb{E}_{\text{num}}[\text{RI}(\mathcal{A},\mathcal{B})]=\frac{S(N-1,K_\mathcal{A})}{S(N,K_\mathcal{A})}\frac{S(N-1,K_\mathcal{B})}{S(N,K_\mathcal{B})}+\left(1-\frac{S(N-1,K_\mathcal{A})}{S(N,K_\mathcal{A})}\right)\left(1-\frac{S(N-1,K_\mathcal{B})}{S(N,K_\mathcal{B})}\right)$$

One-sided

$$\mathbb{E}^1_{\text{num}}[\text{RI}(\mathcal{A},\mathcal{G})]=\left(\frac{S(N-1,K_\mathcal{A})}{S(N,K_\mathcal{A})}\frac{\sum_j\binom{g_j}{2}}{\binom{N}{2}}\right)+\left(1-\frac{S(N-1,K_\mathcal{A})}{S(N,K_\mathcal{A})}\right)\left(1-\frac{\sum_j\binom{g_j}{2}}{\binom{N}{2}}\right)$$

All Clusterings

Two-sided

$$\mathbb{E}_{\text{all}}[\text{RI}(\mathcal{A},\mathcal{B})]=\left(\frac{B_{N-1}}{B_N}\right)^2+\left(1-\frac{B_{N-1}}{B_N}\right)^2$$

One-sided

$$\mathbb{E}^1_{\text{all}}[\text{RI}(\mathcal{A},\mathcal{G})]=\frac{B_{N-1}}{B_N}\frac{\sum_j\binom{g_j}{2}}{\binom{N}{2}}+\left(1-\frac{B_{N-1}}{B_N}\right)\left(1-\frac{\sum_j\binom{g_j}{2}}{\binom{N}{2}}\right)$$

Table 1: The expected Rand index between two random clusterings $\mathcal{A}$ and $\mathcal{B}$ of N elements, or random clustering $\mathcal{A}$ and reference clustering $\mathcal{G}$ under different random models. Details and derivations are given in Section 4.

Adjusted Mutual Information

$$\text{AMI}_{\text{model}}(\mathcal{A},\mathcal{B})=\frac{\text{MI}(\mathcal{A},\mathcal{B})-\mathbb{E}_{\text{model}}[\text{MI}(\mathcal{A},\mathcal{B})]}{\max_{\text{model}}[\text{MI}(\mathcal{A},\mathcal{B})]-\mathbb{E}_{\text{model}}[\text{MI}(\mathcal{A},\mathcal{B})]}$$

Permutation Model

Two-sided = One-sided

$$\mathbb{E}_{\text{perm}}[\text{MI}(\mathcal{A},\mathcal{B})]=\mathbb{E}_{\text{perm}}[H(\mathcal{A})]+\mathbb{E}_{\text{perm}}[H(\mathcal{B})]-\mathbb{E}_{\text{perm}}[H(\mathcal{A},\mathcal{B})]$$

$$\mathbb{E}_{\text{perm}}[H(\mathcal{A})] =-\sum_{i=1}^{K_\mathcal{A}}\frac{a_i}{N}\log\frac{a_i}{N}$$

$$\mathbb{E}_{\text{perm}}[H(\mathcal{A},\mathcal{B})] =\sum_{i=1}^{K_\mathcal{A}}\sum_{j=1}^{K_\mathcal{B}}\sum_n\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{b_j}{n}\binom{N-b_j}{a_i-n}}{\binom{N}{a_i}}$$

Upper Bound

$$\max_{\text{perm}}[\text{MI}(\mathcal{A},\mathcal{B})]=\min\{\log H(\mathcal{A}),H(\mathcal{B})\}\text{ OR }\sqrt{H(\mathcal{A})H(\mathcal{B})}\text{ OR }\frac{1}{2}(H(\mathcal{A})+H(\mathcal{B}))\text{ OR }\max\{H(\mathcal{A}),H(\mathcal{B})\}$$

Fixed Number of Clusters

Two-sided

$$\mathbb{E}_{\text{num}}[\text{MI}(\mathcal{A},\mathcal{B})]=\mathbb{E}_{\text{num}}[H(\mathcal{A})]+\mathbb{E}_{\text{num}}[H(\mathcal{B})]-\mathbb{E}_{\text{num}}[H(\mathcal{A},\mathcal{B})]$$

$$\mathbb{E}_{\text{num}}[H(\mathcal{A})] =-\sum_{k=1}\binom{N}{k}\frac{S(N-k,K_\mathcal{A}-1)}{S(N,K_\mathcal{A})}\frac{k}{N}\log\left(\frac{k}{N}\right)$$

$$\mathbb{E}_{\text{num}}[H(\mathcal{A},\mathcal{B})] =-\sum_{k=1}\binom{N}{k}\frac{S(N-k,K_\mathcal{A}-1)}{S(N,K_\mathcal{A})}\sum_{m=1}\binom{N}{m}\frac{S(N-m,K_\mathcal{B}-1)}{S(N,K_\mathcal{B})}\sum_n\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{m}{n}\binom{N-m}{k-n}}{\binom{N}{k}}$$

One-sided

$$\mathbb{E}^1_{\text{num}}[\text{MI}(\mathcal{A},\mathcal{G})]=\mathbb{E}_{\text{num}}[H(\mathcal{A})]+\mathbb{E}_{\text{perm}}[H(\mathcal{G})]-\mathbb{E}^1_{\text{num}}[H(\mathcal{A},\mathcal{G})]$$

$$\mathbb{E}^1_{\text{num}}[H(\mathcal{A},\mathcal{G})] =-\sum_{k=1}\binom{N}{k}\frac{S(N-k,K_\mathcal{A}-1)}{S(N,K_\mathcal{A})}\sum_{j=1}^{K_\mathcal{G}}\sum_n\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{g_j}{n}\binom{N-g_j}{k-n}}{\binom{N}{k}}$$

Upper Bound

$$\max_{\text{num}}[\text{MI}(\mathcal{A},\mathcal{B})]=\min\{\log K_\mathcal{A},\log K_\mathcal{B}\}\text{ OR }\sqrt{\log K_\mathcal{A}\log K_\mathcal{B}}\text{ OR }\frac{1}{2}\log K_\mathcal{A}K_\mathcal{B}\text{ OR }\max\{\log K_\mathcal{A},\log K_\mathcal{B}\}$$

All Clusterings

Two-sided

$$\mathbb{E}_{\text{all}}[\text{MI}(\mathcal{A},\mathcal{B})]=\mathbb{E}_{\text{all}}[H(\mathcal{A})]+\mathbb{E}_{\text{all}}[H(\mathcal{B})]-\mathbb{E}_{\text{all}}[H(\mathcal{A},\mathcal{B})]$$

$$\mathbb{E}_{\text{all}}[H(\mathcal{A})] =-\sum_{k=1}\binom{N}{k}\frac{B_{N-k}}{B_N}\frac{k}{N}\log\left(\frac{k}{N}\right)$$

$$\mathbb{E}_{\text{all}}[H(\mathcal{A},\mathcal{B})] =-\sum_{k=1}\binom{N}{k}\frac{B_{N-k}}{B_N}\sum_{m=1}\binom{N}{m}\frac{B_{N-m}}{B_N}\sum_n\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{m}{n}\binom{N-m}{k-n}}{\binom{N}{k}}$$

One-sided

$$\mathbb{E}^1_{\text{all}}[\text{MI}(\mathcal{A},\mathcal{G})]=\mathbb{E}_{\text{all}}[H(\mathcal{A})]+\mathbb{E}_{\text{perm}}[H(\mathcal{G})]-\mathbb{E}^1_{\text{all}}[H(\mathcal{A},\mathcal{G})]$$

$$\mathbb{E}^1_{\text{all}}[H(\mathcal{A},\mathcal{B})] =-\sum_{k=1}\binom{N}{k}\frac{B_{N-k}}{B_N}\sum_{j=1}^{K_\mathcal{G}}\sum_n\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{g_j}{n}\binom{N-g_j}{k-n}}{\binom{N}{k}}$$

Upper Bound

$$\max_{\text{all}}[\text{MI}(\mathcal{A},\mathcal{B})]=\log N$$

Table 2: The expected Mutual Information between two random clusterings $\mathcal{A}$ and $\mathcal{B}$ of N elements, or random clustering $\mathcal{A}$ and reference clustering $\mathcal{G}$ under different random models. Details and derivations are given in Section 5.

also introduce one-sided variants of the adjusted Rand index and adjusted Mutual Information when using the M_{num} or M_{all} random models (for M_{perm}, the one-sided similarity is equivalent to the two-sided case).

The impact of our framework is illustrated in the case of two common tasks for adjusted clustering similarity measures: 1) ranking the similarity between pairs of clusterings (or finding the most similar clustering pair), and 2) evaluating the performance of a clustering method with respect to a random baseline. In Section 6, these tasks are demonstrated in the context of several examples: a synthetic clustering example, K-means clustering of a handwritten digits data set (MNIST), and an evaluation of hierarchical clustering applied to gene expression data. Our results demonstrate that both the choice of random model for clusterings and the choice of one-sided comparisons can affect results significantly. Therefore, we argue that clustering comparisons should be accompanied by a proper justification for the random model.

2. Clusterings

We first explicitly introduce a clustering of elements. Given a set of N distinct elements $V = \{v_1, \ldots, v_N\}$ (i.e. data points or vertices), a clustering is a partition of V into a set $\mathcal{C} = \{C_1, \ldots, C_{K_\mathcal{C}}\}$ of $K_\mathcal{C}$ non-empty disjoint subsets of V, the clusters, C_k, such that

1. $\forall C_i, C_j$ if $i \neq j$, then $C_i \cap C_j = \varnothing$
2. $\bigcup_{k=1}^{K_\mathcal{C}} C_k = V$.

Each clustering specifies a sequence of cluster sizes, namely, letting $c_i = |C_i|$ be the size of the i-th cluster, then the sequence of cluster sizes is $[c_1, c_2, \ldots, c_{K_\mathcal{C}}]$.

Throughout this paper, we focus on the similarity of two clusterings over the same set of N labeled elements, $\mathcal{A} = \{A_1, \ldots, A_{K_\mathcal{A}}\}$ (with $K_\mathcal{A}$ clusters of sizes a_i) and $\mathcal{B} = \{B_1, \ldots, B_{K_\mathcal{B}}\}$ (with $K_\mathcal{B}$ clusters of sizes b_j).

3. Correction for Chance

Given a clustering similarity measure s and a random model for clusterings: model, the expected clustering similarity $\mathbb{E}_{\text{model}}[s]$ of pair-wise comparisons within the random ensemble defined by the model corrects s for chance as follows (Hubert and Arabie, 1985)

$$\frac{s - \mathbb{E}_{\text{model}}[s]}{s_{\max} - \mathbb{E}_{model}[s]}. \tag{1}$$

The denominator rescales the adjusted similarity by the maximum similarity of pair-wise comparisons within the ensemble $s_{\max}$ so identical clusterings always have a similarity of 1.0. For some clustering similarity measures, the value of $s_{\max}$ is independent of the random model used; for example, the Rand index is always bounded above by 1.0. However, in the case of mutual information, the value of $s_{\max}$ depends on the random model used.

4. Rand Index

The Rand index (Rand, 1971) compares the number of element pairs which are either co-assigned to the same cluster, or assigned to different clusters in both clusterings, to the

$\mathcal{A}/\mathcal{B}$	B_1	B_2	$\ldots$	$B_{K_\mathcal{B}}$	Sums
A_1	n_{11}	n_{12}	$\ldots$	$n_{1K_\mathcal{B}}$	a_1
A_2	n_{21}	n_{22}	$\ldots$	$n_{2K_\mathcal{B}}$	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$A_{K_\mathcal{A}}$	$n_{K_\mathcal{A}1}$	$n_{K_\mathcal{A}2}$	$\ldots$	$n_{K_\mathcal{A}K_\mathcal{B}}$	$a_{K_\mathcal{A}}$
Sums	b_1	b_2	$\ldots$	$b_{K_\mathcal{B}}$	$\sum_{ij} n_{ij} = N$

Table 3: The contingency table $\mathcal{T}$ for two clusterings $\mathcal{A} = \{A_1, \ldots, A_{K_\mathcal{A}}\}$ and $\mathcal{B} = \{B_1, \ldots, B_{K_\mathcal{B}}\}$ of N elements, where $n_{ij} = |A_i \cap B_j|$ are the number of elements that are in both cluster $A_i \in \mathcal{A}$ and cluster $B_j \in \mathcal{B}$.

total number of element pairs. The most common formulation of the Rand index focuses on the following four sets of the $\binom{N}{2}$ element pairs: N_{11} the number of element pairs which are grouped in the same cluster in both clusterings, N_{10} the number of element pairs which are grouped in the same cluster by $\mathcal{A}$ but in different clusters by $\mathcal{B}$, N_{01} the number of element pairs which are grouped in the same cluster by $\mathcal{B}$ but in different clusters by $\mathcal{A}$, and N_{00} the number of element pairs which are grouped in different clusters by both $\mathcal{A}$ and $\mathcal{B}$. Intuitively, N_{11} and N_{00} are indicators of the agreement between the two clusterings, while N_{10} and N_{01} reflect the disagreement between the clusterings.

The aforementioned pair counts are identified from the contingency table $\mathcal{T}$ between two clusterings, shown in Table 3, by the following set of equations

$$\begin{aligned}
N_{11} &= \sum_{k,m=1}^{K_\mathcal{A},K_\mathcal{B}} \binom{n_{km}}{2} = \frac{1}{2}\left(\sum_{k,m=1}^{K_\mathcal{A},K_\mathcal{B}} n_{km}^2 - N\right) \\
N_{10} &= \sum_{k=1}^{K_\mathcal{A}} \binom{a_k}{2} - N_{11} = \frac{1}{2}\left(\sum_{k=1}^{K_\mathcal{A}} a_k^2 - \sum_{k,m=1}^{K_\mathcal{A},K_\mathcal{B}} n_{km}^2\right) \\
N_{01} &= \sum_{m=1}^{K_\mathcal{B}} \binom{b_m}{2} - N_{11} = \frac{1}{2}\left(\sum_{m=1}^{K_\mathcal{B}} b_m^2 - \sum_{k,m=1}^{K_\mathcal{A},K_\mathcal{B}} n_{km}^2\right) \\
N_{00} &= \binom{N}{2} - N_{11} - N_{10} - N_{01}
\end{aligned} \tag{2}$$

The Rand index between clusterings $\mathcal{A}$ and $\mathcal{B}$, $\text{RI}(\mathcal{A}, \mathcal{B})$ is then given by the function

$$\begin{aligned}
\text{RI}(\mathcal{A}, \mathcal{B}) &= \frac{N_{11} + N_{00}}{\binom{N}{2}} \\
&= \frac{2\sum_{k,m=1}^{K_\mathcal{A},K_\mathcal{B}} \binom{n_{km}}{2} - \sum_{k=1}^{K_\mathcal{A}} \binom{a_k}{2} - \sum_{m=1}^{K_\mathcal{B}} \binom{b_m}{2} + \binom{N}{2}}{\binom{N}{2}}.
\end{aligned} \tag{3}$$

It lies between 0 and 1, where 1 indicates the clusterings are identical and 0 occurs for clusters which do not share a single pair of elements (this only happens when one clustering

is the full set of elements and the other clustering groups each element into its own cluster). As the number of clustered elements increases, the measure becomes dominated by the number of pairs which were classified into different clusters (N_{00}), resulting in decreased sensitivity to co-occurring element pairs (Fowlkes and Mallows, 1983).

Another formulation of the Rand index, used in our later derivations, focuses on a binary representation of the element pairs. Specifically, consider the vector $U_{\mathcal{A}} = [u_1, \ldots, u_{\binom{N}{2}}]$ with binary entries $u_\alpha \in \{-1, 1\}$ corresponding to all possible element pairs. Using α to index over all element pairs by $\alpha = \binom{N}{2} - \binom{N-i+1}{2} + j - i$, for $i < j \leq N$, then $u_\alpha = 1$ if elements v_i and v_j are in the same cluster in $\mathcal{A}$ and $u_\alpha = -1$ if elements v_i and v_j are in different clusters in $\mathcal{A}$. There are $Q_1^{\mathcal{A}}$ 1s in $U_{\mathcal{A}}$ and $Q_{-1}^{\mathcal{A}}$ -1s in $U_{\mathcal{A}}$ with

$$Q_1^{\mathcal{A}} = \sum_{k=1}^{K_{\mathcal{A}}} \binom{a_k}{2}, \quad Q_{-1}^{\mathcal{A}} = \binom{N}{2} - \sum_{k=1}^{K_{\mathcal{A}}} \binom{a_k}{2}. \tag{4}$$

The Rand index is found from the vectors $U_{\mathcal{A}}$ and $U_{\mathcal{B}}$, for clusterings $\mathcal{A}$ and $\mathcal{B}$ respectively, as the number of 1s in their product vector, $U_{\mathcal{A}} \odot U_{\mathcal{B}}$, using element-wise multiplication and normalized by the total size of the vectors, $\binom{N}{2}$.

4.1 Expected Rand Index, Permutation Model (M_{perm})

The expectation of the Rand index with respect to the permutation model follows from drawing the entries in Table 3 from the generalized hypergeometric distribution. Utilizing the previous notation with $Q_1^{\mathcal{A}} = \sum_{k=1}^{K_{\mathcal{A}}} \binom{a_k}{2}$, the expectation $\mathbb{E}_{\text{perm}}[RI(\mathcal{A}, \mathcal{B})]$ of the Rand index with respect to the permutation model for the cluster size sequences of clusterings $\mathcal{A}$ and $\mathcal{B}$ is given by

$$\mathbb{E}_{\text{perm}}[\text{RI}(\mathcal{A}, \mathcal{B})] = \frac{2Q_1^{\mathcal{A}}Q_1^{\mathcal{B}} - \binom{N}{2}\left(Q_1^{\mathcal{A}} + Q_1^{\mathcal{B}}\right) + \binom{N}{2}^2}{\binom{N}{2}^2} \tag{5}$$

(see Fowlkes and Mallows, 1983, Hubert and Arabie, 1985, or Albatineh and Niewiadomska-Bugaj, 2011 for the full derivation).

The commonly used adjusted Rand index (ARI) of Hubert and Arabie (1985) uses M_{perm} to calculate the expectation of the Rand index, $\mathbb{E}_{\text{perm}}[\text{RI}(\mathcal{A}, \mathcal{B})]$, as found in Equation 5. This expectation is then used in Equation 1, along with the fact that the maximum value of the Rand index is $\max_{\text{perm}}[\text{RI}] = 1.0$, to give

$$\text{ARI}_{\text{perm}}(\mathcal{A}, \mathcal{B}) = \frac{\binom{N}{2}\sum_{k,m=1}^{K_{\mathcal{A}}K_{\mathcal{B}}} \binom{n_{km}}{2} - \sum_{k=1}^{K_{\mathcal{A}}} \binom{a_k}{2} \sum_{m=1}^{K_{\mathcal{B}}} \binom{b_m}{2}}{\frac{1}{2}\binom{N}{2}\left[\sum_{k=1}^{K_{\mathcal{A}}} \binom{a_k}{2} + \sum_{m=1}^{K_{\mathcal{B}}} \binom{b_m}{2}\right] - \sum_{k=1}^{K_{\mathcal{A}}} \binom{a_k}{2} \sum_{m=1}^{K_{\mathcal{B}}} \binom{b_m}{2}}. \tag{6}$$

4.2 Expected Rand Index, Fixed Number of Clusters

We follow DuBien and Warde (1981) to calculate the Rand index between two clusterings under the assumptions that both clusterings were independently and uniformly drawn from the ensemble of clusterings with a fixed number of clusters (M_{num}). Recall that the Rand index between two clusterings $\mathcal{A}$ and $\mathcal{B}$ is given by the number of 1s in the element-wise

product of the binary representations vectors $U_{\mathcal{A}}$ and $U_{\mathcal{B}}$. The expected Rand index under any random model is then the expected number of 1s in this product vector, normalized by the total size of the vector

$$\begin{aligned}\mathbb{E}[RI(\mathcal{A}, \mathcal{B})] &= \mathbb{E}\left[\frac{1}{\binom{N}{2}} \sum_{\alpha=1}^{\binom{N}{2}} \mathbf{1}_{u_\alpha^{\mathcal{A}} \cdot u_\alpha^{\mathcal{B}} = 1}\right] \\ &= \frac{1}{\binom{N}{2}} \sum_{\alpha=1}^{\binom{N}{2}} \mathbb{E}\left[\mathbf{1}_{u_\alpha^{\mathcal{A}} \cdot u_\alpha^{\mathcal{B}} = 1}\right] \\ &= \frac{1}{\binom{N}{2}} \sum_{\alpha=1}^{\binom{N}{2}} P(u_\alpha^{\mathcal{A}} \cdot u_\alpha^{\mathcal{B}} = 1)\end{aligned} \tag{7}$$

The product $u_\alpha^{\mathcal{A}} \cdot u_\alpha^{\mathcal{B}}$ equals 1 when either $u_\alpha^{\mathcal{A}} = 1$ and $u_\alpha^{\mathcal{B}} = 1$, or $u_\alpha^{\mathcal{A}} = -1$ and $u_\alpha^{\mathcal{B}} = -1$. Since we assumed both clusterings were independent, this gives

$$P(u_\alpha^{\mathcal{A}} \cdot u_\alpha^{\mathcal{B}} = 1) = P(u_\alpha^{\mathcal{A}} = 1)P(u_\alpha^{\mathcal{B}} = 1) + P(u_\alpha^{\mathcal{A}} = -1)P(u_\alpha^{\mathcal{B}} = -1) \tag{8}$$

where $P(u_\alpha^{\mathcal{A}} = 1)$ is the probability that the two elements v_i and v_j are in the same cluster in clustering $\mathcal{A}$, where the element pair is indexed by $\alpha = \binom{N}{2} - \frac{(N-i)(N-i+1)}{2} + j - i$ with $i < j \leq N$. Likewise, $P(u_\alpha^{\mathcal{A}} = -1)$ is the probability that the two elements v_i and v_j are in different clusters.

Under the assumption of M_{num}, there is a uniform probability of selecting a clustering from the $S(N, K_{\mathcal{A}})$ clusterings of N elements into $K_{\mathcal{A}}$ clusters; we define, $P_{\text{num}}(u_\alpha^{\mathcal{A}} = 1)$ as the proportion of these clusterings with elements v_i and v_j in the same cluster. To find this proportion, notice that we can ensure v_i is in the same cluster as v_j by first partitioning all elements besides v_i into $K_{\mathcal{A}}$ clusters; then, we can add v_i to the same cluster as v_j. Since there are $S(N-1, K_{\mathcal{A}})$ such clusterings without element v_i, this gives

$$P_{\text{num}}(u_\alpha^{\mathcal{A}} = 1) = \frac{S(N-1, K_{\mathcal{A}})}{S(N, K_{\mathcal{A}})} \tag{9}$$

$$P_{\text{num}}(u_\alpha^{\mathcal{A}} = -1) = 1 - \frac{S(N-1, K_{\mathcal{A}})}{S(N, K_{\mathcal{A}})}. \tag{10}$$

Finally, the expected Rand index between two clusterings $\mathcal{A}$ and $\mathcal{B}$ with $K_{\mathcal{A}}$ and $K_{\mathcal{B}}$ clusters assuming M_{num} is given by

$$\begin{aligned}\mathbb{E}_{\text{num}}[\text{RI}(\mathcal{A}, \mathcal{B})] = &\frac{S(N-1, K_{\mathcal{A}})}{S(N, K_{\mathcal{A}})} \frac{S(N-1, K_{\mathcal{B}})}{S(N, K_{\mathcal{B}})} \\ &+ \left(1 - \frac{S(N-1, K_{\mathcal{A}})}{S(N, K_{\mathcal{A}})}\right)\left(1 - \frac{S(N-1, K_{\mathcal{B}})}{S(N, K_{\mathcal{B}})}\right).\end{aligned} \tag{11}$$

When N is large, we can approximate the Stirling numbers of the second kind for a fixed K by $S(N, K) \approx \frac{K^N}{K!}$. This can be inserted into equation (11) to give the following approximation for the mean of the Rand index assuming M_{num}

$$\mathbb{E}_{\text{num}}[\text{RI}(\mathcal{A}, \mathcal{B})] \approx \frac{1}{K_{\mathcal{A}}K_{\mathcal{B}}} + \left(1 - \frac{1}{K_{\mathcal{A}}}\right)\left(1 - \frac{1}{K_{\mathcal{B}}}\right). \tag{12}$$

Interestingly, this suggests that the Rand index goes to 1 at a rate inversely related to the smaller number of clusters $\mathcal{O}(\max\{K_{\mathcal{A}}^{-1}, K_{\mathcal{B}}^{-1}\})$.

4.3 Expected Rand Index, All Clusterings M_{all}

The average of the Rand index between two clusterings under the assumption that the clusterings were drawn with uniform probability from the set of all clusterings directly follows from the random model with a fixed number of clusters previously discussed. Namely, because Bell numbers are related to Stirling numbers of the second kind by $B_N = \sum_{k=1}^{N} S(N,k)$, a similar reasoning as followed for equation (9) gives

$$\begin{aligned} P_{\text{all}}(u_\alpha = 1) &= \sum_{k=1}^{N} \frac{S(N,k)}{B_N} P_{\text{num}}(u_\alpha^k = 1) \qquad (13) \\ &= \frac{1}{B_N} \sum_{k=1}^{N} S(N,k) \frac{S(N-1,k)}{S(N,k)} \\ &= \frac{B_{N-1}}{B_N}. \qquad (14) \end{aligned}$$

Using this probability for the expectation in equation (7) gives the expected Rand index under the assumption that both clusterings were uniformly drawn from the set of all clusterings of N elements

$$\mathbb{E}_{\text{all}}[\text{RI}(\mathcal{A},\mathcal{B})] = \left(\frac{B_{N-1}}{B_N}\right)^2 + \left(1 - \frac{B_{N-1}}{B_N}\right)^2. \qquad (15)$$

When N is large, we can approximate the ratio of successive Bell numbers by $\frac{B_{N+1}}{B_N} \approx \frac{N}{\log N}$. Using this approximation in equation (15) gives the following approximation for the mean of the Rand index in M_{all}

$$\mathbb{E}_{\text{all}}[\text{RI}(\mathcal{A},\mathcal{B})] \approx \left(\frac{\log N}{N}\right)^2 + \left(1 - \frac{\log N}{N}\right)^2. \qquad (16)$$

Interestingly, this suggests that the expected Rand index between two random clusterings goes to 1 at a rate $\mathcal{O}\left(\frac{\log(N)}{N}\right)$, inversely proportional to the number of elements.

4.4 One-Sided Rand

Consider a reference clustering $\mathcal{G}$ that has the cluster size sequence $[g_1, \ldots, g_{K_\mathcal{G}}]$. The binary pair vector representation of $\mathcal{G}$ has $Q_1^{\mathcal{G}}$ 1s and $Q_{-1}^{\mathcal{G}} = \binom{N}{2} - Q_1^{\mathcal{G}}$, -1s. The one-sided expectation of the Rand index under the assumption that clustering $\mathcal{A}$ was randomly drawn from either the M_{num} or M_{all} random models follows from treating the two clusterings independently as in equation (8). Since the cluster sequence for the reference clustering is fixed, the probability that a random entry in the binary pair vector is 1 is given by the

fraction of 1s in the vector

$$\begin{aligned} P_{\text{num}}(u_\alpha^{\mathcal{G}} = 1) &= \frac{1}{\binom{N}{2}} Q_1^{\mathcal{G}} \qquad (17) \\ &= \frac{1}{\binom{N}{2}} \sum_{i=1}^{K_\mathcal{G}} \binom{g_i}{2}. \end{aligned}$$

The one-sided expectation of the Rand index under the assumption that clustering $\mathcal{A}$ was randomly drawn from the set of all clusterings with a fixed number of clusters M_{num}^1 is

$$\mathbb{E}_{\text{num}}^1[\text{RI}(\mathcal{A},\mathcal{G})] = \left(\frac{S(N-1,K_\mathcal{A})}{S(N,K_\mathcal{A})} \frac{Q_1^{\mathcal{G}}}{\binom{N}{2}}\right) + \left(1 - \frac{S(N-1,K_\mathcal{A})}{S(N,K_\mathcal{A})}\right)\left(1 - \frac{Q_1^{\mathcal{G}}}{\binom{N}{2}}\right). \qquad (18)$$

The one-sided expectation of the Rand index with the assumption that the random clustering $\mathcal{A}$ is drawn from the ensemble of all partitions M_{all}^1 is

$$\mathbb{E}_{\text{all}}^1[\text{RI}(\mathcal{A},\mathcal{G})] = \frac{B_{N-1}}{B_N} \frac{Q_1^{\mathcal{G}}}{\binom{N}{2}} + \left(1 - \frac{B_{N-1}}{B_N}\right)\left(1 - \frac{Q_1^{\mathcal{G}}}{\binom{N}{2}}\right). \qquad (19)$$

5. Mutual Information

Another prominent family of clustering similarity measures is based on the Shannon information between probabilistic representations of each clustering. These probability distributions are also calculated from the contingency table $\mathcal{T}$, Table 3. The partition entropy H of a clustering $\mathcal{A}$ is given by

$$H(\mathcal{A}) = -\sum_{k=1}^{K_\mathcal{A}} \frac{a_k}{N} \log \frac{a_k}{N}. \qquad (20)$$

Using this entropy, the mutual information $\text{MI}(\mathcal{A},\mathcal{B})$ between two clusterings $\mathcal{A}$ and $\mathcal{B}$ is given by

$$\begin{aligned} \text{MI}(\mathcal{A},\mathcal{B}) &= H(\mathcal{A}) + H(\mathcal{B}) - H(\mathcal{A},\mathcal{B}) \\ &= \sum_{k,m=1}^{K_\mathcal{A},K_\mathcal{B}} \frac{n_{km}}{N} \log \frac{n_{km} N}{a_k b_m}. \qquad (21) \end{aligned}$$

The mutual information can be interpreted as an inverse measure of independence between the clusterings, or a measure of the amount of information each clustering has about the other. As it can vary in the range $[0, \min\{H(\mathcal{A}), H(\mathcal{B})\}]$, to facilitate comparisons, it is desirable to normalize it to the range $[0,1]$. There are at least six proposals in the literature for this upper bound, each with different advantages and drawbacks

$$\begin{aligned} &\min\{H(\mathcal{A}), H(\mathcal{B})\} \leq \sqrt{H(\mathcal{A})H(\mathcal{B})} \leq \frac{H(\mathcal{A}) + H(\mathcal{B})}{2} \qquad (22) \\ &\leq \max\{H(\mathcal{A}), H(\mathcal{B})\} \leq \max\{\log K_\mathcal{A}, \log K_\mathcal{B}\} \leq \log N. \end{aligned}$$

The resulting measures are all known as normalized mutual information (NMI). This measure has been said to exhibit more desirable properties than the Rand index; for example, it is dependent on the relative proportions of the cluster sizes in each clustering rather than the number of elements. However, due to its dependence on the number of clusters in each clustering, it is known to favor comparisons between clusterings with more clusters regardless of any other shared clustering features (White and Liu, 1994; Vinh et al., 2010; Amelio and Pizzuti, 2015).

5.1 Expected Mutual Information, Permutation Model ($M_{\mathbf{perm}}$)

The mutual information between two clusterings has also previously been studied under the assumption that both clusterings were randomly generated from the permutation model (Vinh et al., 2009; Romano et al., 2014; Vinh et al., 2010). Expanding the definition of the mutual information gives

$$\begin{aligned}\mathbb{E}_{\text{perm}}[\text{MI}(\mathcal{A},\mathcal{B})] &= \mathbb{E}_{\text{perm}}[H(\mathcal{A})] + \mathbb{E}_{\text{perm}}[H(\mathcal{B})] - \mathbb{E}_{\text{perm}}[H(\mathcal{A},\mathcal{B})] \\ &= H(\mathcal{A}) + H(\mathcal{B}) - \mathbb{E}_{\text{perm}}[H(\mathcal{A},\mathcal{B})]\end{aligned} \tag{23}$$

where the second line follows from the fact that all cluster sizes (and hence the entropy) are the same for every clustering in M_{perm}.

The expectation of the joint entropy with respect to M_{perm} for the cluster size distributions of clusterings $\mathcal{A}$ and $\mathcal{B}$ is the average over all possible contingency tables $\mathcal{T}$ with entries n

$$\mathbb{E}_{\text{perm}}[H(\mathcal{A},\mathcal{B})] = -\sum_{\mathcal{T}} p(\mathcal{T}|\mathcal{A},\mathcal{B}) \sum_{k=1}^{K_{\mathcal{A}}} \sum_{m=1}^{K_{\mathcal{B}}} \frac{n}{N} \log\left(\frac{n}{N}\right). \tag{24}$$

Rearranging the summations, and recalling that the entries of the contingency tables are hyper-geometrically distributed such that the probability of each entry

$$p(n) = \frac{\binom{b_m}{n}\binom{N-b_m}{a_k-n}}{\binom{N}{a_k}} \tag{25}$$

is only dependent on the row sum a_k and column sum b_m, gives

$$\mathbb{E}_{\text{perm}}[H(\mathcal{A},\mathcal{B})] = -\sum_{k=1}^{K_{\mathcal{A}}} \sum_{m=1}^{K_{\mathcal{B}}} \sum_{n} \frac{n}{N} \log\left(\frac{n}{N}\right) \frac{\binom{b_m}{n}\binom{N-b_m}{a_k-n}}{\binom{N}{a_k}}. \tag{26}$$

According to the hyper-geometric distribution, the summation over table entries n_{km} occurs between the lower bound: $\max\{0, a_k + b_m - N\}$ and the upper bound: $\min\{a_k, b_m\}$. Combining this expression with the individual entropies $H(\mathcal{A})$ and $H(\mathcal{B})$ gives (Vinh et al., 2009)

$$\mathbb{E}_{\text{perm}}[\text{MI}(\mathcal{A},\mathcal{B})] = \sum_{k=1}^{K_{\mathcal{A}}} \sum_{m=1}^{K_{\mathcal{B}}} \sum_{n} \frac{n}{N} \log\left(\frac{Nn}{a_k b_m}\right) \frac{\binom{b_m}{n}\binom{N-b_m}{a_k-n}}{\binom{N}{a_k}}. \tag{27}$$

As shown in Romano et al. (2014), the computational complexity of calculating the expected mutual information assuming the permutation model is of order $\mathcal{O}(\max\{K_{\mathcal{A}}N, K_{\mathcal{B}}N\})$.

The adjusted mutual information (AMI) of Vinh et al. (2009) uses M_{perm} to correct the MI for chance according to equation (1) and selecting an upper bound max[MI] from equation (22) to give

$$\text{AMI}(\mathcal{A},\mathcal{B}) = \frac{\text{MI}(\mathcal{A},\mathcal{B}) - \mathbb{E}_{\text{perm}}[\text{MI}(\mathcal{A},\mathcal{B})]}{\max_{\text{perm}}[\text{MI}] - \mathbb{E}_{\text{perm}}[\text{MI}(\mathcal{A},\mathcal{B})]}. \tag{28}$$

5.2 Expected Mutual Information, Fixed Number of Clusters ($M_{\mathbf{num}}$)

Next, we consider the Mutual Information between two clusterings under the assumptions that both clusterings were independently and uniformly drawn from the ensemble of clusterings with a fixed number of clusters (M_{num}). In this case, the expected mutual information is dependent on both the average partition entropy and the joint partition entropy

$$\mathbb{E}_{\text{num}}[\text{MI}(\mathcal{A},\mathcal{B})] = \mathbb{E}_{\text{num}}[H(\mathcal{A})] + \mathbb{E}_{\text{num}}[H(\mathcal{B})] - \mathbb{E}_{\text{num}}[H(\mathcal{A},\mathcal{B})]. \tag{29}$$

This expectation can be found by considering the average partition entropy and joint partition entropy separately. Recall that in the permutation model $\mathbb{E}_{\text{perm}}[H(\mathcal{A})] = H(\mathcal{A})$ since the cluster sizes remain unchanged; however, the same does not hold in M_{num}. Denoting a random clustering with $K_{\mathcal{A}}$ clusters as $\pi_{K_{\mathcal{A}}}$, and using the notion $\sum_{\sigma_i \in \pi_{K_{\mathcal{A}}}}$ to indicate the summation over all clusters in the clustering $\pi_{K_{\mathcal{A}}}$, where the cardinality of the cluster is $|\sigma_i| = a$, then the expected partition entropy of a random clustering in M_{num} is

$$\mathbb{E}_{\text{num}}[H(\mathcal{A})] = -\sum_{\pi_{K_{\mathcal{A}}}} p_{\text{num}}(\pi_{K_{\mathcal{A}}}) \sum_{\sigma_i \in \pi_{K_{\mathcal{A}}}} \frac{a}{N} \log\left(\frac{a}{N}\right). \tag{30}$$

Note that this expression only depends on the size $a = |\sigma_i|$ for $\sigma_i \in \pi_{K_{\mathcal{A}}}$ of the clusters in the clustering. This means the expected entropy of a random clustering can be rewritten in terms of the expected contribution to the entropy from a random cluster of size a. A counting argument gives the number of clusters of size a which appear in all of the clusterings in the random ensemble. First, choose a of the N elements to form the cluster. Each clustering in M_{num} must have $K_{\mathcal{A}}$ clusters, so the remaining $N - a$ elements have to be arranged into $K_{\mathcal{A}} - 1$ other clusters. There are $S(N-a, K_{\mathcal{A}}-1)$ ways to partition these remaining elements. This gives $\binom{N}{a} S(N-a, K_{\mathcal{A}}-1)$ clusters of size a (Chern et al., 2014). The expected number of clusters $n_{\text{num}}(a)$ in a random clustering drawn from M_{num} is then $\binom{N}{a}\frac{S(N-a,K_{\mathcal{A}}-1)}{S(N,K_{\mathcal{A}})}$. Therefore, the expected clustering entropy in M_{num} is:

$$\mathbb{E}_{\text{num}}[H(\mathcal{A})] = -\sum_{a=1}^{N-(K_{\mathcal{A}}-1)} \binom{N}{a} \frac{S(N-a, K_{\mathcal{A}}-1)}{S(N,K_{\mathcal{A}})} \frac{a}{N} \log\left(\frac{a}{N}\right). \tag{31}$$

where the summation is over all possible cluster sizes $[1, N-(K_{\mathcal{A}}-1)]$ encountered when partitioning N elements into $K_{\mathcal{A}}$ clusters.

Similarly, the expected joint entropy of two random clusterings drawn independently from M_{num} is given by the expected number of clusters of size a from a clustering with

$K_{\mathcal{A}}$ clusters, the expected number of clusters of size b from a clustering with $K_{\mathcal{B}}$ clusters, and then considering the probability of overlap $p(n_{km}) = \frac{\binom{b}{n_{km}}\binom{N-b}{a-n_{km}}}{\binom{N}{a}}$ from the resulting random contingency table

$$\begin{aligned}\mathbb{E}_{\text{num}}[H(\mathcal{A},\mathcal{B})] &= -\sum_{\pi_{K_{\mathcal{A}}}} p_{\text{num}}(\pi_{K_{\mathcal{A}}}) \sum_{\pi_{K_{\mathcal{B}}}} p_{\text{num}}(\pi_{K_{\mathcal{B}}}) \\ &\times \sum_{k=1}^{K_{\mathcal{A}}}\sum_{m=1}^{K_{\mathcal{B}}}\sum_{n} \frac{n}{N}\log\left(\frac{Nn}{a_k b_m}\right)\frac{\binom{b_m}{n}\binom{N-b_m}{a_k-n}}{\binom{N}{a_k}} \\ &= -\sum_{a=1}^{N-(K_{\mathcal{A}}-1)}\sum_{b=1}^{N-(K_{\mathcal{B}}-1)}\sum_{n}\left[\binom{N}{a}\frac{S(N-a,K_{\mathcal{A}}-1)}{S(N,K_{\mathcal{A}})}\right. \\ &\times \left.\binom{N}{b}\frac{S(N-b,K_{\mathcal{B}}-1)}{S(N,K_{\mathcal{B}})}\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{b}{n}\binom{N-b}{a-n}}{\binom{N}{a}}\right].\end{aligned} \tag{32}$$

Note that, when using equation (1) to adjust the mutual information for chance under the assumption of M_{num}, the maximum value for the measure over the entire ensemble of random clusterings has to be used. When considering clusterings with a fixed number of clusters, we know that $H(\mathcal{A}) \leq \log K_{\mathcal{A}}$. This means that the choices for $\max_{\text{num}}[\text{MI}(\mathcal{A},\mathcal{B})]$ are

$$\min\{\log K_{\mathcal{A}}, \log K_{\mathcal{B}}\} \leq \sqrt{\log K_{\mathcal{A}} \log K_{\mathcal{B}}} \leq \frac{1}{2}\log K_{\mathcal{A}}K_{\mathcal{B}} \leq \max\{\log K_{\mathcal{A}}, \log K_{\mathcal{B}}\}. \tag{33}$$

As is apparent from the summations in equation (32), the computational complexity of exactly calculating the expected mutual information assuming M_{num} is of order $\mathcal{O}(N^3)$.

5.3 Expected Mutual Information, All Clusterings M_{all}

The expected Mutual Information between two clusterings under the assumption that both clusterings were independently and uniformly drawn from the set of all possible clusterings of N elements, M_{all}, has a similar derivation as the previous case of M_{num}. Both the expectations for the entropy of a single clustering and the joint entropy of the two clusterings need to be considered separately and can be rewritten in terms of the contributions from individual clusters of a given size. In M_{all}, the number of clusters of size a is again found by choosing a of the N elements for the cluster and then partitioning the remaining $N-a$ elements; there are now B_{N-a} possible ways to cluster the remaining elements (Chern et al., 2014). This gives

$$\mathbb{E}_{\text{all}}[H(\mathcal{A})] = -\sum_{a=1}^{N}\binom{N}{a}\frac{B_{N-a}}{B_N}\frac{a}{N}\log\left(\frac{a}{N}\right). \tag{34}$$

The expected joint entropy for two clusterings is then

$$\begin{aligned}\mathbb{E}_{\text{all}}[H(\mathcal{A},\mathcal{B})] &= -\sum_{\pi_a} p(\pi_a)\sum_{\pi_b} p(\pi_b)\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{n}\frac{n}{N}\log\left(\frac{Nn}{a_i b_j}\right)\frac{\binom{b_j}{n}\binom{N-b_j}{a_i-n}}{\binom{N}{a_i}} \\ &= -\sum_{a=1}^{N}\binom{N}{a}\frac{B_{N-a}}{B_N}\sum_{b=1}^{N}\binom{N}{b}\frac{B_{N-b}}{B_N}\sum_{n}\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{b}{n}\binom{N-b}{a-n}}{\binom{N}{a}} \\ &= -2\sum_{a=1}^{N}\sum_{b=1}^{a-1}\binom{N}{a}\frac{B_{N-a}}{B_N}\binom{N}{b}\frac{B_{N-b}}{B_N}\sum_{n}\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{b}{n}\binom{N-b}{a-n}}{\binom{N}{a}} \\ &- \sum_{a=1}^{N}\left(\binom{N}{a}\frac{B_{N-a}}{B_N}\right)^2\sum_{n}\frac{n_{ij}}{N}\log\left(\frac{n}{N}\right)\frac{\binom{a}{n}\binom{N-a}{a-n}}{\binom{N}{a}},\end{aligned} \tag{35}$$

with the last simplification resulting from the symmetry of the hyper-geometric term with respect to a and b.

As in the previous case, the maximum bound of the measure must be consider over the entire ensemble of clusterings. Again, we consider the bound $H(\mathcal{A}) \leq \log N$. This reduces to only one choice for $\max_{\text{all}}[\text{MI}(\mathcal{A},\mathcal{B})] = \log N$.

5.4 One-Sided Mutual Information

As was the case for the one-sided expected Rand index, the one-sided expectation of mutual information follows from the fact that the cluster sequence for the reference clustering is fixed. This results in the following one-sided expected joint entropy when the random clustering $\mathcal{A}$ is drawn from the M_{num} model

$$\mathbb{E}^1_{\text{num}}[H(\mathcal{A},\mathcal{G})] = -\sum_{a=1}^{N}\binom{N}{a}\frac{S(N-a,K_{\mathcal{A}}-1)}{S(N,K_{\mathcal{A}})}\sum_{b=1}^{K_{\mathcal{G}}}\sum_{n}\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{g_b}{n}\binom{N-g_b}{a-n}}{\binom{N}{a}}. \tag{36}$$

The corresponding one-sided expected MI assuming M^1_{num} is

$$\begin{aligned}\mathbb{E}^1_{\text{num}}[\text{MI}(\mathcal{A},\mathcal{G})] &= -\sum_{a=1}^{K_{\mathcal{A}}}\binom{N}{a}\frac{S(N-a,K_{\mathcal{A}}-1)}{S(N,K_{\mathcal{A}})}\frac{a}{N}\log\left(\frac{a}{N}\right) - \sum_{b=1}^{K_{\mathcal{G}}}\frac{g_b}{N}\log\left(\frac{g_b}{N}\right) \\ &+ \sum_{a=1}^{N}\binom{N}{a}\frac{S(N-a,K_{\mathcal{A}}-1)}{S(N,K_{\mathcal{A}})}\sum_{b=1}^{K_{\mathcal{G}}}\sum_{n}\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{g_b}{n}\binom{N-g_b}{a-n}}{\binom{N}{a}}.\end{aligned} \tag{37}$$

The one-sided expected joint entropy when the random clustering $\mathcal{A}$ is drawn from the M^1_{all} model is

$$\mathbb{E}^1_{\text{all}}[H(\mathcal{A},\mathcal{G})] = -\sum_{a=1}^{N}\binom{N}{a}\frac{B_{N-a}}{B_N}\sum_{b=1}^{K_{\mathcal{G}}}\sum_{n}\frac{n}{N}\log\left(\frac{n}{N}\right)\frac{\binom{g_b}{n}\binom{N-g_b}{a-n}}{\binom{N}{a}}, \tag{38}$$

and the one-sided expectation of the MI when the random clustering $\mathcal{A}$ is drawn from the M_{all}^1 model is

$$\mathbb{E}_{\text{all}}^1[\text{MI}(\mathcal{A},\mathcal{G})] = -\sum_{a=1}^{K_\mathcal{A}} \binom{N}{a} \frac{B_{N-a}}{B_N} \frac{a}{N} \log\left(\frac{a}{N}\right) - \sum_{b=1}^{K_\mathcal{G}} \frac{g_b}{N} \log\left(\frac{g_b}{N}\right) + \sum_{a=1}^{N} \binom{N}{a} \frac{B_{N-a}}{B_N} \sum_{b=1}^{K_\mathcal{G}} \sum_{n} \frac{n}{N} \log\left(\frac{n}{N}\right) \frac{\binom{g_b}{n}\binom{N-g_b}{a-n}}{\binom{N}{a}}. \tag{39}$$

Again, the maximum bound must be chosen with respect to the measure maximum over the clusterings present in the random model.

6. Results

The choice of random model for clusterings and the choice of one-sided comparisons can significantly affect results of clustering comparisons. We first illustrate that the ranking of similar clustering pairs (or, equivalently, finding the most similar clustering pair) depends on the choices of random models in a hypothetical example (Section 6.1) and K-means clustering of a handwritten digits data set (Section 6.2). One of the primary reasons such strong discrepancies occur is that the cluster size sequences are fixed within samples from M_{perm}. This means that adjusted comparisons using M_{perm} are unable to differentiate random clusterings with drastically different cluster size sequences, as we illustrate through our third example in Section 6.3. Second, we demonstrate that the interpretation of adjusted clustering similarity measures with respect to a random baseline also depends on the random model through an evaluation of hierarchical clustering applied to gene expression data in Section 6.4. Crucially, all of these examples illustrate that conclusions based on corrected similarity measures can change depending on the random model for clusterings.

6.1 Clustering Similarity Ranking

Our first example demonstrates how rankings assigned by the similarity score can change depending on the assumed random model. Consider the four hypothetical clusterings of 20 elements presented in Figure 1a. Clustering $\mathcal{W}$ contains four equally sized clusters; clustering $\mathcal{X}$ is generated by shifting the membership of one element from $\mathcal{W}$; clustering $\mathcal{Y}$ groups the elements into 10 equally sized clusters; and clustering $\mathcal{Z}$ groups the elements into 10 heterogeneous clusters. The similarity (from the most similar at the top to the least similar at the bottom) of all 6 clustering pairs is ranked using the Rand index and each of its three adjusted variants in Figure 1b. Note that the adjusted Rand index can be negative. The unadjusted Rand index ranking serves as a reference to illustrate how the random models change rankings.

As one would expect, all four Rand measures identify clusterings $\mathcal{W}$ and $\mathcal{X}$ as the most similar (Figure 1b). However, the ranking of the other five comparisons varies widely as a result of the underlying random models. These changes can be understood by tracking comparisons to clustering $\mathcal{Z}$. The low Rand index for the three comparisons with clustering $\mathcal{Z}$ (≈ 0.76) reflects the fact that clustering $\mathcal{Z}$ has a drastically different number of clusters or cluster size sequence from the other three clusterings. The permutation model retains these

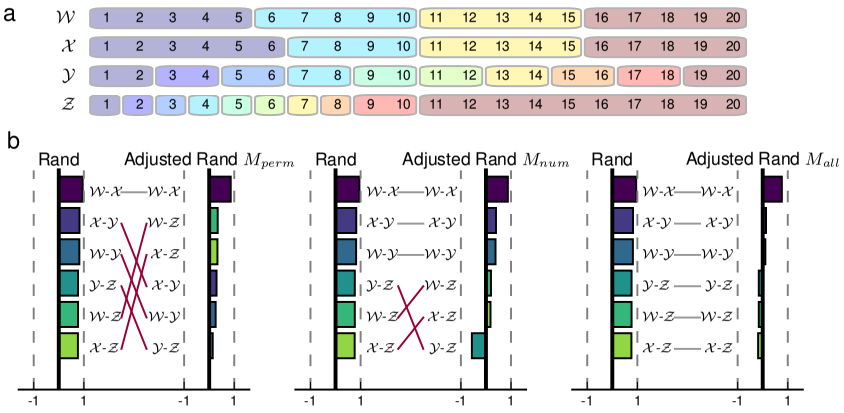


Figure 1: The choice of random model for the Rand index has a significant impact on the rankings of clustering similarity. **a**, Four clusterings of $N = 20$ elements; $\mathcal{W}$ and $\mathcal{X}$ each contain four clusters and differ by the assignment of one element (6), $\mathcal{Y}$ and $\mathcal{Z}$ each contain ten clusters. **b**, Rankings for the similarity of clustering pairs using the Rand index, the Adjusted Rand index assuming M_{perm}, the Adjusted Rand index assuming M_{num}, and the Adjusted Rand index assuming M_{all}. Rankings which change as a function of random model are highlighted in dark red.

differences in all random clusterings; the resulting adjusted index thus treats comparisons between clustering $\mathcal{Z}$ and either $\mathcal{W}$ or $\mathcal{X}$ more favorably than those to clustering $\mathcal{Y}$. On the other hand, the cluster size sequence for clustering $\mathcal{Z}$ is relatively rare in both M_{num} and M_{all}. Since clusterings $\mathcal{Y}$ and $\mathcal{Z}$ have the same number of clusters, the differences in their adjusted scores using M_{num} are a consequence of their cluster size sequences. Finally, all four clusterings are over 20 elements—the only factor that specifies the expected Rand index assuming M_{all}—so they are all adjusted by the same amount when M_{all} is used. Note that in our example, clustering $\mathcal{Z}$ has a negative adjusted Rand score using the M_{all} when compared with all three other clusterings, thus it is less similar to the other three clusterings than one would have expected from comparing two completely random clusterings.

This example illustrates an important property of the M_{all} model. Namely, the ranking provided by the Rand index remains unchanged whenever the M_{all} model is used for adjustment because all clusterings have the same number of elements. However, the corrected baseline now provides a strong interpretation for negative scores: Two randomly selected clusterings are expected to be more similar. This is an important consideration for the evaluation of clustering methods; if the derived clustering is no more similar than would be expected when comparing completely random clusterings, the solution is likely not a meaningful representation of the data.

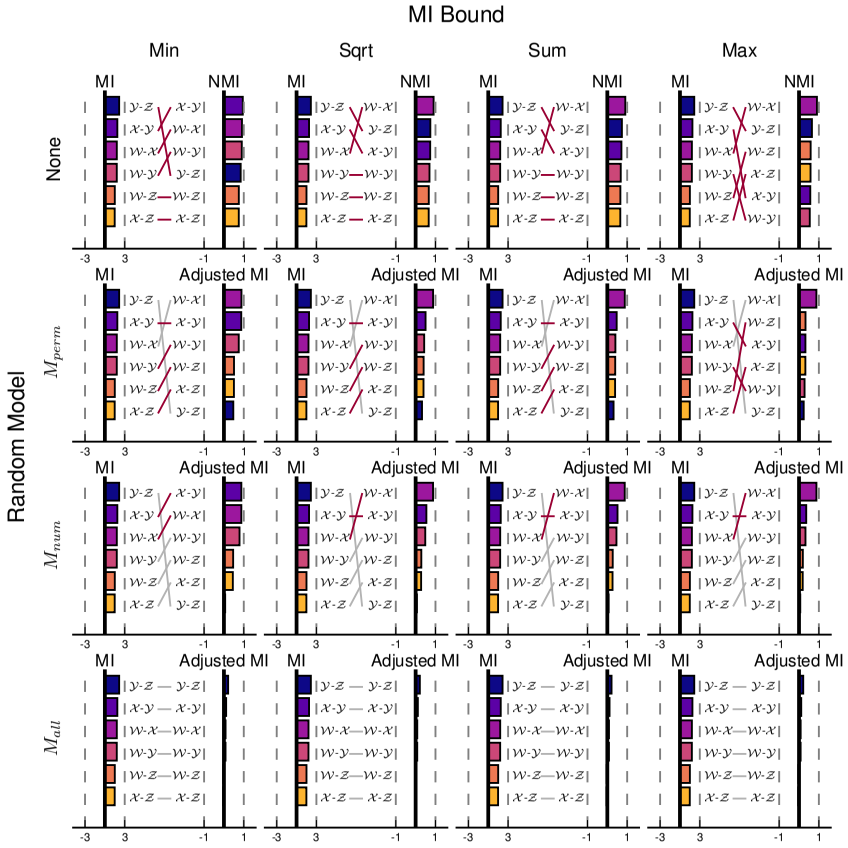


Figure 2: Both the random model and maximum bound have a significant impact on the rankings of clustering similarity using Mutual Information (MI). Rankings for the similarity of clustering pairs from Figure 1a using (vertically) the raw MI, the Adjusted MI assuming M_{perm}, the Adjusted MI assuming M_{num}, and the Adjusted MI assuming M_{all}. MI similarity also depends on the choice of maximum bound (horizontal) as a function of the two clusterings' model entropies; minimum (Min), square-root (Sqrt), average (Sum), and maximum (Max). The MI measures which are normalized but not adjusted by a random model are all members of the common family of normalized MI (NMI). For a given random model, similarity rankings which change as a function of maximum bound are highlighted in dark red.

We then turn our attention to clustering similarity measured by mutual information (MI). Rankings using the adjusted MI depend on two dimensions of variation: the random model and the maximum bound for the measure. This variation is illustrated in Figure 2 using the same 6 comparisons between pairs of clusterings from Figure 1a. We consider four cases for the MI maximum bound: Min, Sqrt, Sum, and Max, corresponding to the minimum of the two model partition entropies, the geometric mean of the two model partition entropies, the average of the two model partition entropies, and the maximum of the two model partition entropies, respectively (see Appendix 5 for details). For the permutation model, the model partition entropies are calculated from the cluster size sequences, while the model partition entropies in M_{num} are bounded by the logarithm of the number of clusters and the model partition entropies in M_{all} are bounded by the logarithm of the number of elements. As a point of reference, all rankings are illustrated in comparison to the raw mutual information score, unnormalized and without a random model adjustment (None). All adjustments of the mutual information without a random model (None, first row) are members of the commonly used family of Normalized Mutual Information (NMI) measures (Danon et al., 2005).

The rankings in Figure 2 demonstrate that both the random model and the maximum bound affect the relative similarity between clusterings when adjusting MI. Firstly, the only random model whose adjustments are independent of MI's maximum bound is M_{all}. This occurs because every choice of the maximum bound reduces to $\log N$ (the entropy of the clustering that places each element into its own cluster). In the other three random model scenarios, the maximum bound depends on the clusterings under comparison. Secondly, MI is highly dependent on the number of clusters in each of the clusterings: When either no normalization and random model adjustment are used, or the M_{all} model is used, MI ranks the similarity of clusterings $\mathcal{Y}$ and $\mathcal{Z}$ above that of $\mathcal{W}$ and $\mathcal{X}$ because of the greater number of clusters in the former case. This bias is mitigated to varying extents by the NMI variations; while NMI using the Sqrt, Sum, and Max normalization terms all produce the intuitive ranking of $\mathcal{W}$ and $\mathcal{X}$ as the most similar pair, NMI using Min for normalization still succumbs to the larger number of clusters in $\mathcal{Y}$, and ranks $\mathcal{X}$ and $\mathcal{Y}$ as the most similar clustering pair. The adjustments provided by both the M_{perm} and M_{num} random models control for the number of clusters; this reduces the impact of the number of clusters when the cluster sizes are regular, but the bias re-occurs when there is a large imbalance between the cluster sizes.

6.2 Appropriate Random Model for Comparing K-means Clusterings

Clustering similarity measures are commonly used to evaluate the results of clustering methods in relation to a known reference clustering. Since the number of clusters can vary between instances, appropriately corrected similarity measures are necessary. However, as we have already seen, the choice of similarity measure and its chance corrected variants can affect the results of the comparisons and suggest drastically different interpretations for the effectiveness of the method.

We demonstrate the importance of the random ensemble assumption through a comparison of the clusterings uncovered by 400 runs of K-means on a collection of hand-written digits (Alimoglu and Alpaydin, 1996, see Appendix B.1 for details). The K-means cluster-

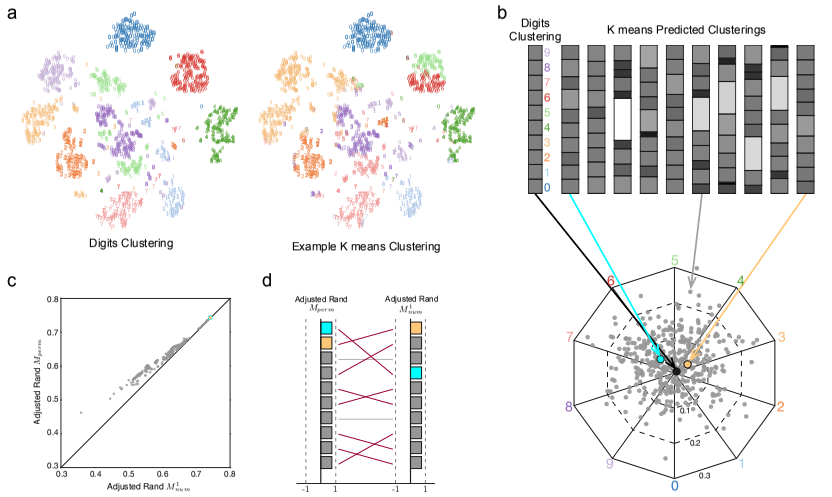


Figure 3: The impact of random model choice on the evaluation of K-means clustering. **a**, The digits data set contains 1,797 points in 64 dimensions (projected to 2 dimensions using t-SNE dimensionality reduction for visualization, Van der Maaten and Hinton, 2008) with a ground truth clustering corresponding to the digit, and an example K-means clustering. **b**, The original cluster size sequence (top left) and 10 cluster size sequences uncovered by K-means clustering with random initialization (top right). Intensity represents cluster sizes that are smaller (darker) or larger (lighter) than the ground truth clusters. (bottom) The cluster size sequence for 400 clusterings uncovered by K-means clustering with random initialization using Barycentric coordinates. The actual clustering size sequence (black) and the most similar clustering determined by the Adjusted Rand index assuming M_{perm} (light blue) and M^1_{num} (light orange). **c**, The similarity between the actual digits clusterings and each of the 400 K-means clusterings as measured by the Adjusted Rand index assuming M_{perm} (y-axis) and M^1_{num} (x-axis). **d**, The ranking of the most similar 10 K-means clusterings as determined by the Adjusted Rand index assuming M_{perm} (left) and M^1_{num} (right).

ing method groups elements so as to minimize the average (Euclidean) distance from the cluster centroid. In most scenarios, it uncovers clusterings with a pre-specified number of clusters (K). For our example, the digits naturally fall into 10 disjoint clusters, shown in Figure 3a, with relative cluster sizes given on the left of Figure 3b. Interestingly, almost all 400 clusterings produced by K-means have a different cluster size sequence (Figure 3b, bottom) and the cluster sizes vary over a wide range (Figure 3b, top). This suggests that both the specific assignment of elements to clusters and the size sequence of the clusters are major factors differentiating the K-means clusterings. Both sources of variation need to be captured by the random model in order to have a meaningful baseline.

Since the number of clusters does not change between runs, but the size sequence of those clusters changes considerably, it is more appropriate to assess similarity within the context of random clusterings with a fixed number of clusters rather than those given by the permutation model. Furthermore, since all of the comparisons are made against the same reference clustering, a one-sided similarity metric better captures the comparison scenario. In Figure 3c, the similarity of the reference clustering compared to each of the 400 uncovered clusterings is shown using the Adjusted Rand index assuming M_{perm} and the Adjusted Rand index assuming M^1_{num}. While the measures are strongly correlated (the black line indicates perfect agreement), the Adjusted Rand index assuming M_{perm} is consistently biased towards higher similarity. Most importantly, the bulk of the uncovered clusterings change their relative ranking when considered in the context of M^1_{num} compared to M_{perm} as demonstrated by the rankings of the top 10 most similar clusterings in Figure 3d.

6.3 Random Models and Inhomogeneous Cluster Sizes

For both the Rand index and MI, the permutation model is invariant to differences in the cluster size sequence. This invariance is explicitly demonstrated in our next example by the difference between the adjusted similarity measures assuming M_{perm} and M_{num}. To generate an increasing disparity in cluster sizes, we use a preferential attachment model of element assignment. At each step of the algorithm, a random element is uniformly chosen for reassignment to a new cluster based on the current sizes of those clusters. A move is rejected if it results in an empty cluster.

In Figure 4, we compare a clustering of 1,000 elements grouped into 50 equally sized clusters and a randomized variant of the same clustering throughout 10^6-steps of our preferential attachment algorithm using the Adjusted Rand index (Figure 4a) and Adjusted MI (Figure 4b). Cluster size inhomogeneity is measured by the entropy of the clustering size sequence; equally sized clusters have the maximum entropy ($\log_2 50 \approx 5.64$), while greater inhomogeneity in cluster sizes decreases the entropy of the cluster size sequence. In both cases, the comparisons assuming M_{perm} are invariant to the inhomogeneity of the cluster size sequences. On the other hand, comparisons assuming M_{num} reflect the changes in the cluster size sequence.

6.4 Performing at Random in Tumor Gene Expression Clustering

Finally, recall that adjusted clustering similarity measures have the added interpretation with respect to a random baseline. Such random baselines play an important role when

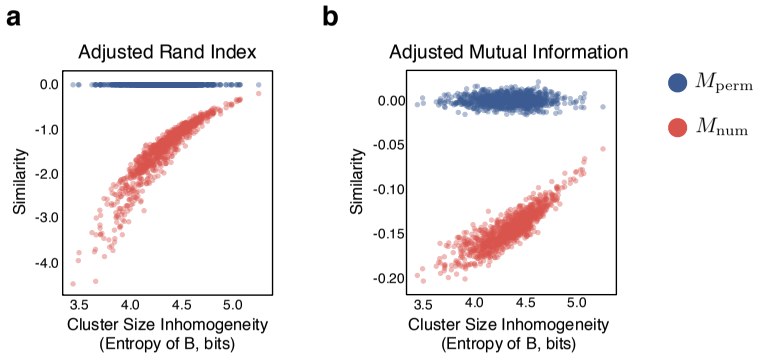
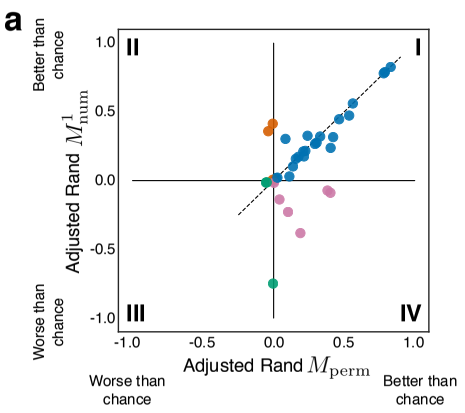


Figure 4: The invariance to inhomogeneous cluster size sequences when assuming M_{perm}. A clustering with 50 equal-sized clusters is compared to a second clustering B generated by a preferential attachment model. The cluster size sequence inhomogenity for clustering B is measured by the cluster size sequence entropy (low entropy is indicative of large cluster size inhomogenity). The similarity is calculated using the adjusted similarity assuming the permutation model M_{perm} and M_{num} for **a**, the Adjusted Rand index, and **b**, the Adjusted Mutual Information. In both cases, the similarity assuming M_{perm} is relatively constant (near 0), while the similarity assuming M_{num} increases with increasing entropy.

evaluating methods in unsupervised learning and classification. In our case, the adjusted similarity measures answer the question: *Is the result of our clustering method more similar to the desired clustering than if we selected a random clustering?* The adjusted similarity measure quantifies an answer to this question: positive scores indicate performance above random, while negative scores indicate a random clustering is more similar.

The interpretation of the adjusted similarity as a random baseline is highly dependent on the assumption of the random model. Critically, if the random model does not reflect the actual ensemble in which the clustering method is searching, the baseline does not accurately reflect the scenario in question. Thus, methods are incorrectly assessed as performing better than randomly generating a clustering.

We illustrate the dependence of adjusted similarity baseline on the choice of random model using a gene expression data set. Specifically, we use a collection of 35 cancer gene expression studies assembled in de Souto et al. (2008). The studies in the collection aim to differentiate the gene expression in cancerous cell tissue samples from those in healthy controls. Each study contains anywhere from 22 to 248 data points (individual tissue samples) for which between 85 and 4,553 features (individual gene expression) were measured after removing the uninformative and missing genes. For details on the individual studies and filtering methodologies, see de Souto et al. (2008) and references therein.



b

vs. M_{perm}

	M^1_{num}	M_{num}	M_{all}	M^1_{all}
I	22	20	1	17
II	3	3	0	5
III	3	3	6	1
IV	7	9	28	12

Figure 5: The impact of random model choice on the evaluation of gene expression clustering with respect to the random baseline. The results of agglomerative hierarchical clustering identified from the gene expression in tissue samples from cancerous and healthy cells in 35 studies. **a**, The uncovered clusterings are compared to the reference clustering using the Adjusted Rand index assuming the permutation model M_{perm} (x-axis), and the one-sided Adjusted Rand index assuming a fixed-number of clusters M^1_{num} (y-axis). The dashed grey line indicates numerical agreement between the similarity measures. There are four possibilities when using two measures to assess similarity with respect to the random baseline: both random models conclude better than chance (blue, quadrant I), both random models conclude worse than chance (green, quadrant III), M_{perm} concludes better than chance but M^1_{num} concludes worse than chance (pink, quadrant IV), and visa-versa (orange, quadrant II). **b**, The assumed random model affects the classification of clustering comparisons with respect to the random baseline in all four random models considered here ($M^1_{\text{num}}, M_{\text{num}}, M_{\text{all}}$ and M^1_{all}) vs. M_{perm}.

Clusterings are identified via agglomerative hierarchical clustering using correlation to compute the average linkage between data points, a common clustering methodology in biology. While many other methods could be used (and indeed, were compared in de Souto et al., 2008), we use hierarchical clustering as a representative example to illustrate the consequences of the random model. Since hierarchical clustering produces a clustering with the user specified number of clusters, its similarity should be adjusted using the one-sided Adjusted Rand index assuming M^1_{num}, where the reference clustering is specified for each study individually.

Figure 5 shows the similarity between the derived clustering and the reference clustering for each of the 35 studies. The Adjusted Rand index assuming M_{perm} is shown on the

x-axis; positive scores (blue and pink points) denote the method performed better than the random baseline, while negative scores (orange and green points) denote the method performed worse than the random baseline. When the Adjusted Rand index assuming M^1_{num} is used (y-axis), a different classification of method performance with respect to the random baseline is found. Of particular note are the seven studies for which the method performed better than chance according to M_{perm}, yet, M^1_{num} concludes the method actually performed worse than chance (pink points). *In this case, a random clustering drawn from the model with a fixed number of clusters would actually perform better than agglomerative hierarchical clustering, yet the practitioner using the permutation model would incorrectly conclude the method was performing better than chance.* This discrepancy occurs even when the values of the Adjusted Rand index assuming M_{perm} are relatively high (> 0.4). Similarly interesting are the three studies in which the method performed worse than chance according to M_{perm}, yet, M^1_{num} concludes the method actually performed better than chance (orange points).

7. Discussion

Given the prevalence of clustering methods for analyzing data, clustering comparison is a fundamental problem that is pertinent to numerous areas of science. In particular, the correction of clustering similarity for chance serves to establish a baseline that facilitates comparisons between different clustering solutions. Expanding previous studies on the selection of an appropriate model for random clusterings (Meila, 2005; Vinh et al., 2009; Romano et al., 2016), our work provides an extensive summary of random models and clearly demonstrates the strong impact of the random model on the interpretation of clustering results.

Our results underpin the importance of selecting the appropriate random model for a given context. To that end, we offer the following guidelines:

1. Consider what is fixed by the clustering method: do all clusterings have a user specified number of clusters (use M_{num}), or is the cluster size sequence fixed (use M_{perm})?

2. Is the comparison against a reference clustering (use a one-sided comparison), or are you comparing two derived clusterings (then use a two-sided comparison)?

The specific comparisons studied here are not meant to establish the superiority of a particular clustering identification technique or a specific random clustering model, rather, they illustrate the importance of the *choice* of the random model. Crucially, conclusions based on corrected similarity measures can change depending on the random model for clusterings. Therefore, previous studies which did promote methods based on evidence from corrected similarity measures should be re-evaluated in the context of the appropriate random model for clusterings (Yeung et al., 2001; de Souto et al., 2008; Yeung and Ruzzo, 2001; Thalamuthu et al., 2006; McNicholas and Murphy, 2010).

Throughout this work, we assumed a uniform probability of selecting a partition given a constraint on the types of partitions in the ensemble. However, other probability distributions could be used which better model the clusterings encountered in practice. For example, instead of using a uniform distribution over the number of clusters, one could consider an inferred distribution for the number of clusters actually uncovered by a given

method (e.g. affinity propagation). This is particularly relevant when considering M_{all}, an extreme case for random partitions. Additionally, given that many systems exhibit clusterings with a heavy-tailed cluster size sequence, clusterings with such skewed cluster size distributions could be favored. Changes to the prior probabilities would likely change the expectations of the clustering similarity measures.

The behavior of the Rand index and Mutual Information in the context of the random clustering models discussed here further reveals problems with both measures. Specifically, the expected similarity of random clusterings increases as the number of elements grows. Intuition would suggest the opposite; the similarity of two randomly selected clusterings should decrease as the number of elements increases because it is harder to match the element memberships to clusters between two random clusterings. Instead, both MI and Rand are dominated by the fact that the expected number of clusters and cluster size distribution are converging with increasing N (Mansour, 2012). Our analysis also illustrates the dependency on the normalization term for MI, which, combined with a previously established bias on the number of clusters, suggests more care should be taken when interpreting the results of MI clustering comparisons.

In conclusion, our framework for the correction of clustering similarity for chance allows for more conscious comparisons between clusterings. The practitioner should always provide justification for their choice of random clustering model and treatment of one-sided comparisons.

Acknowledgments

We thank Ian Wood, Santosh Manicka, James Bagrow, Sune Lehmann, Aaron Clauset, Randall D. Beer, and Luis M. Rocha for helpful discussions. We also thank two anonymous reviewers for their wonderful insights and suggestions to improve the paper. A package to computer the adjusted similarity measures is available on the author's github:
`https://github.com/ajgates42/clusim`

Appendix A. Stirling and Bell Numbers

The Stirling number of the second kind $S(n,k)$ gives the number of ways to partition a set of n elements into k clusters, where

$$S(n,k) = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^n. \tag{40}$$

There are several recurrence relations which also give $S(n,k)$, one of the most useful is the relation

$$\begin{aligned} S(0,0) &= 1 \qquad S(n,0) = S(0,n) = 0 \\ S(n+1,k) &= kS(n,k) + S(n,k-1). \end{aligned} \tag{41}$$

As $n \to \infty$, an asymptotic approximation to the Stirling numbers of the second kind for a fixed k is given by $S(n,k) \approx \frac{k^n}{k!}$.

The Bell number B_n is the total number of clusterings over a set with n elements. It is related the Stirling numbers of the second kind by the summation over k for a fixed n, $B_n = \sum_{k=0}^{n} S(n,k)$. There is also a useful recurrence relation for Bell numbers: $B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$. As $n \to \infty$, an asymptotic approximation to the ratio of the n-th and $(n+1)$-th Bell numbers is $\frac{B_n}{B_{n+1}} \approx \frac{\log n}{n}$. See Mansour (2012) for an extended discussion of both the Stirling numbers of the second kind and the Bell numbers.

In practice, calculating the Bell numbers and Stirling numbers of the second kind from their recurrence relations can be computationally expensive. However, many efficient approximations and implementations are available (Temme, 1993; Mansour, 2012). Here, we make use of the mpmath arbitrary precision library for Python developed by Johansson et al. (2013). This library takes advantage of Dobiński's Formula to approximate the Bell numbers (Dobiński, 1877; Chen and Yeh, 1994).

Appendix B. Application Data Sets

B.1 Digits Data Set

The digits data set is bundled with the sci-kit learn source code and consists of $1,797$ images of 8×8 gray level pixels of handwritten digits. The reference clustering contains 10 clusters corresponding to the true digit. The data set was originally assembled in Alimoglu and Alpaydin (1996). To provide a visualization, the data was projected to 2-d using the t-Distributed Stochastic Neighbor Embedding (t-SNE) dimensionality reduction method (Van der Maaten and Hinton, 2008) initialized from the pca decomposition.

B.2 Gene Expression Data Set

The data was assembled in de Souto et al. (2008) and is freely available from `http://bioinformatics.rutgers.edu/Publications/deSouto2008c/index.html`. The studies represent two prominent methods for determining gene expression in cell tissue samples from cancer tumors or healthy controls, Affymetrix microarrays and cDNA microarrays, which, respectively, measure the number of RNA copies found in the cell and the ratio of the number of copies vs a control sample. Each study contains anywhere from 22 to 248 data points (individual tissue samples) for which between 85 and $4,553$ features (individual gene expression) were measured after removing the uninformative and missing genes. Please see de Souto et al. (2008) for details of this selection process.

References

Ahmed N. Albatineh and Magdalena Niewiadomska-Bugaj. Correcting Jaccard and other similarity indices for chance agreement in cluster analysis. *Advances in Data Analysis and Classification*, 5(3):179–200, 2011.

Ahmed N. Albatineh, Magdalena Niewiadomska-Bugaj, and Daniel Mihalko. On similarity indices and correction for chance agreement. *Journal of Classification*, 23(2):301–313, 2006.

Fevzi Alimoglu and Ethem Alpaydin. Methods of combining multiple classifiers based on different representations for pen-based handwritten digit recognition. In *Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96*, 1996.

Alessia Amelio and Clara Pizzuti. Is normalized mutual information a fair measure for comparing community detection methods? In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 1584–1585. ACM, 2015.

Beifang Chen and Yeong-Nan N. Yeh. Some explanations of dobinski's formula. *Studies in Applied Mathematics*, 92(3):191–199, 1994.

Bobbie Chern, Persi Diaconis, Daniel M. Kane, and Robert C. Rhoades. Closed expressions for averages of set partition statistics. *Research in the Mathematical Sciences*, 1(1):1–32, 2014.

Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005 (09):P09008, 2005.

Marcilio CP de Souto, Ivan G. Costa, Daniel SA de Araujo, Teresa B. Ludermir, and Alexander Schliep. Clustering cancer gene expression data: a comparative study. *BMC Bioinformatics*, 9(1):1, 2008.

Donald G. Dobiński. Summirung der reihe $\sum n^m/n!$ für m= 1, 2, 3, 4, 5,.... *Arch. der Mat. und Physik*, 61:333–336, 1877.

Janice L. DuBien and William D. Warde. Some distributional results concerning a comparative statistic used in cluster analysis. In *ASA Proceedings of the Social Statistics Section*, pages 309–313, 1981.

Janice L. DuBien, William D. Warde, and Seong S. Chae. Moments of Rand's C statistic in cluster analysis. *Statistics & Probability Letters*, 69(3):243–252, 2004.

Edward B. Fowlkes and Colin L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2 2010.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2 (1):193–218, December 1985.

Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31 (8):651–666, 2010.

Fredrik Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.18)*, December 2013. `http://mpmath.org/`.

Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical Review E*, 80(5):056117, August 2009.

Toufik Mansour. *Combinatorics of Set Partitions*. CRC Press, 2012.

Paul D. McNicholas and Thomas Brendan Murphy. Model-based clustering of microarray expression data via latent gaussian mixture models. *Bioinformatics*, 26(21):2705–2712, 2010.

Marina Meila. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 577–584, New York, NY, USA, 2005. ACM.

Darius Pfitzner, Richard Leibbrandt, and David Powers. Characterization and evaluation of similarity measures for pairs of clusterings. *Knowledge and Information Systems*, 19 (3):361–394, 2009.

William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846, 1971.

Simone Romano, James Bailey, Vinh Nguyen, and Karin Verspoor. Standardized mutual information for clustering comparisons: one step further in adjustment for chance. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1143–1151, 2014.

Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. Adjusting for chance clustering comparison measures. *Journal of Machine Learning Research*, 17:1–32, 2016.

James Sethna. *Statistical mechanics: entropy, order parameters, and complexity*, volume 14. Oxford University Press, 2006.

D. Steinley, M. J. Brusco, and L. Hubert. The variance of the adjusted Rand index. *Psychological Methods*, 2016.

Nico M. Temme. Asymptotic estimates of stirling numbers. *Studies in Applied Mathematics*, 89(3):233–243, 1993.

Anbupalam Thalamuthu, Indranil Mukhopadhyay, Xiaojing Zheng, and George C. Tseng. Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22(19):2405–2412, 2006.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML-09)*, pages 1073–1080, 2009.

Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010.

David L. Wallace. A method for comparing two hierarchical clusterings: comment. *Journal of the American Statistical Association*, 78(383):569–576, 1983.

Allan P. White and Wei Zhong Liu. Technical note: bias in information-based measures in decision tree induction. *Machine Learning*, 15(3):321–329, 1994.

Ka Yee Yeung and Walter L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

Ka Yee Yeung, Chris Fraley, Alejandro Murua, Adrian E. Raftery, and Walter L. Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987, 2001.

Pan Zhang. Evaluating accuracy of community detection using the relative normalized mutual information. *Journal of Statistical Mechanics: Theory and Experiment*, 2015(11): P11006, 2015.

Journal of Machine Learning Research 18 (2017) 1-35 Submitted 2/17; Revised 5/17; Published 8/17

Hierarchical Clustering via Spreading Metrics

Aurko Roy AURKO@GATECH.EDU
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA

Sebastian Pokutta SEBASTIAN.POKUTTA@ISYE.GATECH.EDU
ISyE
Georgia Institute of Technology
Atlanta, GA 30332, USA

Editor: Sanjoy Dasgupta

Abstract

We study the cost function for hierarchical clusterings introduced by (Dasgupta, 2016) where hierarchies are treated as first-class objects rather than deriving their cost from projections into flat clusters. It was also shown in (Dasgupta, 2016) that a top-down algorithm based on the *uniform* Sparsest Cut problem returns a hierarchical clustering of cost at most $O(\alpha_n \log n)$ times the cost of the optimal hierarchical clustering, where α_n is the approximation ratio of the Sparsest Cut subroutine used. Thus using the best known approximation algorithm for Sparsest Cut due to Arora-Rao-Vazirani, the top-down algorithm returns a hierarchical clustering of cost at most $O\left(\log^{3/2} n\right)$ times the cost of the optimal solution. We improve this by giving an $O(\log n)$-approximation algorithm for this problem. Our main technical ingredients are a combinatorial characterization of ultrametrics induced by this cost function, deriving an Integer Linear Programming (ILP) formulation for this family of ultrametrics, and showing how to iteratively round an LP relaxation of this formulation by using the idea of *sphere growing* which has been extensively used in the context of graph partitioning. We also prove that our algorithm returns an $O(\log n)$- approximate hierarchical clustering for a generalization of this cost function also studied in (Dasgupta, 2016). Experiments show that the hierarchies found by using the ILP formulation as well as our rounding algorithm often have better projections into flat clusters than the standard linkage based algorithms. We conclude with constant factor inapproximability results for this problem: 1) no polynomial size LP or SDP can achieve a constant factor approximation for this problem and 2) no polynomial time algorithm can achieve a constant factor approximation under the *Small Set Expansion* hypothesis.

Keywords: Hierarchical clustering, clustering, convex optimization, linear programming, approximation algorithms

1. Introduction

Hierarchical clustering is an important method in cluster analysis where a data set is recursively partitioned into clusters of successively smaller size. They are typically represented by rooted trees where the root corresponds to the entire data set, the leaves correspond to individual data points and the intermediate nodes correspond to a cluster of its descendant leaves. Such a hierarchy represents several possible *flat clusterings* of the data at various levels of granularity; indeed every pruning of this tree returns a possible clustering. Therefore in situations where the number of desired clusters is not known beforehand, a hierarchical clustering scheme is often preferred to flat clustering.

The most popular algorithms for hierarchical clustering are bottom-up agglomerative algorithms like *single linkage*, *average linkage* and *complete linkage*. In terms of theoretical guarantees these algorithms are known to correctly recover a ground truth clustering if the similarity function on the data satisfies corresponding stability properties (see, e.g., (Balcan et al., 2008)). Often, however, one wishes to think of a good clustering as optimizing some kind of cost function rather than recovering a hidden "ground truth", since one can then assign a numerical score to an algorithm's performance on every instance. This is useful for objectively comparing the output of different clustering algorithms. Popular objective functions in the classical clustering setting are k-means, k-median, min-sum and k-center (see Chapter 14, (Friedman et al., 2001)). However as pointed out by (Dasgupta, 2016) for a lot of popular hierarchical clustering algorithms including linkage based algorithms, it is hard to pinpoint explicitly the cost function that these algorithms are optimizing. Moreover, much of the existing cost function based approaches towards hierarchical clustering evaluate a hierarchy based on a cost function for flat clustering, e.g., assigning the k-means or k-median cost to a pruning of this tree. Motivated by this, (Dasgupta, 2016) introduced a cost function for hierarchical clustering where the cost takes into account the entire structure of the tree rather than just the projections into flat clusterings. This cost function is shown to recover the intuitively correct hierarchies on several synthetic examples like planted partitions and cliques. In addition, a top-down graph partitioning algorithm based on the *uniform* Sparsest Cut problem is presented that outputs a tree with cost at most $O(\alpha_n \log n)$ times the cost of the optimal tree and where α_n is the approximation guarantee of the Sparsest Cut subroutine used. Thus using the Leighton-Rao algorithm (Leighton and Rao, 1988, 1999) or the Arora-Rao-Vazirani algorithm (Arora et al., 2009) gives an approximation factor of $O\left(\log^2 n\right)$ and $O\left(\log^{3/2} n\right)$ respectively.

In this work we give a polynomial time algorithm to recover a hierarchical clustering of cost at most $O(\log n)$ times the cost of the optimal clustering according to this cost function. We also analyze a generalization of this cost function studied by (Dasgupta, 2016) and show that our algorithm still gives an $O(\log n)$ approximation in this setting. We do this by viewing the cost function in terms of the ultrametric it induces on the data, writing a convex relaxation for it and concluding by analyzing a popular rounding scheme used in graph partitioning algorithms. We also implement the integer program, its LP relaxation, and the rounding algorithm and test it on some synthetic and real world data sets to compare the cost of the rounded solutions to the true optimum as well as to compare its performance to other hierarchical clustering algorithms used in practice. Our experiments suggest that the hierarchies found by this algorithm are often better than the ones found by linkage based algorithms as well as the k-means algorithm in terms of the error of the best pruning of the tree compared to the ground truth.

1.1 Related Work

The immediate precursor to this work is (Dasgupta, 2016) where the cost function for evaluating a hierarchical clustering was introduced. Prior to this there has been a long line of research on hierarchical clustering in the context of phylogenetics and taxonomy (see, e.g., (Jardine and Sibson, 1971; Sneath et al., 1973; Felsenstein and Felenstein, 2004)). Several authors have also given theoretical justifications for the success of the popular linkage based algorithms for hierarchical clustering (see, e.g. (Jardine and Sibson, 1968; Zadeh and Ben-David, 2009; Ackerman et al., 2010)). In terms of cost functions, one approach has been to evaluate a hierarchy in terms of the k-means or k-median cost that it induces (see (Dasgupta and Long, 2005)). The cost function and the top-down algorithm in (Dasgupta, 2016) can also be seen as a theoretical justification for several graph partitioning heuristics that are used in practice.

Besides this prior work on hierarchical clustering, we are also motivated by the long line of work in the classical clustering setting where a popular strategy is to study convex relaxations of these problems and to round an optimal fractional solution into an integral one with the aim of getting a good approximation to the cost function. A long line of work (see, e.g., (Charikar et al., 1999; Jain and Vazirani, 2001; Jain et al., 2003; Charikar and Li, 2012)) has employed this approach on LP relaxations for the k-median problem, including (Li and Svensson, 2013) which gives the best known approximation factor of $1+\sqrt{3}+\varepsilon$. Similarly, a few authors have studied LP and SDP relaxations for the k-means problem (see, e.g., (Peng and Xia, 2005; Peng and Wei, 2007; Awasthi et al., 2015)), while one of the best known algorithms for kernel k-means and

©2017 Aurko Roy and Sebastian Pokutta.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at

spectral clustering is due to (Recht et al., 2012) which approximates the nonnegative matrix factorization (NMF) problem by LPs.

LP relaxations for hierarchical clustering have also been studied in (Ailon and Charikar, 2005) where the objective is to fit a tree metric to a data set given pairwise dissimilarities. While the LP relaxation and rounding algorithm in (Ailon and Charikar, 2005) is similar in flavor, the result is incomparable to ours (see Section 8 for a discussion). Another work that is indirectly related to our approach is (Di Summa et al., 2015) where the authors study an ILP to obtain a closest ultrametric to arbitrary functions on a discrete set. Our approach is to give a combinatorial characterization of the ultrametrics induced by the cost function of (Dasgupta, 2016) which allows us to use the tools from (Di Summa et al., 2015) to model the problem as an ILP. The natural LP relaxation of this ILP turns out to be closely related to LP relaxations considered before for several graph partitioning problems (see, e.g., (Leighton and Rao, 1988, 1999; Even et al., 1999; Krauthgamer et al., 2009)) and we use a rounding technique studied in this context to round this LP relaxation.

Recently, we became aware of independent work by (Charikar and Chatziafratis, 2017) obtaining similar results for hierarchical clustering. In particular (Charikar and Chatziafratis, 2017) improve the approximation factor to $O(\alpha_n)$, where α_n is the approximation factor of the uniform Sparsest Cut problem, by showing how to round a spreading metric SDP relaxation. The analysis of this rounding procedure also enabled them to show that the top-down heuristic of (Dasgupta, 2016) actually returns an $O(\alpha_n)$ approximate clustering rather than an $O(\alpha_n \log n)$ approximate clustering. This improves the approximation factor to $O(\sqrt{\log n})$ using (Arora et al., 2009) for solving the uniform Sparsest Cut problem. (Charikar and Chatziafratis, 2017) also analyzed a similar LP relaxation using the *divide-and-conquer approximation algorithms using spreading metrics* paradigm of (Even et al., 2000) together with a result of (Bartal, 2004) to show an $O(\log n)$ approximation. Finally, (Charikar and Chatziafratis, 2017) also gave similar constant factor inapproximability results for this problem.

1.2 Contribution

While studying convex relaxations of optimization problems is fairly natural, for the cost function introduced in (Dasgupta, 2016) however, it is not immediately clear how one would go about writing such a relaxation. Our first contribution is to give a combinatorial characterization of the family of ultrametrics induced by this cost function on hierarchies. Inspired by the approach in (Di Summa et al., 2015) where the authors study an integer linear program for finding the closest ultrametric, we are able to formulate the problem of finding the minimum cost hierarchical clustering as an integer linear program. Interestingly and perhaps unsurprisingly, the specific family of ultrametrics induced by this cost function give rise to linear constraints studied before in the context of finding balanced separators in weighted graphs. We then show how to round an optimal fractional solution using the *sphere growing* technique first introduced in (Leighton and Rao, 1988) (see also (Garg et al., 1996; Even et al., 1999; Charikar et al., 2003)) to recover a tree of cost at most $O(\log n)$ times the optimal tree for this cost function. The generalization of this cost function involves scaling every pairwise distances by an arbitrary strictly increasing function f satisfying $f(0) = 0$. We modify the integer linear program for this general case and show that the rounding algorithm still finds a hierarchical clustering of cost at most $O(\log n)$ times the optimal clustering in this setting. Complementing the algorithms, we also show a constant factor inapproximability result for this problem under two different settings: 1) for any polynomial time algorithm under the *Small Set Expansion* hypothesis and, 2) unconditionally, for any polynomial sized LP and SDP relaxations. We conclude with an experimental study of the integer linear program and the rounding algorithm on some synthetic and real world data sets to show that the approximation algorithm often recovers clusters close to the true optimum (according to this cost function) and that its projections into flat clusters often has a better error rate than the linkage based algorithms and the k-means algorithm.

2. Preliminaries

A similarity based clustering problem consists of a dataset V of n points and a *similarity function* $\kappa : V \times V \to \mathbb{R}_{\geq 0}$ such that $\kappa(i,j)$ is a measure of the similarity between i and j for any $i,j \in V$. We will assume that the similarity function is symmetric i.e., $\kappa(i,j) = \kappa(j,i)$ for every $i,j \in V$. Note that we do not make any assumptions about the points in V coming from an underlying metric space. For a given instance of a clustering problem we have an associated weighted complete graph K_n with vertex set V and weight function given by κ. A *hierarchical clustering* of V is a tree T with a designated root r and with the elements of V as its leaves, i.e., $\text{leaves}(T) = V$. For any set $S \subseteq V$ we denote the *lowest common ancestor* of S in T by $\text{lca}(S)$. For pairs of points $i,j \in V$ we will abuse the notation for the sake of simplicity and denote $\text{lca}(\{i,j\})$ simply by $\text{lca}(i,j)$. For a node v of T we denote the subtree of T rooted at v by $T[v]$. The following cost function was introduced by (Dasgupta, 2016) to measure the quality of the hierarchical clustering T

$$\text{cost}(T) \coloneqq \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) \left|\text{leaves}(T[\text{lca}(i,j)])\right|. \tag{1}$$

The intuition behind this cost function is as follows. Let T be a hierarchical clustering with designated root r so that r represents the whole data set V. Since $\text{leaves}(T) = V$, every internal node $v \in T$ represents a cluster of its descendant leaves, with the leaves themselves representing singleton clusters of V. Starting from r and going down the tree, every distinct pair of points $i,j \in V$ will be eventually separated at the leaves. If $\kappa(i,j)$ is large, i.e., i and j are very similar to each other then we would like them to be separated as far down the tree as possible if T is a good clustering of V. This is enforced in the cost function (1): if $\kappa(i,j)$ is large then the number of leaves of $\text{lca}(i,j)$ should be small i.e., $\text{lca}(i,j)$ should be far from the root r of T. Such a cost function is not unique however; see Section 8 for some other cost functions of a similar flavor.

Note that while requiring κ to be non-negative might seem like an artificial restriction, cost function (1) breaks down when all the $\kappa(i,j) < 0$, since in this case the trivial clustering r, T^* where T^* is the star graph with V as its leaves is always the minimizer. Therefore in the rest of this work we will assume that $\kappa \geq 0$. This is not a restriction compared to (Dasgupta, 2016), since the Sparsest Cut algorithm used as a subroutine also requires this assumption. Let us now briefly recall the notion of an ultrametric.

Definition 1 (Ultrametric) *An* ultrametric *on a set X of points is a distance function $d : X \times X \to \mathbb{R}$ satisfying the following properties for every $x,y,z \in X$*

1. ***Nonnegativity:*** *$d(x,y) \geq 0$ with $d(x,y) = 0$ iff $x = y$,*
2. ***Symmetry:*** *$d(x,y) = d(y,x)$,*
3. ***Strong triangle inequality:*** *$d(x,y) \leq \max\{d(y,z), d(z,x)\}$.*

Note that due to the requirement of strong triangle inequality, ultrametrics only admit *acute isosceles triangles* and *equilaterial triangles*, in contrast to general metric spaces. Under the cost function (1), one can interpret the tree T as inducing an ultrametric d_T on V given by $d_T(i,j) \coloneqq |\text{leaves}(T[\text{lca}(i,j)])| - 1$. This is an ultrametric since $d_T(i,j) = 0$ iff $i = j$ and for any triple $i,j,k \in V$ we have $d_T(i,j) \leq \max\{d_T(i,k), d_T(j,k)\}$. In the following definition we introduce the notion of *non-trivial ultrametrics*, which turn out to be precisely the ultrametrics that are induced by tree decompositions of V corresponding to cost function (1), as we will show in Corollary 8. We use the notation $x \sim y$ to denote an equivalence relation between x and y. Note that if d is an ultrametric, then $x \sim y$ iff $d(x,y) \leq t$ is an equivalence relation for every $t \in \mathbb{R}$.

Definition 2 *An ultrametric d on a set of points V is* non-trivial *if the following conditions hold.*

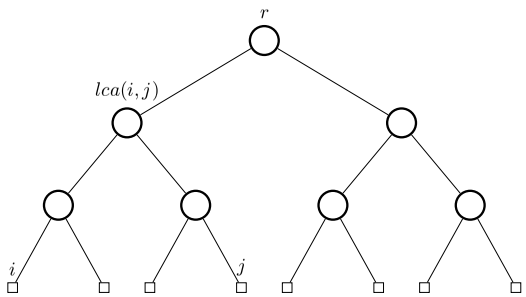


Figure 1: Example hierarchical clustering showing the lca of two data points i and j.

1. *For every non-empty set $S \subseteq V$, there is a pair of points $i, j \in S$ such that $d(i,j) \geq |S| - 1$.*
2. *For any t, if S_t is an equivalence class of V under the relation $i \sim j$ iff $d(i,j) \leq t$, then*

$$\max_{i,j \in S_t} d(i,j) \leq |S_t| - 1.$$

Note that for an equivalence class S_t where $d(i,j) \leq t$ for every $i, j \in S_t$ it follows from Condition 1 that $t \geq |S_t| - 1$. Thus in the case when $t = |S_t| - 1$ the two conditions imply that the maximum distance between any two points in S is t and that there is a pair $i, j \in S$ for which this maximum is attained. The following lemma shows that non-trivial ultrametrics behave well under restrictions to equivalence classes S_t of the form $i \sim j$ iff $d(i,j) \leq t$. All missing proofs from this section can be found in Appendix A.

Lemma 3 *Let d be a non-trivial ultrametric on V and let $S_t \subseteq V$ be an equivalence class under the relation $i \sim j$ iff $d(i,j) \leq t$. Then d restricted to S_t is a non-trivial ultrametric on S_t.*

The intuition behind the two conditions in Definition 2 is as follows. Condition 1 imposes a certain lower bound by ruling out trivial ultrametrics where, e.g., $d(i,j) = 1$ for every distinct pair $i, j \in V$. On the other hand Condition 2 discretizes and imposes an upper bound on d by restricting its range to the set $\{0, 1, \ldots, n-1\}$ (see Lemma 4). This rules out the other spectrum of triviality where for example $d(i,j) = n$ for every distinct pair $i, j \in V$ with $|V| = n$.

Lemma 4 *Let d be a non-trivial ultrametric on the set V as in Definition 2. Then the range of d is contained in the set $\{0, 1, \ldots, n-1\}$ with $|V| = n$.*

3. Ultrametrics and Hierarchical Clusterings

We start with the following easy lemma about the lowest common ancestors of subsets of V in a hierarchical clustering T of V. See Figure 1 for a helpful illustration. All missing proofs from this section can be found in Appendix B.

Lemma 5 *Let $S \subseteq V$ with $|S| \geq 2$. If $r = \mathrm{lca}(S)$ then there is a pair $i, j \in S$ such that $\mathrm{lca}(i,j) = r$.*

We will now show that non-trivial ultrametrics on V as in Definition 2 are exactly those that are induced by hierarchical clusterings on V under cost function (1). The following lemma shows the forward direction: the ultrametric d_T induced by any hierarchical clustering T is non-trivial.

Lemma 6 *Let T be a hierarchical clustering on V and let d_T be the ultrametric on V induced by it. Then d_T is non-trivial.*

The following crucial lemma shows the converse: every non-trivial ultrametric on V is realized by a hierarchical clustering T of V.

Lemma 7 *For every non-trivial ultrametric d on V there is a hierarchical clustering T on V such that for any pair $i, j \in V$ we have*

$$d_T(i,j) = |\mathrm{leaves}(T[\mathrm{lca}\,(i,j)])| - 1 = d(i,j).$$

Moreover this hierarchy can be constructed in time $O\left(n^3\right)$ by Algorithm 1 where $|V| = n$.

Lemmas 6 and 7 together imply the following corollary about the equivalence of hierarchical clusterings and non-trivial ultrametrics.

Corollary 8 *There is a bijection between the set of hierarchical clusterings T on V and the set of non-trivial ultrametrics d on V satisfying the following conditions.*

1. *For every hierarchical clustering T on V, there is a non-trivial ultrametric d_T defined as $d_T(i,j) := |\mathrm{leaves}\, T[\mathrm{lca}(i,j)]| - 1$ for every $i, j \in V$.*
2. *For every non-trivial ultrametric d on V, there is a hierarchical clustering T on V such that for every $i, j \in V$ we have $|\mathrm{leaves}\, T[\mathrm{lca}(i,j)]| - 1 = d(i,j)$.*

Moreover this bijection can be computed in $O(n^3)$ time, where $|V| = n$.

Therefore to find the hierarchical clustering of minimum cost, it suffices to minimize $\langle \kappa, d \rangle$ over non-trivial ultrametrics $d : V \times V \to \{0, \ldots, n-1\}$, where V is the data set. Note that the cost of the ultrametric d_T corresponding to a tree T is an affine offset of $\mathrm{cost}(T)$. In particular, we have $\langle \kappa, d_T \rangle = \mathrm{cost}(T) - \sum_{\{i,j\} \in E(K_n)} \kappa(i,j)$.

A natural approach is to formulate this problem as an Integer Linear Program (ILP) and then study LP or SDP relaxations of it. We consider the following ILP for this problem that is motivated by (Di Summa et al., 2015). We have the variables $x_{ij}^1, \ldots, x_{ij}^{n-1}$ for every distinct pair $i, j \in V$ with $x_{ij}^t = 1$ if and only if $d(i,j) \geq t$. For any positive integer n, let $[n] := \{1, 2, \ldots, n\}$.

$$\min \quad \sum_{t=1}^{n-1} \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) x_{ij}^t \qquad \text{(ILP-ultrametric)}$$

$$\text{s.t.} \quad x_{ij}^t \geq x_{ij}^{t+1} \qquad \forall i, j \in V, t \in [n-2] \tag{2}$$

$$x_{ij}^t + x_{jk}^t \geq x_{ik}^t \qquad \forall i, j, k \in V, t \in [n-1] \tag{3}$$

$$\sum_{i,j \in S} x_{ij}^t \geq 2 \qquad \forall t \in [n-1], S \subseteq V, |S| = t+1 \tag{4}$$

$$\sum_{i,j \in S} x_{ij}^{|S|} \leq |S|^2 \left(\sum_{i,j \in S} x_{ij}^t + \sum_{\substack{i \in S \\ j \notin S}} \left(1 - x_{ij}^t\right) \right) \quad \forall t \in [n-1], S \subseteq V \tag{5}$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i, j \in V, t \in [n-1] \tag{6}$$

$$x_{ii}^t = 0 \qquad \forall i \in V, t \in [n-1] \tag{7}$$

$$x_{ij}^t \in \{0, 1\} \qquad \forall i, j \in V, t \in [n-1] \tag{8}$$

Input: Data set V of n points, non-trivial ultrametric $d : V \times V \to \mathbb{R}_{\geq 0}$
Output: Hierarchical clustering T of V with root r

$r \leftarrow$ arbitrary choice of designated root in V
$X \leftarrow \{r\}$
$E \leftarrow \emptyset$
if $n = 1$ **then**
  $T \leftarrow (X, E)$
  return r, T
else
  Partition V into $\{V_1, \dots V_m\}$ under the equivalence relation $i \sim j$ iff $d(i,j) < n-1$
  for $l \in \{1, \dots, m\}$ **do**
    Let r_l, T_l be output of Algorithm 1 on V_l, $d|_{V_l}$
    $X \leftarrow X \cup V(T_l)$
    $E \leftarrow E \cup \{r, r_l\}$
  end
  $T \leftarrow (X, E)$
  return r, T
end

Algorithm 1: Hierarchical clustering of V from non-trivial ultrametric

Constraints (2) and (7) follow from the interpretation of the variables x_{ij}^t: if $d(i,j) \geq t$, i.e., $x_{ij}^t = 1$ then clearly $d(i,j) \geq t-1$ and so $x_{ij}^{t-1} = 1$. Furthermore, for any $i \in V$ we have $d(i,i) = 0$ and so $x_{ii}^t = 0$ for every $t \in [n-1]$. Note that constraint (3) is the same as the *strong triangle inequality* (Definition 1) since the variables x_{ij}^t are in $\{0,1\}$. Constraint (6) ensures that the ultrametric is symmetric. Constraint (4) ensures the ultrametric satisfies Condition 1 of non-triviality: for every $S \subseteq V$ of size $t+1$ we know that there must be points $i, j \in S$ such that $d(i,j) = d(j,i) \geq t$ or in other words $x_{ij}^t = x_{ji}^t = 1$. Constraint (5) ensures that the ultrametric satisfies Condition 2 of non-triviality. To see this note that the constraint is active only when $\sum_{i,j \in S} x_{ij}^t = 0$ and $\sum_{i \in S, j \notin S}(1 - x_{ij}^t) = 0$. In other words $d(i,j) \leq t-1$ for every $i, j \in S$ and S is a maximal such set since if $i \in S$ and $j \notin S$ then $d(i,j) \geq t$. Thus S is an equivalence class under the relation $i \sim j$ iff $d(i,j) \leq t-1$ and so for every $i, j \in S$ we have $d(i,j) \leq |S| - 1$ or equivalently $x_{ij}^{|S|} = 0$. The ultrametric d represented by a feasible solution x_{ij}^t is given by $d(i,j) = \sum_{t=1}^{n-1} x_{ij}^t$.

Definition 9 *For any* $\left\{x_{ij}^t \mid t \in [n-1], i, j \in V\right\}$ *let* E_t *be defined as* $E_t := \left\{\{i,j\} \mid x_{ij}^t = 0\right\}$. *Note that if* x_{ij}^t *is feasible for ILP-ultrametric then* $E_t \subseteq E_{t+1}$ *for any* t *since* $x_{ij}^t \geq x_{ij}^{t+1}$. *The sets* $\{E_t\}_{t=1}^{n-1}$ *induce a natural sequence of graphs* $\{G_t\}_{t=1}^{n-1}$ *where* $G_t = (V, E_t)$ *with* V *being the data set.*

For a fixed $t \in \{1, \dots, n-1\}$, it is instructive to study the combinatorial properties of the *layer-t problem*, where we restrict ourselves to the constraints corresponding to that particular t and drop constraints (2) and

(5) since they involve different layers in their expression.

$$\min \quad \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) x_{ij}^t \qquad \text{(ILP-layer)}$$

$$\text{s.t.} \quad x_{ij}^t + x_{jk}^t \geq x_{ik}^t \qquad \forall i,j,k \in V \qquad (9)$$

$$\sum_{i,j \in S} x_{ij}^t \geq 2 \qquad \forall S \subseteq V, |S| = t+1 \qquad (10)$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V \qquad (11)$$

$$x_{ii}^t = 0 \qquad \forall i \in V \qquad (12)$$

$$x_{ij}^t \in \{0,1\} \qquad \forall i,j \in V \qquad (13)$$

The following lemma provides a combinatorial characterization of feasible solutions to the layer-t problem.

Lemma 10 *Let* $G_t = (V, E_t)$ *be the graph as in Definition 9 corresponding to a solution* x_{ij}^t *to the layer-t problem ILP-layer. Then* G_t *is a disjoint union of cliques of size* $\leq t$. *Moreover this exactly characterizes all feasible solutions of ILP-layer.*

Proof We first note that $G_t = (V, E_t)$ must be a disjoint union of cliques since if $\{i,j\} \in E_t$ and $\{j,k\} \in E_t$ then $\{i,k\} \in E_t$ since $x_{ik}^t \leq x_{ij}^t + x_{jk}^t = 0$ due to constraint (9). Suppose there is a clique in G_t of size $> t$. Choose a subset S of this clique of size $t+1$. Then $\sum_{i,j \in S} x_{ij}^t = 0$ which violates constraint (10). Conversely, let E_t be a subset of edges such that $G_t = (V, E_t)$ is a disjoint union of cliques of size $\leq t$. Let $x_{ij}^t = 0$ if $\{i,j\} \in E_t$ and 1 otherwise. Clearly $x_{ij}^t = x_{ji}^t$ by definition. Suppose x_{ij}^t violates constraint (9), so that there is a pair $i, j, k \in V$ such that $x_{ik}^t = 1$ but $x_{ij}^t = x_{jk}^t = 0$. However this implies that G_t is not a disjoint union of cliques since $\{i,j\}, \{j,k\} \in E_t$ but $\{i,k\} \notin E_t$. Suppose x_{ij}^t violates constraint (10) for some set S of size $t+1$. Therefore for every $i, j \in S$, we have $x_{ij}^t = 0$ since $x_{ij}^t = x_{ji}^t$ for every $i, j \in V$ and so S must be a clique of size $t+1$ in G_t which is a contradiction. ■

By Lemma 10 the layer-t problem is to find a subset $\overline{E}_t \subseteq E(K_n)$ of minimum weight under κ, such that the complement graph $G_t = (V, E_t)$ is a disjoint union of cliques of size $\leq t$. Note that this implies that the number of components in the complement graph is $\geq \lceil n/t \rceil$.The converse however, is not necessarily true: when $t = n-1$ then the layer t-problem is the minimum (weighted) cut problem whose partitions may have size larger than 1. Our algorithmic approach is to solve an LP relaxation of ILP-ultrametric and then round the solution to obtain a feasible solution to ILP-ultrametric. The rounding however proceeds iteratively in a layer-wise manner and so we need to make sure that the rounded solution satisfies the inter-layer constraints (2) and (5). The following lemma gives a combinatorial characterization of solutions that satisfy these two constraints.

Lemma 11 *For every* $t \in [n-1]$, *let* x_{ij}^t *be feasible for the layer-t problem ILP-layer. Let* $G_t = (V, E_t)$ *be the graph as in Definition 9 corresponding to* x_{ij}^t, *so that by Lemma 10,* G_t *is a disjoint union of cliques* $K_1^t, \dots, K_{l_t}^t$ *each of size at most t. Then* x_{ij}^t *is feasible for ILP-ultrametric if and only if the following conditions hold.*

Nested cliques *For any* $s \leq t$ *every clique* K_p^s *for some* $p \in [l_s]$ *in* G_s *is a subclique of some clique* K_q^t *in* G_t *where* $q \in [l_t]$.

Realization *If* $\left|K_p^t\right| = s$ *for some* $s \leq t$, *then* G_s *contains* K_p^t *as a component clique, i.e.,* $K_q^s = K_p^t$ *for some* $q \in [l_s]$.

Proof Since x^t_{ij} is feasible for the layer-t problem ILP-layer it is feasible for ILP-ultrametric if and only if it satisfies constraints (2) and (5). The solution x^t_{ij} satisfies constraint (2) if and only if $E_t \subseteq E_{t+1}$ by definition and so Condition Nested cliques follows.

Let us now assume that x^t_{ij} is feasible for ILP-ultrametric, so that by the above argument Condition Nested cliques is satisfied. Note that every clique K^t_p in the clique decomposition of G_t corresponds to an equivalence class S_t under the relation $i \sim j$ iff $x^t_{ij} = 0$. Moreover, by Lemma 10 we have $|S_t| \leq t$. Constraint (5) implies that $x^{|S_t|}_{ij} = 0$ for every $i, j \in S_t$. In other words, if $|S_t| = s \leq t$, then $x^s_{ij} = 0$ for every $i, j \in S_t$ and so S_t is a subclique of some clique K^s_q in the clique decomposition of G_s. However by Condition Nested cliques, K^s_q must be a subclique of a clique $K^t_{p'}$ in the clique decomposition of G_t, since $s \leq t$. However, as $K^t_p \cap K^t_{p'} = S_t$ and the clique decomposition decomposes G_t into a disjoint union of cliques, it follows that $S_t \subseteq K^s_q \subseteq K^t_{p'} = K^t_p = S_t$ and so $K^s_q = K^t_p$. Therefore Condition Realization is satisfied.

Conversely, suppose that x^t_{ij} satisfies Conditions Nested cliques and Realization, so that by the argument in the paragraph above x^t_{ij} satisfies constraint (2). Let us assume for the sake of contradiction that for a set $S \subseteq V$ and a $t \in [n-1]$ constraint (5) is violated, i.e.,

$$\sum_{i,j \in S} x^{|S|}_{ij} > |S| \left(\sum_{i,j \in S} x^t_{ij} + \sum_{\substack{i \in S \\ j \notin S}} \left(1 - x^t_{ij}\right) \right).$$

Since $x^t_{ij} \in \{0,1\}$ it follows that $x^t_{ij} = 0$ for every $i, j \in S$ and $x^t_{ij} = 1$ for every $i \in S, j \notin S$ so that S is a clique in G_t. Note that $|S| < t$ since $\sum_{i,j \in S} x^{|S|}_{ij} > 0$. This contradicts Condition Realization however, since S is clearly not a clique in $G_{|S|}$. ■

The combinatorial interpretation of the individual layer-t problems allow us to simplify the formulation of ILP-ultrametric by replacing the constraints for sets of a specific size (constraint (4)) by a global constraint about all sets (constraint (14)). These global constraints are also referred to as *spreading constraints*.

Lemma 12 *We may replace constraint* (4) *of ILP-ultrametric by the following equivalent* spreading constraint*:*

$$\sum_{j \in S} x^t_{ij} \geq |S| - t \qquad \forall t \in [n-1], S \subseteq V, i \in S. \tag{14}$$

Proof Let x^t_{ij} be a feasible solution to ILP-ultrametric. Note that if $|S| \leq t$ then the constraints are redundant since $x^t_{ij} \in \{0,1\}$. Thus we may assume that $|S| > t$ and let i be any vertex in S. Let us suppose for the sake of a contradiction that $\sum_{j \in S} x^t_{ij} < |S| - t$. This implies that there is a t sized subset $S' \subseteq S \setminus \{i\}$ such that for every $j \in S'$ we have $x^t_{ij'} = 0$. In other words $\{i, j'\}$ is an edge in $G_t = (V, E_t)$ for every $j' \in S'$ and since G_t is a disjoint union of cliques (constraint (3)), this implies the existence of a clique of size $t+1$. Thus by Lemma 10, x^t_{ij} could not have been a feasible solution to ILP-ultrametric.

Conversely, suppose x^t_{ij} is feasible for the modified ILP where constraint (4) is replaced by constraint (14). Then again $G_t = (V, E_t)$ is a disjoint union of cliques since x^t_{ij} satisfies constraint (3). Assume for contradiction that constraint (4) is violated: there is a set S of size $t+1$ such that $\sum_{i,j \in S} x^t_{ij} < 2$. Note that this implies that $\sum_{i,j} x^t_{ij} = 0$ since $x^t_{ij} = x^t_{ji}$ for every $i, j \in V$ and $t \in [n-1]$. Fix any $i \in S$, then $\sum_{j \in S} x^t_{ij} < 1 = |S| - t$ since $x^t_{ij} = x^t_{ji}$ by constraint (6), a violation of constraint (14). Thus x^t_{ij} is feasible for ILP-ultrametric since it satisfies every other constraint by assumption. ■

4. Rounding an LP relaxation

In this section we consider the following natural LP relaxation for ILP-ultrametric. All missing proofs from this section can be found in Appendix C. We keep the variables x^t_{ij} for every $t \in [n-1]$ and $i, j \in V$ but relax the integrality constraint on the variables as well as drop constraint (5).

$$\min \quad \sum_{t=1}^{n-1} \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) x^t_{ij} \tag{LP-ultrametric}$$

$$\text{s.t.} \quad x^t_{ij} \geq x^{t+1}_{ij} \qquad \forall i, j \in V, t \in [n-2] \tag{15}$$

$$x^t_{ij} + x^t_{jk} \geq x^t_{ik} \qquad \forall i, j, k \in V, t \in [n-1] \tag{16}$$

$$\sum_{j \in S} x^t_{ij} \geq |S| - t \qquad \forall t \in [n-1], S \subseteq V, i \in S \tag{17}$$

$$x^t_{ij} = x^t_{ji} \qquad \forall i, j \in V, t \in [n-1] \tag{18}$$

$$x^t_{ii} = 0 \qquad \forall i, j \in V, t \in [n-1] \tag{19}$$

$$0 \leq x^t_{ij} \leq 1 \qquad \forall i, j \in V, t \in [n-1] \tag{20}$$

A feasible solution x^t_{ij} to LP-ultrametric induces a sequence $\{d_t\}_{t \in [n-1]}$ of distance metrics over V defined as $d_t(i,j) \coloneqq x^t_{ij}$. Constraint (17) enforces an additional structure on this metric: informally points in a "large enough" subset S should be spread apart according to the metric d_t. Metrics of type d_t are called *spreading metrics* and were first studied in (Even et al., 1999, 2000) in relation to graph partitioning problems. The following lemma gives a technical interpretation of spreading metrics (see, e.g., (Even et al., 1999, 2000; Krauthgamer et al., 2009)); we include a proof for completeness.

Lemma 13 *Let x^t_{ij} be feasible for LP-ultrametric and for a fixed $t \in [n-1]$, let d_t be the induced spreading metric. Let $i \in V$ be an arbitrary vertex and let $S \subseteq V$ be a set with $i \in S$ such that $|S| > (1+\varepsilon)t$ for some $\varepsilon > 0$. Then $\max_{j \in S} d_t(i,j) > \frac{\varepsilon}{1+\varepsilon}$.*

Proof For the sake of a contradiction suppose that for every $j \in S$ we have $d_t(i,j) = x^t_{ij} \leq \frac{\varepsilon}{1+\varepsilon}$. This implies that x^t_{ij} violates constraint (17) leading to a contradiction:

$$\sum_{j \in S} x^t_{ij} \leq \frac{\varepsilon}{1+\varepsilon} |S| < |S| - t,$$

where the last inequality follows from $|S| > (1+\varepsilon)t$. ■

Note that there are exponentially many inequalities for constraint (17); however the following lemma shows that we can still optimize over LP-ultrametric in polynomial time using the Ellipsoid method.

Lemma 14 *An optimal solution to LP-ultrametric can be computed in time polynomial in n and* $\log\left(\max_{i,j} \kappa(i,j)\right)$.

Proof We argue in the standard fashion via the application of the Ellipsoid method (see e.g., (Schrijver, 1998)). As such it suffices to verify that the encoding length of the numbers is small (which is indeed the case here) and that the constraints can be separated in polynomial time in the size of the input, i.e., in n and the logarithm of the absolute value of the largest coefficient. Since constraints of type (15), (16), (18), and (19) are polynomially many in n, we only need to check separation for constraints of type (17). Given a claimed solution x^t_{ij} we can check constraint (17) by iterating over all $t \in [n-1]$, vertices $i \in V$, and sizes m of the set S from $t+1$ to n. For a fixed t, i, and set size m sort the vertices in $V \setminus \{i\}$ in increasing order of distance from i (according to the metric d_t) and let $\overline{S}$ be the first m vertices in this ordering. If $\sum_{j \in \overline{S}} x^t_{ij} < m - t$ then

clearly x^t_{ij} is not feasible for LP-ultrametric, so we may assume that $\sum_{j\in\overline{S}} x^t_{ij} \geq m-t$. Moreover this is the only set to check: for any set $S \subseteq V$ containing i such that $|S| = m$, $\sum_{j\in S} x^t_{ij} \geq \sum_{j\in\overline{S}} x^t_{ij} \geq m-t$. Thus for a fixed $t \in [n-1]$, $i \in V$ and set size m, it suffices to check that x^t_{ij} satisfies constraint (17) for this subset $\overline{S}$. ■

From now on we will simply refer to a feasible solution to LP-ultrametric by the sequence of spreading metrics $\{d_t\}_{t\in[n-1]}$ it induces. The following definition introduces the notion of an open ball $\mathcal{B}_U(i,r,t)$ of radius r centered at $i \in V$ according to the metric d_t and restricted to the set $U \subseteq V$.

Definition 15 *Let $\{d_t \mid t \in [n-1]\}$ be the sequence of spreading metrics feasible for LP-ultrametric. Let $U \subseteq V$ be an arbitrary subset of V. For a vertex $i \in U$, $r \in \mathbb{R}$, and $t \in [n-1]$ we define the* open ball *$\mathcal{B}_U(i,r,t)$ of radius r centered at i as*

$$\mathcal{B}_U(i,r,t) \coloneqq \{j \in U \mid d_t(i,j) < r\} \subseteq U.$$

If $U = V$ then we denote $\mathcal{B}_U(i,r,t)$ simply by $\mathcal{B}(i,r,t)$.

Remark 16 *For every pair $i,j \in V$ we have $d_t(i,j) \geq d_{t+1}(i,j)$ by constraint* (15)*. Thus for any subset $U \subseteq V$, $i \in U$, $r \in \mathbb{R}$, and $t \in [n-2]$, it holds $\mathcal{B}_U(i,r,t) \subseteq \mathcal{B}_U(i,r,t+1)$.*

To round LP-ultrametric to get a feasible solution for ILP-ultrametric, we will use the technique of *sphere growing* which was introduced in (Leighton and Rao, 1988) to show an $O(\log n)$ approximation for the maximum multicommodity flow problem. Recall from Lemma 10 that a feasible solution to ILP-layer consists of a decomposition of the graph G_t into a set of disjoint cliques of size at most t. One way to obtain such a decomposition is to choose an arbitrary vertex, grow a ball around this vertex until the expansion of this ball is below a certain threshold, chop off this ball and declare it as a partition and then recurse on the remaining vertices. This is the main idea behind sphere growing, and the parameters are chosen depending on the constraints of the specific problem (see, e.g., (Garg et al., 1996; Even et al., 1999; Charikar et al., 2003) for a few representative applications of this technique). The first step is to associate to every ball $\mathcal{B}_U(i,r,t)$ a volume $\mathrm{vol}(\mathcal{B}_U(i,r,t))$ and a boundary $\partial\mathcal{B}_U(i,r,t)$ so that its expansion is defined. For any $t \in [n-1]$ and $U \subseteq V$ we denote by γ^U_t the value of the layer-t objective for solution d_t restricted to the set U, i.e.,

$$\gamma^U_t \coloneqq \sum_{\substack{i,j\in U\\ i<j}} \kappa(i,j) d_t(i,j).$$

When $U = V$ we refer to γ^U_t simply by γ_t. Since $\kappa : V \times V \to \mathbb{R}_{\geq 0}$, it follows that $\gamma^U_t \leq \gamma_t$ for any $U \subseteq V$. We are now ready to define the volume, boundary, and expansion of a ball $\mathcal{B}_U(i,r,t)$. We use the definition of (Even et al., 1999) modified for restrictions to arbitrary subsets $U \subseteq V$.

Definition 17 *(Even et al., 1999) Let U be an arbitrary subset of V. For a vertex $i \in U$, radius $r \in \mathbb{R}_{\geq 0}$, and $t \in [n-1]$, let $\mathcal{B}_U(i,r,t)$ be the ball of radius r as in Definition 15. Then we define its* volume *as*

$$\mathrm{vol}(\mathcal{B}_U(i,r,t)) \coloneqq \frac{\gamma^U_t}{n\log n} + \sum_{\substack{j,k\in\mathcal{B}_U(i,r,t)\\ j<k}} \kappa(j,k) d_t(j,k) + \sum_{\substack{j\in\mathcal{B}_U(i,r,t)\\ k\notin\mathcal{B}_U(i,r,t)\\ k\in U}} \kappa(j,k)\left(r - d_t(i,j)\right).$$

The boundary *of the ball $\partial\mathcal{B}_U(i,r,t)$ is the partial derivative of volume with respect to the radius:*

$$\partial\mathcal{B}_U(i,r,t) \coloneqq \frac{\partial\,\mathrm{vol}(\mathcal{B}_U(i,r,t))}{\partial r} = \sum_{\substack{j\in\mathcal{B}_U(i,r,t)\\ k\notin\mathcal{B}_U(i,r,t)\\ k\in U}} \kappa(j,k).$$

The expansion *$\phi(\mathcal{B}_U(i,r,t))$ of the ball $\mathcal{B}_U(i,r,t)$ is defined as the ratio of its boundary to its volume, i.e.,*

$$\phi\left(\mathcal{B}_U(i,r,t)\right) \coloneqq \frac{\partial\mathcal{B}_U(i,r,t)}{\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)}.$$

The following lemma shows that the volume of a ball $\mathcal{B}_U(i,r,t)$ is differentiable with respect to r in the interval $(0,\Delta]$ except at finitely many points (see e.g., (Even et al., 1999)).

Lemma 18 *Let $\mathcal{B}_U(i,r,t)$ be the ball corresponding to a set $U \subseteq V$, vertex $i \in U$, radius $r \in \mathbb{R}$ and $t \in [n-1]$. Then $\mathrm{vol}(\mathcal{B}_U(i,r,t))$ is differentiable with respect to r in the interval $(0,\Delta]$ except at finitely many points.*

Input: Data set V, $\{d_t\}_{t\in[n-1]} : V \times V$, $\varepsilon > 0$, $\kappa : V \times V \to \mathbb{R}_{\geq 0}$
Output: A solution set of the form $\left\{x^t_{ij} \in \{0,1\} \mid t \in \left[\left\lfloor\frac{n-1}{1+\varepsilon}\right\rfloor\right], i,j \in V\right\}$

1 $m_\varepsilon \leftarrow \left\lfloor\frac{n-1}{1+\varepsilon}\right\rfloor$
2 $t \leftarrow m_\varepsilon$
3 $\mathcal{C}_{t+1} \leftarrow \{V\}$
4 $\Delta \leftarrow \frac{\varepsilon}{1+\varepsilon}$
5 **while** $t \geq 1$ **do**
6 $\mathcal{C}_t \leftarrow \emptyset$
7 **for** $U \in \mathcal{C}_{t+1}$ **do**
8 **if** $|U| \leq (1+\varepsilon)t$ **then**
9 $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \{U\}$
10 Go to line 7
11 **end**
12 **while** $U \neq \emptyset$ **do**
13 Let i be arbitrary in U
14 Let $r \in (0,\Delta]$ be s.t. $\phi\left(\mathcal{B}_U(i,r,t)\right) \leq \frac{1}{\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)$
15 $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \{\mathcal{B}_U(i,r,t)\}$
16 $U \leftarrow U \setminus \mathcal{B}_U(i,r,t)$
17 **end**
18 **end**
19 $x^t_{ij} = 1$ if $i \in U_1 \in \mathcal{C}_t$, $j \in U_2 \in C_t$ and $U_1 \neq U_2$, else $x^t_{ij} = 0$
20 $t \leftarrow t-1$
21 **end**
22 **return** $\left\{x^t_{ij} \mid t \in [m_\varepsilon], i,j \in V\right\}$

Algorithm 2: Iterative rounding algorithm to find a low cost ultrametric

The following theorem establishes that the rounding procedure of Algorithm 2 ensures that the cliques in $\mathcal{C}_t$ are "small" and that the cost of the edges removed to form them, is not too high. It also shows that Algorithm 2 can be implemented to run in time polynomial in n.

Theorem 19 *Let $m_\varepsilon \coloneqq \left\lfloor\frac{n-1}{1+\varepsilon}\right\rfloor$ as in Algorithm 2 and let $\left\{x^t_{ij} \mid t \in [m_\varepsilon], i,j \in V\right\}$ be the output of Algorithm 2 run on a feasible solution $\{d_t\}_{t\in[n-1]}$ of LP-ultrametric and any choice of $\varepsilon \in (0,1)$. For any*

$t \in [m_\varepsilon]$, *we have that* x^t_{ij} *is feasible for the layer-*$\lfloor(1+\varepsilon)\,t\rfloor$ *problem ILP-layer and there is a constant* $c(\varepsilon) > 0$ *depending only on* ε *such that*

$$\sum_{\{i,j\}\in E(K_n)} \kappa(i,j)x^t_{ij} \leq c(\varepsilon)(\log n)\gamma_t.$$

Moreover, Algorithm 2 can be implemented to run in time polynomial in n.

Remark 20 *A discrete version of the volumetric argument for region growing can be found in (Gupta, 2005).*

We are now ready to prove the main theorem showing that we can obtain a low cost non-trivial ultrametric from Algorithm 2.

Theorem 21 *Let* $\left\{x^t_{ij} \mid t \in [m_\varepsilon]\,, i,j \in V\right\}$ *be the output of Algorithm 2 on an optimal solution* $\{d_t\}_{t\in[n-1]}$ *of LP-ultrametric for any choice of* $\varepsilon \in (0,1)$. *Define the sequence* $\left\{y^t_{ij}\right\}$ *for every* $t \in [n-1]$ *and* $i,j \in V$ *as*

$$y^t_{ij} := \begin{cases} x^{\lfloor t/(1+\varepsilon)\rfloor}_{ij} & \text{if } t > 1+\varepsilon \\ 1 & \text{if } t \leq 1+\varepsilon. \end{cases}$$

Then y^t_{ij} *is feasible for ILP-ultrametric and satisfies*

$$\sum_{t=1}^{n-1}\sum_{\{i,j\}\in E(K_n)} \kappa(i,j)y^t_{ij} \leq (2c(\varepsilon)\log n)\,\mathrm{OPT}$$

where OPT *is the optimal solution to ILP-ultrametric and* $c(\varepsilon)$ *is the constant in the statement of Theorem 19.*

Proof Note that by Theorem 19 for every $t \in [m_\varepsilon]$, x^t_{ij} is feasible for the layer-$\lfloor(1+\varepsilon)t\rfloor$ problem ILP-layer and that there is a constant $c(\varepsilon) > 0$ such that for every $t \in [m_\varepsilon]$, we have $\sum_{\{i,j\}\in E(K_n)} \kappa(i,j)x^t_{ij} \leq (c(\varepsilon)\log n)\,\gamma_t$.
Let y^t_{ij} be as in the statement of the theorem. The graph $G_t = (V, E_t)$ as in Definition 9 corresponding to y^t_{ij} for $t \leq 1+\varepsilon$ consists of isolated vertices, i.e., cliques of size 1: By definition y^t_{ij} is feasible for the layer-t problem ILP-layer. The collection $\mathcal{C}_1$ corresponding to x^1_{ij} consists of cliques of size at most $1+\varepsilon$, however since $0 < \varepsilon < 1$ it follows that the cliques in $\mathcal{C}_1$ are isolated vertices and so $x^1_{ij} = 1$ for every $\{i,j\} \in E(K_n)$. Thus $\sum_{i,j}\kappa(i,j)y^t_{ij} = \sum_{i,j}\kappa(i,j)x^1_{ij} \leq (c(\varepsilon)\log n)\,\gamma_1$ for $t \leq 1+\varepsilon$ by Theorem 19. Moreover for every $t > 1+\varepsilon$, we have $\sum_{i,j}\kappa(i,j)y^t_{ij} \leq (c(\varepsilon)\log n)\gamma_{\lfloor t/(1+\varepsilon)\rfloor}$ again by Theorem 19. We claim that y^t_{ij} is feasible for ILP-ultrametric. The solution y^t_{ij} corresponds to the collection $\mathcal{C}_{\lfloor\frac{t}{1+\varepsilon}\rfloor}$ for $t > 1+\varepsilon$ or to the collection $\mathcal{C}_1$ for $t \leq 1+\varepsilon$ from Algorithm 2. For any $t < m_\varepsilon$, every ball $\mathcal{B}_U(i,r,t) \in \mathcal{C}_t$ comes from the refinement of a ball $\mathcal{B}_{U'}(i',r',t')$ for some $i' \in V$, $r' \geq r$, $t' \geq t$ and $U' \supseteq U$. Thus y^t_{ij} satisfies Condition Nested cliques of Lemma 11. On the other hand line 8 ensures that if $|\mathcal{B}_U(i,r,t)| = \lfloor(1+\varepsilon)s\rfloor$ for some $U \subseteq V$ and $s < t$ then $\mathcal{B}_U(i,r,t)$ also appears as a ball in $\mathcal{C}_s$. Therefore y^t_{ij} also satisfies Condition Realization of Lemma 11 and so is feasible for ILP-ultrametric. The cost of y^t_{ij} is at most

$$\begin{aligned}\sum_{t=1}^{n-1}\sum_{\{i,j\}\in E(K_n)} \kappa(i,j)y^t_{ij} &\leq (c(\varepsilon)\log n)\left(\gamma_1 + \sum_{t=2}^{n-1}\gamma_{\lfloor t/(1+\varepsilon)\rfloor}\right) \\ &\leq 2c(\varepsilon)\log n\sum_{t=1}^{n-1}\gamma_t \\ &\leq 2c(\varepsilon)\log n\,\mathrm{OPT},\end{aligned}$$

where we use the fact that $\sum_{t=1}^{n-1}\gamma_t = \mathrm{OPT}(LP) \leq \mathrm{OPT}$ since LP-ultrametric is a relaxation of ILP-ultrametric. ■

Theorem 21 implies the following corollary where we put everything together to obtain a hierarchical clustering of V in time polynomial in n with $|V| = n$. Let $\mathcal{T}$ denote the set of all possible hierarchical clusterings of V.

Input: Data set V of n points, similarity function $\kappa : V \times V \to \mathbb{R}_{\geq 0}$
Output: Hierarchical clustering of V
1 Solve LP-ultrametric to obtain optimal sequence of spreading metrics $\{d_t \mid d_t : V \times V \to [0,1]\}$
2 Fix a choice of $\varepsilon \in (0,1)$
3 $m_\varepsilon \leftarrow \lfloor\frac{n-1}{1+\varepsilon}\rfloor$
4 Let $\left\{x^t_{ij} \mid t \in [m_\varepsilon]\right\}$ be the output of Algorithm 2 on $V, \kappa, \{d_t\}_{t\in[n-1]}$
5 Let $y^t_{ij} := \begin{cases} x^{\lfloor t/(1+\varepsilon)\rfloor}_{ij} & \text{if } t > 1+\varepsilon \\ 1 & \text{if } t \leq 1+\varepsilon \end{cases}$ for every $t \in [n-1], i,j \in E(K_n)$
6 $d(i,j) \leftarrow \sum_{t=1}^{n-1} y^t_{ij}$ for every $i,j \in E(K_n)$
7 $d(i,i) \leftarrow 0$ for every $i \in V$
8 Let r, T be the output of Algorithm 1 on V, d
9 **return** r, T

Algorithm 3: Hierarchical clustering of V for cost function (1)

Corollary 22 *Given a data set* V *of* n *points and a similarity function* $\kappa : V \times V \to \mathbb{R}_{\geq 0}$, *Algorithm 3 returns a hierarchical clustering* T *of* V *satisfying*

$$\mathrm{cost}(T) \leq O(\log n)\min_{T'\in\mathcal{T}}\mathrm{cost}(T').$$

Moreover Algorithm 3 runs in time polynomial in n *and* $\log\left(\max_{i,j\in V}\kappa(i,j)\right)$.

Proof Let $\widehat{T}$ be the optimal hierarchical clustering according to cost function (1). By Corollary 8 and Theorem 21 we can find a hierarchical clustering T satisfying

$$\sum_{\{i,j\}\in E(K_n)}\kappa(i,j)(|\mathrm{leaves}(T[\mathrm{lca}(i,j)])| - 1) \leq O(\log n)\left(\sum_{\{i,j\}\in E(K_n)}\kappa(i,j)\left(\left|\mathrm{leaves}(\widehat{T}[\mathrm{lca}(i,j)])\right| - 1\right)\right).$$

Let $K := \sum_{\{i,j\}\in E(K_n)}\kappa(i,j)$. Then it follows from the above expression that $\mathrm{cost}(T) \leq O(\log n)\,\mathrm{cost}(\widehat{T}) - O(\log n)K + K \leq O(\log n)\,\mathrm{cost}(\widehat{T})$.
We can find an optimal solution to LP-ultrametric due to Lemma 14 using the Ellipsoid algorithm in time polynomial in n and $\log\left(\max_{i,j\in V}\kappa(i,j)\right)$. Algorithm 2 runs in time polynomial in n due to Theorem 19. Finally, Algorithm 1 runs in time $O\left(n^3\right)$ due to Lemma 7. ■

5. Generalized Cost Function

In this section we study the following natural generalization of cost function (1) also introduced by (Dasgupta, 2016) where the distance between the two points is scaled by a function $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, i.e.,

$$\mathrm{cost}_f(T) := \sum_{\{i,j\}\in E(K_n)}\kappa(i,j)f\left(|\mathrm{leaves}\,T[\mathrm{lca}(i,j)]|\right). \tag{21}$$

In order that cost function (21) makes sense, f should be strictly increasing and satisfy $f(0) = 0$. Possible choices for f could be $\{x^2, e^x - 1, \log(1+x)\}$. The top-down heuristic in (Dasgupta, 2016) finds the optimal hierarchical clustering up to an approximation factor of $c_n \log n$ with c_n being defined as

$$c_n \coloneqq 3\alpha_n \max_{1 \le n' \le n} \frac{f(n')}{f\left(\lceil n'/3 \rceil\right)}$$

and where α_n is the approximation factor from the algorithm used for solving uniform Sparsest Cut problem. A naive approach to solving this problem using the ideas of Algorithm 2 would be to replace the objective function of ILP-ultrametric by

$$\sum_{\{i,j\} \in E(K_n)} \kappa(i,j) f\left(\sum_{t=1}^{n-1} x_{ij}^t\right).$$

This makes the corresponding analogue of LP-ultrametric non-linear however, and for a general κ and f it is not clear how to compute an optimum solution in polynomial time. One possible solution is to assume that f is convex and use the Frank-Wolfe algorithm to compute an optimum solution. That still leaves the problem of how to relate $f\left(\sum_{t=1}^{n-1} x_{ij}^t\right)$ to $\sum_{t=1}^{n-1} f\left(x_{ij}^t\right)$ as one would have to do to get a corresponding version of Theorem 21. The following simple observation provides an alternate way of tackling this problem.

Observation 23 *Let $d : V \times V \to \mathbb{R}$ be an ultrametric and $f : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$ be a strictly increasing function such that $f(0) = 0$. Define the function $f(d) : V \times V \to \mathbb{R}$ as $f(d)(i,j) \coloneqq f(d(i,j))$. Then $f(d)$ is also an ultrametric on V.*

Therefore by Corollary 8 to find a minimum cost hierarchical clustering T of V according to the cost function (21), it suffices to minimize $\langle \kappa, d \rangle$ where d is the f-image of a non-trivial ultrametric as in Definition 2. The following lemma lays down the analogue of Conditions 1 and 2 from Definition 2 that the f-image of a non-trivial ultrametric satisfies. As in Definition 2, we will use the notation $x \sim y$ to denote an equivalence relation between x and y.

Lemma 24 *Let $f : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$ be a strictly increasing function satisfying $f(0) = 0$. An ultrametric d on V is the f-image of a non-trivial ultrametric on V iff*

1. *for every non-empty set $S \subseteq V$, there is a pair of points $i, j \in S$ such that $d(i,j) \ge f(|S| - 1)$,*
2. *for any t, if S_t is an equivalence class of V under the relation $i \sim j$ iff $d(i,j) \le t$, then $\max_{i,j \in S_t} d(i,j) \le f(|S_t| - 1)$.*

Proof If d is the f-image of a non-trivial ultrametric d' on V then clearly d satisfies Conditions 1 and 2. Conversely, let d be an ultrametric on V satisfying Conditions 1 and 2. Note that f is strictly increasing and V is a finite set and thus f^{-1} exists and is strictly increasing as well, with $f^{-1}(0) = 0$. Define d' as $d'(i,j) \coloneqq f^{-1}(d(i,j))$ for every $i, j \in V$. By Observation 23 d' is an ultrametric on V satisfying Conditions 1 and 2 of Definition 2 and so d' is a non-trivial ultrametric on V. ■

Lemma 24 allows us to write the analogue of ILP-ultrametric for finding the minimum cost ultrametric that is the f-image of a non-trivial ultrametric on V. Note that by Lemma 4 the range of such an ultrametric is the set $\{f(0), f(1), \ldots, f(n-1)\}$. We have the binary variables x_{ij}^t for every distinct pair $i, j \in V$ and

$t \in [n-1]$, where $x_{ij}^t = 1$ if $d(i,j) \ge f(t)$ and $x_{ij}^t = 0$ if $d(i,j) < f(t)$.

$$\min \quad \sum_{t=1}^{n-1} \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) \left(f(t) - f(t-1)\right) x_{ij}^t \qquad \text{(f-ILP-ultrametric)}$$

$$\text{s.t.} \quad x_{ij}^t \ge x_{ij}^{t+1} \qquad \forall i,j \in V, t \in [n-2] \tag{22}$$

$$x_{ij}^t + x_{jk}^t \ge x_{ik}^t \qquad \forall i,j,k \in V, t \in [n-1] \tag{23}$$

$$\sum_{i,j \in S} x_{ij}^t \ge 2 \qquad \forall t \in [n-1], S \subseteq V, |S| = t+1 \tag{24}$$

$$\sum_{i,j \in S} x_{ij}^{|S|} \le |S|^2 \left(\sum_{i,j \in S} x_{ij}^t + \sum_{\substack{i \in S \\ j \notin S}} \left(1 - x_{ij}^t\right) \right) \quad \forall t \in [n-1], S \subseteq V \tag{25}$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V, t \in [n-1] \tag{26}$$

$$x_{ii}^t = 0 \qquad \forall i \in V, t \in [n-1] \tag{27}$$

$$x_{ij}^t \in \{0,1\} \qquad \forall i,j \in V, t \in [n-1] \tag{28}$$

If x_{ij}^t is a feasible solution to f-ILP-ultrametric then the ultrametric represented by it is defined as

$$d(i,j) \coloneqq \sum_{t=1}^{n-1} (f(t) - f(t-1)) x_{ij}^t.$$

Constraint (24) ensures that d satisfies Condition 1 of Lemma 24, since for every $S \subseteq V$ of size $t+1$ we have a pair $i, j \in S$ such that $d(i,j) \ge f(t)$. Similarly constraint (25) ensures that d satisfies Condition 2 of Lemma 24 since it is active if and only if S is an equivalence class of V under the relation $i \sim j$ iff $d(i,j) < f(t)$. In this case Condition 2 of Lemma 24 requires $\max_{i,j \in S} d(i,j) \le f(|S| - 1)$ or in other words $x_{ij}^{|S|} = 0$ for every $i, j \in S$.

Similar to ILP-layer we define an analogous *layer-t problem* where we fix a choice of $t \in [n-1]$ and drop the constraints that relate the different layers to each other.

$$\min \quad \sum_{\{i,j\} \in E(K_n)} \kappa(i,j) \left(f(t) - f(t-1)\right) x_{ij}^t \qquad \text{(f-ILP-layer)}$$

$$\text{s.t.} \quad x_{ij}^t + x_{jk}^t \ge x_{ik}^t \qquad \forall i,j,k \in V \tag{29}$$

$$\sum_{i,j \in S} x_{ij}^t \ge 2 \qquad \forall S \subseteq V, |S| = t+1 \tag{30}$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V \tag{31}$$

$$x_{ii}^t = 0 \qquad \forall i \in V \tag{32}$$

$$x_{ij}^t \in \{0,1\} \qquad \forall i,j \in V \tag{33}$$

Note that f-ILP-ultrametric and f-ILP-layer differ from ILP-ultrametric and ILP-layer respectively only in the objective function. Therefore Lemmas 10 and 11 also give a combinatorial characterization of the set of feasible solutions to f-ILP-layer and f-ILP-ultrametric respectively. Similarly, by Lemma 12 we may replace constraint (24) by the following equivalent constraint over all subsets of V

$$\sum_{j \in S} x_{ij}^t \ge |S| - t \qquad \forall t \in [n-1], S \subseteq V, i \in S.$$

This provides the analogue of LP-ultrametric in which we drop constraint (25) and enforce it in the rounding procedure.

$$\min \quad \sum_{t=1}^{n-1} \sum_{\{i,j\}\in E(K_n)} \kappa(i,j)\left(f(t)-f(t-1)\right) x_{ij}^t \qquad \text{(f-LP-ultrametric)}$$

$$\text{s.t.} \quad x_{ij}^t \geq x_{ij}^{t+1} \qquad \forall i,j \in V, t \in [n-2] \tag{34}$$

$$x_{ij}^t + x_{jk}^t \geq x_{ik}^t \qquad \forall i,j,k \in V, t \in [n-1] \tag{35}$$

$$\sum_{j\in S} x_{ij}^t \geq |S| - t \qquad \forall t \in [n-1], S \subseteq V, i \in S \tag{36}$$

$$x_{ij}^t = x_{ji}^t \qquad \forall i,j \in V, t \in [n-1] \tag{37}$$

$$x_{ii}^t = 0 \qquad \forall i \in V, t \in [n-1] \tag{38}$$

$$0 \leq x_{ij}^t \leq 1 \qquad \forall i,j \in V, t \in [n-1] \tag{39}$$

Since f-LP-ultrametric differs from LP-ultrametric only in the objective function, it follows from Lemma 14 that an optimum solution to f-LP-ultrametric can be computed in time polynomial in n. As before, a feasible solution x_{ij}^t of f-LP-ultrametric induces a sequence $\{d_t\}_{t\in[n-1]}$ of spreading metrics on V defined as $d_t(i,j) \coloneqq x_{ij}^t$. Note that in contrast to the ultrametric d, the spreading metrics $\{d_t\}_{t\in[n-1]}$ are independent of the function f.

Let $\mathcal{B}_U(i,r,t)$ be a ball of radius r centered at $i \in U$ for some set $U \subseteq V$ as in Definition 15. For a subset $U \subseteq V$, let γ_t^U be defined as before to be the value of the layer-t objective corresponding to a solution d_t of f-LP-ultrametric restricted to U, i.e.,

$$\gamma_t^U \coloneqq \sum_{\substack{i,j\in U \\ i<j}} \left(f(t)-f(t-1)\right)\kappa(i,j)d_t(i,j).$$

As before, we denote γ_t^V by γ_t. We will associate a volume $\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)$ and a boundary $\partial\mathcal{B}_U(i,r,t)$ to the ball $\mathcal{B}_U(i,r,t)$ as in Section 4.

Definition 25 *Let U be an arbitrary subset of V. For a vertex $i \in U$, radius $r \in \mathbb{R}_{\geq 0}$, and $t \in [n-1]$, let $\mathcal{B}_U(i,r,t)$ be the ball of radius r as in Definition 15. Then we define its* volume *as*

$$\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right) \coloneqq \frac{\gamma_t^U}{n\log n} + \left(f(t)-f(t-1)\right)\left(\sum_{\substack{j,k\in\mathcal{B}_U(i,r,t)\\ j<k}} \kappa(j,k)d_t(j,k) + \sum_{\substack{j\in\mathcal{B}_U(i,r,t)\\ k\notin\mathcal{B}_U(i,r,t)\\ k\in U}} \kappa(j,k)\left(r-d_t(i,j)\right)\right).$$

The boundary *of the ball $\partial\mathcal{B}_U(i,r,t)$ is the partial derivative of volume with respect to the radius:*

$$\partial\mathcal{B}_U(i,r,t) \coloneqq \left(f(t)-f(t-1)\right)\left(\frac{\partial\,\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)}{\partial r}\right) = \left(f(t)-f(t-1)\right)\left(\sum_{\substack{j\in\mathcal{B}_U(i,r,t)\\ k\notin\mathcal{B}_U(i,r,t)\\ k\in U}} \kappa(j,k)\right).$$

The expansion *$\phi\left(\mathcal{B}_U(i,r,t)\right)$ of the ball $\mathcal{B}_U(i,r,t)$ is defined as the ratio of its boundary to its volume, i.e.,*

$$\phi\left(\mathcal{B}_U(i,r,t)\right) \coloneqq \frac{\partial\mathcal{B}_U(i,r,t)}{\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)}.$$

Note that the expansion $\phi\left(\mathcal{B}_U(i,r,t)\right)$ of Definition 25 is the same as in Definition 17 since the $\left(f(t)-f(t-1)\right)$ term cancels out. Thus one could run Algorithm 2 with the same notion of volume as in Definition 17, however in that case the analogous versions of Theorems 19 and 21 do not follow as naturally. The following is then a simple corollary of Theorem 19.

Corollary 26 *Let $m_\varepsilon \coloneqq \left\lfloor \frac{n-1}{1+\varepsilon} \right\rfloor$ as in Algorithm 2. Let $\left\{x_{ij}^t \mid t \in [n-1], i,j \in V\right\}$ be the output of Algorithm 2 using the notion of volume, boundary and expansion as in Definition 25, on a feasible solution to f-LP-ultrametric and any choice of $\varepsilon \in (0,1)$. For any $t \in [m_\varepsilon]$, we have that x_{ij}^t is feasible for the layer-$\lfloor(1+\varepsilon)t\rfloor$ problem f-ILP-layer and there is a constant $c(\varepsilon) > 0$ depending only on ε such that*

$$\sum_{\{i,j\}\in E(K_n)} \kappa(i,j)\left(f(t)-f(t-1)\right)x_{ij}^t \leq \left(c(\varepsilon)\log n\right)\gamma_t.$$

Corollary 26 allows us to prove the analogue of Theorem 21, i.e., we can use Algorithm 2 to get an ultrametric that is an f-image of a non-trivial ultrametric and whose cost is at most $O(\log n)$ times the cost of an optimal hierarchical clustering according to cost function (21).

Theorem 27 *Let $\{x_{ij}^t \mid t \in [m_\varepsilon], i,j \in V\}$ be the output of Algorithm 2 using the notion of volume, boundary, and expansion as in Definition 25 on an optimal solution $\{d_t\}_{t\in[n-1]}$ of f-LP-ultrametric for any choice of $\varepsilon \in (0,1)$. Define the sequence $\left\{y_{ij}^t\right\}$ for every $t \in [n-1]$ and $i,j \in V$ as*

$$y_{ij}^t \coloneqq \begin{cases} x_{ij}^{\lfloor t/(1+\varepsilon)\rfloor} & \text{if } t > 1+\varepsilon \\ 1 & \text{if } t \leq 1+\varepsilon. \end{cases}$$

Then y_{ij}^t is feasible for f-ILP-ultrametric and there is a constant $c(\varepsilon) > 0$ such that

$$\sum_{t=1}^{n-1} \sum_{\{i,j\}\in E(K_n)} \kappa(i,j)\left(f(t)-f(t-1)\right)y_{ij}^t \leq \left(c(\varepsilon)\log n\right)\mathrm{OPT}$$

where OPT *is the optimal solution to f-ILP-ultrametric.*

Proof Immediate from Corollary 26 and Theorem 21. ■

Finally we put everything together to obtain the corresponding Algorithm 4 that outputs a hierarchical clustering of V of cost at most $O(\log n)$ times the optimal clustering according to cost function (21).

Corollary 28 *Given a data set V of n points and a similarity function $\kappa : V \times V \to \mathbb{R}$, Algorithm 4 returns a hierarchical clustering T of V satisfying*

$$\mathrm{cost}_f(T) \leq O\left(c_n + \log n\right)\min_{T'\in\mathcal{T}} \mathrm{cost}_f(T'),$$

where $c_n \coloneqq \max_{n'\in[n]} f(n') - f(n'-1)$. Moreover Algorithm 4 runs in time polynomial in n, $\log f(n)$ and $\log\left(\max_{i,j\in V}\kappa(i,j)\right)$.

Proof Let $\widehat{T}$ be an optimal hierarchical clustering according to cost function (21). By Corollary 8, Lemma 24 and Theorem 27 it follows that we can find a hierarchical clustering T satisfying

$$\sum_{\{i,j\}\in E(K_n)} \kappa(i,j) f\left(|\mathrm{leaves}(T[\mathrm{lca}(i,j)]|-1\right) \leq O(\log n)\left(\sum_{\{i,j\}\in E(K_n)} \kappa(i,j) f\left(\left|\mathrm{leaves}(\widehat{T}[\mathrm{lca}(i,j)]\right|-1\right)\right).$$

Recall that $\text{cost}_f(T) := \sum_{\{i,j\}\in E(K_n)} \kappa(i,j) f(|\text{leaves}(T[\text{lca}(i,j)]|)$. Let $K := \sum_{\{i,j\}\in E(K_n)} \kappa(i,j)$. Note that for any hierarchical clustering T' we have $K \leq \text{cost}_f(T')$ since f is an increasing function. From the above expression we infer that

$$\text{cost}_f(T) - c_n K \leq \sum_{\{i,j\}\in E(K_n)} \kappa(i,j) f(|\text{leaves}(T[\text{lca}(i,j)]| - 1) \leq O(\log n)\, \text{cost}_f(\widehat{T}),$$

and so $\text{cost}_f(T) \leq O(\log n)\, \text{cost}_f(\widehat{T}) + c_n K \leq O(c_n + \log n)\, \text{cost}_f(\widehat{T})$. We can find an optimal solution to f-LP-ultrametric due to Lemma 14 using the Ellipsoid algorithm in time polynomial in n, $\log f(n)$, and $\log\left(\max_{i,j\in V} \kappa(i,j)\right)$. Note the additional $\log f(n)$ in the running time since now we need to binary search over the interval $\left[0, \max_{i,j\in V} \kappa(i,j) \cdot f(n) \cdot n\right]$. Algorithm 2 runs in time polynomial in n due to Theorem 19. Finally, Algorithm 1 runs in time $O\left(n^3\right)$ due to Lemma 7. ■

Input: Data set V of n points, similarity function $\kappa : V \times V \to \mathbb{R}_{\geq 0}$, $f : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ strictly increasing with $f(0) = 0$
Output: Hierarchical clustering of V

Solve f-LP-ultrametric to obtain optimal sequence of spreading metrics $\{d_t \mid d_t : V \times V \to [0,1]\}$
Fix a choice of $\varepsilon \in (0,1)$
$m_\varepsilon \leftarrow \left\lfloor \frac{n-1}{1+\varepsilon} \right\rfloor$
Let $\left\{x^t_{ij} \mid t \in [m_\varepsilon]\right\}$ be the output of Algorithm 2 on $V, \kappa, \{d_t\}_{t\in[n-1]}$
Let $y^t_{ij} := \begin{cases} x^{\lfloor t/(1+\varepsilon)\rfloor}_{ij} & \text{if } t > 1+\varepsilon \\ 1 & \text{if } t \leq 1+\varepsilon \end{cases}$ for every $t \in [n-1], i,j \in E(K_n)$
$d(i,j) \leftarrow \sum_{t=1}^{n-1} (f(t) - f(t-1))\, y^t_{ij}$ for every $i,j \in E(K_n)$
$d(i,i) \leftarrow 0$ for every $i \in V$
Let r, T be the output of Algorithm 1 on $V, f^{-1}(d)$
return r, T

Algorithm 4: Hierarchical clustering of V for cost function (21)

6. Experiments

Finally, we describe the experiments we performed. For small data sets ILP-ultrametric and f-ILP-ultrametric describe integer programming formulations that allow us to compute the exact optimal hierarchical clustering for cost functions (1) and (21) respectively. We implement f-ILP-ultrametric where one can plug in any strictly increasing function f satisfying $f(0) = 0$. In particular, setting $f(x) = x$ gives us ILP-ultrametric. We use the Mixed Integer Programming (MIP) solver Gurobi 6.5 (Gurobi Optimization, 2015). Similarly, we also implement Algorithms 1, 2, and 4 using Gurobi as our LP solver. Note that Algorithm 4 needs to fix a parameter choice $\varepsilon \in (0,1)$. In Sections 4 and 5 we did not discuss the effect of the choice of the parameter ε in detail. In particular, we need to choose an ε small enough such that for every $U \subseteq V$ encountered in Algorithm 2, $\text{vol}\left(\mathcal{B}_U(i,\Delta,t)\right)$ is of the same sign as $\text{vol}\left(\mathcal{B}_U(i,0,t)\right)$ for every $t \in [n-1]$, so that $\log\left(\frac{\text{vol}(\mathcal{B}_U(i,\Delta,t))}{\text{vol}(\mathcal{B}_U(i,0,t))}\right)$ is defined. In our experiments we start with a particular value of ε (say 0.5) and halve it till the volumes have the same sign. For the sake of exposition, we limit ourselves to the following choices for the function f

$$\left\{x, x^2, \log(1+x), e^x - 1\right\}.$$

By Lemma 14 we can optimize over f-LP-ultrametric in time polynomial in n using the Ellipsoid method. In practice however, we use the *dual simplex* method where we separate triangle inequality constraints (35) and spreading constraints (36) to obtain fast computations. For the similarity function $\kappa : V \times V \to \mathbb{R}$ we limit ourselves to using *cosine similarity* and the *Gaussian kernel* with $\sigma = 1$. They are defined formally below.

Definition 29 (Cosine similarity) *Given a data set $V \in \mathbb{R}^m$ for some $m \geq 0$, the cosine similarity κ_{cos} is defined as $\kappa_{cos}(x,y) := \frac{\langle x,y\rangle}{\|x\|\,\|y\|}$, for any pair of points $x, y \in V$.*

Since the LP rounding Algorithm 2 assumes that $\kappa \geq 0$ in practice we implement $1 + \kappa_{cos}$ rather than κ_{cos}.

Definition 30 (Gaussian kernel) *Given a data set $V \in \mathbb{R}^m$ for some $m \geq 0$, the Gaussian kernel κ_{gauss} with standard deviation σ is defined as $\kappa_{gauss}(x,y) := \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$, for any pair of points $x, y \in V$.*

The main aim of our experiments was to answer the following two questions.

1. How good is the hierarchal clustering obtained from Algorithm 4 as opposed to the true optimal output by f-ILP-ultrametric?
2. How good does Algorithm 4 perform compared to other hierarchical clustering methods?

For the first question, we are restricted to working with small data sets since computing an optimum solution to f-ILP-ultrametric is expensive. In this case we consider synthetic data sets of small size and samples of some data sets from the UCI database (Lichman, 2013). The synthetic data sets we consider are mixtures of Gaussians in various small dimensional spaces. Figure 2 shows a comparison of the cost of the hierarchy (according to cost function (21)) returned by solving f-ILP-ultrametric and by Algorithm 4 for various forms of f when the similarity function is κ_{cos} and κ_{gauss}. Note that we normalize the cost of the tree returned by f-ILP-ultrametric and Algorithm 4 by the cost of the trivial clustering r, T^* where T^* is the star graph with V as its leaves and r as the internal node. In other words $d_{T^*}(i,j) = n-1$ for every distinct pair $i,j \in V$ and so the normalized cost of any tree lies in the interval $(0,1]$.
For the study of the second question, we consider some of the popular algorithms for hierarchical clustering such as *single linkage*, *average linkage*, *complete linkage*, and *Ward's method* (Ward Jr, 1963). To get a numerical handle on how good a hierarchical clustering T of V is, we prune the tree to get the *best* k flat clusters and measure its error relative to the target clustering. We use the following notion of error also known as *Classification Error* that is standard in the literature for hierarchical clustering (see, e.g., (Meilă and Heckerman, 2001)). Note that we may think of a flat k-clustering of the data V as a function h mapping elements of V to a label set $\mathcal{L} := \{1, \ldots, k\}$. Let S_k denote the group of permutations on k letters.

Definition 31 (Classification Error) *Given a proposed clustering $h : V \to \mathcal{L}$ its* classification error *relative to a target clustering $g : V \to \mathcal{L}$ is denoted by* $\text{err}(g,h)$ *and is defined as*

$$\text{err}(g,h) := \min_{\sigma\in S_k} \left[\Pr_{x\in V}[h(x) \neq \sigma(g(x))\right].$$

We compare the error of Algorithm 4 with the various linkage based algorithms that are commonly used for hierarchical clustering, as well as Ward's method and the k-means algorithm. We test Algorithm 4 most extensively for $f(x) = x$ while doing a smaller number of tests for $f(x) \in \left\{x^2, \log(1+x), e^x - 1\right\}$. Note that both Ward's method and the k-means algorithm work on the squared Euclidean distance $\|x-y\|_2^2$ between two points $x, y \in V$, i.e., they both require an embedding of the data points into a normed vector space which provides extra information that can be potentially exploited. For the linkage based algorithms we use the same notion of similarity $1 + \kappa_{cos}$ or κ_{gauss} that we use for Algorithm 4. For comparison we use a mix of synthetic data sets as well as the Wine, Iris, Soybean-small, Digits, Glass, and Wdbc data sets from the

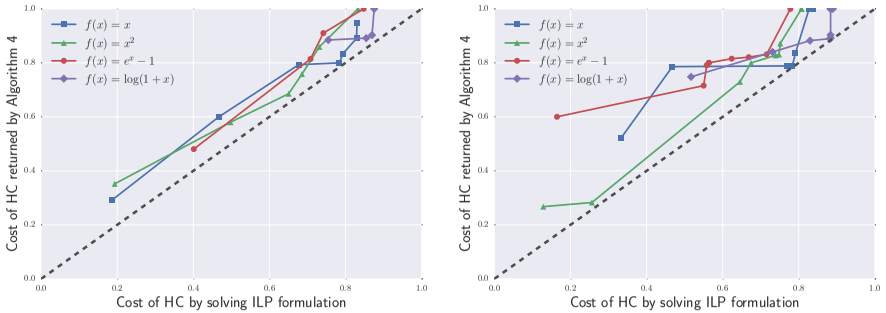


Figure 2: Comparison of f-ILP-ultrametric and Algorithm 4 for $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

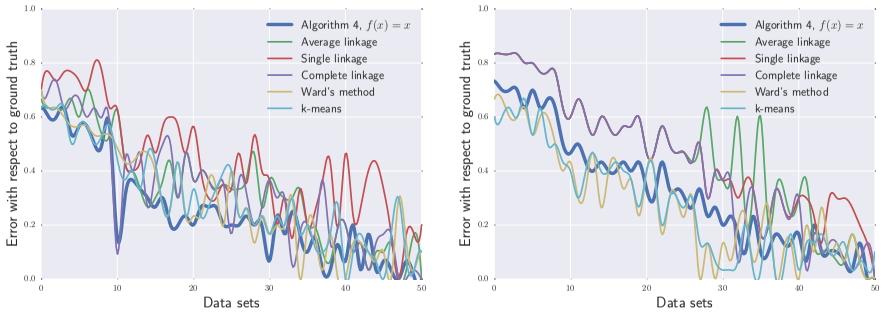


Figure 3: Comparison of Algorithm 4 using $f(x) = x$, with other algorithms for clustering using $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

UCI repository (Lichman, 2013). For some of the larger data sets, we sample uniformly at random a smaller number of data points and take the average of the error over the different runs. Figures 3, 4, 5, and 6 plot the performance of the various algorithms, with the x-axis representing the data sets (sorted in decreasing order of average classification error) and the y-axis representing the classification error. From the figures, we see that the hierarchical clustering returned by Algorithm 4 with $f(x) \in \{x, x^2, \log(1+x), e^x - 1\}$ often has better projections into flat clusterings than the other algorithms. This is especially true when we compare it to the linkage based algorithms, since they use the same pairwise similarity function as Algorithm 4, as opposed to Ward's method and k-means.

7. Hardness of finding the optimal hierarchical clustering

In this section we study the hardness of finding the optimal hierarchical clustering according to cost function (1). We show that under the assumption of the *Small Set Expansion* (SSE) hypothesis there is no constant factor approximation algorithm for this problem. We also show that no polynomial sized Linear Program

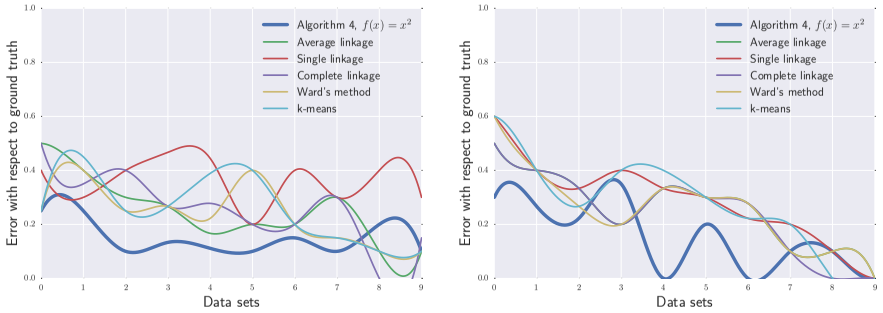


Figure 4: Comparison of Algorithm 4 using $f(x) = x^2$, with other algorithms for clustering using $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

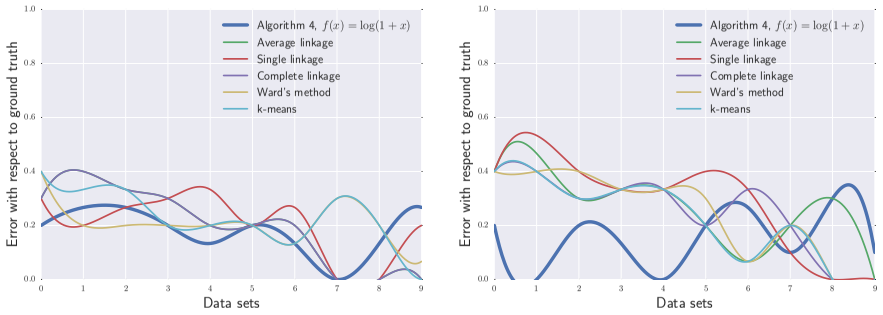


Figure 5: Comparison of Algorithm 4 using $f(x) = \log(1+x)$, with other algorithms for clustering using $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

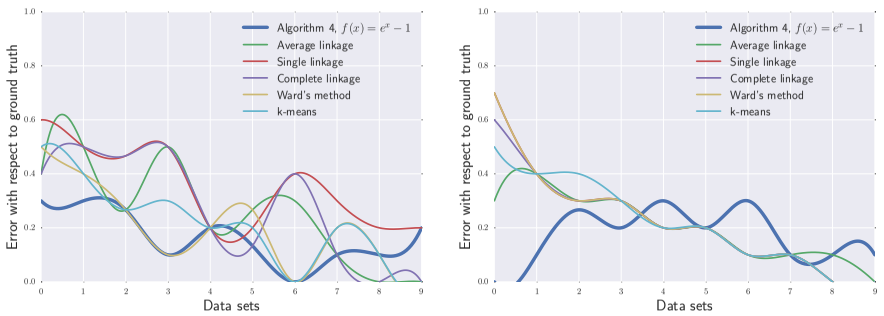


Figure 6: Comparison of Algorithm 4 using $f(x) = e^x - 1$, with other algorithms for clustering using $1 + \kappa_{cos}$ (left) and κ_{gauss} (right)

(LP) or Semidefinite Program (SDP) can give a constant factor approximation for this problem without the need for any complexity theoretic assumptions. Both these results make use of the similarity of this problem with the *minimum linear arrangement* problem. To show hardness under Small Set Expansion, we make use of the result of (Raghavendra et al., 2012) showing that there is no constant factor approximation algorithm for the Minimum Linear Arrangement problem under the assumption of SSE. To show the LP and SDP inapproximability results, we make use of the reduction framework of (Braun et al., 2015) together with the NP-hardness proof for Minimum Linear Arrangement due to (Garey et al., 1976). We also note that both these hardness results hold even for unweighted graphs (i.e., when $\kappa \in \{0,1\}$).
Note that the individual layer-t problem f-ILP-layer for $t = \lfloor n/2 \rfloor$ is equivalent to the *minimum bisection problem* for which the best known approximation is $O(\log n)$ due to (Räcke, 2008), while the best known bi-criteria approximation is $O\left(\sqrt{\log n}\right)$ due to (Arora et al., 2009) and improving these approximation factors is a major open problem. However it is not clear if an improved approximation algorithm for hierarchical clustering under cost function (1) would imply an improved algorithm for every layer-t problem, which is why a constant factor inapproximability result is of interest. We recall the following setup for LP and SDP reductions between optimization problems from (Braun et al., 2015). We denote by $\mathcal{P} = (\mathcal{S}, \mathcal{I}, \mathrm{val})$, a generic optimization problem consisting of *solutions*, *feasible instances* and an *objective function* respectively. The (C,S)-approximate LP and SDP complexity of a problem $\mathcal{P}$ is denoted by $\mathrm{fc}_{\mathrm{LP}}(\mathcal{P},C,S)$ and $\mathrm{fc}_{\mathrm{SDP}}(\mathcal{P},C,S)$ respectively. A reduction from problem $\mathcal{P}_1$ with approximation guarantees (C_1,S_1) to a problem $\mathcal{P}_2$ with approximation guarantees (C_2,S_2) is formally defined as:

Definition 32 (Reduction) *(Braun et al., 2015) Let $\mathcal{P}_1 = (\mathcal{S}_1, \mathcal{I}_1, \mathrm{val})$ and $\mathcal{P}_2 = (\mathcal{S}_2, \mathcal{I}_2, \mathrm{val})$ be optimization problems with guarantees C_1, S_1 and C_2, S_2, respectively. Let $\tau_1 = +1$ if $\mathcal{P}_1$ is a maximization problem, and $\tau_1 = -1$ if $\mathcal{P}_1$ is a minimization problem. Similarly, let $\tau_2 = \pm 1$ depending on whether $\mathcal{P}_2$ is a maximization problem or a minimization problem.*
A reduction *from $\mathcal{P}_1$ to $\mathcal{P}_2$ respecting the guarantees consists of*

1. *two mappings: $*\colon \mathcal{I}_1 \to \mathcal{I}_2$ and $*\colon \mathcal{S}_1 \to \mathcal{S}_2$ translating instances and feasible solutions independently;*

2. *two nonnegative $\mathcal{I}_1 \times \mathcal{S}_1$ matrices M_1, M_2*

subject to the conditions

$$\tau_1 \left[C_1(\mathcal{I}_1) - \mathrm{val}_{\mathcal{I}_1}(s_1)\right] = \tau_2 \left[C_2(\mathcal{I}_1^*) - \mathrm{val}_{\mathcal{I}_1^*}(s_1^*)\right] M_1(\mathcal{I}_1, s_1) + M_2(\mathcal{I}_1, s_1) \qquad \text{(40-complete)}$$

$$\tau_2 \,\mathrm{OPT}\,(\mathcal{I}_1^*) \le \tau_2 S_2(\mathcal{I}_1^*) \qquad \text{if } \tau_1 \,\mathrm{OPT}\,(\mathcal{I}_1) \le \tau_1 S_1(\mathcal{I}_1). \qquad \text{(40-sound)}$$

The matrices M_1 and M_2 control the parameters of the reduction relating the integrality gap of relaxations for $\mathcal{P}_1$ to the integrality gap of corresponding relaxations for $\mathcal{P}_2$. For a matrix A, let $\mathrm{rk}_+ A$ and $\mathrm{rk}_{\mathrm{psd}} A$ denote the nonnegative rank and psd rank of A respectively. The following theorem is a restatement of Theorem 3.2 from (Braun et al., 2015) ignoring constants.

Theorem 33 *(Braun et al., 2015) Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be optimization problems with a reduction from $\mathcal{P}_1$ to $\mathcal{P}_2$ respecting the completeness guarantees C_1, C_2 and soundness guarantees S_1, S_2 of $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Then*

$$\mathrm{fc}_{\mathrm{LP}}(\mathcal{P}_1, C_1, S_1) \le \mathrm{rk}_+ M_2 + \mathrm{rk}_+ M_1 + \mathrm{rk}_+ M_1 \cdot \mathrm{fc}_{\mathrm{LP}}(\mathcal{P}_2, C_2, S_2), \qquad (41)$$

$$\mathrm{fc}_{\mathrm{SDP}}(\mathcal{P}_1, C_1, S_1) \le \mathrm{rk}_{\mathrm{psd}} M_2 + \mathrm{rk}_{\mathrm{psd}} M_1 + \mathrm{rk}_{\mathrm{psd}} M_1 \cdot \mathrm{fc}_{\mathrm{SDP}}(\mathcal{P}_2, C_2, S_2), \qquad (42)$$

where M_1 and M_2 are the matrices in the reduction as in Definition 32.

Therefore to obtain a lower bound for problem $\mathcal{P}_2$, it suffices to find a source problem $\mathcal{P}_1$ and matrices M_1 and M_2 of low nonnegative rank and low psd rank, satisfying Definition 32.
Below, we cast the hierarchical clustering problem (HCLUST) as an optimization problem. We also recall a different formulation of cost function (1) due to (Dasgupta, 2016) that will be useful in the analysis of the reduction.

Definition 34 (HCLUST **as optimization problem**) *The minimization problem* HCLUST *of size n consists of*

instances *similarity function $\kappa : E(K_n) \to \mathbb{R}_{\ge 0}$*
feasible solutions *hierarchical clustering r, T of $V(K_n)$*
measure $\mathrm{val}_\kappa(T) = \sum_{\{i,j\} \in E(K_n)} \kappa(i,j)\, |\mathrm{leaves}(T[\mathrm{lca}(i,j)])|$.

We will also make use of the following alternate interpretation of cost function (1) given by (Dasgupta, 2016). Let $\kappa\colon V \times V \to \mathbb{R}_{\ge 0}$ be an instance of HCLUST. For a subset $S \subseteq V$, a split $S_1, \ldots, S_k$ is a partition of S into k disjoint pieces. For a binary split S_1, S_2 we can define $\kappa(S_1, S_2) := \sum_{i \in S_1, j \in S_2} \kappa(i,j)$. This can be extended to k-way splits in the natural way:

$$\kappa(S_1, \ldots, S_k) := \sum_{1 \le i \le j \le k} \kappa(S_i, S_j).$$

Then the cost of a tree T is the sum over all the internal nodes of the splitting costs at the nodes, as follows.

$$\mathrm{cost}(T) = \sum_{\text{splits } S \to (S_1, \ldots, S_k) \text{ in } T} |S|\, \kappa(S_1, \ldots, S_k).$$

We now briefly recall the MAXCUT problem.

Definition 35 (MAXCUT **as optimization problem**) *The maximization problem* MAXCUT *of size n consists of*

instances *all graphs G with $V(G) \subseteq [n]$*
feasible solutions *all subsets X of $[n]$*
measure $\mathrm{val}_G(X) = |\delta_G(X)|$.

Similarly, the Minimum Linear Arrangement problem can be phrased as an optimization problem as follows.

Definition 36 (MLA **as optimization problem**) *The minimization problem* MLA *of size n consists of*

instances *weight function $w : E(K_n) \to \mathbb{R}_{\geq 0}$*
feasible solutions *all permutations $\pi : V(K_n) \to [n]$*
measure $\mathrm{val}_w(\pi) \coloneqq \sum_{\{i,j\} \in E(K_n)} w(i,j)\,|\pi(i) - \pi(j)|$.

We now describe the reduction from MAXCUT to HCLUST which is a modification of the reduction from MAXCUT to MLA due to (Garey et al., 1976). Note that an instance of MAXCUT maps to an unweighted instance of HCLUST, i.e., $\kappa \in \{0,1\}$.

Mapping instances Given an instance $G = (V,E)$ of MAXCUT of size n, let $r = n^4$ and $U = \{u_1, u_2, \ldots, u_r\}$. The instance κ of HCLUST is on the graph with vertex set $V' \coloneqq V \cup U$ and has weights in $\{0,1\}$. For any distinct pair $i,j \in V'$, if $\{i,j\} \in E$ then we define $\kappa(i,j) \coloneqq 0$ and otherwise we set $\kappa(i,j) \coloneqq 1$.

Mapping solutions Given a cut $X \subseteq V$ of MAXCUT we map it to the clustering r, T of V' where the root r has the following children: n^4 leaves corresponding to U, and 2 internal vertices corresponding to X and $\overline{X}$. The internal vertices for X and $\overline{X}$ are split into $|X|$ and $\left|\overline{X}\right|$ leaves respectively at the next level.

The following lemma relates the LP and SDP formulations for MAXCUT and MLA.

Lemma 37 *For any completeness and soundness guarantee (C,S), we have the following*

$$\mathrm{fc}_{\mathrm{LP}}\left(\mathsf{MAXCUT}, C, S\right) \leq \mathrm{fc}_{\mathrm{LP}}\left(\mathsf{HCLUST}, C', S'\right) + O(n^2)$$
$$\mathrm{fc}_{\mathrm{SDP}}\left(\mathsf{MAXCUT}, C, S\right) \leq \mathrm{fc}_{\mathrm{SDP}}\left(\mathsf{HCLUST}, C', S'\right) + O(n^2).$$

where $C' \coloneqq \frac{(n^4+n)^3-(n^4+n)}{3} - C(n^4+n)$ and $S' \coloneqq \binom{n^4+n+1}{3} - Sn^4$.

Proof To show completeness, we analyze the cost of the tree T that a cut X maps to, using the alternate interpretation of the cost function (1) due to (Dasgupta, 2016) (see above). Let H be the graph on vertex set V' induced by κ, i.e. $\{i,j\} \in E(H)$ iff $\kappa(i,j) = 1$. Let $\overline{H}$ denote the complement graph of H and let $\overline{\kappa}$ be the similarity function induced by it, i.e., $\overline{\kappa}(i,j) = 1$ iff $\{i,j\} \notin E(H)$ and $\overline{\kappa}(i,j) = 0$ otherwise. For a hierarchical clustering T of V', we denote by $\mathrm{cost}_H(T)$ and $\mathrm{cost}_{\overline{H}}(T)$ the cost of T induced by κ and $\overline{\kappa}$ respectively, i.e., $\mathrm{cost}_H(T) \coloneqq \sum_{\{i,j\} \in E(H)} |\mathrm{leaves}(T[\mathrm{lca}(i,j)])|$ and $\mathrm{cost}_{\overline{H}}(T) \coloneqq \sum_{\{i,j\} \notin E(H)} |\mathrm{leaves}(T[\mathrm{lca}(i,j)])|$. Let $\overline{X} \coloneqq V' \setminus X$. The cost of the tree T that the cut X maps to, is given by

$$\begin{aligned}
\mathrm{cost}(T) &= \mathrm{cost}_H(T) \\
&= \frac{\left(n+n^4\right)^3 - (n+n^4)}{3} - \mathrm{cost}_{\overline{H}}(T) \\
&= \frac{\left(n+n^4\right)^3 - (n+n^4)}{3} - \sum_{\text{splits } S \to (S_1,\ldots,S_k) \text{ in } T} |S|\,\overline{\kappa}(S_1,\ldots,S_k) \\
&= \frac{\left(n+n^4\right)^3 - (n+n^4)}{3} - \left(n+n^4\right)\mathrm{val}_G(X) - \left(|X|\,|E[X]| + \left|\overline{X}\right|\,\left|E[\overline{X}]\right|\right),
\end{aligned}$$

where $E[X]$ and $E[\overline{X}]$ are the edges of $E(H)$ induced on the set X and $\overline{X}$ respectively. Therefore, we have the following completeness relationship between the two problems

$$C - \mathrm{val}_G(X) = \frac{1}{n+n^4}\left(\mathrm{cost}(T) - \left(\frac{(n+n^4)^3 - (n+n^4)}{3} - C(n+n^4)\right)\right) + \frac{|X|\,|E[X]| + \left|\overline{X}\right|\,\left|E[\overline{X}]\right|}{n^4+n}.$$

We now define the matrices M_1 and M_2 as $M_1(H,X) \coloneqq \frac{1}{n+n^4}$ and $M_2(H,X) \coloneqq |X|\,|E[X]| + \left|\overline{X}\right|\,\left|E[\overline{X}]\right|$. Clearly, M_1 has $O(1)$ nonnegative rank and psd rank. We claim that the nonnegative rank of M_2 is at most $2^{\binom{n}{2}}$. The vectors $v_H \in \mathbb{R}^{2^{\binom{n}{2}}}$ corresponding to the instances H is defined as the concatenation $[u_H, w_H]$ of two vectors $u_H, w_H \in \mathbb{R}^{\binom{n}{2}}$. Both the vectors u_H, w_H encode the edges of H scaled by n^4+n, i.e., $u_H(\{i,j\}) = w_H(\{i,j\}) = 1/(n^4+n)$ iff $\{i,j\} \in E(H)$ and 0 otherwise. The vectors $v_X \in \mathbb{R}^{2^{\binom{n}{2}}}$ corresponding to the solutions are also defined as the concatenation $[u_X, w_X]$ of two vectors $u_X, w_X \in \mathbb{R}^n$. The vector u_X encodes the vertices in X scaled by $|X|$ i.e., $u_X(\{i,j\}) = |X|$ iff $i,j \in X$ and 0 otherwise. The vector w_X encodes the vertices in $\overline{X}$ scaled by $\left|\overline{X}\right|$ i.e., $w_X(\{i,j\}) = \left|\overline{X}\right|$ iff $i,j \in \overline{X}$ and 0 otherwise. Clearly, we have $M_2(H,X) = \langle v_H, v_X \rangle$ and so the nonnegative (and psd) rank of M_2 is at most $2^{\binom{n}{2}}$. Soundness follows due to the analysis in (Garey et al., 1976) and by noting that the cost of a linear arrangement obtained by projecting the leaves of T is a lower bound on $\mathrm{cost}(T)$. By the analysis in (Garey et al., 1976) if the optimal value $\mathrm{OPT}(G)$ of MAXCUT is at most S, then the optimal value of MLA on V', κ is at least $\binom{n^4+n+1}{3} - Sn^4$. Therefore, it follows that the optimal value of HCLUST on V', κ is also at least $\binom{n^4+n+1}{3} - Sn^4$. ■

The constant factor inapproximability result for HCLUST now follows due to the following theorems.

Theorem 38 ((Chan et al., 2013, Theorem 3.2)) *For any $\varepsilon > 0$ there are infinitely many n such that*

$$\mathrm{fc}_{\mathrm{LP}}\left(\mathsf{MAXCUT}, 1-\varepsilon, \frac{1}{2} + \frac{\varepsilon}{6}\right) \geq n^{\Omega(\log n / \log\log n)}.$$

Theorem 39 ((Braun et al., 2015, Theorem 7.1)) *For any $\delta, \varepsilon > 0$ there are infinitely many n such that*

$$\mathrm{fc}_{\mathrm{SDP}}\left(\mathsf{MAXCUT}, \frac{4}{5} - \varepsilon, \frac{3}{4} + \delta\right) = n^{\Omega(\log n / \log\log n)}. \tag{43}$$

Thus we have the following corollary about the LP and SDP inapproximability for the problem HCLUST.

Corollary 40 (LP and SDP hardness for HCLUST**)** *For any constant $c \geq 1$,* HCLUST *is LP-hard and SDP-hard with an inapproximability factor of c.*

Proof Straightforward by using Theorems 38 and 39 together with Lemma 37 and by choosing n large enough. ■

The following lemma shows that a minor modification of the argument in (Raghavendra et al., 2012) also implies a constant factor inapproximability result under the *Small Set Expansion* (SSE) hypothesis. Note that this reduction is also true for unit capacity graphs, i.e., $\kappa \in \{0,1\}$. The SSE hypothesis implies the well-known Unique Games Conjecture (UGC) and it is believed that the two may be equivalent. We briefly recall the formulation of the Small Set Expansion hypothesis. Informally, given a graph $G = (V,E)$ the problem is to decide whether all "small" sets in the graph are expanding. Let $d(i)$ denote the degree of a vertex $i \in V$. For a subset $S \subseteq V$ let $\mu(S) \coloneqq |S| \,/\, |V|$ be the volume of S, and let $\phi(S) \coloneqq E(S,\overline{S}) / \sum_{i \in S} d(i)$ be the expansion of S. Then the SSE problem is defined as follows.

Definition 41 (Small set expansion (SSE) hypothesis (Raghavendra et al., 2012)) *For every constant $\eta > 0$, there exists sufficiently small $\delta > 0$ such that given a graph $G = (V, E)$, it is NP-hard to decide the following cases,*

Completeness *there exists a subset $S \subseteq V$ with volume $\mu(S) = \delta$ and expansion $\phi(S) \leq \eta$,*

Soundness *every subset $S \subseteq V$ of volume $\mu(S) = \delta$ has expansion $\phi(S) \geq 1 - \eta$.*

Under this assumption, (Raghavendra et al., 2012) proved the following amplification result about the expansion of small sets in the graph.

Theorem 42 (Theorem 3.5 (Raghavendra et al., 2012)) *For all $q \in \mathbb{N}$ and $\varepsilon', \gamma > 0$ it is SSE-hard to distinguish the following for a given graph $H = (V_H, E_H)$*

Completeness *There exist disjoint sets $S_1, \ldots, S_q \subseteq V_H$ satisfying $\mu(S_i) = \frac{1}{q}$ and $\phi(S_i) \leq \varepsilon' + o(\varepsilon')$ for all $i \in [n]$,*

Soundness *For all sets $S \subseteq V_H$ we have $\phi(S) \geq \phi_{\mathcal{G}}(1 - \varepsilon'/2)(\mu(S)) - \gamma/\mu(S)$,*

where $\phi_{\mathcal{G}}(1 - \varepsilon'/2)(\mu(S))$ is the expansion of sets of volume $\mu(S)$ in the infinite Gaussian graph $\mathcal{G}(1 - \varepsilon'/2)$.

The following lemma establishes that it is SSE-hard to approximate HCLUST to within any constant factor. The argument closely parallels Corollary A.5 of (Raghavendra et al., 2012) where it was shown that it is SSE-hard to approximate MLA to within any constant factor.

Lemma 43 *Let $G = (V, E)$ be a graph on V with κ induced by the edges E i.e., $\kappa(i, j) = 1$ iff $\{i, j\} \in E$ and 0 otherwise. Then it is SSE-hard to distinguish between the following two cases*

Completeness *There exists a hierarchical clustering T of V with $\text{cost}(T) \leq \varepsilon n |E|$,*

Soundness *Every hierarchical clustering T of V satisfies $\text{cost}(T) \geq c\sqrt{\varepsilon} n |E|$*

for some constant c not depending on n.

Proof Apply Theorem 42 on the graph G with the following choice of parameters: $q = \lceil 2/\varepsilon \rceil$, $\varepsilon' = \varepsilon/3$ and $\gamma = \varepsilon$. Suppose there exist $S_1, \ldots, S_q \subseteq V$ satisfying $\phi(S_i) \leq \varepsilon' + o(\varepsilon')$ and $|S_i| = |V| / q \leq \varepsilon |V| / 2$. Then consider the tree r, T with the root r having q children corresponding to each S_i, and each S_i being further separated into $|S_i|$ leaves at the next level. We claim that $\text{cost}(T) \leq \varepsilon n |E|$. We analyze this using the alternate interpretation of cost function (1) (see above). Every crossing edge between S_i, S_j for distinct $i, j \in [q]$ incurs a cost of n, but by assumption there are at most $\varepsilon |E| / 2$ such edges. Further, any edge in S_i incurs a cost $\frac{n}{q} \leq \varepsilon n/2$ and thus their contribution is upper bounded by $\varepsilon n |E|$.
The analysis for soundness follows by the argument of Corollary A.5 in (Raghavendra et al., 2012). In particular, if for every $S \subseteq V$ we have $\phi(S) \geq \phi_{\mathcal{G}}(1 - \varepsilon'/2)(\mu(S)) - \gamma/\mu(S)$ then the cost of the optimal linear arrangement on G is at most $\sqrt{\varepsilon} n |E|$. Since the cost of any tree (including the optimal tree) is at least the cost of the linear arrangement induced by projecting the leaf vertices, the claim about soundness follows. ■

8. Discussion

In this work we have studied the cost functions (1) and (21) for hierarchical clustering given a pairwise similarity function over the data and shown an $O(\log n)$ approximation algorithm for this problem. As briefly mentioned in Section 2 however, such a cost function is not unique. Further, there is an intimate connection between hierarchical clusterings and ultrametrics over discrete sets which points to other directions for formulating a cost function over hierarchies. In particular we briefly mention the related notion of *hierarchically well-separated trees* (HST) as defined in (Bartal, 1996) (see also (Bartal et al., 2001, 2003)). A k-HST for $k \geq 1$ is a tree T such that each vertex $u \in T$ has a label $\Delta(u) \geq 0$ such that $\Delta(u) = 0$ if and only if u is a leaf of T. Further, if u is a child of v in T then $\Delta(u) \leq \Delta(v)/k$. It is well known that any ultrametric d on a finite set V is equivalent to a 1-HST where V is the set of leaves of T and $d(i, j) = \Delta(\text{lca}(i, j))$ for every $i, j \in V$. Thus in the special case when $\Delta(u) = |\text{leaves } T[u]| - 1$ we get the cost function (1), while if $\Delta(u) = f(|\text{leaves } T[u]| - 1)$ for a strictly increasing function f with $f(0) = 0$ then we get cost function (21). It turns out this assumption on Δ enables us to prove the combinatorial results of Section 3 and give an $O(\log n)$ approximation algorithm to find the optimal cost tree according to these cost functions. It is an interesting problem to investigate cost functions and algorithms for hierarchical clustering induced by other families of Δ that arise from a k-HST on V, i.e., if the cost of T is defined as

$$\text{cost}_{\Delta}(T) \coloneqq \sum_{\{i,j\} \in E(K_n)} \kappa(i, j) \Delta(\text{lca}(i, j)). \tag{44}$$

Note that not all choices of Δ lead to a meaningful cost function. For example, choosing $\Delta(u) = \text{diam}(T[u]) - 1$ gives rise to the following cost function

$$\text{cost}(T) \coloneqq \sum_{\{i,j\} \in E(K_n)} \kappa(i, j) \, \text{dist}_T(i, j) \tag{45}$$

where $\text{dist}_T(i, j)$ is the length of the unique path from i to j in T. In this case, the trivial clustering r, T^* where T^* is the star graph with V as its leaves and r as the root is always a minimizer; in other words, there is no incentive for spreading out the hierarchical clustering. Also worth mentioning is a long line of related work on fitting tree metrics to metric spaces (see e.g., (Ailon and Charikar, 2005; Räcke, 2008; Fakcharoenphol et al., 2003)). In this setting, the data points V are assumed to come from a metric space d_V and the objective is to find a hierarchical clustering T so as to minimize $\|d_V - d_T\|_p$. If the points in V lie on the unit sphere and the similarity function κ is the cosine similarity $\kappa_{cos}(i, j) = 1 - d_V(i, j)/2$, then the problem of fitting a tree metric with $p = 2$ minimizes the same objective as cost function (45). Since $d_V \leq 1$ in this case, the minimizer is the trivial tree r, T^* (as remarked above). In general, when the points in V are not constrained to lie on the unit sphere, the two problems are incomparable.

Acknowledgments

Research reported in this paper was partially supported by NSF CAREER award CMMI-1452463 and NSF grant CMMI-1333789. We would like to thank Kunal Talwar and Mohit Singh for helpful discussions and anonymous reviewers for helping improve the presentation of this paper.

Appendix A. Missing proofs from Section 2

Lemma 44 (Restatement of Lemma 3.) *Let d be a non-trivial ultrametric on V and let $S_t \subseteq V$ be an equivalence class under the relation $i \sim j$ iff $d(i, j) \leq t$. Then d restricted to S_t is a non-trivial ultrametric on S_t.*

Proof Clearly d restricted to S_t is an ultrametric on S_t and so we need to establish that it satisfies Conditions 1 and 2 of Definition 2. Let $S \subseteq S_t$ be any set. Since d is a non-trivial ultrametric on V it follows that there is a pair $i, j \in S$ with $d(i,j) \geq |S| - 1$, and so d restricted to S_t satisfies Condition 1.
If S'_r is an equivalence class in S_t under the relation $i \sim j$ iff $d(i,j) \leq r$ then clearly $S'_r = S_t$ if $r > t$. Since d is a non-trivial ultrametric on V, it follows that $\max_{i,j \in S'_r} d(i,j) = \max_{i,j \in S_t} d(i,j) \leq |S_t| - 1 = |S'_r| - 1$. Thus we may assume that $r \leq t$. Consider an $i \in S'_r$ and let $j \in V$ be such that $d(i,j) \leq r$. Since $r \leq t$ and $i \in S_t$, it follows that $j \in S_t$ and so $j \in S'_r$. In other words S'_r is an equivalence class in V under the relation $i \sim j$ iff $d(i,j) \leq r$. Since d is an ultrametric on V it follows that $\max_{i,j \in S'_r} d(i,j) \leq |S'_r| - 1$. Thus d restricted to S_t satisfies Condition 2. ■

Lemma 45 (Restatement of Lemma 4.) *Let d be a non-trivial ultrametric on the set V as in Definition 2. Then the range of d is contained in the set $\{0, 1, \ldots, n-1\}$ with $|V| = n$.*

Proof We will prove this by induction on $|V|$. The base case when $|V| = 1$ is trivial. Therefore, we now assume that $|V| > 1$. By Condition 1 there is a pair $i, j \in V$ such that $d(i,j) \geq n-1$. Let $t = \max_{i,j \in V} d(i,j)$, then the only equivalence class under the relation $i \sim j$ iff $d(i,j) \leq t$ is V. By Condition 2 it follows that $\max_{i,j \in V} d(i,j) = t = n-1$. Let $V_1, \ldots V_m$ denote the set of equivalence classes of V under the relation $i \sim j$ iff $d(i,j) \leq n-2$. Note that $m > 1$ as there is a pair $i, j \in V$ with $d(i,j) = n-1$, and therefore each $V_l \subsetneq V$. By Lemma 3, d restricted to each of these V_l's is a non-trivial ultrametric on those sets. The claim then follows immediately: for any $i, j \in V$ either $i, j \in V_l$ for some V_l in which case by the induction hypothesis $d(i,j) \in \{0, 1, \ldots, |V_l| - 1\}$, or $i \in V_l$ and $j \in V_{l'}$ for $l \neq l'$ in which case $d(i,j) = n-1$. ■

Appendix B. Missing proofs from Section 3

Lemma 46 (Restatement of Lemma 5) *Let $S \subseteq V$ with $|S| \geq 2$. If $r = \mathrm{lca}(S)$ then there is a pair $i, j \in S$ such that $\mathrm{lca}(i,j) = r$.*

Proof We will proceed by induction on $|S|$. If $|S| = 2$ then the claim is trivial and so we may assume $|S| > 2$. Let $i \in S$ be an arbitrary point and let $r' = \mathrm{lca}(S \setminus \{i\})$. We claim that $r = \mathrm{lca}(i, r')$. Clearly the subtree rooted at $\mathrm{lca}(i, r')$ contains S and since $T[r]$ is the smallest such tree it follows that $r \in T[\mathrm{lca}(i, r')]$. Conversely, $T[r]$ contains $S \setminus \{i\}$ and so $r' \in T[r]$ and since $i \in T[r]$, it follows that $\mathrm{lca}(i, r') \in T[r]$. Thus we conclude that $r = \mathrm{lca}(i, r')$.
If $\mathrm{lca}(i, r') = r'$, then we are done by the induction hypothesis. Thus we may assume that $i \notin T[r']$. Consider any $j \in S$ such that $j \in T[r']$. Then we have that $\mathrm{lca}(i,j) = r$ as $\mathrm{lca}(i, r') = r$ and $j \in T[r']$ and $i \notin T[r']$. ■

Lemma 47 (Restatement of Lemma 6) *Let T be a hierarchical clustering on V and let d_T be the ultrametric on V induced by it. Then d_T is non-trivial.*

Proof Let $S \subseteq V$ be arbitrary and $r = \mathrm{lca}(S)$, then $T[r]$ has at least $|S|$ leaves. By Lemma 5 there must be a pair $i, j \in S$ such that $r = \mathrm{lca}(i,j)$ and so $d_T(i,j) \geq |S| - 1$. This satisfies Condition 1 of non-triviality.
For any t, let S_t be a non-empty equivalence class under the relation $i \sim j$ iff $d_T(i,j) \leq t$. Since d_T satisfies Condition 1 it follows that $|S_t| - 1 \leq t$. Let us assume for the sake of contradiction that there is a pair $i, j \in S_t$ such that $d_T(i,j) > |S_t| - 1$. Let $r = \mathrm{lca}(S_t)$; using the definition of d_T it follows that $t + 1 \geq |\mathrm{leaves}(T[r])| > |S_t|$ since $i, j \in S_t$. Let $k \in \mathrm{leaves}(T[r]) \setminus S_t$ be an arbitrary point, then for every $l \in S_t$ it follows that $d_T(k,l) \leq |\mathrm{leaves}(T[r])| - 1 \leq t$ since the subtree rooted at r contains both k and l. This is a contradiction to S_t being an equivalence class under $i \sim j$ iff $d_T(i,j) \leq t$ since $k \notin S_t$. Thus d_T also satisfies Condition 2 of Definition 2. ■

Lemma 48 (Restatement of Lemma 7) *For every non-trivial ultrametric d on V there is a hierarchical clustering T on V such that for any pair $i, j \in V$ we have*

$$d_T(i,j) = |\mathrm{leaves}(T[\mathrm{lca}\,(i,j)])| - 1 = d(i,j).$$

Moreover this hierarchy can be constructed in time $O\left(n^3\right)$ by Algorithm 1 where $|V| = n$.

Proof The proof is by induction on n. The base case when $n = 1$ is straightforward. We now suppose that the statement is true for sets of size $< n$. Note that $i \sim j$ iff $d(i,j) \leq n-2$ is an equivalence relation on V and thus partitions V into m equivalence classes $V_1, \ldots, V_m$. We first observe that $m > 1$ since by Condition 1 there is a pair of points $i, j \in V$ such that $d(i,j) \geq n-1$ and in particular $|V|_l < n$ for every $l \in \{1, \ldots, m\}$. By Lemma 3, d restricted to any V_l is a non-trivial ultrametric on V_l and there is a pair of points $i, j \in V_l$ such that $d(i,j) = |V_l| - 1$ by Conditions 1 and 2. Therefore by the induction hypothesis we construct trees $T_1, \ldots, T_m$ such that for every $l \in \{1, \ldots, m\}$ we have $\mathrm{leaves}(T_l) = V_l$. Further for any pair of points $i, j \in V_l$ for some $l \in \{1, \ldots, m\}$, we also have $d(i,j) = d_{T_l}(i,j)$.
We construct the tree T as follows: we first add a root r and then connect the root r_l of T_l to r for every $l \in \{1, \ldots, m\}$. Consider a pair of points $i, j \in V$. If $i, j \in V_l$ for some $l \in \{1, \ldots, m\}$ then we are done since $d_{T_l}(i,j) = d_T(i,j)$ as $\mathrm{lca}(i,j) \in T_l$. If $i \in V_l$ and $j \in V_{l'}$ for some $l \neq l'$ then $d(i,j) = n-1$ since $d(i,j) \geq n-1$ by definition of the equivalence relation and the range of d lies in $\{0, 1, \ldots, n-1\}$ by Lemma 4. Moreover i and j are leaves in T_l and $T_{l'}$ respectively, and thus by construction of T we have $\mathrm{lca}(i,j) = r$, i.e., $d_T(i,j) = n-1$ and so the claim follows. Algorithm 1 simulates this inductive argument can be easily implemented to run in time $O\left(n^3\right)$. ■

Appendix C. Missing proofs from Section 4

Lemma 49 (Restatement of Lemma 18) *Let $\mathcal{B}_U(i,r,t)$ be the ball corresponding to a set $U \subseteq V$, vertex $i \in U$, radius $r \in \mathbb{R}$ and $t \in [n-1]$. Then $\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)$ is differentiable with respect to r in the interval $(0, \Delta]$ except at finitely many points.*

Proof Note that for any fixed $U \subseteq V$, $\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)$ is a monotone non-decreasing function in r since for a pair $j, k \in U$ such that $j \in \mathcal{B}_U(i,r,t)$ and $k \notin \mathcal{B}_U(i,r,t)$ we have $r - d_t(i,j) \leq d_t(j,k)$ otherwise $r - d_t(i,j) > d_t(j,k)$ so that $r > d_t(i,j) + d_t(j,k) \geq d_t(i,k)$, a contradiction to the fact that $k \notin \mathcal{B}_U(i,r,t)$. Therefore adding the vertex k to the ball centered at i is only going to increase its volume as $r - d_t(i,j) \leq d_t(j,k)$ (see Definition 15). Thus $\mathrm{vol}\left(\mathcal{B}_U(i,r,t)\right)$ is differentiable with respect to r in the interval $(0, \Delta]$ except at finitely many points which correspond to a new vertex from U being added to the ball. ■

Theorem 50 (Restatement of Theorem 19) *Let $m_\varepsilon := \left\lfloor \frac{n-1}{1+\varepsilon} \right\rfloor$ as in Algorithm 2 and let $\left\{ x^t_{ij} \mid t \in [m_\varepsilon], i, j \in V \right\}$ be the output of Algorithm 2 run on a feasible solution $\{d_t\}_{t \in [n-1]}$ of LP-ultrametric and any choice of $\varepsilon \in (0,1)$. For any $t \in [m_\varepsilon]$, we have that x^t_{ij} is feasible for the layer-$\lfloor (1+\varepsilon)\, t \rfloor$ problem ILP-layer and there is a constant $c(\varepsilon) > 0$ depending only on ε such that*

$$\sum_{\{i,j\} \in E(K_n)} \kappa(i,j) x^t_{ij} \leq c(\varepsilon)(\log n)\gamma_t.$$

Moreover, Algorithm 2 can be implemented to run in time polynomial in n.

Proof We first show that for a fixed t, the constructed solution x^t_{ij} is feasible for the layer-$\lfloor(1+\varepsilon)t\rfloor$ problem ILP-layer. Let $\mathcal{C}_t$ be as in Algorithm 2 so that $x^t_{ij}=1$ if i,j belong to different sets in $\mathcal{C}_t$ and $x^t_{ij}=0$ otherwise. Let $G_t=(V,E_t)$ be as in Definition 9 corresponding to x^t_{ij}. Note that for any $t\in[m_\varepsilon]$, every $V_i\in\mathcal{C}_t$ is a clique in G_t by construction (line 19) and for every distinct pair $V_i,V_j\in\mathcal{C}_t$ we have $V_i\cap V_j=\emptyset$ (lines 15 and 16). Therefore by Lemma 10, it suffices to prove that for any $V_i\in\mathcal{C}_t$, it holds $|V_i|\le\lfloor(1+\varepsilon)t\rfloor$. If V_i is added to $\mathcal{C}_t$ in line 9 then there is nothing to prove.

Thus let us assume that V_i is of the form $\mathcal{B}_U(i,r,t)$ for some $U\subseteq V$ as in line 14 so that $\phi(\mathcal{B}_U(i,r,t))\le\frac{1}{\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)$. Note that by Lemma 13 it suffices to show that there is such an $r\in(0,\Delta]$. This property follows from the rounding scheme due to (Even et al., 1999) as we will explain now.

By Lemma 18 $\mathrm{vol}(\mathcal{B}_U(i,r,t))$ is differentiable everywhere in the interval $(0,\Delta]$ except at finitely many points X. Let the set of discontinuous points be $X=\{x_1,x_2,\ldots,x_{k-1}\}$ with $x_0=0<x_1<x_2\ldots x_{k-1}<x_k=\Delta$. We claim that there must be an $r\in(0,\Delta]\setminus X$ such that $\phi(\mathcal{B}_U(i,r,t))\le\frac{1}{\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)$. Let us assume for the sake of a contradiction that for every $r\in(0,\Delta]\setminus X$ we have $\phi(\mathcal{B}_U(i,r,t))>\frac{1}{\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)$. However integrating both sides from 0 to Δ results in a contradiction:

$$\begin{aligned}
\int_{r=0}^{\Delta}\phi(\mathcal{B}_U(i,r,t))\,dr &= \int_{r=0}^{\Delta}\frac{\partial\mathcal{B}_U(i,r,t)}{\mathrm{vol}(\mathcal{B}_U(i,r,t))}dr && (46)\\
&= \sum_{i=1}^{k}\int_{r=x_{i-1}}^{x_i}\frac{\partial\mathcal{B}_U(i,r,t)}{\mathrm{vol}(\mathcal{B}_U(i,r,t))}dr && (47)\\
&= \sum_{i=1}^{k}\int_{r=x_{i-1}}^{x_i}\frac{d(\mathrm{vol}(\mathcal{B}_U(i,r,t)))}{\mathrm{vol}(\mathcal{B}_U(i,r,t))} && (48)\\
&\le \log\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))-\log\mathrm{vol}(\mathcal{B}_U(i,0,t)) && (49)\\
&= \int_{r=0}^{\Delta}\frac{1}{\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)dr, && (50)
\end{aligned}$$

where line 49 follows since f is monotonic increasing. For any $t\in[m_\varepsilon]$ the set $\mathcal{C}_t$ is a disjoint partition of V with balls of the form $\mathcal{B}_U(i,r,t')$ for some $t'\ge t$ and $U\subseteq U_l\in\mathcal{C}_{t'+1}$: this is easily seen by induction since $\mathcal{C}_{m_\varepsilon+1}$ is initialized as V. Further, a cluster V_i is added to $\mathcal{C}_t$ either in line 15 in which case it is a ball of the form $\mathcal{B}_U(i,r,t)$ for some $U\in\mathcal{C}_{t+1}$, $i\in U$, and $r\in\mathbb{R}$ or it is added in line 9 in which case it must have been a ball $\mathcal{B}_U(i',r',t')$ for some $t'>t$, $U\subseteq U_l\in\mathcal{C}_{t'+1}$, $i'\in V$, and $r'\in\mathbb{R}$. Note that for any $t'\ge t$ and $U\subseteq V$, it holds $\gamma^U_{t'}\le\gamma^U_t$ since for every pair $i,j\in V$ we have $\kappa(i,j)\ge0$ and $d_t(i,j)\ge d_{t'}(i,j)$ because of constraint (15). Moreover, for any subset $U\subseteq V$ we have $\gamma^U_t\le\gamma_t$ since $\kappa,d_t\ge0$. We claim that for any $t\in[m_\varepsilon]$ the total volume of the balls in $\mathcal{C}_t$ is at most $\left(2+\frac{1}{\log n}\right)\gamma_t$. First note that the affine term $\frac{\gamma^U_{t'}}{n\log n}$ in the volume of a ball $\mathcal{B}_U(i,r,t')$ in $\mathcal{C}_t$ is upper bounded by $\frac{\gamma_t}{n\log n}$ and appears at most n times. Next we claim that the contribution to the total volume from the term involving the edges inside and crossing a ball $\mathcal{B}_U(i,r,t')\in\mathcal{C}_t$ is at most $2\gamma_t$. This is because the balls are disjoint, $r-d_{t'}(i,k)\le d_{t'}(j,k)\le d_t(j,k)$ for the crossing edges of a ball $\mathcal{B}_U(i,r,t')\in\mathcal{C}_t$ and a crossing edge contributes to the volume of at most 2 balls in $\mathcal{C}_t$. Note that for any $U\subseteq V$, $i\in U$, and $r\in\mathbb{R}_{\ge0}$ we have $\mathrm{vol}(\mathcal{B}_U(i,r,t))\in\left[\frac{\gamma^U_t}{n\log n},\left(1+\frac{1}{n\log n}\right)\gamma^U_t\right]$.

Using this observation and the stopping condition of line 14 it follows that

$$\begin{aligned}
\sum_{\{i,j\}\in E(K_n)}\kappa(i,j)x^t_{ij} &= \sum_{\substack{\{i,j\}\in E(K_n):\\ i,j\text{ separated in }\mathcal{C}_t}}\kappa(i,j)\\
&= \frac{1}{2}\underbrace{\sum_{\substack{\mathcal{B}_U(i,r,t')\in\mathcal{C}_t:\\ t'\ge t\\ U\subseteq U_l\in\mathcal{C}_{t'+1}}}\ \sum_{\substack{j\in\mathcal{B}_U(i,r,t')\\ k\notin\mathcal{B}_U(i,r,t')}}\kappa(j,k)}_{\text{Since }\kappa\text{ is symmetric}}\\
&= \frac{1}{2}\sum_{\substack{\mathcal{B}_U(i,r,t')\in\mathcal{C}_t:\\ t'\ge t\\ U\subseteq U_l\in\mathcal{C}_{t'+1}}}\partial\mathcal{B}_U(i,r,t')\\
&= \frac{1}{2}\sum_{\substack{\mathcal{B}_U(i,r,t')\in\mathcal{C}_t:\\ t'\ge t\\ U\subseteq U_l\in\mathcal{C}_{t'+1}}}\phi(\mathcal{B}_U(i,r,t'))\,\mathrm{vol}(\mathcal{B}_U(i,r,t'))\\
&\le \sum_{\substack{\mathcal{B}_U(i,r,t')\in\mathcal{C}_t:\\ t'\ge t\\ U\subseteq U_l\in\mathcal{C}_{t'+1}}}\frac{1}{2\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t'))}{\mathrm{vol}(\mathcal{B}_U(i,0,t'))}\right)\mathrm{vol}(\mathcal{B}_U(i,r,t'))\\
&\le \frac{1}{2\Delta}\underbrace{(\log(n\log n+1))}_{\text{via interval bounds}}\sum_{\substack{\mathcal{B}_U(i,r,t')\in\mathcal{C}_t:\\ t'\ge t\\ U\subseteq U_l\in\mathcal{C}_{t'+1}}}\mathrm{vol}(\mathcal{B}_U(i,r,t'))\\
&\le \frac{1+\varepsilon}{2\varepsilon}(\log(n\log n+1))\underbrace{\left(2+\frac{1}{\log n}\right)\gamma_t}_{\substack{\text{contribution of affine term}\le\frac{\gamma_t}{\log n}\\ \text{contribution of edge terms}\le2\gamma_t}}\\
&\le c(\varepsilon)(\log n)\gamma_t,
\end{aligned}$$

for some constant $c(\varepsilon)>0$ depending only on ε.

For the run time of Algorithm 2 note that the loop in line 5 runs for at most $n-1$ steps, while the loop in line 7 runs for at most n steps. For a set $U\subseteq V$, to compute the ball $\mathcal{B}_U(i,r,t)$ of least radius r such that $\phi(\mathcal{B}_U(i,r,t))\le\frac{1}{\Delta}\log\left(\frac{\mathrm{vol}(\mathcal{B}_U(i,\Delta,t))}{\mathrm{vol}(\mathcal{B}_U(i,0,t))}\right)$, sort the vertices in $U\setminus\{i\}$ in increasing order of distance from i according to d_t. Let the vertices in $U\setminus\{i\}$ in this sorted order be $\{j_1,\ldots,j_{|U|-1}\}$. Then it suffices to check the expansion of the balls $\{i\}$ and $\{i\}\cup\{j_1,\ldots,j_k\}$ for every $k\in[|U|-1]$. It is straightforward to see that all the other steps in Algorithm 2 run in time polynomial in n. ■

References

Margareta Ackerman, Shai Ben-David, and David Loker. Characterization of linkage-based clustering. In *COLT*, pages 270–281. Citeseer, 2010.

Nir Ailon and Moses Charikar. Fitting tree metrics: Hierarchical clustering and phylogeny. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 73–82. IEEE, 2005.

Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009.

Pranjal Awasthi, Afonso S Bandeira, Moses Charikar, Ravishankar Krishnaswamy, Soledad Villar, and Rachel Ward. Relax, no need to round: Integrality of clustering formulations. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 191–200. ACM, 2015.

Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. A discriminative framework for clustering via similarity functions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 671–680. ACM, 2008.

Yair Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*, pages 184–193. IEEE, 1996.

Yair Bartal. Graph decomposition lemmas and their role in metric embedding methods. In *European Symposium on Algorithms*, pages 89–97. Springer, 2004.

Yair Bartal, Béla Bollobás, and Manor Mendel. A ramsey-type theorem for metric spaces and its applications for metrical task systems and related problems. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 396–405. IEEE, 2001.

Yair Bartal, Nathan Linial, Manor Mendel, and Assaf Naor. On metric ramsey-type phenomena. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 463–472. ACM, 2003.

Gábor Braun, Sebastian Pokutta, and Aurko Roy. Strong reductions for extended formulations. *CoRR*, abs/1512.04932, 2015. URL `http://arxiv.org/abs/1512.04932`.

Siu On Chan, James Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large lp relaxations. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 350–359. IEEE, 2013.

Moses Charikar and Vaggos Chatziafratis. Approximate hierarchical clustering via sparsest cut and spreading metrics. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 841–854. SIAM, 2017.

Moses Charikar and Shi Li. A dependent lp-rounding approach for the k-median problem. In *Automata, Languages, and Programming*, pages 194–205. Springer, 2012.

Moses Charikar, Sudipto Guha, Éva Tardos, and David B Shmoys. A constant-factor approximation algorithm for the k-median problem. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 1–10. ACM, 1999.

Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 524–533. IEEE, 2003.

Sanjoy Dasgupta. A cost function for similarity-based hierarchical clustering. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 118–127. ACM, 2016. ISBN 978-1-4503-4132-5. doi: 10.1145/2897518.2897527. URL `http://doi.acm.org/10.1145/2897518.2897527`.

Sanjoy Dasgupta and Philip M Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005.

Marco Di Summa, David Pritchard, and Laura Sanità. Finding the closest ultrametric. *Discrete Applied Mathematics*, 180:70–80, 2015.

Guy Even, Joseph Naor, Satish Rao, and Baruch Schieber. Fast approximate graph partitioning algorithms. *SIAM Journal on Computing*, 28(6):2187–2214, 1999.

Guy Even, Joseph Seffi Naor, Satish Rao, and Baruch Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *Journal of the ACM (JACM)*, 47(4):585–616, 2000.

Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 448–455. ACM, 2003.

Joseph Felsenstein and Joseph Felenstein. *Inferring phylogenies*, volume 2. Sinauer Associates Sunderland, 2004.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin, 2001.

Michael R Garey, David S. Johnson, and Larry Stockmeyer. Some simplified np-complete graph problems. *Theoretical computer science*, 1(3):237–267, 1976.

Naveen Garg, Vijay V Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.

Anupam Gupta. Lecture notes on approximation algorithms. *Available at* `https://www.cs.cmu.edu/afs/cs/academic/class/15854-f05/www/scribe/lec20.pdf`, 2005.

Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2015. URL `http://www.gurobi.com`.

Kamal Jain and Vijay V Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM (JACM)*, 48(2):274–296, 2001.

Kamal Jain, Mohammad Mahdian, Evangelos Markakis, Amin Saberi, and Vijay V Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing lp. *Journal of the ACM (JACM)*, 50 (6):795–824, 2003.

Nicholas Jardine and Robin Sibson. The construction of hierarchic and non-hierarchic classifications. *The Computer Journal*, 11(2):177–184, 1968.

Nicholas Jardine and Robin Sibson. Mathematical taxonomy. *London etc.: John Wiley*, 1971.

Robert Krauthgamer, Joseph Seffi Naor, and Roy Schwartz. Partitioning graphs into balanced components. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 942–949. Society for Industrial and Applied Mathematics, 2009.

Tom Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 422–431. IEEE, 1988.

Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999.

Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 901–910. ACM, 2013.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Marina Meilă and David Heckerman. An experimental comparison of model-based clustering methods. *Machine learning*, 42(1-2):9–29, 2001.

Jiming Peng and Yu Wei. Approximating k-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 18(1):186–205, 2007.

Jiming Peng and Yu Xia. A new theoretical framework for k-means-type clustering. In *Foundations and advances in data mining*, pages 79–96. Springer, 2005.

Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 255–264. ACM, 2008.

Prasad Raghavendra, David Steurer, and Madhur Tulsiani. Reductions between expansion problems. In *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on*, pages 64–73. IEEE, 2012.

Ben Recht, Christopher Re, Joel Tropp, and Victor Bittorf. Factoring nonnegative matrices with linear programs. In *Advances in Neural Information Processing Systems*, pages 1214–1222, 2012.

Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

Peter HA Sneath, Robert R Sokal, et al. *Numerical taxonomy. The principles and practice of numerical classification.* 1973.

Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

Reza Bosagh Zadeh and Shai Ben-David. A uniqueness theorem for clustering. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 639–646. AUAI Press, 2009.

Journal of Machine Learning Research 18 (2017) 1-5 Submitted 3/17; Revised 5/17; Published 8/17

The MADP Toolbox: An Open Source Library for Planning and Learning in (Multi-)Agent Systems

Frans A. Oliehoek FAO@LIVERPOOL.AC.UK
Department of Computer Science, University of Liverpool
Liverpool, United Kingdom

Matthijs T. J. Spaan M.T.J.SPAAN@TUDELFT.NL
Delft University of Technology
Delft, The Netherlands

Bas Terwijn B.TERWIJN@UVA.NL
Informatics Institute, University of Amsterdam
Amsterdam, The Netherlands

Philipp Robbel ROBBEL@ALUM.MIT.EDU
Media Lab, Massachusetts Institute of Technology
Cambridge, MA, USA

João V. Messias JMESSIAS@UVA.NL
Informatics Institute, University of Amsterdam
Amsterdam, The Netherlands

Editor: Alexandre Gramfort

Abstract

This article describes the Multiagent Decision Process (MADP) Toolbox, a software library to support planning and learning for intelligent agents and multiagent systems in uncertain environments. Key features are that it supports partially observable environments and stochastic transition models; has unified support for single- and multiagent systems; provides a large number of models for decision-theoretic decision making, including one-shot and sequential decision making under various assumptions of observability and cooperation, such as Dec-POMDPs and POSGs; provides tools and parsers to quickly prototype new problems; provides an extensive range of planning and learning algorithms for single- and multiagent systems; is released under the GNU GPL v3 license; and is written in C++ and designed to be extensible via the object-oriented paradigm.

Keywords: software, decision-theoretic planning, reinforcement learning, multiagent systems

1. Introduction

Decision making is a core topic of research in the fields of artificial intelligence and machine learning. Especially research on planning and learning in stochastic, partially observable and/or multiagent systems (MASs) has received much attention in the last decade, since these settings hold great promise to deal with challenging decision-making problems encountered in the real world. For instance, *Markov decision processes (MDPs)* can be used for aircraft collision avoidance (Kochenderfer and Chryssanthacopoulos, 2011), *partially observable MDPs (POMDPs)* may enable active perception or certain robotics applications (Kaelbling et al., 1998; Spaan, 2012; Hsu et al., 2008), *decentralized POMDPs (Dec-POMDPs)* (Bernstein et al., 2000; Oliehoek and Amato, 2016) enable reasoning about the behavior of a team of robots or other decision makers acting on local information (Amato et al., 2015), and *partially observable stochastic games (POSGs)* allow for representing situations with self-interested agents (Hansen et al., 2004).

©2017 Frans A. Oliehoek, Matthijs T. J. Spaan, Bas Terwijn, Philipp Robbel and João V. Messias.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-156.html.

All these models are closely related and instances of what we refer to as *multiagent decision processes (MADPs)* (Oliehoek and Amato, 2016). While many software libraries are available for planning or learning in specific sub-classes of MADPs—e.g., there are many toolboxes focusing on single-agent, fully observable reinforcement learning—no comprehensive libraries are available that support the more complex partially observable multiagent settings. The MADP Toolbox aims to fill this void by providing the building blocks for developing planning and learning algorithms for existing and novel instances of MADPs. Some important features of the toolbox are:

Toolkit and Library. It can be used in two ways: as a toolbox containing a number of planning and learning algorithms and as a software library to develop one's own models and algorithms.

Modular design. It is object oriented, making it easy to extend to new problem classes, and implement new planning and learning algorithms.

Flexible. It was designed for multiagent problems, providing support for mixing different types of agents. It is also suitable for single-agent problems, which are just a special case.

Feature rich. It includes many code-intensive features, such as support for factored models (including variable elimination and max-sum optimization), for basic inference in partially observable settings, and for simulations of MASs under different communication assumptions. Finally, it comes equipped with a large number of state-of-the-art methods particularly for (Dec-)POMDPs.

Connected. It reads a number of problem formats: ProbModelXML (Arias et al., 2012), `.pomdp` file format, `.dpomdp` file format, and exports SPUDD (Hoey et al., 1999) format.

Documented. It comes with fairly extensive user and developer documentation. This is complemented by an API reference, as well as a user email list.

2. Supported Problem Classes and Algorithms

Here we give a concise overview of the different MADP frameworks that are currently supported by the toolbox along with implemented algorithms.[1] The MADP Toolbox is designed to support both *planning*, i.e., settings where the model is given in advance, and reinforcement learning, i.e., settings where no such model is available, but the agents *learn* to interact during simulations.

Decentralized POMDPs. One of the most general models that the toolbox represents is the Dec-POMDP. Formally, a Dec-POMDP is a tuple $\langle \mathcal{D}, \mathcal{S}, \boldsymbol{\mathcal{A}}, T, R, \boldsymbol{\mathcal{O}}, O, h, b^0 \rangle$, where $\mathcal{D} = \{1, \ldots, n\}$ is the set of n agents, $\mathcal{S}$ is a set of states s, $\boldsymbol{\mathcal{A}}$ is the set of joint actions $\boldsymbol{a} = \langle a_1, \ldots a_n \rangle$, T is the transition function that specifies $\Pr(s^{t+1}|s^t, \boldsymbol{a}^t)$, $R(s, \boldsymbol{a})$ is the immediate reward function, $\boldsymbol{\mathcal{O}}$ is the set of joint observations $\boldsymbol{o} = \langle o_1, \ldots, o_n \rangle$, O is the observation function: $\Pr(\boldsymbol{o}^{t+1}|\boldsymbol{a}^t, s^{t+1})$, h is the horizon of the problem, $b^0 \in \triangle(\mathcal{S})$ is the initial state distribution at time $t = 0$. The typical goal of the agents is to optimize the expected (possibly discounted) cumulative reward. The toolbox supports a range of solution methods for Dec-POMDPs, including JESP (Nair et al., 2003), DP-LPC (Boularias and Chaib-draa, 2008), (Generalized) Multiagent A* (GMAA) and variants (Szer et al., 2005; Oliehoek et al., 2013), and direct CE (DICE) policy search (Oliehoek et al., 2008a). In general, GMAA-ICE is the fastest optimal solution method included, while DICE is the most scalable approximate method for Dec-POMDPs in the toolbox. The toolbox also includes methods to deal with 1-step delayed communication (Oliehoek et al., 2008b).

POMDPs. A Dec-POMDP with just a single agent or with instantaneous communication reduces to a POMDP (referred to as a *multiagent POMDP* in the latter case). The MADP Toolbox includes a number of solution methods for these: Perseus (Spaan and Vlassis, 2005), Monahan's algorithm (Monahan, 1982), incremental pruning (Cassandra et al., 1997) with accelerated vector pruning (Walraven and Spaan, 2017).

1. For more details on these frameworks, please see, e.g., Oliehoek and Amato (2016).

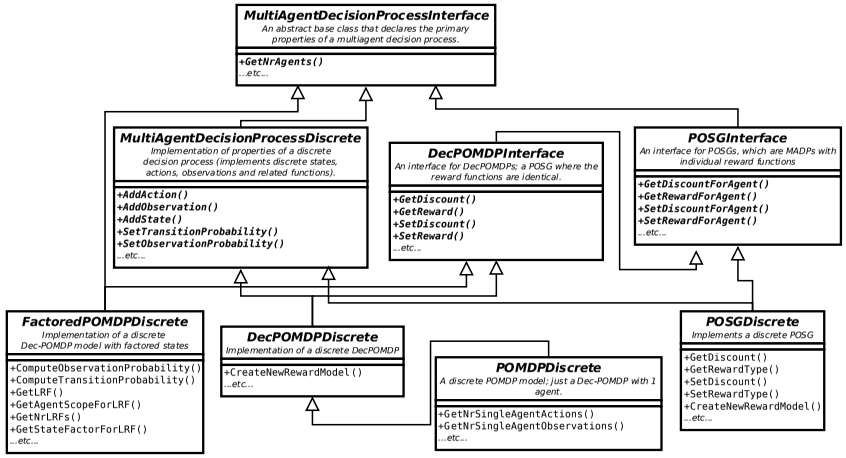


Figure 1: A simplified view of the object-oriented approach to representing problems. Illustrated is how both DecPOMDPDiscrete and POSGDiscrete inherit the implementation of states, actions, observations, etc. from MultiagentDecisionProcessDiscrete. In contrast, FactoredDecPOMDPDiscrete does not inherit this implementation.

MDPs. When the problem additionally is fully observable, the model simplifies further to the standard MDP. For these settings, the toolbox provides algorithms such as value iteration and policy iteration, as well as a GPU accelerated version of the latter. Additionally, there is a Q-learner agent and an agent performing on-line MDP planning.

Factored Models. Since the aforementioned models can be quite large for real problems, compact specification is an important issue. The toolbox includes support for compact *factored* representations (e.g., Oliehoek et al., 2008c, and references therein). For instance, it includes the GMAA-ELSI factored Dec-POMDP solver (Oliehoek et al., 2008c) and exports to SPUDD (Hoey et al., 1999).

Inheritance and Extensibility. The toolbox is designed to be extensible via the object-oriented paradigm. To illustrate this Figure 1 gives a (slightly simplified) illustration of how the discussed problem classes inherit from one another. The figure also shows that the toolbox support POSGs, which generalize Dec-POMDPs to include individual payoff functions.

Additional Functionality. A key functionality that the toolbox provides is the ability to conveniently specify MADPs. Three methods currently exist: 1) using Anthony Cassandra's plain text `.pomdp` file format or a multiagent `.dpomdp` extension, 2) using the graphical OpenMarkov[2] editor for ProbModelXML files, or 3) writing a C++ class that implements the appropriate interface.

The MADP Toolbox also provides a collection of support classes that represent histories, beliefs, value functions, and policies, etc., as well as some algorithms to perform belief updates and approximate inference. Additionally, as part of GMAA, the toolbox also provides classes to represent and solve collaborative (Graphical) Bayesian games (Oliehoek et al., 2008b, 2012).

2. `http://www.openmarkov.org`

Name	Type	MAS	PO	Language	URL
APPL	planning	–	√	C++	`http://bigbird.comp.nus.edu.sg/pmwiki/farm/appl/`
BURLAP	planning/RL	√	–	Java	`http://burlap.cs.brown.edu`
AI-Toolbox	planning/RL	–	√	C++	`https://github.com/Svalorzen/AI-Toolbox`
RLlib	RL	–	–	C++	`https://github.com/HerveFrezza-Buet/RLlib`
RLPy	RL	–	–	Python	`https://bitbucket.org/rlpy/rlpy`
MARL Toolbox	RL	√	–	Matlab	`http://busoniu.net/repository.php`
PyBrain	ML/RL	–	–	Python	`http://pybrain.org`

Table 1: Comparison to related software and toolboxes

3. Technical Details

The software can be used in two ways: as a toolbox and as a library. The fact that it can be used as a library provides flexibility to, for instance, support the deployment and real-time execution of decision-theoretic control policies for autonomous agents (such as the markov_decision_making ROS package mentioned below). Since it has been implemented in C++ and is organized according to the object-oriented paradigm, it is easy to extend. Furthermore, as the toolbox aims to support core planning tasks, C++ provides the additional advantage that it is very fast for such computationally expensive routines.

The MADP Toolbox is released under the GNU GPL v3 license and can be downloaded from `https://github.com/MADPToolbox/MADP`. It is developed under Debian GNU/Linux, has been tested under various recent Linux distributions and also supports Mac OS X. In order to provide reliability, the toolbox includes a basic set of tests which cover most of the included functionality. MADP Toolbox 0.4.1 adheres to the C++98 standard and is relatively self-contained: required Boost header files and required libraries are included. It comes with extensive documentation both in form of a user/developer manual as well as API documentation (also available online).

4. Related Software

Many toolboxes exists for decision making, but few deal with MASs or partially observable settings. Closely related software and toolboxes for planning, reinforcement learning (RL) and machine learning (ML) applications are shown in Table 1. The columns 'MAS' and 'PO' indicate whether the respective toolbox includes algorithms for multiagent or partially observable settings, respectively. Closest in scope is BURLAP, which does not support partially observable multiagent settings.

The MADP software distribution uses functionality from and therefore includes boost and libDAI (Mooij, 2010). The MADP Toolbox is the back-end of a number of solution methods on the THINC lab solver page.[3] It is also used in the markov_decision_making ROS package in order to parse model files and query agent policies in real time.[4]

Acknowledgments

We are grateful to Joris Mooij for allowing to include libDAI and to many others that made contributions. The MADP Toolbox has been developed over the course of many years, and has thus been supported by various funding agencies including: The Dutch Ministry of Economic Affairs, AFOSR, NWO, FP7 Marie Curie, FCT. F.O. was also affiliated with the University of Amsterdam, supported by NWO Veni grant #639.021.336, during the writing of this article.

3. `http://lhotse.cs.uga.edu/pomdp/`
4. `http://wiki.ros.org/markov_decision_making`

References

C. Amato, G. D. Konidaris, G. Cruz, C. A. Maynor, J. P. How, and L. P. Kaelbling. Planning for decentralized control of multiple robots under uncertainty. In *IEEE Int. Conference on Robotics and Automation*, pages 1241–1248, 2015.

M. Arias, F. J. Díez, M. A. Palacios-Alonso, M. Yebra, and J. Fernández. POMDPs in OpenMarkov and ProbModelXML. In *Multi-Agent Sequential Decision Making in Uncertain Domains*, 2012.

D. S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of Markov decision processes. In *Uncertainty in Artificial Intelligence*, pages 32–37, 2000.

A. Boularias and B. Chaib-draa. Exact dynamic programming for decentralized POMDPs with lossless policy compression. In *Int. Conference on Automated Planning and Scheduling*, pages 20–28, 2008.

A. Cassandra, M. L. Littman, and N. L. Zhang. Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes. In *Uncertainty in Artificial Intelligence*, pages 54–61, 1997.

E. A. Hansen, D. S. Bernstein, and S. Zilberstein. Dynamic programming for partially observable stochastic games. In *National Conference on Artificial Intelligence*, pages 709–715, 2004.

J. Hoey, R. St-Aubin, A. J. Hu, and C. Boutilier. SPUDD: Stochastic planning using decision diagrams. In *Uncertainty in Artificial Intelligence*, pages 279–288, 1999.

D. Hsu, W. S. Lee, and N. Rong. A point-based POMDP planner for target tracking. In *IEEE Int. Conference on Robotics and Automation*, pages 2644–2650, 2008.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

M. J. Kochenderfer and J. P. Chryssanthacopoulos. Collision avoidance using partially controlled Markov decision processes. In *Int. Conference on Agents and Artificial Intelligence*, pages 86–100, 2011.

G. E. Monahan. A survey of partially observable Markov decision processes: theory, models and algorithms. *Management Science*, 28(1):1–16, 1982.

J. M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11:2169–2173, 2010.

R. Nair, M. Tambe, M. Yokoo, D. V. Pynadath, and S. Marsella. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *Int. Joint Conference on Artificial Intelligence*, pages 705–711, 2003.

F. A. Oliehoek and C. Amato. *A Concise Introduction to Decentralized POMDPs*. Springer Briefs in Intelligent Systems. Springer, 2016.

F. A. Oliehoek, J. F. Kooi, and N. Vlassis. The cross-entropy method for policy search in decentralized POMDPs. *Informatica*, 32:341–357, 2008a.

F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008b.

F. A. Oliehoek, M. T. J. Spaan, S. Whiteson, and N. Vlassis. Exploiting locality of interaction in factored Dec-POMDPs. In *Int. Conference on Autonomous Agents and Multiagent Systems*, pages 517–524, 2008c.

F. A. Oliehoek, S. Whiteson, and M. T. J. Spaan. Exploiting structure in cooperative Bayesian games. In *Uncertainty in Artificial Intelligence*, pages 654–664, 2012.

F. A. Oliehoek, M. T. J. Spaan, C. Amato, and S. Whiteson. Incremental clustering and expansion for faster optimal planning in decentralized POMDPs. *Journal of Artificial Intelligence Research*, 46:449–509, 2013.

M. T. J. Spaan. Partially observable Markov decision processes. In M. Wiering and M. van Otterlo, editors, *Reinforcement Learning: State of the Art*, pages 387–414. Springer Verlag, 2012.

M. T. J. Spaan and N. Vlassis. Perseus: Randomized point-based value iteration for POMDPs. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

D. Szer, F. Charpillet, and S. Zilberstein. MAA*: A heuristic search algorithm for solving decentralized POMDPs. In *Uncertainty in Artificial Intelligence*, pages 576–583, 2005.

E. Walraven and M. T. J. Spaan. Accelerated vector pruning for optimal POMDP solvers. In *AAAI Conference on Artificial Intelligence*, pages 3672–3678, 2017.

Journal of Machine Learning Research 18 (2017) 1-50 Submitted 10/14; Revised 11/15; Published 8/17

A Survey of Algorithms and Analysis for Adaptive Online Learning

H. Brendan McMahan
mcmahan@google.com
Google, Inc.
651 N 34th St
Seattle, WA 98103 USA

Editor: Shie Mannor

Abstract

We present tools for the analysis of Follow-The-Regularized-Leader (FTRL), Dual Averaging, and Mirror Descent algorithms when the regularizer (equivalently, prox-function or learning rate schedule) is chosen adaptively based on the data. Adaptivity can be used to prove regret bounds that hold on every round, and also allows for data-dependent regret bounds as in AdaGrad-style algorithms (e.g., Online Gradient Descent with adaptive per-coordinate learning rates). We present results from a large number of prior works in a unified manner, using a modular and tight analysis that isolates the key arguments in easily re-usable lemmas. This approach strengthens previously known FTRL analysis techniques to produce bounds as tight as those achieved by potential functions or primal-dual analysis. Further, we prove a general and exact equivalence between adaptive Mirror Descent algorithms and a corresponding FTRL update, which allows us to analyze Mirror Descent algorithms in the same framework. The key to bridging the gap between Dual Averaging and Mirror Descent algorithms lies in an analysis of the FTRL-Proximal algorithm family. Our regret bounds are proved in the most general form, holding for arbitrary norms and non-smooth regularizers with time-varying weight.

Keywords: online learning, online convex optimization, regret analysis, adaptive algorithms, follow-the-regularized-leader, mirror descent, dual averaging

1. Introduction

We consider the problem of online convex optimization over a series of rounds $t \in \{1, 2, \dots\}$. On each round the algorithm selects a point (e.g., a predictor or an action) $x_t \in \mathbb{R}^n$, and then an adversary selects a convex loss function f_t, and the algorithm suffers loss $f_t(x_t)$. The goal is to minimize

$$\text{Regret}_T(x^*, f_t) \equiv \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*), \tag{1}$$

©2017 H. Brendan McMahan.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/14-428.html.

Algorithm 1 General Template for Adaptive FTRL

Parameters: Scheme for selecting convex r_t s.t. $\forall x,\ r_t(x) \geq 0$ for $t = 0, 1, 2, \dots$
$x_1 \leftarrow \arg\min_{x \in \mathbb{R}^n}\ r_0(x)$
for $t = 1, 2, \dots$ **do**
  Observe convex loss function $f_t : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$
  Incur loss $f_t(x_t)$
  Choose incremental convex regularizer r_t, possibly based on $f_1, \dots f_t$
  Update

$$x_{t+1} \leftarrow \arg\min_{x \in \mathbb{R}^n} \sum_{s=1}^{t} f_s(x) + \sum_{s=0}^{t} r_s(x)$$

end for

the difference between the algorithm's loss and the loss of a fixed point x^*, potentially chosen with full knowledge of the sequence of f_t up through round T. When the functions f_t and round T are clear from the context we write $\text{Regret}(x^*)$. The "adversary" choosing the f_t need not be malicious, for example the f_t might be drawn from a distribution. The name "online convex optimization" was introduced by Zinkevich (2003), though the setting was introduced earlier by Gordon (1999). When a particular set of comparators $\mathcal{X}$ is fixed in advance, one is often interested in $\text{Regret}(\mathcal{X}) \equiv \sup_{x^* \in \mathcal{X}} \text{Regret}(x^*)$; since $\mathcal{X}$ is often a norm ball, frequently we bound $\text{Regret}(x^*)$ by a function of $\|x^*\|$.

Online algorithms with good regret bounds (that is, bounds that are sublinear in T) can be used for a wide variety of prediction and learning tasks (Cesa-Bianchi and Lugosi, 2006; Shalev-Shwartz, 2012). The case of online logistic regression, where one predicts the probability of a binary outcome, is typical. Here, on each round a feature vector $a_t \in \mathbb{R}^n$ arrives, and we make a prediction $p_t = \sigma(a_t \cdot x_t) \in (0, 1)$ using the current model coefficients $x_t \in \mathbb{R}^n$, where $\sigma(z) = 1/(1 + e^{-z})$. The adversary then reveals the true outcome $y_t \in \{0, 1\}$, and we measure loss with the negative log-likelihood, $\ell(p_t, y_t) = -y_t \log p_t - (1 - y_t) \log(1 - p_t)$. We encode this problem as online convex optimization by taking $f_t(x) = \ell(\sigma(a_t \cdot x), y_t)$; these f_t are in fact convex. Linear Support Vector Machines (SVMs), linear regression, and many other learning problems can be encoded in a similar manner; Shalev-Shwartz (2012) and many of the other works cited here contain more details and examples.

We consider the family of Follow-The-Regularized-Leader (FTRL, or FoReL) algorithms as shown in Algorithm 1 (Shalev-Shwartz, 2007; Shalev-Shwartz and Singer, 2007; Rakhlin, 2008; McMahan and Streeter, 2010; McMahan, 2011). Shalev-Shwartz (2012) and Hazan (2015) provide a comprehensive survey of analysis techniques for non-adaptive members of this algorithm family, where the regularizer is fixed for all rounds and chosen with knowledge of the horizon T. In this survey,

we allow the regularizer to change adaptively. Given a sequence of incremental regularization functions $r_0, r_1, r_2, \ldots$, we consider the algorithm that selects

$$x_1 \in \arg\min_{x \in \mathbb{R}^n} r_0(x)$$
$$x_{t+1} = \arg\min_{x \in \mathbb{R}^n} f_{1:t}(x) + r_{0:t}(x) \qquad \text{for } t = 1, 2, \ldots, \tag{2}$$

where we use the compressed summation notation $f_{1:t}(x) = \sum_{s=1}^{t} f_s(x)$ (we also use this notation for sums of scalars or vectors). The argmin in Eq. (2) is over all $\mathbb{R}^n$, but it is often necessary to constrain the selected points x_t to a convex feasible set $\mathcal{X}$. This can be accomplished in our framework by including the indicator function $I_{\mathcal{X}}$ as a term in r_0 ($I_{\mathcal{X}}$ is a convex function defined by $I_{\mathcal{X}}(x) = 0$ for $x \in \mathcal{X}$ and ∞ otherwise); details are given in Section 2.4. The algorithms we consider are adaptive in that each r_t can be chosen based on $f_1, f_2, \ldots, f_t$. For convenience, we define functions h_t by

$$h_0(x) = r_0(x)$$
$$h_t(x) = f_t(x) + r_t(x) \qquad \text{for } t = 1, 2, \ldots$$

so $x_{t+1} = \arg\min_x h_{0:t}(x)$. Generally we will assume the f_t are convex, and the r_t are chosen so that $r_{0:t}$ (or $h_{0:t}$) is strongly convex for all t, e.g., $r_{0:t}(x) = \frac{1}{2\eta_t}\|x\|_2^2$ (Sections 2.3 and 4.2 review important definitions and results from convex analysis).

FTRL algorithms generalize the Follow-The-Leader (FTL) approach (Hannan, 1957; Kalai and Vempala, 2005), which selects $x_{t+1} = \arg\min_x f_{1:t}(x)$. FTL can provide sublinear regret in the case of strongly convex functions (as we will show), but for general convex functions additional regularization is needed.

Adaptive regularization can be used to construct practical algorithms that provide regret bounds that hold on all rounds T, rather than only on a single round T which is chosen in advance. The framework is also particularly suitable for analyzing AdaGrad-style algorithms that adapt their regularization or norms based on the observed data, for example those of McMahan and Streeter (2010) and Duchi et al. (2010a, 2011). This approach leads to regret bounds that depend on the actual observed sequence of functions f_t (usually via $\nabla f_t(x_t)$), rather than purely worst-case bounds. These tighter bounds translate to much better performance in practice, especially for high-dimensional but sparse problem (e.g., bag-of-words feature vectors). Examples of such algorithms are analyzed in Sections 3.4 and 3.5.

We also study Mirror Descent algorithms, for example updates like

$$x_{t+1} = \arg\min_{x \in \mathcal{X}} \nabla f_t(x_t) \cdot x + \lambda\|x\|_1 + \frac{1}{2\eta_t}\|x - x_t\|_2^2$$

where η_t is an adaptive non-increasing learning rate. This update generalizes Online Gradient Descent with a non-smooth regularization term; Mirror Descent also encompasses the use of an arbitrary Bregman divergence in place of the $\|\cdot\|_2^2$ penalty

above. We will discuss this family of algorithms at length in Section 6. In fact, Mirror Descent algorithms can be expressed as particular members of the FTRL family, though generally not the most natural ones. In particular, since the state maintained by Mirror Descent is essentially only the current feasible point x_t, we will see that Mirror Descent algorithms are forced to linearize penalties like $\lambda\|x\|_1$ from previous rounds, while the more natural FTRL algorithms can keep these terms in closed form, leading to practical advantages such as producing sparser models when L_1 regularization is used.

While we focus on online algorithms and regret bounds, the development of many of the algorithms considered rests heavily on work in general convex optimization and stochastic optimization. As a few starting points, we refer the reader to Nemirovsky and Yudin (1983) and Nesterov (2004, 2007). Going the other way, the algorithms presented here can be applied to batch optimization problems of the form

$$\arg\min_{x \in \mathbb{R}^n} F(x) \qquad \text{where} \qquad F(x) \equiv \sum_{t=1}^{T} f_t(x) \tag{3}$$

by running the online algorithm for one or more passes over the set of f_t and returning a suitable point (usually the last x_t or an average of past x_t). Using online-to-batch conversion techniques (e.g., Cesa-Bianchi et al. (2004), Shalev-Shwartz (2012, Chapter 5)), one can convert the regret bounds given here to convergence bounds for the batch problem. Many state-of-the-art algorithms for batch optimization over very large datasets can be analyzed in this fashion.

Outline In Section 2, we elaborate on the family of algorithms encompassed by the update of Eq. (2). We then state two regret bounds, Theorems 1 and 2, which are flexible enough to cover many known results for general and strongly convex functions; in Section 3 we use them to derive concrete bounds for many standard online algorithms.

In Section 4 we break the analysis of adaptive FTRL algorithms into three main components, which helps to modularize the arguments. In Section 4.1 we prove the *Strong FTRL Lemma* which lets us express the regret through round T as a regularization term on the comparator x^*, namely $r_{0:T}(x^*)$, plus a sum of per-round stability terms. This reduces the problem of bounding regret to that of bounding these per-round terms. In Section 4.2 we review some standard results from convex analysis, and prove lemmas that make bounding the per-round terms relatively straightforward. The general regret bounds are then proved in Section 4.3 as corollaries of these results.

Section 5 considers the special case of a composite objective, where for example $f_t(x) = \ell_t(x) + \Psi(x)$ with ℓ_t is a smooth loss on the t'th training example and Ψ is a possibly non-smooth regularizer (e.g., $\Psi(x) = \|x\|_1$). Finally, Section 6 proves the

Algorithm 2 General Template for Adaptive Linearized FTRL

Parameters: Scheme for selecting convex r_t s.t. $\forall x,\ r_t(x) \geq 0$ for $t = 0, 1, 2, \ldots$
$z \leftarrow \mathbf{0} \in \mathbb{R}^n$ *//Maintains* $g_{1:t}$
$x_1 \leftarrow \arg\min_{x \in \mathbb{R}^n}\ z \cdot x + r_0(x)$
for $t = 1, 2, \ldots$ **do**
  Select x_t, observe loss function f_t, incur loss $f_t(x_t)$
  Compute a subgradient $g_t \in \partial f_t(x_t)$
  Choose incremental convex regularizer r_t, possibly based on $g_1, \ldots, g_t$
  $z \leftarrow z + g_t$
  $x_{t+1} \leftarrow \arg\min_{x \in \mathbb{R}^n}\ z \cdot x + r_{0:t}(x)$ *//Often solved in closed form*
end for

equivalence of an arbitrary adaptive Mirror Descent algorithm and a certain FTRL algorithm, and uses this to prove regret bounds for Mirror Descent.

New Contributions The principal goal of this work is to provide a useful survey of central results in the analysis of adaptive algorithms for online convex optimization; whenever possible we provide precise references to earlier results that we re-prove or strengthen. Achieving this goal in a concise fashion requires some new results, which we summarize here.

The FTRL style of analysis is both modular and intuitive, but in previous work resulted in regret bounds that are not the tightest possible; we remedy this by introducing the Strong FTRL Lemma in Section 4.1. This also relates the FTRL analysis technique to the primal-dual style of analysis.

By analyzing both FTRL-Proximal algorithms (introduced in the next section) and Dual Averaging algorithms in a unified manner, it is much easier to contrast the strengths and weaknesses of each approach. This highlights a technical but important "off-by-one" difference between the two families in the adaptive setting, as well as an important difference when the algorithm is unconstrained (any $x_t \in \mathbb{R}^n$ is feasible).

Perhaps the most significant new contribution is given in Section 6, where we show that Mirror Descent algorithms (including adaptive algorithms for composite objectives) are in fact particular instances of the FTRL-Proximal algorithm schema, and can be analyzed using the general tools developed for the analysis of FTRL.

2. The FTRL Algorithm Family and General Regret Bounds

We begin by considering two important dimensions in the space of FTRL algorithms. First, the algorithm designer has significant flexibility in deciding whether the sum of previous loss functions is optimized exactly as $f_{1:t}(x)$ in Eq. (2), or if the true losses should be replaced by appropriate lower bounds, $\bar{f}_{1:t}(x)$, for computational efficiency.

Second, we consider whether the incremental regularizers r_t are all minimized at a fixed stationary point x_1, or are chosen so they are minimized at the current x_t. After discussing these options, we state general regret bounds.

2.1 Linearization and the Optimization of Lower Bounds

In practice, it may be infeasible to solve the optimization problem of Eq. (2), or even represent it as t becomes sufficiently large. A key point is that we can derive a wide variety of first-order algorithms by linearizing the f_t, and running the algorithm on these linear functions. Algorithm 2 gives the general scheme. For convex f_t, let x_t be defined as above, and let $g_t \in \partial f_t(x_t)$ be a subgradient (e.g., $g_t = \nabla f_t(x_t)$ for differentiable f_t). Convexity implies for any comparator x^*, $f_t(x_t) - f_t(x^*) \leq g_t \cdot (x_t - x^*)$. A key observation of Zinkevich (2003) is that if we let $\bar{f}_t(x) = g_t \cdot x$, then for any algorithm the regret against the functions $\bar{f}_t$ upper bounds the regret against the original f_t:

$$\text{Regret}(x^*, f_t) \leq \text{Regret}(x^*, \bar{f}_t).$$

Note we can construct the functions $\bar{f}_t$ on the fly (after observing x_t and f_t) and then present them to the algorithm.

Thus, rather than solving $x_{t+1} = \arg\min_x f_{1:t}(x) + r_{0:t}(x)$ on each round t, we now solve $x_{t+1} = \arg\min_x g_{1:t} \cdot x + r_{0:t}(x)$. Note that $g_{1:t} \in \mathbb{R}^n$, and we will generally choose the r_t so that $r_{0:t}(x)$ can also be represented in constant space. Thus, we have at least ensured our storage requirements stay constant even as $t \to \infty$. Further, we will usually be able to choose r_t so the optimization with $g_{1:t}$ can be solved in closed form. For example, if we take $r_{0:t}(x) = \frac{1}{2\eta}\|x\|_2^2$ then we can solve $x_{t+1} = \arg\min_x g_{1:t} \cdot x + r_{0:t}(x)$ in closed form, yielding $x_{t+1} = -\eta g_{1:t}$ (that is, this FTRL algorithm is exactly constant learning rate Online Gradient Descent).

However, we will usually state our results in terms of general f_t, since one can always simply take $f_t = \bar{f}_t$ when appropriate. In fact, an important aspect of our analysis is that it does not depend on linearization; our regret bounds hold for the the general update of Eq. (2) as well as applying to linearized variants.

More generally, we can run the algorithm on any $\bar{f}_t$ that satisfy $\bar{f}_t(x_t) - \bar{f}_t(x^*) \geq f_t(x_t) - f_t(x^*)$ for all x^* and have the regret bound achieved for the $\bar{f}$ also apply to the original f. This is generally accomplished by constructing a lower bound $\bar{f}_t$ that is tight at x_t, that is $\bar{f}_t(x) \leq f_t(x)$ for all x and further $\bar{f}_t(x_t) = f_t(x_t)$. A tight linear lower bound is always possible for convex functions, but for example if the f_t are all strongly convex, better algorithms are possible by taking $\bar{f}_t$ to be an appropriate quadratic lower bound.

A more in-depth introduction to the linearization of convex function can be found in Shalev-Shwartz (2012, Sec 2.4). We also note that the idea of replacing the loss function on each round with an appropriate lower bound ("linearization of

convex functions") is distinct from the modeling decision to replace a non-convex loss function (e.g., the zero-one loss for classification) with a convex upper bound (e.g., the hinge loss). This "convexification by surrogate loss" approach is described in detail by (Shalev-Shwartz, 2012, Sec 2.1).

2.2 Regularization in FTRL Algorithms

The term "regularization" can have multiple meanings, and so in this section we clarify the different roles regularization plays in the present work.

We refer to the functions $r_{0:t}$ as regularization functions, with r_t the incremental increase in regularization on round t (we assume $r_t(x) \geq 0$). This is the regularization in the name Follow-The-Regularized-Leader, and these r_t terms should be viewed as part of the algorithm itself—analogous (and in some cases exactly equivalent) to the learning rate schedule in an Online Gradient Descent algorithm, for example. The adaptive choice of these regularizers is the principle topic of the current work. We study two main classes of regularizers:

- In *FTRL-Centered* algorithms, each r_t (and hence $r_{0:t}$) is minimized at a fixed point, $x_1 = \arg\min_x r_0(x)$. An example is Dual Averaging (which also linearizes the losses), where $r_{0:t}$ is called the *prox-function* (Nesterov, 2009).

- In *FTRL-Proximal* algorithms, each incremental regularization function r_t is minimized by x_t, and we call such r_t incremental proximal regularizers.

When we make neither a proximal nor centered assumption on the r_t, we refer to general FTRL algorithms. Theorem 1 (below) allows us to analyze regularization choices that do not fall into either of these two categories, but the Centered and Proximal cases cover the algorithms of practical interest.

There are a number of reasons we might wish to add additional regularization terms to the objective function in the FTRL update. In many cases this is handled immediately by our general theory by grouping the additional regularization terms with either the f_t or the r_t. However, in some cases it will be advantageous to handle this additional regularization more explicitly. We study this situation in detail in Section 5.

2.3 General Regret Bounds

In this section we introduce two general regret bounds that can be used to analyze many different adaptive online algorithms. First, we introduce some additional notation and definitions.

Notation and Definitions An extended-value convex function $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ satisfies

$$\psi(\theta x + (1-\theta)y) \leq \theta\psi(x) + (1-\theta)\psi(y),$$

for $\theta \in (0,1)$, and the domain of ψ is the convex set $\operatorname{dom}\psi \equiv \{x : \psi(x) < \infty\}$ (e.g., Boyd and Vandenberghe (2004, Sec. 3.1.2)); ψ is proper if $\exists x \in \mathbb{R}^n$ s.t. $\psi(x) < +\infty$ and $\forall x \in \mathbb{R}^n, \psi(x) > -\infty$. We refer to extended-value proper convex functions as simply "convex functions."

We write $\partial\psi(x)$ for the subdifferential of ψ at x; a subgradient $g \in \partial\psi(x)$ satisfies

$$\forall y \in \mathbb{R}^n,\ \psi(y) \geq \psi(x) + g \cdot (y - x).$$

The subdifferential $\partial\psi(x)$ for a convex ψ is always non-empty for $x \in \operatorname{int}(\operatorname{dom}\psi)$, and typically non-empty for any $x \in \operatorname{dom}\psi$ for the functions ψ considered in this work; $\partial\psi(x)$ is empty for $x \notin \operatorname{dom}\psi$ (Rockafellar, 1970, Thm. 23.2).

Working with extended convex functions lets us encode constraints seamlessly by using $I_{\mathcal{X}}$, the indicator function on a convex set $\mathcal{X} \subseteq \mathbb{R}^n$ given by

$$I_{\mathcal{X}}(x) = \begin{cases} 0 & x \in \mathcal{X} \\ \infty & \text{otherwise}\ , \end{cases} \tag{4}$$

since $I_{\mathcal{X}}$ is itself an extended convex function. Generally we assume $\mathcal{X}$ is a closed convex set. This approach makes it convenient to write $\arg\min_x$ as shorthand for $\arg\min_{x \in \mathbb{R}^n}$.

A function $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is *σ-strongly convex* w.r.t. a norm $\|\cdot\|$ if for all $x, y \in \mathbb{R}^n$,

$$\forall g \in \partial\psi(x), \quad \psi(y) \geq \psi(x) + g \cdot (y - x) + \tfrac{\sigma}{2}\|y - x\|^2. \tag{5}$$

If some ψ only satisfies Eq. (5) for $x, y \in \mathcal{X}$ for a convex set $\mathcal{X}$, then the function $\psi' = \psi + I_{\mathcal{X}}$ satisfies Eq. (5) for all $x, y \in \mathbb{R}^n$, and so is strongly convex by our definition. Thus, we can work with ψ' without any need to explicitly refer to $\mathcal{X}$.

The *convex conjugate* (or Fenchel conjugate) of an arbitrary function $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is

$$\psi^\star(g) \equiv \sup_x\ g \cdot x - \psi(x). \tag{6}$$

For a norm $\|\cdot\|$, the dual norm is given by

$$\|x\|_\star \equiv \sup_{y : \|y\| \leq 1}\ x \cdot y.$$

It follows from this definition that for any $x, y \in \mathbb{R}^n$, $x \cdot y \leq \|x\|\|y\|_\star$, a generalization of Hölder's inequality. We make heavy use of norms $\|\cdot\|_{(t)}$ that change as a function of the round t; the dual norm of $\|\cdot\|_{(t)}$ is $\|\cdot\|_{(t),\star}$.

Our basic assumptions correspond to the framework of Algorithm 1, which we summarize together with a few technical conditions as follows:

Setting 1 *We consider the algorithm that selects points according to Eq.* (2) *based on convex r_t that satisfy $r_t(x) \geq 0$ for $t \in \{0, 1, 2, \dots\}$, against a sequence of convex*

loss functions $f_t : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$. Further, letting $h_{0:t} = r_{0:t} + f_{1:t}$ we assume $\operatorname{dom} h_{0:t}$ is non-empty. Recalling $x_t = \arg\min_x h_{0:t-1}(x)$, we further assume $\partial f_t(x_t)$ is non-empty.

The minor technical assumptions made here do not rule out any practical applications. We can now introduce the theorems which will be our main focus. The first will typically be applied to FTRL-Centered algorithms such as Dual Averaging:

Theorem 1 General FTRL Bound *Consider Setting 1, and suppose the r_t are chosen such that $h_{0:t} + f_{t+1} = r_{0:t} + f_{1:t+1}$ is 1-strongly-convex w.r.t. some norm $\|\cdot\|_{(t)}$. Then, for any $x^* \in \mathbb{R}^n$ and for any $T > 0$,*

$$\text{Regret}_T(x^*) \leq r_{0:T-1}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_{(t-1),\star}.$$

Our second theorem handles proximal regularizers:

Theorem 2 FTRL-Proximal Bound *Consider Setting 1, and further suppose the r_t are chosen such that $h_{0:t} = r_{0:t} + f_{1:t}$ is 1-strongly-convex w.r.t. some norm $\|\cdot\|_{(t)}$, and further the r_t are proximal, that is x_t is a minimizer of r_t. Then, choosing any $g_t \in \partial f_t(x_t)$ on each round, for any $x^* \in \mathbb{R}^n$ and for any $T > 0$,*

$$\text{Regret}_T(x^*) \leq r_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_{(t),\star}.$$

We state these bounds in terms of strong convexity conditions on $h_{0:t}$ in order to also cover the case where the f_t are themselves strongly convex. In fact, if each f_t is strongly convex, then we can choose $r_t(x) = 0$ for all t, and Theorems 1 and 2 produce *identical* bounds (and algorithms).[1] When it is not known a priori whether the loss functions f_t are strongly convex, the r_t can be chosen adaptively to add only as much strong convexity as needed, following Bartlett et al. (2007). On the other hand, when the f_t are not strongly convex (e.g., linear), a sufficient condition for both theorems is choosing the r_t such that $r_{0:t}$ is 1-strongly-convex w.r.t. $\|\cdot\|_{(t)}$.

It is worth emphasizing the "off-by-one" difference between Theorems 1 and 2 in this case: we can choose r_t based on g_t, and when using proximal regularizers, this lets us influence the norm we use to measure g_t in the final bound (namely the $\|g_t\|^2_{(t),\star}$ term); this is not possible using Theorem 1, since we have $\|g_t\|^2_{(t-1),\star}$. This makes constructing AdaGrad-style adaptive learning rate algorithms for FTRL-Proximal easier (McMahan and Streeter, 2010), whereas with FTRL-Centered algorithms one must start with slightly more regularization. We will see this in more detail in Section 3.

1. To see this, note in Theorem 1 the norm in $\|g_t\|_{(t-1),\star}$ is determined by the strong convexity of $f_{1:t}$, and in Theorem 2 the norm in $\|g_t\|_{(t),\star}$ is again determined by the strong convexity of $f_{1:t}$.

Theorem 1 leads immediately to a bound for Dual Averaging algorithms (Nesterov, 2009), including the Regularized Dual Averaging (RDA) algorithm of Xiao (2009), and its AdaGrad variant (Duchi et al., 2011) (in fact, this statement is equivalent to Duchi et al. (2011, Prop. 2) when we assume the f_t are not strongly convex). As in these cases, Theorem 1 is usually applied to FTRL-Centered algorithms where x_1 (often the origin) is a global minimizer of $r_{0:t}$ for each t. The theorem does not require this; however, such a condition is usually necessary to bound $r_{0:T-1}(x^*)$ and hence $\text{Regret}(x^*)$ in terms of $\|x^*\|$.

Less general versions of these theorems often assume that each $r_{0:t}$ is α_t-strongly-convex with respect to a fixed norm $\|\cdot\|$. Our results include this as a special case, see Section 3 and Lemma 3 in particular.

Non-Adaptive Algorithms These theorems can also be used to analyze non-adaptive algorithms. If we choose $r_0(x)$ to be a fixed non-adaptive regularizer (perhaps chosen with knowledge of T) that is 1-strongly convex w.r.t. $\|\cdot\|$, and all $r_t(x) = 0$ for $t \geq 1$, then we have $\|x\|_{(t),\star} = \|x\|_\star$ for all t, and so both theorems provide the identical statement

$$\text{Regret}(x^*) \leq r_0(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|^2_\star. \tag{7}$$

This matches Shalev-Shwartz (2012, Theorem 2.11), though we improve by a constant factor due to the use of the Strong FTRL Lemma.

2.4 Incorporating a Feasible Set

We have introduced the FTRL update as an unconstrained optimization over $x \in \mathbb{R}^n$. For many learning problems, where x_t is a vector of model parameters, this may be fine, but in other applications we need to enforce constraints. These could correspond to budget constraints, structural constraints like $\|x_t\|_2 \leq R$ or $\|x_t\|_1 \leq R_1$, a constraint that x_t is a flow on a graph, or that x_t is a probability distribution. In all of these cases, this amounts to the constraint that $x_t \in \mathcal{X}$ where $\mathcal{X}$ is a suitable convex feasible set. Further, for FTRL-Proximal algorithms a constraint like $\|x_t\|_2 \leq R$ is generally needed in order to bound $r_{0:T}(x^*)$; see Section 3.3.

Such constraints can be addressed immediately in our setting by adding the additional regularizer $I_{\mathcal{X}}$ to r_0, based on the equivalence

$$\arg\min_{x \in \mathbb{R}^n} f_{1:t}(x) + r_{0:t}(x) + I_{\mathcal{X}}(x) \quad = \quad \arg\min_{x \in \mathcal{X}} f_{1:t}(x) + r_{0:t}(x).$$

Further, if $r_{0:t}$ satisfies the conditions of Theorem 1, then so does $r_{0:t} + I_{\mathcal{X}}$. Similarly, for Theorem 2, adding $I_{\mathcal{X}}$ to r_0 will generally still produce a scheme where r_t has x_t as a minimizer, and so the theorem will still apply. We apply this technique to specific algorithms in Section 3.

Note that while the theorems still apply, the regret bounds change in an important way, since $I_{\mathcal{X}}(x^*)$ now appears in the regret bound: that is, if Theorem 1 on functions $r_0, r_1, \ldots,$ gives a bound $\text{Regret}(x^*) \leq r_{0:T-1}(x^*) + \frac{1}{2}\sum_{t=1}^T \|g_t\|^2_{(t-1),\star}$, then the version constrained to select from $\mathcal{X}$ by adding $I_{\mathcal{X}}$ to r_0 has regret bound

$$\text{Regret}_T(x^*) \leq I_{\mathcal{X}}(x^*) + r_{0:T-1}(x^*) + \frac{1}{2}\sum_{t=1}^T \|g_t\|^2_{(t-1),\star}.$$

This bound is *vacuous* for $x^* \notin \mathcal{X}$, but identical to the unconstrained bound for $x^* \in \mathcal{X}$. This makes sense: one can show that any online algorithm constrained to select $x_t \in \mathcal{X}$ cannot in general hope to have sublinear regret against some $x^* \notin \mathcal{X}$. Thus, if we believe some $x^* \notin \mathcal{X}$ could perform very well, incorporating the constraint $x_t \in \mathcal{X}$ is a significant sacrifice that should only be made if external considerations really require it.

3. Application to Specific Algorithms and Settings

Before proving these theorems, we apply them to a variety of specific algorithms. We will use the following lemma, which collects some facts for the sequence of incremental regularizers r_t. These claims are immediate consequences of the relevant definitions.

Lemma 3 *Consider a sequence of r_t as in Setting 1. Then, since $r_t(x) \geq 0$, we have $r_{0:t}(x) \geq r_{0:t-1}(x)$, and so $r^\star_{0:t}(x) \leq r^\star_{0:t-1}(x)$, where $r^\star_{0:t}$ is the convex-conjugate of $r_{0:t}$. If each r_t is σ_t-strongly convex w.r.t. a norm $\|\cdot\|$ for $\sigma_t \geq 0$, then, $r_{0:t}$ is $\sigma_{0:t}$-strongly convex w.r.t. $\|\cdot\|$, or equivalently, is 1-strongly-convex w.r.t. $\|x\|_{(t)} = \sqrt{\sigma_{0:t}}\|x\|$, which has dual norm $\|x\|_{(t),\star} = \frac{1}{\sqrt{\sigma_{0:t}}}\|x\|$.*

For reasons that will become clear, it is natural to define a learning rate schedule η_t to be the inverse of the cumulative strong convexity,

$$\eta_t = \frac{1}{\sigma_{0:t}}.$$

In fact, in many cases it will be more natural to define the learning rate schedule, and infer the sequence of σ_t,

$$\sigma_t = \frac{1}{\eta_t} - \frac{1}{\eta_{t-1}},$$

with $\sigma_0 = \frac{1}{\eta_0}$.

For simplicity, in this section we assume the loss functions have already been linearized, that is, $f_t(x) = g_t \cdot x$, unless otherwise stated. Figure 1 summarizes most of the FTRL algorithms analyzed in this section.

3.1 Constant Learning Rate Online Gradient Descent

As a warm-up, we first consider a non-adaptive algorithm, unconstrained constant learning rate Online Gradient Descent, which selects $x_1 = 0$ and thereafter

$$x_{t+1} = x_t - \eta g_t, \tag{8}$$

where the parameter $\eta > 0$ is the learning rate. Iterating this update, we see $x_{t+1} = -\eta g_{1:t}$. There is a close connection between Online Gradient Descent and FTRL, which we will use to analyze this algorithm. If we take FTRL with $r_0(x) = \frac{1}{2\eta}\|x\|_2^2$ and $r_t(x) = 0$ for $t \geq 1$, we have the update

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + \frac{1}{2\eta}\|x\|_2^2, \tag{9}$$

which we can solve in closed form to see $x_{t+1} = -\eta g_{1:t}$ as well. Applying either Theorem 1 or 2 (recall they are equivalent when the regularizer is fixed) gives the bound of Eq. (7), in this case

$$\text{Regret}_T(x^*) \leq \frac{1}{2\eta}\|x^*\|_2^2 + \frac{1}{2}\sum_{t=1}^T \eta\|g_t\|_2^2, \tag{10}$$

using Lemma 3 for $\|x\|_{(t),\star} = \sqrt{\eta}\|x\|_2$. Suppose we are concerned with x^* where $\|x^*\|_2 \leq R$, the g_t satisfy $\|g_t\|_2 \leq G$, and we want to minimize regret after T' rounds. Then, choosing $\eta = \frac{R}{G\sqrt{T'}}$ minimizes Eq. (10) when $T = T'$, and we have

$$\text{Regret}_T(x^*) \leq \frac{RG}{2}\sqrt{T'} + \frac{RG}{2}\frac{T}{\sqrt{T'}},$$

or $\text{Regret}(x^*) \leq RG\sqrt{T}$ when $T = T'$. However, this bound is only $\mathcal{O}(\sqrt{T})$ when $T = \mathcal{O}(T')$. For $T \ll T'$, or $T \gg T'$ the bound is no longer interesting, and in fact the algorithm will likely perform poorly. This deficiency can be addressed via the "doubling trick", where we double T' and restart the algorithm each time T grows larger than T' (c.f., Shalev-Shwartz (2012, 2.3.1)). However, adaptively choosing the learning rate without restarting will allow us to achieve better bounds than the doubling trick (by a constant factor) with a more practically useful algorithm. We do this in Sections 3.2 and 3.3 below.

Constant Learning Rate Online Gradient Descent with a Feasible Set Above we assumed $\|x^*\|_2 \leq R$, but there is no a priori bound on the magnitude of the x_t selected by the algorithm. Following the approach of Section 2.4, we can incorporate a feasible set by taking $r_0(x) = \frac{1}{2\eta}\|x\|_2^2 + I_{\mathcal{X}}(x)$, so the update becomes

$$x_{t+1} = \arg\min_{x\in\mathbb{R}^n} g_{1:t} \cdot x + \frac{1}{2\eta}\|x\|_2^2 + I_{\mathcal{X}}(x) = \arg\min_{x\in\mathcal{X}} g_{1:t} \cdot x + \frac{1}{2\eta}\|x\|_2^2. \tag{11}$$

Following Shalev-Shwartz (2012, Sec. 2.6), this update is equivalent to the two-step update where we first solve the unconstrained problem and then project onto the feasible set, namely

$$u_{t+1} = \arg\min_{x\in\mathbb{R}^n} g_{1:t}\cdot x + \frac{1}{2\eta}\|x\|_2^2$$
$$x_{t+1} = \Pi_{\mathcal{X}}(u_{t+1}) \qquad \text{where} \qquad \Pi_{\mathcal{X}}(u) \equiv \arg\min_{x\in\mathcal{X}} \|x-u\|_2.$$

Many FTRL algorithms on feasible sets can in this way be interpreted as lazy-projection algorithms, where we find (or maintain) the solution to the unconstrained problem, and then project onto the feasible set when needed.

Theorem 1 can be used to analyze the constrained algorithm of Eq. (11) in exactly the same way we analyzed Eq. (9): adding $I_{\mathcal{X}}$ does not change the strong convexity of the $\|x\|_2^2$ terms in the regularizer, and so the only difference is in the $r_{0:T}(x^*)$ term. Instead of Eq. (10), we have

$$\forall x^*\in\mathcal{X},\ \text{Regret}_T(x^*) \le \frac{1}{2\eta}\|x^*\|_2^2 + \frac{1}{2}\sum_{t=1}^T \eta\|g_t\|_2^2,$$

where we have chosen to use the explicit $\forall x^*\in\mathcal{X}$ rather than the equivalent choice of including $I_{\mathcal{X}}(x^*)$ on the right-hand side.

Interestingly, the update of Eq. (11) is no longer equivalent to the standard projected Online Gradient Descent update $x_{t+1} = \Pi_{\mathcal{X}}(x_t - \eta g_t)$; this issue is discussed in the context of more general Mirror Descent updates in Appendix C.2. We will be able to analyze this algorithm using techniques from Section 6.

3.2 Dual Averaging

Dual Averaging is an adaptive FTRL-Centered algorithm with linearized loss functions; the adaptivity allows us to prove regret bounds that are $\mathcal{O}(\sqrt{T})$ for all T. We choose $r_t(x) = \frac{\sigma_t}{2}\|x\|_2^2$ for constants $\sigma_t \ge 0$, so $r_{0:t}$ is 1-strongly-convex w.r.t. the norm $\|x\|_{(t)} = \sqrt{\sigma_{0:t}}\|x\|_2$, which has dual norm $\|x\|_{(t),\star} = \frac{1}{\sqrt{\sigma_{0:t}}}\|x\|_2 = \sqrt{\eta_t}\|x\|_2$, using Lemma 3. Plugging into Theorem 1 then gives

$$\forall T,\ \text{Regret}_T(x^*) \le \frac{1}{2\eta_{T-1}}\|x^*\|_2^2 + \frac{1}{2}\sum_{t=1}^T \eta_{t-1}\|g_t\|_2^2.$$

Suppose we know $\|g_t\|_2 \le G$, and we consider x^* where $\|x^*\|_2 \le R$. Then, with the choice $\eta_t = \frac{R}{\sqrt{2}G\sqrt{t+1}}$, using the inequality $\sum_{t=1}^T \frac{1}{\sqrt{t}} \le 2\sqrt{T}$, we arrive at

$$\forall T,\ \text{Regret}_T(x^*) \le \frac{\sqrt{2}}{2}\left(R + \frac{\|x^*\|_2^2}{R}\right)G\sqrt{T}. \tag{12}$$

When in fact $\|x^*\| \le R$, we have Regret $\le \sqrt{2}RG\sqrt{T}$, but the bound of Eq. (12) is valid (and meaningful) for arbitrary $x^* \in \mathbb{R}^n$. Observe that on a particular round T, this bound is a factor $\sqrt{2}$ worse than the bound of $RG\sqrt{T}$ shown in Section 3.1 when the learning rate is tuned for exactly round T; this is the (small) price we pay for a bound that holds uniformly for all T.

As in the previous example, Dual Averaging can also be restricted to select from a feasible set $\mathcal{X}$ by including $I_{\mathcal{X}}$ in r_0. Additional non-smooth regularization can also be applied by adding the appropriate terms to r_0 (or any of the r_t); for example, we can add an L_1 and L_2 penalty by adding the terms $\lambda_1\|x\|_1 + \lambda_2\|x\|_2^2$. When in addition the f_t are linearized, this produces the Regularized Dual Averaging algorithm of Xiao (2009). Note that our result of $\sqrt{2}RG\sqrt{T}$ improves on the bound of $2RG\sqrt{T}$ achieved by Xiao (2009, Cor. 2(a)). We consider the case of such additional regularization terms in more detail in Section 5.

3.3 FTRL-Proximal

Suppose $\mathcal{X} \subseteq \{x \mid \|x\|_2 \le R\}$, and we choose $r_0(x) = I_{\mathcal{X}}(x)$ and for $t > 1$, $r_t(x) = \frac{\sigma_t}{2}\|x - x_t\|_2^2$. It is worth emphasizing that unlike in the previous examples, for FTRL-Proximal the inclusion of the feasible set $\mathcal{X}$ is essential to proving regret bounds. With this constraint we have $r_{0:t}(x^*) \le \frac{\sigma_{1:t}}{2}(2R)^2$ for any $x^* \in \mathcal{X}$, since each $x_t \in \mathcal{X}$. Without forcing $x_t \in \mathcal{X}$, however, the terms $\|x^* - x_t\|_2^2$ in $r_{0:t}(x^*)$ cannot be usefully bounded.

With these choices, $r_{0:t}$ is 1-strongly-convex w.r.t. the norm $\|x\|_{(t)} = \sqrt{\sigma_{1:t}}\|x\|_2$, which has dual norm $\|x\|_{(t),\star} = \frac{1}{\sqrt{\sigma_{1:t}}}\|x\|_2$. Thus, applying Theorem 2, we have

$$\forall x^*\in\mathcal{X},\quad \text{Regret}(x^*) \le \frac{1}{2\eta_T}(2R)^2 + \frac{1}{2}\sum_{t=1}^T \eta_t\|g_t\|^2, \tag{13}$$

where again $\eta_t = \frac{1}{\sigma_{1:t}}$. Choosing $\eta_t = \frac{\sqrt{2}R}{G\sqrt{t}}$ and assuming $\|x^*\| \le R$ and $\|g_t\|_2 \le G$,

$$\text{Regret}(x^*) \le 2\sqrt{2}RG\sqrt{T}. \tag{14}$$

Note that we are a factor of 2 worse than the corresponding bound for Dual Averaging. However, this is essentially an artifact of loosely bounding $\|x^* - x_t\|_2^2$ by $(2R)^2$, whereas for Dual Averaging we can bound $\|x^* - 0\|_2^2$ with R^2. In practice one would hope x_t is closer to x^* than 0, and so it is reasonable to believe that the FTRL-Proximal bound will actually be tighter post-hoc in many cases. Empirical evidence also suggests FTRL-Proximal can work better in practice (McMahan, 2011).

3.4 FTRL-Proximal with Diagonal Matrix Learning Rates

We now consider an AdaGrad FTRL-Proximal algorithm which is adaptive to the observed sequence of gradients g_t, improving on the previous result. For simplicity,

first consider a one-dimensional problem. Let $r_0 = I_{\mathcal{X}}$ with $\mathcal{X} = [-R, R]$, and fix a learning rate schedule for FTRL-Proximal where

$$\eta_t = \frac{\sqrt{2}R}{\sqrt{\sum_{s=1}^t g_s^2}}$$

for use in Eq. (13). This gives

$$\text{Regret}(x^*) \leq 2\sqrt{2}R\sqrt{\sum_{t=1}^T g_t^2}, \tag{15}$$

where we have used the following lemma, which generalizes $\sum_{t=1}^T 1/\sqrt{t} \leq 2\sqrt{T}$:

Lemma 4 *For any non-negative real numbers $a_1, a_2, \ldots, a_n$,*

$$\sum_{i=1}^n \frac{a_i}{\sqrt{\sum_{j=1}^i a_j}} \leq 2\sqrt{\sum_{i=1}^n a_i}\ .$$

For a proof see Auer et al. (2002) or Streeter and McMahan (2010, Lemma 1). The bound of Eq. (15) gives us a fully adaptive version of Eq. (14): not only do we not need to know T in advance, we also do not need to know a bound on the norms of the gradients G. Rather, the bound is fully adaptive and we see, for example, that the bound only depends on rounds t where the gradient is nonzero (as one would hope). We do, however, require that R is chosen in advance; for algorithms that avoid this, see Streeter and McMahan (2012); Orabona (2013); McMahan and Abernethy (2013), and McMahan and Orabona (2014).

To arrive at an AdaGrad-style algorithm for n-dimensions we need only apply the above technique on a per-coordinate basis, namely using learning rate

$$\eta_{t,i} = \frac{\sqrt{2}R_\infty}{\sqrt{\sum_{s=1}^t g_{s,i}^2}}$$

for coordinate i, where we assume $\mathcal{X} \subseteq [-R_\infty, R_\infty]^n$. Streeter and McMahan (2010) take the per-coordinate approach directly; the more general approach here allows us to handle arbitrary feasible sets and L_1 or other non-smooth regularization.

We take $r_0 = I_{\mathcal{X}}$, and for $t \geq 1$ define $r_t(x) = \frac{1}{2}\|Q_t^{\frac{1}{2}}(x - x_t)\|_2^2$ where $Q_t = \text{diag}\left(\sigma_{t,i}\right)$, the diagonal matrix with entries $\sigma_{t,i} = \eta_{t,i}^{-1} - \eta_{t-1,i}^{-1}$. This Q_t is positive semi-definite, and for any such Q_t, we have that $r_{0:t}$ is 1-strongly-convex w.r.t. the norm $\|x\|_{(t)} = \|(Q_{1:t})^{\frac{1}{2}}x\|_2$, which has dual norm $\|g\|_{(t),\star} = \|(Q_{1:t})^{-\frac{1}{2}}g\|_2$. Then, plugging into Theorem 2 gives

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^T \|(Q_{1:t})^{-\frac{1}{2}}g_t\|_2.$$

which improves on McMahan and Streeter (2010, Theorem 2) by a constant factor.

Essentially, this bound amounts to summing Eq. (15) across all n dimensions; McMahan and Streeter (2010, Cor. 9) show this bound is at least as good (and often better) than that of Eq. (14). Full matrix learning rates can be derived using a matrix generalization of Lemma 4, e.g., Duchi et al. (2011, Lemma 10); however, since this requires $\mathcal{O}(n^2)$ space and potentially $\mathcal{O}(n^2)$ time per round, in practice these algorithms are often less useful than the diagonal varieties.

It is perhaps not immediately clear that the diagonal FTRL-Proximal algorithm is easy and efficient to implement. However, taking the linear approximation to f_t, one can see $h_{1:t}(x) = g_{1:t} \cdot x + r_{1:t}(x)$ is itself just a quadratic which can be represented using two length n vectors, one to maintain the linear terms ($g_{1:t}$ plus adjustment terms) and one to maintain $\sum_{s=1}^t g_{s,i}^2$, from which the diagonal entries of $Q_{1:t}$ can be constructed. That is, the update simplifies to

$$x_{t+1} = \operatorname*{arg\,min}_{x \in \mathcal{X}} \ (g_{1:t} - a_{1:t}) \cdot x + \sum_{i=1}^n \frac{1}{2\eta_{t,i}} x_i^2 \qquad \text{where} \qquad a_t = \sigma_t x_t.$$

This update can be solved in closed-form on a per-coordinate basis when $\mathcal{X} = [-R_\infty, R_\infty]^n$. For a general feasible set, it is equivalent to a lazy-projection algorithm that first solves for the unconstrained solution and then projects it onto $\mathcal{X}$ using norm $\|(Q_{1:t})^{\frac{1}{2}} \cdot \|$ (see McMahan and Streeter (2010, Eq. 7)). Pseudo-code which also incorporates L_1 and L_2 regularization is given in McMahan et al. (2013).

3.5 AdaGrad Dual Averaging

Similar ideas can be applied to Dual Averaging (where we center each r_t at x_1), but one must use some care due to the "off-by-one" difference in the bounds. For example, for the diagonal algorithm, it is necessary to choose per-coordinate learning rates

$$\eta_t \approx \frac{R}{\sqrt{G^2 + \sum_{s=1}^t g_s^2}},$$

where $|g_t| \leq G$. Thus, we arrive at an algorithm that is almost (but not quite) fully adaptive in the gradients, since a modest dependence on the initial guess G of the maximum per-coordinate gradient remains in the bound. This offset appears, for example, as the δI terms added to the learning rate matrix H_t in Figure 1 of Duchi et al. (2011). We will see this issue again in Section 3.7.

3.6 Strongly Convex Functions

Suppose each loss function f_t is 1-strongly-convex w.r.t. a norm $\|\cdot\|$, and let $r_t(x) = 0$ for all t (that is, we use the Follow-The-Leader (FTL) algorithm). Define $\|x\|_{(t)} =$

Non-Adaptive FTRL Algorithms (fixed regularizer r_0, with $r_t(x) = 0$ for $t \geq 1$)

Constant Learning Rate Unprojected Online Gradient Descent

$$\begin{aligned} x_{t+1} &= x_t - \eta g_t \\ &= \arg\min_x g_{1:t} \cdot x_t + \frac{1}{2\eta}\|x\|_2^2 \\ &= -\eta g_{1:t} \end{aligned}$$

Follow-The-Leader where the f_t are 1-strongly-convex w.r.t. $\|\cdot\|$

$$x_{t+1} = \arg\min_x f_{1:t}(x)$$

Online Gradient Descent for strongly-convex functions

$$\begin{aligned} x_{t+1} &= \arg\min_x g_{1:t} \cdot x + \frac{1}{2}\sum_{s=1}^t \|x - x_s\|^2 \qquad \text{where } g_t \in \partial f_t(x_t) \\ &= x_t - \eta_t g_t \qquad \text{where } \eta_t = \frac{1}{t} \end{aligned}$$

Adaptive FTRL-Centered Algorithms (r_t chosen adaptively and minimized at x_1)

Unconstrained Dual Averaging (adaptive to t)

$$\begin{aligned} x_{t+1} &= \arg\min_x g_{1:t} \cdot x + \frac{1}{2\eta_t}\|x\|_2^2 \qquad \text{where } \eta_t = \frac{R}{\sqrt{2}G\sqrt{t+1}} \\ &= -\eta_t g_{1:t} \end{aligned}$$

FTRL with the entropic regularizer over the probability simplex Δ (adaptive to g_t)

$$x_{t+1} = \arg\min_{x\in\Delta} g_{1:t} \cdot x + \frac{1}{2\eta_t}\sum_{i=1}^n x_i \log x_i \qquad \text{where } \eta_t = \frac{\sqrt{\log n}}{\sqrt{G_\infty^2 + \sum_{s=1}^t \|g_s\|_\infty^2}}, \text{ or}$$

$$x_{t+1,i} = \frac{\exp(-\eta_t g_{1:t,i})}{\sum_{i=1}^n \exp(-\eta_t g_{1:t,i})} \qquad \text{in closed form}$$

Adaptive FTRL-Proximal Algorithms (r_t chosen adaptively and minimized at x_t)

FTRL-Proximal (adaptive to t) with $\sigma_s = \eta_s^{-1} - \eta_{s-1}^{-1}$

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} g_{1:t} \cdot x + \sum_{s=1}^t \frac{\sigma_s}{2}\|x - x_s\|_2^2 \qquad \text{where } \eta_t = \frac{\sqrt{2}R}{G\sqrt{t}}$$

AdaGrad FTRL-Proximal (adaptive to g_t) with $\sigma_{s,i} = \eta_{s,i}^{-1} - \eta_{s-1,i}^{-1}$.

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} g_{1:t} \cdot x + \sum_{s=1}^t \frac{1}{2}\left\| \operatorname{diag}\left(\sigma_{s,i}^{\frac{1}{2}}\right)(x - x_s)\right\|_2^2 \qquad \text{where } \eta_{t,i} = \frac{\sqrt{2}R}{\sqrt{\sum_{s=1}^t g_{s,i}^2}}$$

Figure 1: Example updates for algorithms in different branches of the FTRL family.

$\sqrt{t}\|x\|$, and observe $h_{0:t}(x)$ is 1-strongly-convex w.r.t. $\|\cdot\|_{(t)}$ (by Lemma 3). Then, applying either Theorem 1 or 2 (recalling they coincide when all $r_t(x) = 0$),

$$\text{Regret}(x^*) \leq \frac{1}{2}\sum_{t=1}^T \|g_t\|_{(t),\star}^2 = \frac{1}{2}\sum_{t=1}^T \frac{1}{t}\|g_t\|^2 \leq \frac{G^2}{2}(1 + \log T),$$

where we have used the inequality $\sum_{t=1}^T 1/t \leq 1 + \log T$ and assumed $\|g_t\| \leq G$. This recovers, e.g., Kakade and Shalev-Shwartz (2008, Cor. 1) for the the exact FTL algorithm. This algorithm requires optimizing over $f_{1:t}$ exactly, which may be computationally prohibitive.

For a 1-strongly-convex f_t with $g_t \in \partial f_t(x_t)$ we have by definition

$$f_t(x) \geq \underbrace{f_t(x_t) + g_t \cdot (x - x_t) + \frac{1}{2}\|x - x_t\|^2}_{=\bar{f}_t}.$$

Thus, we can define a $\bar{f}_t$ equal to the right-hand-side of the above inequality, so $\bar{f}_t(x) \leq f_t(x)$ and $\bar{f}_t(x_t) = f_t(x_t)$. The $\bar{f}_t$ are also 1-strongly-convex w.r.t. $\|\cdot\|$, and so running FTL on these functions produces an identical regret bound. Theorem 11 will show that the update $x_{t+1} = \arg\min_x \bar{f}_{1:t}(x)$ is equivalent to the Online Gradient Descent update

$$x_{t+1} = x_t - \frac{1}{t}g_t,$$

showing this update is essentially the Online Gradient Descent algorithm for strongly convex functions given by Hazan et al. (2007).[2]

3.7 Adaptive Dual Averaging with the Entropic Regularizer

We consider problems where the algorithm selects a probability distribution (e.g., in order to sample an action from a discrete set of n choices), that is $x_t \in \Delta_n$ with

$$\Delta_n = \left\{ x \;\middle|\; \sum_{i=1}^n x_i = 1 \text{ and } x_i \geq 0 \right\}.$$

We assume gradients are bounded so that $\|g_t\|_\infty \leq G_\infty$, which is natural for example if each action has a cost in the range $[-G_\infty, G_\infty]$, so $g_t \cdot x$ gives the expected cost of choosing an action from the distribution x. This is the classic problem of prediction from expert advice (Vovk, 1990; Littlestone and Warmuth, 1994; Freund and Schapire, 1995; Cesa-Bianchi and Lugosi, 2006).

2. Again, the constraint to select from a fixed feasible set $\mathcal{X}$ can be added easily in either case; however, the natural way to add the constraint to the FTRL expression produces a "lazy-projection" algorithm, whereas adding the constraint to the Online Gradient Descent update produces a "greedy-projection" algorithm. This issue is discussed in some depth in Appendix C.2.

The previously introduced algorithms can be applied by enforcing the constraint $x \in \Delta_n$ by adding I_{Δ_n} to r_0, but to instantiate their bounds we can only bound $\|g_t\|_2$ by $\sqrt{n}G_\infty$ in this case, leading to bounds like $\mathcal{O}(G_\infty\sqrt{nT})$. By using a more appropriate regularizer, we can reduce the dependence on the dimension from $\sqrt{n}$ to $\sqrt{\log n}$. In particular, we use the entropic regularizer,

$$H(x) = I_\Delta(x) + \log n + \sum_{i=1}^n x_i \log x_i,$$

from which we define the following adaptive regularization schedule:

$$r_{0:t}(x) = \frac{1}{\eta_t} H(x) \qquad \text{where} \qquad \eta_t = \frac{\sqrt{\log n}}{\sqrt{G_\infty^2 + \sum_{s=1}^t \|g_s\|_\infty^2}}$$

for $t \geq 0$. Note that as in AdaGrad Dual Averaging, we make the learning rate schedule η_t a function of the observed g_t. The function H (and hence each $r_{0:t}$) is minimized by the uniform distribution $x_1 = (1/n, \dots, 1/n)$ where $H(x) = 0$, and so these regularizers are centered at x_1. Note also that h is maximized at the corners of Δ_n (e.g., $x = (1, 0, \dots, 0)$) where it has value $\log n$.

The entropic regularizer H is 1-strongly-convex with respect to the L_1 norm over the probability simplex $\mathcal{X}$ (e.g., Shalev-Shwartz (2012, Ex 2.5)), and it follows that $r_{0:t}$ is 1-strongly convex with respect to the norm $\|x\|_{(t)} = \frac{1}{\sqrt{\eta_t}}\|x\|_1$, and $\|g\|_{(t),\star}^2 = \eta_t\|g\|_\infty^2$. Then, applying Theorem 1, we have

$$\begin{aligned}
\text{Regret}(x^*) &\leq r_{0:T-1}(x^*) + \frac{1}{2}\sum_{t=1}^T \|g_t\|_{(t-1),\star}^2 \\
&\leq \frac{\log n}{\eta_{T-1}} + \frac{1}{2}\sum_{t=1}^T \eta_{t-1}\|g_t\|_\infty^2 \\
&\leq \frac{\log n}{\eta_{T-1}} + \frac{\sqrt{\log n}}{2}\sum_{t=1}^T \frac{\|g_t\|_\infty^2}{\sqrt{\sum_{s=1}^t \|g_s\|_\infty^2}} \qquad \text{since } \forall t,\ \|g_t\|_\infty \leq G_\infty \\
&\leq 2\sqrt{\left(G_\infty^2 + \sum_{t=1}^{T-1}\|g_t\|_\infty^2\right)\log n} \qquad \text{Lemma 4 and } \|g_T\|_\infty \leq G_\infty \\
&\leq 2G_\infty\sqrt{T\log n}.
\end{aligned}$$

The last line gives an adaptive ($\forall T$) version of Shalev-Shwartz (2012, Cor. 2.14 and Cor 2.16), but the version of the bound in terms of $\|g_t\|_\infty$ may be much tighter if there are many rounds where the maximum magnitude cost is much less than G_∞. For similar adaptive algorithms, see Stoltz (2005, Thm 2.3) and Stoltz (2011, Thm 1.4, Eq. (1.22)).

4. A General Analysis Technique

In this section, we prove Theorems 1 and 2; the analysis techniques developed will also be used in subsequent sections to analyze composite objectives and Mirror Descent algorithms.

4.1 Inductive Lemmas

In this section we prove the following lemma that lets us analyze arbitrary FTRL-style algorithms:

Lemma 5 (Strong FTRL Lemma) *Let f_t be a sequence of arbitrary (possibly non-convex) loss functions, and let r_t be arbitrary non-negative regularization functions, such that $x_{t+1} = \arg\min_x h_{0:t}(x)$ is well defined, where $h_{0:t}(x) \equiv f_{1:t}(x) + r_{0:t}(x)$. Then, the algorithm that selects these x_t achieves*

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \sum_{t=1}^T h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t). \tag{16}$$

This lemma can be viewed as a stronger form of the more well-known standard FTRL Lemma (see Kalai and Vempala (2005); Hazan (2008), Hazan (2010, Lemma 1), McMahan and Streeter (2010, Lemma 3), and Shalev-Shwartz (2012, Lemma 2.3)). The strong version has three main advantages over the standard version: 1) it is essentially tight, which improves the final bounds by a constant factor, 2) it can be used to analyze adaptive FTRL-Centered algorithms in addition to FTRL-Proximal, and 3) it relates directly to the primal-dual style of analysis. For completeness, in Appendix A we present the standard version of the lemma, along with the proof of a bound analogous to Theorem 2 (but weaker by a constant factor).

The Strong FTRL Lemma bounds regret by the sum of two factors:

- **Stability** The terms in the sum over t measure how much better x_{t+1} is for the cumulative objective function $h_{0:t}$ than the point actually selected, x_t: namely $h_{0:t}(x_t) - h_{0:t}(x_{t+1})$. These per-round terms can be seen as measuring the stability of the algorithm, an online analog to the role of stability in the stochastic setting (Bousquet and Elisseeff, 2002; Rakhlin et al., 2005; Shalev-Shwartz et al., 2010).
- **Regularization** The term $r_{0:T}(x^*)$ quantifies how much regularization we have added, measured at the comparator point x^*. This captures the intuitive fact that if we could center our regularization at x^* it should not increase regret.

Adding strongly convex regularizers will increase stability (and hence decrease the cost of the stability terms), at the expense of paying a larger regularization penalty

$r_{0:T}(x^*)$. At the heart of the adaptive algorithms we study is the ability to dynamically balance these two competing goals.

The following corollary relates the above statement to the primal-dual style of analysis:

Corollary 6 *Consider the same conditions as Lemma 5, and further suppose the loss functions are linear, $f_t(x) = g_t \cdot x_t$. Then,*

$$h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) = r^\star_{0:t}(-g_{1:t}) - r^\star_{0:t-1}(-g_{1:t-1}) + g_t \cdot x_t, \tag{17}$$

which implies

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \sum_{t=1}^T r^\star_{0:t}(-g_{1:t}) - r^\star_{0:t-1}(-g_{1:t-1}) + g_t \cdot x_t.$$

We make a few remarks before proving these results at the end of this section. Corollary 6 can easily be proved directly using the Fenchel-Young inequality. Our statement directly matches the first claim of Orabona (2013, Lemma 1), and in the non-adaptive case re-arrangement shows equivalence to Shalev-Shwartz (2007, Lemma 1) and Shalev-Shwartz (2012, Lemma 2.20); see also Kakade et al. (2012, Corollary 4). McMahan and Orabona (2014, Thm. 1) give a closely related duality result for regret and reward, and discuss several interpretations for this result, including the potential function view, the connection to Bregman divergences, and an interpretation of $r^\star$ as a benchmark target for reward.

Note, however, that Lemma 5 is strictly stronger than Corollary 6: it applies to non-convex f_t and r_t. Further, even for convex f_t, it can be more useful: for example, we can directly analyze strongly convex f_t with all $r_t(x) = 0$ using the first statement. Lemma 5 is also arguably simpler, in that it does not require the introduction of convexity or the Fenchel conjugate. We now prove the Strong FTRL Lemma:

Proof of Lemma 5 First, we bound a quantity that is essentially our regret if we had used the FTL algorithm against the functions $h_1, \dots h_T$ (for convenience, we include a $-h_0(x^*)$ term as well):

$$\begin{aligned}
&\sum_{t=1}^T h_t(x_t) - h_{0:T}(x^*) \\
&\quad = \sum_{t=1}^T (h_{0:t}(x_t) - h_{0:t-1}(x_t)) - h_{0:T}(x^*) \\
&\quad \leq \sum_{t=1}^T (h_{0:t}(x_t) - h_{0:t-1}(x_t)) - h_{0:T}(x_{T+1}) \quad \text{Since } x_{T+1} \text{ minimizes } h_{0:T} \\
&\quad \leq \sum_{t=1}^T (h_{0:t}(x_t) - h_{0:t}(x_{t+1})),
\end{aligned}$$

where the last line follows by simply re-indexing the $-h_{0:t}$ terms and dropping the the non-positive term $-h_0(x_1) = -r_0(x_1) \leq 0$. Expanding the definition of h on the left-hand-side of the above inequality gives

$$\sum_{t=1}^T (f_t(x_t) + r_t(x_t)) - f_{1:T}(x^*) - r_{0:T}(x^*) \leq \sum_{t=1}^T (h_{0:t}(x_t) - h_{0:t}(x_{t+1})).$$

Re-arranging the inequality proves the lemma. ■

We remark it is possible to make Lemma 5 an *equality* if we include the term $h_{0:T}(x_{T+1}) - h_{0:T}(x^*)$ on the RHS, since we can assume $r_0(x_1) = 0$ without loss of generality. In this case, we do not need the assumption that $x_{t+1} = \arg\min_x h_{0:t}(x)$, and so the lemma applies to an arbitrary sequence of points $x_1, \dots, x_T$. On the other hand, if one is actually interested in the performance of the Follow-The-Leader (FTL) algorithm against the h_t (e.g., if all the r_t are uniformly zero), then running the FTL algorithm and choosing $x^* = x_{T+1}$ is particularly natural.

Proof of Corollary 6 Using the definition of the Fenchel conjugate and of x_{t+1},

$$r^\star_{0:t}(-g_{1:t}) = \max_x \; -g_{1:t} \cdot x - r_{0:t}(x) = -\big(\min_x \; g_{1:t} \cdot x + r_{0:t}(x)\big) = -h_{0:t}(x_{t+1}). \tag{18}$$

Now, observe that

$$\begin{aligned}
h_{0:t}(x_t) - r_t(x_t) &= g_{1:t} \cdot x_t + r_{0:t}(x_t) - r_t(x_t) \\
&= g_{1:t-1} \cdot x_t + r_{0:t-1}(x_t) + g_t \cdot x_t \\
&= h_{0:t-1}(x_t) + g_t \cdot x_t \\
&= -r^\star_{0:t-1}(-g_{1:t-1}) + g_t \cdot x_t,
\end{aligned}$$

where the last line uses Eq. (18) with $t \to t-1$. Combining this with Eq. (18) again $(-h_{0:t}(x_{t+1}) = r^\star_{0:t}(-g_{1:t}))$ proves Eq. (17). ■

4.2 Tools from Convex Analysis

Here we highlight a few key tools from convex analysis that will be used to bound the per-round stability terms that appear in the Strong FTRL Lemma. For more background on convex analysis, see Rockafellar (1970) and Shalev-Shwartz (2007, 2012). The next result generalizes arguments found in earlier proofs for FTRL algorithms:

Lemma 7 *Let $\phi_1 : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ be a convex function such that $x_1 = \arg\min_x \phi_1(x)$ exists. Let ψ be a convex function such that $\phi_2(x) = \phi_1(x) + \psi(x)$ is strongly convex w.r.t. norm $\|\cdot\|$. Let $x_2 = \arg\min_x \phi_2(x)$. Then, for any $b \in \partial\psi(x_1)$, we have*

$$\|x_1 - x_2\| \leq \|b\|_\star, \tag{19}$$

and for any x',

$$\phi_2(x_1) - \phi_2(x') \leq \frac{1}{2}\|b\|_\star^2.$$

We defer the proofs of the results in this section to Appendix B. When ϕ_1 and ψ are quadratics (with ψ possibly linear) and the norm is the corresponding L_2 norm, both statements in the above lemma hold with equality. For the analysis of composite updates (Section 5), it will be useful to split the change ψ in the objective function ϕ into two components:

Corollary 8 *Let $\phi_1 : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ be a convex function such that $x_1 = \arg\min_x \phi_1(x)$ exists. Let ψ and Ψ be convex functions such that $\phi_2(x) = \phi_1(x) + \psi(x) + \Psi(x)$ is strongly convex w.r.t. norm $\|\cdot\|$. Let $x_2 = \arg\min_x \phi_2(x)$. Then, for any $b \in \partial\psi(x_1)$ and any x',*

$$\phi_2(x_1) - \phi_2(x') \leq \frac{1}{2}\|b\|_\star^2 + \Psi(x_1) - \Psi(x_2).$$

The concept of strong smoothness plays a key role in the proof of the above lemma, and can also be used directly in the application of Corollary 6. A function ψ is *σ-strongly-smooth* with respect to a norm $\|\cdot\|$ if it is differentiable and for all x, y we have

$$\psi(y) \leq \psi(x) + \nabla\psi(x) \cdot (y - x) + \tfrac{\sigma}{2}\|y - x\|^2. \tag{20}$$

There is a fundamental duality between strongly convex and strongly smooth functions:

Lemma 9 *Let ψ be closed and convex. Then ψ is σ-strongly convex with respect to the norm $\|\cdot\|$ if and only if $\psi^\star$ is $\frac{1}{\sigma}$-strongly smooth with respect to the dual norm $\|\cdot\|_\star$.*

For the strong convexity implies strongly smooth direction see Shalev-Shwartz (2007, Lemma 15), and for the other direction see Kakade et al. (2012, Theorem 3).

4.3 Regret Bound Proofs

In this section, we prove Theorems 1 and 2 using Lemma 5. Stating these two analyses in a common framework makes clear exactly where the "off-by-one" issue arises for FTRL-Centered, and how assuming proximal r_t resolves this issue. The key tool is Lemma 7, though for comparison we also provide a proof of Theorem 1 for linearized functions from Corollary 6 directly using strong smoothness.

General FTRL including FTRL-Centered (Proof of Theorem 1) In order to apply Lemma 5, we work to bound the stability terms in the sum in Eq. (16). Fix a particular round t. For Lemma 7 take $\phi_1(x) = h_{0:t-1}(x)$ and $\phi_2(x) = h_{0:t-1}(x) + f_t(x)$, so $x_t = \arg\min_x \phi_1(x)$, and by assumption ϕ_2 is 1-strongly-convex w.r.t. $\|\cdot\|_{(t-1)}$. Then, applying Lemma 7 to ϕ_2 (with $x' = x_{t+1}$), we have $\phi_2(x_t) - \phi_2(x_{t+1}) \leq \frac{1}{2}\|g_t\|_{(t-1),\star}^2$ for $g_t \in \partial f_t(x_t)$, and so

$$\begin{aligned} h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) &= \phi_2(x_t) + r_t(x_t) - \phi_2(x_{t+1}) - r_t(x_{t+1}) - r_t(x_t) \\ &\leq \frac{1}{2}\|g_t\|_{(t-1),\star}^2 \end{aligned}$$

where we have used the assumption that $r_t(x) \geq 0$ to drop the $-r_t(x_{t+1})$ term. We can now plug this bound into Lemma 5. However, we need to make one additional observation: the choice of r_T only impacts the bound by increasing $r_{0:T}(x^*)$. Further, r_T does not influence any of the points $x_1, \ldots, x_T$ selected by the algorithm. Thus, for analysis purposes, we can take $r_T(x) = 0$ without loss of generality, and hence replace $r_{0:T}(x^*)$ with $r_{0:T-1}(x^*)$ in the final bound. ■

FTRL-Proximal (Proof of Theorem 2) The key is again to bound the stability terms in the sum in Eq. (16). Fix a particular round t, and take $\phi_1(x) = f_{1:t-1}(x) + r_{0:t}(x) = h_{0:t}(x) - f_t(x)$. Since the r_t are proximal (so x_t is a global minimizer of r_t) we have $x_t = \arg\min_x \phi_1(x)$, and $x_{t+1} = \arg\min_x \phi_1(x) + f_t(x)$. Thus,

$$\begin{aligned} h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) &\leq h_{0:t}(x_t) - h_{0:t}(x_{t+1}) && \text{Since } r_t(x) \geq 0 \\ &= \phi_1(x_t) + f_t(x_t) - \phi_1(x_{t+1}) - f_t(x_{t+1}) \\ &\leq \frac{1}{2}\|g_t\|_{(t),\star}^2, && (21) \end{aligned}$$

where the last line follows by applying Lemma 7 to ϕ_1 and $\phi_2(x) = \phi_1(x) + f_t(x) = h_{0:t}(x)$. Plugging into Lemma 5 completes the proof. ■

Primal-dual Analysis of General FTRL on Linearized Functions We give an alternative proof of Theorem 1 for linear functions, $f_t(x) = g_t \cdot x$, using Eq. (17). We remark that in this case $x_t = \nabla r_{1:t-1}^\star(-g_{1:t-1})$ (see Lemma 15 in Appendix B).

By Lemma 9, $r^\star_{1:t-1}$ is 1-strongly-smooth with respect to $\|\cdot\|_{(t-1),\star}$, and so

$$r^\star_{1:t-1}(-g_{1:t}) \le r^\star_{1:t-1}(-g_{1:t-1}) - x_t \cdot g_t + \frac{1}{2}\|g_t\|^2_{(t-1),\star}, \tag{22}$$

and we can bound the per-round terms in Eq. (17) by

$$\begin{aligned} r^\star_{1:t}(-g_{1:t}) - r^\star_{1:t-1}(-g_{1:t-1}) + x_t \cdot g_t &\le r^\star_{1:t}(-g_{1:t}) - r^\star_{1:t-1}(-g_{1:t}) + \frac{1}{2}\|g_t\|^2_{(t-1),\star} \\ &\le \frac{1}{2}\|g_t\|^2_{(t-1),\star}, \end{aligned}$$

where we use Eq. (22) to bound $-r^\star_{1:t-1}(-g_{1:t-1}) + x_t \cdot g_t$, and then used the fact that $r^\star_{1:t-1}(-g_{1:t}) \ge r^\star_{1:t}(-g_{1:t})$ from Lemma 3. ■

5. Additional Regularization Terms and Composite Objectives

In this section, we consider generalized FTRL algorithms where we introduce an additional regularization term $\alpha_t\Psi(x)$ on each round, where Ψ is a convex function taking on only non-negative values, and the weights $\alpha_t \ge 0$ for $t \ge 1$ are non-increasing in t. We further assume Ψ and r_0 are both minimized at x_1, and w.l.o.g. $\Psi(x_1) = 0$ (as usual, additive constant terms do not impact regret). We generalize our definition of h_t to $h_0(x) = r_0(x)$ and

$$h_t(x) = g_t \cdot x + \alpha_t\Psi(x) + r_t(x), \tag{23}$$

so the FTRL update is

$$x_{t+1} = \arg\min_x h_{0:t}(x) = \arg\min_x g_{1:t} \cdot x + \alpha_{1:t}\Psi(x) + r_{0:t}(x). \tag{24}$$

In applications, generally the $g_t \cdot x_t$ terms come from the linearization of a loss ℓ_t, that is $g_t = \partial\ell_t(x_t)$. Here ℓ_t is for example a loss function measuring the prediction error on the tth training example for a model parameterized by x_t. (It is straightforward to replace $g_t \cdot x$ with $\ell_t(x)$ in this section, but for simplicity we assume linearization has been applied).

The Ψ terms often encode a non-smooth regularizer, and might be added for a variety of reasons. For example, the actual convex optimization problem we are solving may itself contain regularization terms. This is perhaps most clear in the case of applying an online algorithm to a batch problem as in Eq. (3). For example:

- An L_2 penalty $\Psi(x) = \|x\|_2^2$ might be added in order to promote generalization in a statistical setting, as in regularized empirical risk minimization.

- An L_1 penalty $\Psi(x) = \|x\|_1$ (as in the LASSO method) might be added to encourage sparse solutions and improve generalization in the high-dimensional setting ($n \gg T$).
- An indicator function might be added by taking $\Psi(x) = I_{\mathcal{X}}(x)$ to force $x \in \mathcal{X}$ where $\mathcal{X}$ is a convex set of feasible solutions.

As discussed in Section 2.4, the case of $\Psi = I_{\mathcal{X}}$ can be handled by our existing results. However, for other choices of Ψ it is generally preferable to only apply the linearization to the part of the objective where it is necessary computationally; in the L_1 case, given loss functions $\ell_t(x) + \lambda_1\|x\|_1$, we might partially linearize by taking $\bar{f}_t(x) = g_t \cdot x + \lambda_1\|x\|_1$, where $g_t \in \partial\ell_t(x_t)$. Recall that the primary motivation for linearization was to reduce the computation and storage requirements of the algorithm. Storing and optimizing over $\ell_{1:t}$ might be prohibitive; however, for common choices of Ψ and r_t, the optimization of Eq. (24) can be represented and solved efficiently (often in closed form). Thus, it is advantageous to consider such a composite representation.

Further, even in the case of a feasible set $\Psi = I_{\mathcal{X}}$, a careful consideration of if and when Ψ is linearized is critical to understanding the connection between Mirror Descent and FTRL. We will see that Mirror Descent *always* linearizes the past penalties $\alpha_{1:t-1}\Psi$, while with FTRL it is possible to avoid this additional linearization as in Eq. (24)—to make this distinction more clear, we will refer to the direct application of Eq. (24) as the Native FTRL algorithm. For $\Psi = I_{\mathcal{X}}$ this gives rise to the distinction between "lazy-projection" and "greedy-projection" algorithms, as discussed in Appendix C.2. And for $\Psi(x) = \|x\|_1$, this distinction makes Native FTRL algorithms preferable to composite-objective Mirror Descent for generating sparse models using L_1 regularization (see Section 6.2).

There are two types of regret bounds we may wish to prove in this setting, depending on whether we group the Ψ terms with the objective g_t, or with the regularizer r_t. We discuss these below.

In the objective We may view the $\alpha_t\Psi(x)$ terms as part of the objective, in that we desire a bound on regret against the functions $f^\Psi_t(x) \equiv g_t \cdot x + \alpha_t\Psi(x)$, that is

$$\text{Regret}(x^*, f^\Psi) \equiv \sum_{t=1}^T f^\Psi_t(x_t) - f^\Psi_t(x^*).$$

This setting is studied by Xiao (2009) and Duchi et al. (2010b, 2011), though in the less general setting where all $\alpha_t = 1$. We can directly apply Theorem 1 or Theorem 2 to the f^Ψ in this case, but this gives us bounds that depend on terms like $\|g_t + g_t^{(\Psi)}\|^2_{(t),\star}$ where $g_t^{(\Psi)} \in \partial(\alpha_t\Psi)(x_t)$; this is fine for $\Psi = I_{\mathcal{X}}$ since we can then always take $g_t^{(\Psi)} = 0$ since $x_t \in \mathcal{X}$, but for general Ψ this bound may be harder

to interpret. Further, adding a fixed known penalty like Ψ should intuitively make the problem no harder, and we would like to demonstrate this in our bounds.

In the regularizer We may wish to measure loss only against the functions $f_t(x) = g_t \cdot x$, that is,

$$\text{Regret}(x^*, g_t) \equiv \sum_{t=1}^{T} g_t \cdot x_t - g_t \cdot x^*,$$

even though we include the terms $\alpha_t \Psi$ in the update of Eq. (24). This approach is natural when we are only concerned with regret on the learning problem, $f_t(x) = \ell_t(x)$, but wish to add (for example) additional L_1 regularization in order to produce sparse models, as in McMahan et al. (2013).

In this case we can apply Theorem 1 to $f_t(x) \leftarrow g_t \cdot x$ and $r_t(x) \leftarrow r_t(x) + \alpha_t \Psi(x)$, noting that if the original $r_{0:t}$ is strongly convex w.r.t. $\|\cdot\|_{(t)}$, then $r_{0:t} + \alpha_{1:t}\Psi$ is as well, since Ψ is convex. However, if r_t is proximal, $r_t + \alpha_t \Psi$ generally will not be, and so a modified result is needed in place of Theorem 2. The following theorem provides this as well as a bound on $\text{Regret}(x^*, f^\Psi)$.

Theorem 10 FTRL-Proximal Bounds for Composite Objectives *Let Ψ be a non-negative convex function minimized at x_1 with $\Psi(x_1) = 0$. Let $\alpha_t \geq 0$ be a non-increasing sequence of constants. Consider Setting 1, and define h_t as in Eq.* (23)*. Suppose the r_t are chosen such that $h_{0:t}$ is 1-strongly-convex w.r.t. some norm $\|\cdot\|_{(t)}$, and further the r_t are proximal, that is x_t is a global minimizer of r_t.*

When we consider regret against $f_t^\Psi(x) = g_t \cdot x + \alpha_t \Psi(x)$, we have

$$\text{Regret}(x^*, f^\Psi) \leq r_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_{(t),\star}^2. \tag{25}$$

When we consider regret against only the functions $f_t(x) = g_t \cdot x$, we have

$$\text{Regret}(x^*, g_t) \leq r_{0:T}(x^*) + \alpha_{1:T}\Psi(x^*) + \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_{(t),\star}^2. \tag{26}$$

Proof The proof closely follows the proof of Theorem 2 in Section 4.3, with the key difference that we use Corollary 8 in place of Lemma 7. We will use Lemma 5 to prove both claims. First, observe that the stability terms $h_{0:t}(x_t) - h_{0:t}(x_{t+1})$ depend only on h, and so we can bound them in the same way in both cases.

Take $\phi_1(x) = h_{0:t-1}(x) + r_t(x)$. Since the r_t are proximal (so x_t is a global minimizer of r_t) we have $x_t = \arg\min_x \phi_1(x)$, and $x_{t+1} = \arg\min_x \phi_2(x)$ where

$\phi_2(x) = \phi_1(x) + g_t \cdot x + \alpha_t \Psi(x) = h_{0:t}(x)$. Then, using Corollary 8 lets us replace Eq. (21) with

$$h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) \leq \frac{1}{2}\|g_t\|_{(t),\star}^2 + \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}).$$

To apply Lemma 5 we sum over t. Considering only the Ψ terms, we have

$$\sum_{t=1}^{T} \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}) = \alpha_1 \Psi(x_1) - \alpha_T \Psi(x_{T+1}) + \sum_{t=2}^{T} \alpha_t \Psi(x_t) - \alpha_{t-1}\Psi(x_t) \leq 0,$$

since $\Psi(x) \geq 0$, $\alpha_t \leq \alpha_{t-1}$, and $\Psi(x_1) = 0$. Thus,

$$\sum_{t=1}^{T} h_{0:t}(x_t) - h_{0:t}(x_{t+1}) - r_t(x_t) \leq \frac{1}{2}\sum_{t=1}^{T} \|g_t\|_{(t),\star}^2.$$

Using this with Lemma 5 applied to $f_t(x) \leftarrow g_t \cdot x + \alpha_t \Psi(x)$ and $r_t \leftarrow r_t$ proves Eq. (25). For Eq. (26), we apply Lemma 5 taking $f_t(x) \leftarrow g_t \cdot x$ and $r_t(x) \leftarrow \alpha_t \Psi(x) + r_t(x)$. ■

For FTRL-Centered algorithms, Theorem 1 immediately gives a bound for $\text{Regret}(x^*, g_t)$. For the $\text{Regret}(x^*, f^\Psi)$ case, we can prove a bound matching Theorem 1 using arguments analogous to the above.

6. Mirror Descent, FTRL-Proximal, and Implicit Updates

Recall Section 3.1 showed the equivalence between constant learning rate Online Gradient Descent and a fixed-regularizer FTRL algorithm. This equivalence is well-known in the case where $r_t(x) = 0$ for $t \geq 1$, that is, there is a fixed stabilizing regularizer r_0 independent of t, and further we take $\mathcal{X} = \mathbb{R}^n$ (e.g., Rakhlin (2008); Hazan (2010); Shalev-Shwartz (2012)). Observe that in this case FTRL-Centered and FTRL-Proximal coincide. In this section, we show how this equivalence extends to adaptive regularizers (equivalently, adaptive learning rates) and composite objectives. This builds on the work of McMahan (2011), but we make some crucial improvements in order to obtain an exact equivalence result for a much broader class of Mirror Descent algorithms and then use this result to derive regret bounds.[3]

3. Subsequent to this work, Sra et al. (2016) analyzed AdaDelay, an adaptive stochastic gradient descent algorithm that allows for potentially increasing learning rates, and Joulani et al. (2016) provided a more general analysis of Mirror Descent algorithms with non-monotonic regularizers in the online setting. Extending the FTRL view presented here to handle such algorithms is an interesting direction for future work.

Adaptive Mirror Descent Even in the non-adaptive case, Mirror Descent can be expressed as a variety of different updates, some of which are equivalent but some of which are not;[4] in particular, the inclusion of the feasible set constraint $I_{\mathcal{X}}$ gives rise to distinct "lazy projection" vs "greedy projection" algorithms—this issue is discussed in detail in Appendix C. To define the adaptive Mirror Descent family of algorithms we first define the Bregman divergence with respect to a convex differentiable function[5] ϕ:

$$\mathcal{B}_\phi(u, v) = \phi(u) - \big(\phi(v) + \nabla\phi(v) \cdot (u - v)\big).$$

The Bregman divergence is the difference at u between ϕ and ϕ's first-order Taylor expansion taken at v. For example, if we take $\phi(u) = \|u\|^2$, then $\mathcal{B}_\phi(u, v) = \|u-v\|^2$.

An adaptive Mirror Descent algorithm is defined by a sequence of continuously differentiable incremental regularizers $r_0, r_1, \ldots,$ chosen so $r_{0:t}$ is strongly convex. From this, we define the time-indexed Bregman divergence $\mathcal{B}_{r_{0:t}}$,

$$\mathcal{B}_{r_{0:t}}(u, v) = r_{0:t}(u) - \big(r_{0:t}(v) + \nabla r_{0:t}(v) \cdot (u - v)\big).$$

The adaptive Mirror Descent update is then given by

$$\begin{aligned} \hat{x}_1 &= \arg\min_x r_0(x) \\ \hat{x}_{t+1} &= \arg\min_x \; g_t \cdot x + \alpha_t \Psi(x) + \mathcal{B}_{r_{0:t}}(x, \hat{x}_t). \end{aligned} \tag{27}$$

We use $\hat{x}$ to distinguish this update from an FTRL update we will introduce shortly. Building on the previous section, we allow the update to include an additional regularization term $\alpha_t\Psi(x)$. As before, typically $g_t \cdot x$ should be viewed as a subgradient approximation to a loss function ℓ_t; it will become clear that a key question is to what extent Ψ is also linearized.

Mirror Descent algorithms were introduced in Nemirovsky and Yudin (1983) for the optimization of a fixed non-smooth convex function, and generalized to Bregman divergences by Beck and Teboulle (2003). Bounds for the online case appeared in Warmuth and Jagota (1997); a general treatment in the online case for composite objectives (with a non-adaptive learning rate) is given by Duchi et al. (2010b). Following this existing literature, we might term the update of Eq. (27) Adaptive Composite-Objective Online Mirror Descent; for simplicity we simply refer to Mirror Descent in this work.

4. In particular, it is common to see updates written in terms of $\nabla r^\star(\theta)$ for a strongly convex regularizer r, based on the fact that $\nabla r^\star(-\theta) = \arg\min_x \theta \cdot x + r(x)$ (see Lemma 15 in Appendix B).
5. Certain properties of Bregman divergences require ϕ to be strictly convex, but it provides a convenient notation to define $\mathcal{B}_\phi(u, v)$ for any differentiable convex ϕ.

Implicit updates For the moment, we neglect the Ψ terms and consider convex per-round losses ℓ_t. While standard Online Gradient Descent (or Mirror Descent) linearizes the ℓ_t to arrive at the update $\hat{x}_{t+1} = \arg\min_x \; g_t \cdot x_t + \mathcal{B}_{r_{0:t}}(x, \hat{x}_t)$, we can define the alternative update

$$\hat{x}_{t+1} = \arg\min_x \; \ell_t(x) + \mathcal{B}_{r_{0:t}}(x, \hat{x}_t), \tag{28}$$

where we avoid linearizing the loss ℓ_t. This is often referred to as an implicit update, since for general convex ℓ_t it is no longer possible to solve for $\hat{x}_{t+1}$ in closed form. The implicit update was introduced by Kivinen and Warmuth (1997), and has more recently been studied by Kulis and Bartlett (2010).

Again considering the Ψ terms, the Mirror Descent update of Eq. (27) can be viewed as a partial implicit update: if the real loss per round is $\ell_t(x) + \alpha_t\Psi(x)$, we linearize the $\ell_t(x)$ term but not the $\Psi(x)$ term, taking $f_t(x) = g_t \cdot x + \alpha_t\Psi(x)$. Generally this is done for computational reasons, as for common choices of Ψ such as $\Psi(x) = \|x\|_1$ or $\Psi(x) = I_{\mathcal{X}}(x)$, the update can still be solved in closed form (or at least in a computationally efficient manner, e.g., by projection). However, while $\alpha_t\Psi$ is handled without linearization, we shall see that echoes of the past $\alpha_{1:t-1}\Psi$ are encoded in a linearized fashion in the current state $\hat{x}_t$.

On terminology In the unprojected and non-adaptive case, the Mirror Descent update $\hat{x}_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_r(x, \hat{x}_t)$ is equivalent to the FTRL update $x_{t+1} = \arg\min_x g_{1:t} \cdot x + r(x)$ (see Appendix C). In fact, Shalev-Shwartz (2012, Sec. 2.6) refers to this update (with linearized losses) explicitly as Mirror Descent.

In our view, the key property that distinguishes Mirror Descent from FTRL is that for Mirror Descent, the state of the algorithm is exactly $\hat{x}_t \in \mathbb{R}^n$, the current feasible point. For FTRL on the other hand, the state is a different vector in $\mathbb{R}^n$, for example $g_{1:t}$ for Dual Averaging. The indirectness of the FTRL representation makes it more flexible, since for example multiple values of $g_{1:t}$ can all map to the same coefficient value x_t.

6.1 Mirror Descent is an FTRL-Proximal Algorithm

We will show that the Mirror Descent update of Eq. (27) can be expressed as the FTRL-Proximal update given in Figure 2. In particular, consider a Mirror Descent algorithm defined by the choice of r_t for $t \geq 0$. Then, we define the FTRL-Proximal update

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + g^{(\Psi)}_{1:t-1} \cdot x + \alpha_t\Psi(x) + r^{\mathcal{B}}_{0:t}(x) \tag{29}$$

Mirror Descent

$$\hat{x}_{t+1} = \arg\min_x g_t \cdot x + \alpha_t \Psi(x) + \mathcal{B}_{r_{0:t}}(x, \hat{x}_t) \tag{27}$$

Mirror Descent as FTRL-Proximal

$$\hat{x}_{t+1} = \arg\min_x g_{1:t} \cdot x + g_{1:t-1}^{(\Psi)} \cdot x + \alpha_t \Psi(x) + r_0(x) + \sum_{s=1}^{t} \mathcal{B}_{r_s}(x, x_s)$$

$$= \arg\min_x g_{1:t} \cdot x + g_{1:t}^{(\Psi)} \cdot x + r_0(x) + \sum_{s=1}^{t} \mathcal{B}_{r_s}(x, x_s)$$

where $g_s^{(\Psi)}$ is a suitable subgradient from $\partial(\alpha_s \Psi)(x_{s+1})$

Figure 2: Mirror Descent as normally presented, and expressed as an equivalent FTRL-Proximal update.

for an appropriate choice $g_t^{(\Psi)} \in \partial(\alpha_t \Psi)(x_{t+1})$ (given below), where $r_t^{\mathcal{B}}$ is an incremental proximal regularizer defined in terms of r_t, namely

$$r_0^{\mathcal{B}}(x) \equiv r_0(x)$$
$$r_t^{\mathcal{B}}(x) \equiv \mathcal{B}_{r_t}(x, x_t) = r_t(x) - \big(r_t(x_t) + \nabla r_t(x_t) \cdot (x - x_t)\big) \qquad \text{for } t \geq 1.$$

Note that $r_t^{\mathcal{B}}$ is indeed minimized by x_t and $r_t^{\mathcal{B}}(x_t) = 0$. We require $g_t^{(\Psi)} \in \partial(\alpha_t \Psi)(x_{t+1})$ such that

$$g_{1:t} + g_{1:t}^{(\Psi)} + \nabla r_{0:t}^{\mathcal{B}}(x_{t+1}) = 0. \tag{30}$$

The dependence of $g_t^{(\Psi)}$ on x_{t+1} is not problematic, as $g_t^{(\Psi)}$ is not necessary to compute x_{t+1} using Eq. (29). To see (inductively) that we can always find a a $g_t^{(\Psi)}$ satisfying Eq. (30), note the subdifferential of the objective of Eq. (29) at x is

$$g_{1:t} + g_{1:t-1}^{(\Psi)} + \partial(\alpha_t \Psi)(x) + \nabla r_{0:t}^{\mathcal{B}}(x). \tag{31}$$

Since x_{t+1} is a minimizer, we know 0 is a subgradient, which implies there must be a subgradient $g_t^{(\Psi)} \in \partial(\alpha_t \Psi)(x_{t+1})$ that satisfies Eq. (30). The fact we use a subgradient of Ψ at x_{t+1} rather than x_t is a consequence of the fact we are replicating the behavior of a (partial) implicit update algorithm.

Finally, note the update

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + g_{1:t}^{(\Psi)} \cdot x + r_{0:t}^{\mathcal{B}}(x) \tag{32}$$

is equivalent to Eq. (29), since Equations (30) and (31) imply 0 is in the subgradient of the objective Eq. (29) at the x_{t+1} given by Eq. (32). This update is exactly an FTRL-Proximal update on the functions $f_t(x) = (g_t + g_t^{(\Psi)}) \cdot x$.

With these definitions in place, we can now state and prove the main result of this section, namely the equivalence of the two updates given in Figure 2:

Theorem 11 *The Mirror Descent update of Eq.* (27) *and the FTRL-Proximal update of Eq.* (29) *select identical points.*

Proof The proof is by induction on the hypothesis that $\hat{x}_t = x_t$. This holds trivially for $t = 1$, so we proceed by assuming it holds for t.

First we consider the x_t selected by the FTRL-Proximal algorithm of Eq. (29). Since x_t minimizes this objective, zero must be a subgradient at x_t. Letting $g_s^{(r)} = \nabla r_s(x_s)$ and noting $\nabla r_t^{\mathcal{B}}(x) = \nabla r_t(x) - \nabla r_t(x_t)$, we have $g_{1:t-1} + g_{1:t-1}^{(\Psi)} + \nabla r_{0:t-1}(x_t) - g_{0:t-1}^{(r)} = 0$ following Eq. (31). Since $x_t = \hat{x}_t$ by induction hypothesis, we can rearrange and conclude

$$-\nabla r_{0:t-1}(\hat{x}_t) = g_{1:t-1} + g_{1:t-1}^{(\Psi)} - g_{0:t-1}^{(r)}. \tag{33}$$

For Mirror Descent, the gradient of the objective in Eq. (27) must be zero for $\hat{x}_{t+1}$, and so there exists a $\hat{g}_t^{(\Psi)} \in \partial(\alpha_t \Psi)(\hat{x}_{t+1})$ such that

$$\begin{aligned}
0 &= g_t + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}(\hat{x}_{t+1}) - \nabla r_{0:t}(\hat{x}_t) \\
&= g_t + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}(\hat{x}_{t+1}) - \nabla r_{0:t-1}(\hat{x}_t) - g_t^{(r)} && \text{IH and } \nabla r_t(x_t) = g_t^{(r)} \\
&= g_t + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}(\hat{x}_{t+1}) + g_{1:t-1} + g_{1:t-1}^{(\Psi)} - g_{0:t-1}^{(r)} - g_t^{(r)} && \text{Using Eq. (33)} \\
&= g_{1:t} + g_{1:t-1}^{(\Psi)} + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}(\hat{x}_{t+1}) - g_{0:t}^{(r)} \\
&= g_{1:t} + g_{1:t-1}^{(\Psi)} + \hat{g}_t^{(\Psi)} + \nabla r_{0:t}^{\mathcal{B}}(\hat{x}_{t+1}).
\end{aligned}$$

The last line implies zero is a subgradient of the objective of Eq. (29) at $\hat{x}_{t+1}$, and so $\hat{x}_{t+1}$ is a minimizer. Since $r_{0:t}$ is strongly convex, this solution is unique and so $\hat{x}_{t+1} = x_{t+1}$. ■

6.2 Comparing Mirror Descent to the Native FTRL-Proximal Algorithm, and the Application to L_1 Regularization

Since we can write Mirror Descent as a particular FTRL update, we can now do a careful comparison to the direct application of Section 5 which gives the Native FTRL-Proximal algorithm. These two algorithms are given in Figure 3, expressed in a way that facilitates comparison.

Mirror Descent

$$\hat{x}_{t+1} = \arg\min_x \quad g_{1:t} \cdot x \quad + \quad g^{(\Psi)}_{1:t-1} \cdot x + \alpha_t \Psi(x) \quad + r^{\mathcal{B}}_{0:t}(x)$$

Native FTRL-Proximal

$$x_{t+1} = \arg\min_x \quad g_{1:t} \cdot x \quad + \quad \alpha_{1:t} \Psi(x) \quad + r^{\mathcal{B}}_{0:t}(x)$$

$$(A) \qquad (B) \qquad (C)$$

Figure 3: Mirror Descent expressed as an FTRL-Proximal algorithm compared to the Native FTRL-Proximal algorithm.

Both algorithms use a linear approximation to the loss functions ℓ_t, as seen in column (A) of Figure 3, and the same proximal regularization terms (C). The key difference is in how the non-smooth terms Ψ are handled: Mirror Descent approximates the past $\alpha_s \Psi(x)$ terms for $s < t$ using a subgradient approximation $g^{(\Psi)}_s \cdot x$, keeping only the current $\alpha_t \Psi(x)$ term explicitly. In Native FTRL-Proximal, on the other hand, we represent the full weight of the Ψ terms exactly as $\alpha_{1:t}\Psi(x)$. That is, Mirror Descent is applying significantly more linearization than Native FTRL-Proximal.

Why does this matter? As we will see in Section 6.3, there is no difference in the regret bounds, even though intuitively avoiding unnecessary linearization should be preferable. However, there can be a substantial practical differences for some choices of Ψ. In particular, we focus on the common and practically important case of L_1 regularization, where we take $\Psi(x) = \|x\|_1$. Such regularization terms are often used to produce sparse solutions (x_t where many $x_{t,i} = 0$). Models with few non-zeros can be stored, transmitted, and evaluated much more cheaply than the corresponding dense models.

As discussed in McMahan (2011), it is precisely the explicit representation of the full $\alpha_{1:t}\|x\|_1$ terms that lets Native FTRL produce much sparser solutions when compared with the composite-objective Mirror Descent update with L_1 regularization (equivalent to the FOBOS algorithm of Duchi and Singer (2009)). This argument also applies to Regularized Dual Averaging (RDA, a Native FTRL-Centered algorithm); Xiao (2009) presents experiments showing the advantages of RDA for producing sparse solutions. In the remainder of this section, we explore the application to L_1 regularization in more detail, in order to illustrate the effect of the additional linearization of the $\|x\|_1$ terms used by Mirror Descent as compared to the Native FTRL-Proximal algorithm.

Another way to understand this distinction is the previously mentioned difference in how the two algorithms maintain state. Mirror Descent has exactly one way to

represent a zero coefficient in the ith coordinate, namely $\hat{x}_{t,i} = 0$. The FTRL representation is significantly more flexible, since many state values, say any $g_{1:t,i} \in [-\lambda, \lambda]$, can all correspond to a zero coefficient. This means that FTRL can represent both "we have lots of evidence that $x_{t,i}$ should be zero" (as $g_{1:t,i} = 0$ for example), as well as "we think $x_{t,i}$ is zero right now, but the evidence is very weak" (as $g_{1:t,i} = \lambda$ for example). This means there may be a memory cost for training FTRL, as $g_{1:t,i} \neq 0$ still needs to be stored when $x_{t,i} = 0$, but the obtained models typically provide much better sparsity-accuracy tradeoffs (McMahan, 2011; McMahan et al., 2013).

This distinction is critical even in the non-adaptive case, and so we consider the simplest possible setting: a fixed regularizer $r_0(x) = \frac{1}{2\eta}\|x\|_2^2$ (with $r_t(x) = 0$ for $t \geq 1$), and $\alpha_t \Psi(x) = \lambda \|x\|_1$ for all t. The updates of Figure 3 then simplify to:

Mirror Descent

$$x_{t+1} = \arg\min_x \quad g_{1:t} \cdot x \quad + g^{(\Psi)}_{1:t-1} \cdot x + \lambda\|x\|_1 \quad + \frac{1}{2\eta}\|x\|_2^2 \tag{34}$$

Native FTRL

$$x_{t+1} = \arg\min_x \quad g_{1:t} \cdot x \quad + t\lambda\|x\|_1 \quad + \frac{1}{2\eta}\|x\|_2^2. \tag{35}$$

The key point is the Native FTRL algorithm uses a much stronger explicit L_1 penalty, $\alpha_{1:t} = t\lambda$ instead of just $\alpha_t = \lambda$.

The closed-form update We can write the update of Eq. (34) as a standard Mirror Descent update (that is, as an optimization over f_t and a regularizer centered at the current x_t):

$$\begin{aligned} x_{t+1} &= \arg\min_x g_t \cdot x + \lambda\|x\|_1 + \frac{1}{2\eta}\|x - x_t\|_2^2 \\ &= \arg\min_x \left(g_t - \frac{x_t}{\eta}\right) \cdot x + \lambda\|x\|_1 + \frac{1}{2\eta}\|x\|_2^2. \end{aligned} \tag{36}$$

The above update decomposes on a per-coordinate basis. Subgradient calculations show that for constants $a > 0$, $b \in \mathbb{R}$, and $\lambda \geq 0$, we have

$$\arg\min_{x \in \mathbb{R}} b \cdot x + \lambda|x| + \frac{a}{2}|x|^2 = \begin{cases} 0 & \text{when } |b| \leq \lambda \\ -\frac{1}{a}(b - \text{sign}(b)\lambda) & \text{otherwise.} \end{cases} \tag{37}$$

Thus, we can simplify Eq. (36) to

$$x_{t+1} = \begin{cases} 0 & \text{when } |g_t - \frac{x_t}{\eta}| \leq \lambda \\ x_t - \eta(g_t - \lambda) & \text{when } g_t - \frac{x_t}{\eta} > \lambda \quad (\text{implying } x_{t+1} < 0) \\ x_t - \eta(g_t + \lambda) & \text{otherwise} \quad (\text{i.e., } g_t - \frac{x_t}{\eta} < -\lambda \text{ and } x_{t+1} > 0). \end{cases}$$

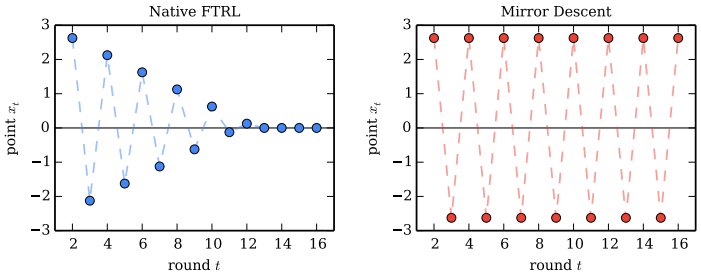


Figure 4: The points selected by Native FTRL and Mirror Descent on the one-dimensional example, using $\alpha_t\Psi(x) = \frac{1}{2}\|x\|_1$. Native FTRL quickly converges to $x^* = 0$, but Mirror Descent oscillates indefinitely.

If we choose $g_t^{(\Psi)} \in \partial\lambda\|x_{t+1}\|_1$ as

$$g_t^{(\Psi)} = \begin{cases} -\lambda & \text{when } x_{t+1} < 0 \\ \lambda & \text{when } x_{t+1} > 0\,, \\ x_t/\eta - g_t & \text{when } x_{t+1} = 0 \end{cases}$$

then Eq. (30) is satisfied, and the update becomes

$$x_{t+1} = x_t - \eta\big(g_t + g_t^{(\Psi)}\big)$$

in all cases, showing how the implicit update can be re-written in terms of a subgradient update using an appropriate subgradient approximation at the *next* point.

A One-Dimensional Example To illustrate the practical significance of the stronger explicit L_1 penalty used by Native FTRL, we compare the updates of Eq. (34) and Eq. (35) on a simple one-dimensional example. The gradients g_t satisfy $\|g_t\|_2 \le G$, and we use a feasible set of radius $R = 2G$. Both algorithms use the theory-recommended fixed learning rate $\eta = \frac{R}{G\sqrt{T}} = \frac{2}{\sqrt{T}}$ (see Section 3), against an adaptive adversary that selects gradients g_t as a function of x_t:

$$g_t = \begin{cases} -\frac{1}{2}(G+\lambda) & \text{when } t = 1 \\ -G & \text{when } t > 1 \text{ and } x_t \le 0 \\ G & \text{when } t > 1 \text{ and } x_t > 0\,. \end{cases}$$

Both algorithms select $x_1 = 0$, and since $g_1 = -\frac{1}{2}(G+\lambda)$ both algorithms select $x_2 = (G-\lambda)/\sqrt{T}$. After this, however, their behavior diverges: Mirror Descent will indefinitely oscillate between x_2 and $-x_2$ for any $\lambda < G$. On the other hand, FTRL learns that $x^* = 0$ is the optimal solution after a constant number of rounds, selecting $x_{t+1} = 0$ for any $t > \frac{G}{2\lambda} + \frac{1}{2}$. The details of this example are worked out in Appendix D

Figure 4 plots the points selected by the algorithms as a function of t, taking $G = 11$, $T = 16$, and $\lambda = 0.5$. This example clearly demonstrates that, though Mirror Descent and Native FTRL have the same regret bounds, Native FTRL is much more likely to produce sparse solutions and can also incur less actual regret.

6.3 Analysis of Mirror Descent as FTRL-Proximal

Having established the equivalence between Mirror Descent and a particular FTRL-Proximal update as given in Figure 2, we now use the general analysis techniques for FTRL developed in this work to prove regret bounds for Mirror Descent algorithm. This is accomplished by applying the Strong FTRL lemma to the FTRL-Proximal expression for Mirror Descent.

First, we observe that in the non-composite case (i.e., all $\alpha_t = 0$), then all $g_t^{(\Psi)} = 0$, and we can apply Theorem 2 directly to Eq. (29) for the loss functions $f_t(x) = g_t \cdot x$, which gives us

$$\text{Regret}(x^*, g_t) \le r^{\mathcal{B}}_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^T \|g_t\|^2_{(t),\star} = \sum_{t=1}^T \mathcal{B}_{r_t}(x^*, x_t) + \frac{1}{2}\sum_{t=1}^T \|g_t\|^2_{(t),\star}.$$

In the case of a composite-objective (nontrivial Ψ terms, including feasible set constraints such as $I_{\mathcal{X}}$), we will arrive at the same bound, but must refine our analysis somewhat to encompass the partial implicit update of Eq. (29). This is accomplished in the following theorem:

Theorem 12 *We consider the Mirror Descent update of Eq.* (27) *under the same conditions as Theorem 10. When we consider regret against* $f_t^{\Psi}(x) = g_t \cdot x + \alpha_t\Psi(x)$*, we have*

$$\text{Regret}(x^*, f^{\Psi}) \le r^{\mathcal{B}}_{0:T}(x^*) + \frac{1}{2}\sum_{t=1}^T \|g_t\|^2_{(t),\star}. \tag{38}$$

When we consider regret against only the functions $f_t(x) = g_t \cdot x$*, we have*

$$\text{Regret}(x^*, g_t) \le r^{\mathcal{B}}_{0:T}(x^*) + \alpha_{1:T}\Psi(x^*) + \frac{1}{2}\sum_{t=1}^T \|g_t\|^2_{(t),\star}. \tag{39}$$

The bound of Eq. (38) matches Duchi et al. (2011, Prop. 3),[6] and also encompasses Theorem 2 of Duchi et al. (2010b).[7]

Proof First, by Theorem 11, this algorithm can equivalently be expressed as in Eq. (32). To simplify bookkeeping, we define

$$\bar{f}_t(x) = g_t \cdot x + \bar{\Psi}_t(x) \qquad \text{where} \qquad \bar{\Psi}_t(x) = \alpha_t \Psi(x_{t+1}) + g_t^{(\Psi)} \cdot (x - x_{t+1}),$$

Then, the update

$$x_{t+1} = \arg\min_x \bar{f}_{1:t}(x) + r^{\mathcal{B}}_{0:t}(x) \tag{40}$$

is equivalent to Eq. (32), since the objectives differ only in constant terms. Note

$$\bar{\Psi}_t(x_{t+1}) = \alpha_t \Psi(x_{t+1}) \qquad \text{and} \qquad \forall x,\ \alpha_t \Psi(x) \geq \bar{\Psi}_t(x), \tag{41}$$

where the second claim uses the convexity of $\alpha_t \Psi$.

Observe that Eq. (40) defines an FTRL-Proximal algorithm—we can imagine the $\bar{f}_t$ are computed by a black-box given f_t which solves the optimization problem of Eq. (29) in order to compute $g_t^{(\Psi)}$. Thus, we can apply the Strong FTRL Lemma (Lemma 5). Again, the key is bounding the stability terms. Using $h_t(x) = \bar{f}_t(x) + r_t^{\mathcal{B}}(x)$, we have

$$\sum_{t=1}^T h_{1:t}(x_t) - h_{1:t}(x_{t+1}) - r_t(x_t) \leq \sum_{t=1}^T \frac{1}{2}\|g_t\|^2_{(t),\star} + \bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}),$$

using Corollary 8 as in Theorem 10.

We first consider regret against the functions $f_t^{\Psi}(x) = g_t \cdot x + \alpha_t \Psi(x)$. We can apply Lemma 5 to the functions $\bar{f}_t$, yielding

$$\text{Regret}(x^*, \bar{f}_t) \leq r^{\mathcal{B}}_{0:T}(x^*) + \sum_{t=1}^T \frac{1}{2}\|g_t\|^2_{(t),\star} + \bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}).$$

6. Mapping our notation to their notation, we have $f_t(x) = \ell_t(x) + \alpha_t \Psi(x) \Rightarrow \phi_t(x) = f_t(x) + \varphi(x)$ and $r_{1:t}(x) \Rightarrow \frac{1}{\eta}\psi_t(x)$. Dividing their Update (4) by η and using our notation, we arrive at exactly the update of Eq. (27). We can take $\eta = 1$ in their bound w.l.o.g.. Then, using the fact that ψ_t in their notation is $r_{1:t}$ in our notation, we have

$$\begin{aligned}
\mathcal{B}_{\psi_{t+1}}(x^*, x_{t+1}) - \mathcal{B}_{\psi_t}(x^*, x_{t+1}) &= \psi_{t+1}(x^*) - (\psi_{t+1}(x_{t+1}) + \nabla\psi_{t+1}(x_{t+1}) \cdot (x - x_{t+1})) \\
&\quad - \big(\psi_t(x^*) - (\psi_t(x_{t+1}) + \nabla\psi_t(x_{t+1}) \cdot (x - x_{t+1}))\big) \\
&= r_{t+1}(x^*) - \big(r_{t+1}(x_{t+1}) + \nabla r_{t+1}(x_{t+1}) \cdot (x - x_{t+1})\big) \\
&= \mathcal{B}_{r_{t+1}}(x^*, x_{t+1}).
\end{aligned}$$

7. We can take their $\alpha = 1$ and $\eta = 1$ w.l.o.g., and also assume our $\Psi(x_1) = 0$. Their r is our Ψ, and the implicitly take our $\alpha_t = 1$; their ψ is our r_0 (with our $r_1, \ldots, r_T$ all uniformly zero). Thus, their bound amounts (in our notation) to: Regret $\leq \mathcal{B}_{r_0}(x^*, x_1) + \frac{1}{2}\sum_{t=1}^T \|g_t\|^2_\star$, matching exactly the bound of our Theorem 12 (noting $r^{\mathcal{B}}_{0:t}(x^*) = \mathcal{B}_{r_0}(x^*, x_1)$ in this case).

However, this does not immediately yield a bound on regret against the f_t^{Ψ}. While $\bar{f}_t(x^*) \leq f_t^{\Psi}(x^*)$, our actual loss $f_t^{\Psi}(x_t)$ could be larger than $\bar{f}_t(x_t)$. Thus, in order to bound regret against f_t^{Ψ}, we must add terms $f_t^{\Psi}(x_t) - \bar{f}_t(x_t) = \alpha_t \Psi(x_t) - \bar{\Psi}_t(x_t)$. This gives

$$\begin{aligned}
\text{Regret}(x^*, f_t^{\Psi}) &\leq \text{Regret}(x^*, \bar{f}_t) + \sum_{t=1}^T \alpha_t \Psi(x_t) - \bar{\Psi}_t(x_t) \\
&\leq r^{\mathcal{B}}_{0:T}(x^*) + \sum_{t=1}^T \frac{1}{2}\|g_t\|^2_{(t),\star} + \bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}) + \alpha_t \Psi(x_t) - \bar{\Psi}_t(x_t) \\
&= r^{\mathcal{B}}_{0:T}(x^*) + \sum_{t=1}^T \frac{1}{2}\|g_t\|^2_{(t),\star} + \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}),
\end{aligned}$$

where the equality uses $\bar{\Psi}_t(x_{t+1}) = \alpha_t \Psi(x_{t+1})$. Recalling $\sum_{t=1}^T \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1}) \leq 0$ from the proof of Theorem 10 completes the proof of Eq. (38).

For Eq. (39), applying Lemma 5 with $r_t \leftarrow \bar{\Psi}_t + r_t^{\mathcal{B}}$ and $f_t(x) \leftarrow g_t \cdot x$ yields

$$\text{Regret}(x^*, g_t) \leq r^{\mathcal{B}}_{0:T}(x^*) + \bar{\Psi}_{1:t}(x^*) + \sum_{t=1}^T \frac{1}{2}\|g_t\|^2_{(t),\star} + \bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}).$$

Eq. (41) implies $\bar{\Psi}_t(x_t) - \bar{\Psi}_t(x_{t+1}) \leq \alpha_t \Psi(x_t) - \alpha_t \Psi(x_{t+1})$, and so the sum of these terms again vanishes. Finally, observing $\bar{\Psi}_{1:t}(x^*) \leq \alpha_{1:t} \Psi(x^*)$ completes the proof. ■

7. Conclusions

Using a general and modular analysis, we have presented a unified view of a wide family of algorithms for online convex optimization that includes Dual Averaging, Mirror Descent, FTRL, and FTRL-Proximal, recovering and sometimes improving regret bounds from many earlier works. Our emphasis has been on the case of adaptive regularizers, but the results recover those for a fixed learning rate or regularizer as well.

Acknowledgments

The author thanks Daniel Golovin, Francesco Orabona, and the anonymous reviewers for useful feedback on this work.

References

Peter Auer, Nicolò Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 2002.

Peter L. Bartlett, Elad Hazan, and Alexander Rakhlin. Adaptive online gradient descent. In *Neural Information Processing Systems (NIPS)*, 2007.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3), 2003.

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research (JMLR)*, 2, 2002.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

Nicolò Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50, 2004.

John Duchi and Yoram Singer. Efficient learning using forward-backward splitting. In *Neural Information Processing Systems (NIPS)*. 2009.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Conference On Learning Theory (COLT)*, 2010a.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *Conference On Learning Theory (COLT)*, 2010b.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 12, 2011.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, 1995.

Geoffrey J. Gordon. Regret bounds for prediction problems. In *Conference on Computational Learning Theory (COLT)*, 1999.

J. Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games, Volume III*, 1957.

Elad Hazan. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *Conference On Learning Theory (COLT)*, 2008.

Elad Hazan. The convex optimization approach to regret minimization, 2010.

Elad Hazan. Introduction to online convex optimization, 2015.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69, December 2007.

Pooria Joulani, Andras Gyorgy, and Csaba Szepesvari. A unified modular analysis of online and stochastic optimization: Adaptivity, optimism, non-convexity. In *OPT 2016 (NIPS Workshop on Optimization)*, 2016.

Sham M. Kakade and Shai Shalev-Shwartz. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *Neural Information Processing Systems (NIPS)*, 2008.

Sham M. Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Regularization techniques for learning with matrices. *Journal of Machine Learning Research (JMLR)*, 2012.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and Systems Sciences*, 71(3), 2005.

Jyrki Kivinen and Manfred Warmuth. Exponentiated Gradient Versus Gradient Descent for Linear Predictors. *Journal of Information and Computation*, 132, 1997.

Brian Kulis and Peter Bartlett. Implicit online learning. In *International Conference on Machine Learning (ICML)*, 2010.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2), February 1994.

H. Brendan McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

H. Brendan McMahan and Jacob Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In *Neural Information Processing Systems (NIPS)*, 2013.

H. Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In *Conference on Learning Theory (COLT)*, 2014.

H. Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. In *Conference On Learning Theory (COLT)*, 2010.

H. Brendan McMahan, Gary Holt, David Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Mar Hrafnkelsson, Tom Boulos, and Jeremy Kubica. Ad click prediction: a view from the trenches. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.

A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. 1983.

Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*. Applied optimization. Kluwer Academic Publ., Boston, Dordrecht, London, 2004.

Yurii Nesterov. Gradient methods for minimizing composite objective function. Technical Report Technical Report 2007/76, Catholic University of Louvain, Center for Operations Research and Econometrics, 2007.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Math. Program.*, 120(1), April 2009.

Francesco Orabona. Dimension-free exponentiated gradient. In *Neural Information Processing Systems (NIPS)*, 2013.

Alexander Rakhlin. Lecture notes on online learning, 2008.

Alexander Rakhlin, Sayan Mukherjee, and Tomaso Poggio. Stability results in learning theory. *Analysis and Applications*, 2005.

Ralph T. Rockafellar. *Convex Analysis (Princeton Landmarks in Mathematics and Physics)*. Princeton University Press, 1970.

Shai Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.

Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 2012.

Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 2007.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research (JMLR)*, 2010.

Suvrit Sra, Adams Wei Yu, Mu Li, and Alex Smola. Adadelay: Delay adaptive distributed stochastic optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 957–965, 2016.

Gilles Stoltz. *Incomplete information and internal regret in prediction of individual sequences*. PhD thesis, Paris-Sud XI University, 2005.

Gilles Stoltz. *Contributions to the sequential prediction of arbitrary sequences: applications to the theory of repeated games and empirical studies of the performance of the aggregation of experts*. Habilitation à diriger des recherches, Université Paris-Sud, 2011.

Matthew Streeter and H. Brendan McMahan. Less regret via online conditioning. 2010. URL `http://arxiv.org/abs/1002.4862`.

Matthew Streeter and H. Brendan McMahan. No-regret algorithms for unconstrained online convex optimization. In *Neural Information Processing Systems (NIPS)*, 2012.

Volodimir G. Vovk. Aggregating strategies. In *Workshop on Computational Learning Theory*, 1990.

M. K. Warmuth and A. K. Jagota. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. In *International Symposium on Artificial Intelligence and Mathematics*, 1997.

Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. In *Neural Information Processing Systems (NIPS)*, 2009.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, 2003.

Appendix A. The Standard FTRL Lemma

The following lemma is a well-known tool for the analysis of FTRL algorithms (see Kalai and Vempala (2005); Hazan (2008), Hazan (2010, Lemma 1), and Shalev-Shwartz (2012, Lemma 2.3)):

Lemma 13 (Standard FTRL Lemma) *Let f_t be a sequence of arbitrary (possibly non-convex) loss functions, and let r_t be arbitrary non-negative regularization functions, such that $x_{t+1} = \arg\min_x h_{0:t}(x)$ is well defined (recall $h_{0:t}(x) = f_{1:t}(x) + r_{0:t}(x)$). Then, the algorithm that selects these x_t achieves*

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \sum_{t=1}^{T} f_t(x_t) - f_t(x_{t+1}).$$

The proof of this lemma (e.g., McMahan and Streeter (2010, Lemma 3)) relies on showing that if one could run the Be-The-Leader algorithm by selecting $x_t = \arg\min_x f_{1:t}(x)$ (which requires peaking ahead at f_t to choose x_t), then the algorithm's regret is bounded above by zero.

However, as we see by comparing Theorem 2 and 14 (stated below), this analysis loses a factor of $1/2$ on one of the terms. The key is that being the leader is actually *strictly better* than always using the post-hoc optimal point, a fact that is not captured by the Standard FTRL Lemma. To prove the Strong FTRL Lemma, rather than first analyzing the Be-The-Leader algorithm and showing it has no regret, the key is to directly analyze the FTL algorithm (using a similar inductive argument). The proofs are also similar in that in both the basic bound is proved first for regret against the functions h_t (equivalently, the regret for FTL without regularization), and this bound is then applied to the regularized functions and re-arranged to bound regret against the f_t.

Using Lemma 13, we can prove the following weaker version of Theorem 2:

Theorem 14 Weak FTRL-Proximal Bound *Consider Setting 1, and further suppose the r_t are chosen such that $h_{0:t} = r_{0:t} + f_{1:t}$ is 1-strongly-convex w.r.t. some norm $\|\cdot\|_{(t)}$, and further the r_t are proximal, that is x_t is a global minimizer of r_t. Then, choosing any $g_t \in \partial f_t(x_t)$ on each round, for any $x^* \in \mathbb{R}^n$,*

$$\text{Regret}(x^*) \leq r_{0:T}(x^*) + \sum_{t=1}^{T} \|g_t\|^2_{(t),\star}.$$

We prove Theorem 14 using strong smoothness via Lemma 7. An alternative proof that uses strong convexity directly is also possible, closely following Shalev-Shwartz (2012, Sec. 2.5.2).

Proof of Theorem 14 Applying Lemma 13, it is sufficient to consider a fixed t and upper bound $f_t(x_t) - f_t(x_{t+1})$. For this fixed t, define a helper function $\phi_1(x) = f_{1:t-1}(x) + r_{0:t}(x)$. Observe $x_t = \arg\min_x \phi_1(x)$ since x_t is a minimizer of $r_t(x)$, and by definition of the update x_t is a minimizer of $f_{1:t-1}(x) + r_{0:t-1}(x)$. Let $\phi_2(x) = \phi_1(x) + f_t(x) = h_{0:t}(x)$, so ϕ_2 is 1-strongly convex with respect to $\|\cdot\|_{(t)}$ by assumption, and $x_{t+1} = \arg\min_x \phi_2(x)$. Then, we have

$$\begin{aligned}
f_t(x_t) - f_t(x_{t+1}) &\leq g_t \cdot (x_t - x_{t+1}) && \text{Convexity of } f_t \text{ and } g_t \in \partial f_t(x_t) \\
&\leq \|g_t\|_{(t),\star} \|x_t - x_{t+1}\|_{(t)} && \text{Property of dual norms} \\
&\leq \|g_t\|_{(t),\star} \|g_t\|_{(t),\star} = \|g_t\|^2_{(t),\star}. && \text{Using Eq. (19) from Lemma 7}
\end{aligned}$$

■

Interestingly, it appears difficult to achieve a tight (up to constant factors) analysis of non-proximal FTRL algorithms (e.g., FTRL-Centered algorithms like Dual Averaging) using Lemma 13. The Strong FTRL Lemma, however, allowed us to accomplish this.

Appendix B. Proofs For Section 4.2

We first state a standard technical result (see Shalev-Shwartz (2007, Lemma 15)):

Lemma 15 *Let ψ be 1-strongly convex w.r.t. $\|\cdot\|$, so $\psi^\star$ is 1-strongly smooth with respect to $\|\cdot\|_\star$. Then,*

$$\|\nabla\psi^\star(z) - \nabla\psi^\star(z')\| \leq \|z - z'\|_\star, \tag{42}$$

and

$$\arg\min_x g \cdot x + \psi(x) = \nabla\psi^\star(-g). \tag{43}$$

In order to prove Lemma 7, we first prove a somewhat easier result:

Lemma 16 *Let $\phi_1 : \mathbb{R}^n \to \mathbb{R}$ be strongly convex w.r.t. norm $\|\cdot\|$, and let $x_1 = \arg\min_x \phi_1(x)$, and define $\phi_2(x) = \phi_1(x) + b \cdot x$ for $b \in \mathbb{R}^n$. Letting $x_2 = \arg\min_x \phi_2(x)$, we have*

$$\phi_2(x_1) - \phi_2(x_2) \leq \frac{1}{2}\|b\|^2_\star, \qquad \text{and} \qquad \|x_1 - x_2\| \leq \|b\|_\star.$$

Proof We have

$$-\phi_1^\star(0) = -\max_x 0 \cdot x - \phi_1(x) = \min_x \phi_1(x) = \phi_1(x_1).$$

and similarly,

$$-\phi_1^\star(-b) = -\max_x -b \cdot x - \phi_1(x) = \min_x b \cdot x + \phi_1(x) = b \cdot x_2 + \phi_1(x_2).$$

Since $x_1 = \nabla\phi_1^\star(0)$ and $\phi_1^\star$ is strongly-smooth (Lemma 9), Eq. (20) gives

$$\phi_1^\star(-b) \le \phi_1^\star(0) + x_1 \cdot (-b - 0) + \frac{1}{2}\|b\|_\star^2.$$

Combining these facts, we have

$$\begin{aligned}\phi_1(x_1) + b \cdot x_1 - \phi_1(x_2) - b \cdot x_2 &= -\phi_1^\star(0) + b \cdot x_1 + \phi_1^\star(-b) \\ &\le -\phi_1^\star(0) + b \cdot x_1 + \phi_1^\star(0) + x_1 \cdot (-b) + \frac{1}{2}\|b\|_\star^2 \\ &= \frac{1}{2}\|b\|_\star^2.\end{aligned}$$

For the second part, observe $\nabla\phi_1^\star(0) = x_1$, and $\nabla\phi_1^\star(-b) = x_2$ and so $\|x_1 - x_2\| \le \|b\|_\star$, using both parts of Lemma 15. ■

Proof of Lemma 7 We are given that $\phi_2(x) = \phi_1(x) + \psi(x)$ is 1-strongly convex w.r.t. $\|\cdot\|$. The key trick is to construct an alternative ϕ_1' that is also 1-strongly convex with respect to this same norm, but has x_1 as a minimizer. Fortunately, this is easily possible: define $\phi_1'(x) = \phi_1(x) + \psi(x) - b \cdot x$, and note ϕ_1 is 1-strongly convex w.r.t. $\|\cdot\|$ since it differs from ϕ_2 only by a linear function. Since $b \in \partial\psi(x_1)$ it follows that 0 is in $\partial(\psi(x) - b \cdot x)$ at $x = x_1$, and so $x_1 = \arg\min \phi_1'(x)$. Note $\phi_2(x) = \phi_1'(x) + b \cdot x$. Applying Lemma 16 to ϕ_1' and ϕ_2 completes the proof, noting for any x' we have $\phi_2(x_1) - \phi_2(x') \le \phi_2(x_1) - \phi_2(x_2)$. ■

Proof of Corollary 8 Let $x_2' = \arg\min_x \phi_1(x) + \psi(x)$, so by Lemma 7, we have

$$\phi_1(x_1) + \psi(x_1) - \phi_1(x_2') - \psi(x_2') \le \frac{1}{2}\|b\|_\star^2, \tag{44}$$

Then, noting $\phi_1(x_2') + \psi(x_2') \le \phi_1(x_2) + \psi(x_2)$ by definition, we have

$$\begin{aligned}\phi_2(x_1) - \phi_2(x_2) &= \phi_1(x_1) + \psi(x_1) + \Psi(x_1) - \phi_1(x_2) - \psi(x_2) - \Psi(x_2) \\ &\le \phi_1(x_1) + \psi(x_1) + \Psi(x_1) - \phi_1(x_2') - \psi(x_2') - \Psi(x_2) \\ &\le \frac{1}{2}\|b\|_\star^2 + \Psi(x_1) - \Psi(x_2). && \text{Using Eq. (44).}\end{aligned}$$

Noting that $\phi_2(x_1) - \phi_2(x') \le \phi_2(x_1) - \phi_2(x_2)$ for any x' completes the proof. ■

Appendix C. Non-Adaptive Mirror Descent and Projection

Non-adaptive Mirror Descent algorithms have appeared in the literature in a variety of forms, some equivalent and some not. In this section we briefly review these connections. We first consider the unconstrained case, where the domain of the convex functions is taken to be $\mathbb{R}^n$, and there is no constraint that $x_t \in \mathcal{X}$.

Explicit	$\theta_{t+1} = \theta_t - g_t$ $x_{t+1} = \nabla R^\star(\theta_{t+1})$	$\theta_{t+1} = \nabla R(x_t) - g_t$ $x_{t+1} = \nabla R^\star(\theta_{t+1})$
Implicit	$x_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_R(x, x_t)$	
FTRL	$x_{t+1} = \arg\min_x g_{1:t} \cdot x + R(x)$	

Figure 5: Four equivalent expressions for unconstrained Mirror Descent defined by a strongly convex regularizer R. The top-right expression is from by Beck and Teboulle (2003), while the top-left expression matches the presentation of Shalev-Shwartz (2012, Sec 2.6).

C.1 The Unconstrained Case

Figure 5 summarizes a set of equivalent expressions for the unconstrained non-adaptive Mirror Descent algorithm. Here we assume R is a strongly-convex regularizer which is differentiable on $\mathbb{R}^n$ so that the corresponding Bregman divergence $\mathcal{B}_R$ is defined. Recall from Lemma 15,

$$\nabla R^\star(-g) = \arg\min_x g \cdot x + R(x). \tag{45}$$

We now prove that these updates are equivalent:

Theorem 17 *The four updates in Figure 5 are equivalent.*

Proof It is sufficient to prove three equivalences:

- The two explicit formulations are equivalent. For the right-hand version, we have $x_t = \nabla R^\star(\theta_t) = \arg\min_x -\theta_t \cdot x + R(x)$ using Eq. (45). The optimality of x_t for this minimization implies $0 = -\theta_t + \nabla R(x_t)$, or $\nabla R(x_t) = \theta_t$.
- Explicit $\Leftrightarrow$ FTRL: Immediate from Eq. (45) and the fact that $\theta_{t+1} = -g_{1:t}$.
- Implicit $\Leftrightarrow$ FTRL: That is,

$$\hat{x}_{t+1} = \arg\min_x g_t \cdot x + \mathcal{B}_R(x, \hat{x}_t) \qquad \text{and} \tag{46}$$

$$x_{t+1} = \arg\min_x g_{1:t} \cdot x + R(x) \tag{47}$$

are equivalent. The proof is by induction on the hypothesis $x_t = \hat{x}_t$. We must have from Eq. (46) and the IH that $g_t + \nabla R(\hat{x}_{t+1}) - \nabla R(x_t) = 0$, and from Eq. (47) applied to $t-1$ we must have $\nabla R(x_t) = -g_{1:t-1}$, and so $\nabla R(\hat{x}_{t+1}) = -g_{1:t}$. Then, we have the gradient of the objective of Eq. (47) at $\hat{x}_{t+1}$ is $g_{1:t} + \nabla R(\hat{x}_{t+1}) = 0$, and since the optimum of Eq. (47) is unique, we must have $\hat{x}_{t+1} = x_{t+1}$. The same general technique is used to prove the more general result for adaptive composite Mirror Descent in Theorem 11.

■

C.2 The Constrained Case: Projection onto $\mathcal{X}$

Even in the non-adaptive case (fixed R), the story is already more complicated when we constrain the algorithm to select from a convex set $\mathcal{X}$. For this section we take $R(x) = r(x) + I_{\mathcal{X}}(x)$ where r is continuously differentiable on $\operatorname{dom} I_{\mathcal{X}} = \mathcal{X}$.

In this setting, the two explicit algorithms given in the previous table are no longer equivalent. Figure 6 gives the two resulting families of updates. The classic Mirror Descent algorithm corresponds to the right-hand column, and follows the presentation of Beck and Teboulle (2003). This algorithm can be expressed as a greedy projection, and when $r(x) = \frac{1}{2\eta}\|x\|_2^2$ gives a constant learning rate version of the projected Online Gradient Descent algorithm of Zinkevich (2003). The Lazy column corresponds for example to the "Online Gradient Descent with lazy projections" algorithm (Shalev-Shwartz, 2012, Cor. 2.16).

The relationship to these projection algorithms is made explicit by the last row in the table. We define the projection operator onto $\mathcal{X}$ with respect to Bregman divergence $\mathcal{B}_r$ by

$$\Pi^r_{\mathcal{X}}(u) \equiv \operatorname*{arg\,min}_{x \in \mathcal{X}} \mathcal{B}_r(x, u).$$

Expanding the definition of the Bregman divergence, dropping terms independent of x since they do not influence the arg min, and replacing the explicit $x \in \mathcal{X}$ constraint with an $I_{\mathcal{X}}$ term in the objective, we have the equivalent expression

$$\Pi^r_{\mathcal{X}}(u) = \operatorname*{arg\,min}_x r(x) - \nabla r(u) \cdot x + I_{\mathcal{X}}(x). \tag{48}$$

The names Lazy and Greedy come from the manner in which the projection is used. For Lazy-Projection, the state of the algorithm is simply $g_{1:t}$ which can be updated without any need for projection; projection is applied lazily when we need to calculate x_{t+1}. For the Greedy-Projection algorithm on the other, the state of the algorithm is essentially x_t, and in particular u_{t+1} cannot be calculated without knowledge of x_t, the result of greedily applying projection on the previous round. If the g_t are really linear approximations to some f_t, however, a projection is needed on each round for both algorithms to produce x_t so $g_t \in \partial f_t(x_t)$ can be computed.

Both the Lazy and Greedy families can be analyzed (including in the more general adaptive case) using the techniques introduced in this paper. The Lazy family corresponds to the Native FTRL update of Section 5, namely

$$x_{t+1} = \operatorname*{arg\,min}_x g_{1:t} \cdot x + I_{\mathcal{X}}(x) + r_{0:t}(x),$$

which we encode as a single fixed non-smooth penalty $\Psi = I_{\mathcal{X}}$ which arrives on the first round: $\alpha_1 = 1$ and $\alpha_t = 0$ for $t > 1$.

	Lazy	Greedy
Explicit	$\theta_{t+1} = \theta_t - g_t$ $x_{t+1} = \nabla R^\star(\theta_{t+1})$	$\theta_{t+1} = \nabla r(x_t) - g_t$ $x_{t+1} = \nabla R^\star(\theta_{t+1})$
Implicit		$x_{t+1} =$ $\operatorname*{arg\,min}_x g_t \cdot x + \mathcal{B}_r(x, x_t) + I_{\mathcal{X}}(x)$
FTRL	$x_{t+1} = \operatorname*{arg\,min}_x g_{1:t} \cdot x + R(x)$	$x_{t+1} =$ $\operatorname*{arg\,min}_x \left(g_{1:t} + g^{(\Psi)}_{1:t-1}\right) \cdot x + R(x)$
Projection	$u_{t+1} = \operatorname*{arg\,min}_x g_{1:t} \cdot x + r(x)$ $x_{t+1} = \Pi^r_{\mathcal{X}}(u_{t+1})$	$u_{t+1} = \operatorname*{arg\,min}_u g_t \cdot u + \mathcal{B}_r(u, x_t)$ $= \nabla r^\star(\nabla r(x_t) - g_t)$ $x_{t+1} = \Pi^r_{\mathcal{X}}(u_{t+1})$

Figure 6: The Lazy and Greedy families of Mirror Descent algorithms, defined via $R(x) = r(x) + I_{\mathcal{X}}(x)$, where r is a differentiable strongly-convex regularizer. These families are not equivalent, but the different updates in each column are equivalent.

The Greedy-Projection Mirror Descent algorithms, on the other hand, can be thought of us receiving loss functions $g_t \cdot x + I_{\mathcal{X}}(x)$ on each round: that is, we have $\alpha_t = 1$ for all t. This family is analyzed using the techniques from Section 6. In this setting, embedding $I_{\mathcal{X}}(x)$ inside R can be seen as a convenience for defining $\nabla R^\star$,

$$\nabla R^\star(-g) = \operatorname*{arg\,min}_x g \cdot x + r(x) + I_{\mathcal{X}}(x). \tag{49}$$

We have the following equivalence results:

Theorem 18 *The Lazy-Explicit, Lazy-FTRL, and Lazy-Projection updates from the left column of Figure 6 are equivalent.*

Proof First, we show Lazy-Explicit is equivalent to Lazy-FTRL. Iterating the definition of θ_{t+1} in the explicit version gives $\theta_{t+1} = -g_{1:t}$, and so the second line in the update becomes exactly $x_{t+1} = \arg\min_x g_{1:t} \cdot x + R(x)$.

Next, we show that Lazy-Projection is equivalent to the Lazy-Explicit update. Optimality conditions for the minimization that defines u_{t+1} imply $\nabla r(u_{t+1}) = -g_{1:t}$. Then, the second equation in the Lazy-Projection update becomes

$$\begin{aligned} x_{t+1} = \Pi^r_{\mathcal{X}}(u_{t+1}) &= \operatorname*{arg\,min}_x r(x) - \nabla r(u_{t+1}) \cdot x + I_{\mathcal{X}}(x) && \text{Using Eq. (48).} \\ &= \operatorname*{arg\,min}_x g_{1:t} \cdot x + r(x) + I_{\mathcal{X}}(x), && \text{Since } \nabla r(u_{t+1}) = -g_{1:t}. \end{aligned}$$

which is exactly the Lazy-FTRL update (recalling $R(x) = r(x) + I_{\mathcal{X}}(x)$). ■

Theorem 19 *The Explicit, Implicit, FTRL, and Projected updates in the "Greedy" column of Figure 6 are equivalent.*

Proof We prove the result via the following chain of equivalences:

- Greedy-Explicit ⇔ Greedy-Implicit (c.f. Beck and Teboulle (2003, Prop 3.2)). We again use $\hat{x}$ for the points selected by the implicit version,

$$\begin{aligned}\hat{x}_{t+1} &= \arg\min_x g_t \cdot x + \mathcal{B}_r(x, x_t) + I_{\mathcal{X}}(x) \\ &= \arg\min_x g_t \cdot x + r(x) - \nabla r(x_t) \cdot x + I_{\mathcal{X}}(x),\end{aligned}$$

  where we have dropped terms independent of x in the arg min. On the other hand, plugging in the definition of θ_{t+1}, the explicit update is

$$x_{t+1} = \arg\min_x -(\nabla r(x_t) - g_t) \cdot x + r(x) + I_{\mathcal{X}}(x), \tag{50}$$

  which is equivalent.

- Greedy-Implicit ⇔ Greedy-FTRL: This is a special case of Theorem 11, taking $r_0 \leftarrow r + I_{\mathcal{X}}$, $r_t(x) = r_t^{\mathcal{B}}(x) = 0$ for $t \geq 1$, and $\alpha_t \Psi(x) = I_{\mathcal{X}}(x)$ for $t \geq 1$.

- When $I_{\mathcal{X}} = I_{\mathcal{X}}$, Projection is equivalent to the Greedy-Explicit expression. First, note we can re-write the Greedy-Projection update as

$$\begin{aligned}u_{t+1} &= \arg\min_u -(\nabla r(x_t) - g_t) \cdot u + r(u) \\ x_{t+1} &= \arg\min_{x \in \mathcal{X}} \mathcal{B}_r(x, u_{t+1}).\end{aligned}$$

  Optimality conditions for the first expression imply $\nabla r(u_{t+1}) = \nabla r(x_t) - g_t$. Then, the second update becomes

$$\begin{aligned}x_{t+1} &= \Pi_{\mathcal{X}}^r(u_{t+1}) \\ &= \arg\min_x r(x) - \nabla r(u_{t+1}) \cdot x + I_{\mathcal{X}}(x) && \text{Using Eq. (48).} \\ &= \arg\min_x r(x) - (\nabla r(x_t) - g_t) \cdot x + I_{\mathcal{X}}(x), && \text{Since } \nabla r(u_{t+1}) = \nabla r(x_t) - g_t.\end{aligned}$$

  which is equivalent to the Greedy-Explicit update, e.g., Eq. (50). ■

Appendix D. Details for the One-Dimensional L_1 Example

In this section we provide details for the one-dimensional example presented in Section 6.2. Suppose gradients g_t satisfy $\|g_t\|_2 \leq G$, and we use a feasible set of radius $R = 2G$, so the theory-recommended fixed learning rate is $\eta = \frac{R}{G\sqrt{T}} = \frac{2}{\sqrt{T}}$ (see Section 3).

We first consider the behavior of Mirror Descent: we construct the example so that the algorithm oscillates between two points, $\hat{x}$ and $-\hat{x}$ (allowing the possibility that $\hat{x} = -\hat{x} = 0$). Given alternating gradients of $+G$ and $-G$, in such an oscillation the distance one update takes us must be $\eta(G - \lambda)$, assuming $\lambda < G$. Thus, we can cause the algorithm to oscillate between $\hat{x} = (G - \lambda)/\sqrt{T}$ and $-\hat{x}$. We assume an initial $g_1 = -\frac{1}{2}(G + \lambda)$, which gives us $x_2 = \hat{x}$ for both Mirror Descent and FTRL when $x_1 = 0$.

This construction implies that for any constant L_1 penalty $\lambda < G$, Mirror Descent will never learn the optimal solution $x^* = 0$ (note that after the first round, we can view the g_t as being for example the subgradients of $f_t(x) = G\|x\|_1$). The points x_t selected by Mirror Descent, the gradients, and the subgradients of the L_1 penalty are given by the following table:

t	1	2	3	4	5	$\cdots$
g_t	g_1	G	$-G$	G	$-G$	$\cdots$
x_t	0	$\hat{x}$	$-\hat{x}$	$\hat{x}$	$-\hat{x}$	$\cdots$
$g_t^{(\Psi)}$	λ	$-\lambda$	λ	$-\lambda$	λ	$\cdots$

While we have worked from the standard Mirror Descent update, Eq. (36), it is instructive to verify the FTRL-Proximal representation is indeed equivalent. For example, using the values from the table, for x_5 we have

$$\begin{aligned}x_5 &= \arg\min_x g_{1:4} \cdot x + g_{1:3}^{(\Psi)} \cdot x + \lambda\|x\|_1 + \frac{1}{2\eta}\|x\|_2^2 \\ &= \arg\min_x (g_1 + G) \cdot x + \lambda \cdot x + \lambda\|x\|_1 + \frac{1}{2\eta}\|x\|_2^2 = -\frac{G - \lambda}{\sqrt{T}} = -\hat{x},\end{aligned}$$

where we solve the argmin by applying Eq. (37) with $b = g_1 + G + \lambda$.

Now, contrast this with the FTRL update of Eq. (35); we can solve this update in closed form using Eq. (37). First, note that FTRL will not oscillate in the same way, unless $\lambda = 0$. We have that $x_{t+1} = 0$ whenever $|g_{1:t}| < t\lambda$. Note that $g_{1:t}$ oscillates between $g_{1:t} = g_1 = -\frac{1}{2}(G + \lambda)$ on odd rounds t, and $g_{1:t} = g_1 + G = \frac{1}{2}G - \frac{1}{2}\lambda$ on even rounds. Since the magnitude of $g_{1:t}$ is larger on odd rounds, if we have $\frac{1}{2}(G+\lambda) \leq t\lambda$ then x_{t+1} will always be zero; re-arranging, this amounts to $\lambda \geq \frac{G}{2t-1}$. Thus, as with Mirror Descent, we need $\lambda \geq G$ to have $x_2 = 0$ (plugging in $t = 1$) but on subsequent rounds a *much* smaller λ is sufficient to produce sparsity. In the extreme case, taking $\lambda = G/(2T-1)$ is sufficient to ensure $x_T = 0$, whereas we need a λ value almost $2T$ times larger in order to get $x_T = 0$ from Mirror Descent.

Journal of Machine Learning Research 18 (2017) 1-35 Submitted 11/14; Revised 11/16; Published 9/17

A distributed block coordinate descent method for training l_1 regularized linear classifiers

Dhruv Mahajan DHRUVM@FB.COM
Applied Machine Learning Group
Facebook Research
Menlo Park, CA 94025, USA

S. Sathiya Keerthi KEERTHI@MICROSOFT.COM
Office Data Science Group
Microsoft
Mountain View, CA 94043, USA

S. Sundararajan SSRAJAN@MICROSOFT.COM
Microsoft Research
Bangalore, India

Editor: Vishwanathan S V N

Abstract

Distributed training of l_1 regularized classifiers has received great attention recently. Most existing methods approach this problem by taking steps obtained from approximating the objective by a quadratic approximation that is decoupled at the individual variable level. These methods are designed for multicore systems where communication costs are low. They are inefficient on systems such as Hadoop running on a cluster of commodity machines where communication costs are substantial. In this paper we design a distributed algorithm for l_1 regularization that is much better suited for such systems than existing algorithms. A careful cost analysis is used to support these points and motivate our method. The main idea of our algorithm is to do block optimization of many variables on the actual objective function within each computing node; this increases the computational cost per step that is matched with the communication cost, and decreases the number of outer iterations, thus yielding a faster overall method. Distributed Gauss-Seidel and Gauss-Southwell greedy schemes are used for choosing variables to update in each step. We establish global convergence theory for our algorithm, including Q-linear rate of convergence. Experiments on two benchmark problems show our method to be much faster than existing methods.

Keywords: Distributed learning, l_1 regularization

1. Introduction

The design of sparse linear classifiers using l_1 regularization is an important problem that has received great attention in recent years. This is due to its value in scenarios where the number of features is large and the classifier representation needs to be kept compact. Big data is becoming common nowadays. For example, in online advertising one comes across datasets with about a billion examples and a billion features. A substantial fraction of the features is usually irrelevant; and, l_1 regularization offers a systematic way to choose the small fraction of relevant features and form the classifier using them. In the future, one can foresee even bigger sized datasets to arise in this and other applications. For such big data, distributed storage of data over a cluster of commodity machines becomes necessary. Thus, fast training of l_1 regularized classifiers over distributed data is an important problem.

A number of algorithms have been recently proposed for parallel and distributed training of l_1 regularized classifiers; see section 3 for a review.[1] Most of these algorithms are based on coordinate-descent and they assume the data to be feature-partitioned. They are designed for multicore systems in which data communication costs are negligible. Recently, distributed systems with Hadoop running on a cluster of commodity machines have become popular. In such systems, communication costs are generally high; current methods for l_1 regularization are not optimally designed for such systems. Recently there has been been increased attention given to designing communication-efficient algorithms (Jaggi et al., 2014; Ma et al., 2015). *In this paper we develop a distributed block coordinate descent (DBCD) method that is efficient on distributed platforms in which communication costs are high.*

Following are the main contributions of this paper.

1. Most methods for the parallel training of l_1 regularized classifiers (including the ones proposed in this paper) fall into a generic algorithm format (see algorithm 1 in section 2). We make careful choices for the three key steps of this algorithm, leading to the development of a distributed block coordinate descent (DBCD) method that is very efficient on distributed platforms with high communication cost.

2. We provide a detailed cost analysis (section 5) that brings out the computation and communication costs of the generic algorithm clearly for different methods. In the process we motivate the need for new efficient methods such as DBCD that are suited to communication heavy settings.

3. We establish convergence theory (subsection 4.4) for our method using the results of Tseng and Yun (2009) and Yun et al. (2011). It is worth noting the following: (a) though Tseng and Yun (2009) and Yun et al. (2011) cover algorithms using quadratic approximations for the total loss, we use a simple trick to apply them to general nonlinear approximations, thus bringing more power to their results; and (b) even these two works use only per-feature decoupled quadratic models in their implementations whereas we work with more powerful approximations that couple features.

4. We give an experimental evaluation (section 6) that shows the strong performance of DBCD against key current methods in scenarios where communication cost is significant. Based on the experiments we make a final recommendation for the best method to employ for such scenarios.

The paper is organized as follows. The generic algorithm format is described in section 2. This gives a clear view of existing methods and allows us to motivate the new method. In section 3 we discuss the key related work in some detail. In section 4 we describe the DBCD method in detail and prove its convergence. The analysis of computation and communication costs in section 5 gives a firmer motivation of our DBCD method. Experiments

1. In this paper we only consider synchronous distributed training algorithms in which various computing nodes share their information and complete one update. Asynchronous methods (Li et al., 2014) form another important class of methods that needs a separate study.

©2017 Mahajan, Keerthi and Sundararajan.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/14-484.html.

comparing our method with several existing methods on a few large scale datasets are given in section 6. These experiments strongly demonstrate the efficiency of one version of our method that chooses update variables greedily. This best version of the DBCD method is described in section 7. Section 8 contains some concluding comments.

2. A generic algorithm

Various distributed solvers of the l_1 regularization problem can be put in a generic algorithm format. We begin this section by describing the problem formulation. Then we state the generic algorithm format. We follow this by discussing the choices various methods make for the steps and point out how new choices for the steps can lead to a better design.

2.1 Problem formulation

Let w be the weight vector with m variables, w_j, $j = 1, \ldots, m$, and $x_i \in R^m$ denote the i-th example. Let there be n training examples and let X denote the $n \times m$ data matrix, whose i-th row is x_i^T. Note that we have denoted vector components by subscripts, e.g., w_j is the j-th component of w; we have also used subscripts for indexing examples, e.g., x_i is the i-th example, which itself is a vector. But this will not cause confusion anywhere. A linear classifier produces the output $y_i = w^T x_i$. The loss is a nonlinear convex function applied on the output. For binary class label $c_i \in \{1, -1\}$, the loss is given by $\ell(y_i; c_i)$. Let us simply view $\ell(y_i; c_i)$ as a function of y_i with c_i acting as a parameter. We will assume that ℓ is non-negative and convex, $\ell \in \mathcal{C}^1$, the class of continuously differentiable functions, and that ℓ' is Lipschitz continuous[2]. Loss functions such as least squares loss, logistic loss, SVM squared hinge loss and Huber loss satisfy these assumptions. All experiments reported in this paper use the squared hinge loss, $\ell(y_i; c_i) = \max\{0, 1 - c_i y_i\}^2$. The total loss function, $f : R^m \to R$ is $f(w) = \frac{1}{n} \sum_i \ell(y_i; c_i)$. Let u be the l_1 regularizer given by $u(w) = \lambda \sum_j |w_j|$, where $\lambda > 0$ is the regularization constant. Our aim is to solve the problem

$$\min_{w \in R^m} F(w) = f(w) + u(w). \tag{1}$$

Let $g = \nabla f$. The optimality conditions for (1) are:

$$\forall j: \quad g_j + \lambda \,\mathrm{sign}(w_j) = 0 \text{ if } |w_j| > 0; \quad |g_j| \leq \lambda \text{ if } w_j = 0. \tag{2}$$

For problems with a large number of features, it is natural to randomly partition the columns of X and place the parts in P computing nodes. Let $\{B_p\}_{p=1}^P$ denote this partition of $\mathcal{M} = \{1, \ldots, m\}$, i.e., $B_p \subset \mathcal{M}$ $\forall p$ and $\cup_p B_p = \mathcal{M}$. *We will assume that this feature partitioning is given and that all algorithms operate within that constraint.* The variables associated with a particular partition get placed in one node. Given a subset of variables S, let X_S be the submatrix of X containing the columns corresponding to S. For a vector $z \in R^m$, z_S will denote the vector containing the components of z corresponding to S.

2. A function h is Lipschitz continuous if there exists a (Lipschitz) constant $L \geq 0$ such that $\|h(a) - h(b)\| \leq L\|a - b\|$ $\forall\, a, b$.

2.2 Generic algorithm

Algorithm 1 gives the generic algorithm. In each iteration t, the following steps happen, in parallel, in each node (p): (a) a subset of variables, S_p^t is chosen; (b) a suitable approximating function, f_p^t is formed and the chosen variables are optimized so as to define a direction; (c) a step size is chosen along that direction to update the weight vector on the chosen variables. The outputs of all examples are then computed using an *AllReduce* operation (step (d)) and the algorithm is terminated if optimality conditions are satisfied (step (e)).

Items such as B_p, S_p^t, w_{B_p}, $d_{B_p}^t$, X_{B_p} stay local in node p and do not need to be communicated. Step (d) can be carried out using an *AllReduce* operation (Agarwal et al., 2013) over the nodes and then y becomes available in all the nodes. The gradient subvector $g_{B_p}^t$ (which is needed for solving (3)) can then be computed locally as $g_{B_p}^t = X_{B_p}^T b$ where $b \in R^n$ is a vector with $\{\ell'(y_i)\}$ as its components.

Algorithm 1: A generic distributed algorithm

Choose w^0 and compute $y^0 = Xw^0$;
for $t = 0, 1 \ldots$ **do**
  for $p = 1, \ldots, P$ **do**
    (a) Select a working subset of variables[3], $S_p^t \subset B_p$;
    (b) Form $f_p^t(w_{B_p})$, an approximation of f and minimize, exactly or approximately, $f_p^t + u$ over only the weights corresponding to S_p^t:

$$\min f_p^t(w_{B_p}) + u(w_{B_p}) \quad \text{s.t.} \quad w_j = w_j^t \;\forall\, j \in B_p \setminus S_p^t \tag{3}$$

    to get $\bar{w}_{B_p}^t$ and set direction: $d_{B_p}^t = \bar{w}_{B_p}^t - w_{B_p}^t$;
    (c) Choose α^t and update: $w_{B_p}^{t+1} = w_{B_p}^t + \alpha^t d_{B_p}^t$;
  end
  (d) Update $y^{t+1} = y^t + \alpha^t \sum_p X_{B_p} d_{B_p}^t$;
  (e) Terminate if optimality conditions hold;
end

Steps (d) and (e) of Algorithm 1 are quite straight-forward. But the first three steps, (a)-variable sampling, (b)-function approximation, and (c)-step size determination, can be implemented in various ways and require a detailed discussion.

Step (a) - variable sampling. Some choices are:

- **(a.1)** random selection (Bradley et al., 2011; Richtárik and Takáč, 2014);
- **(a.2)** random cyclic: over a set of consecutive iterations (t) all variables are touched once (Bian et al., 2013);
- **(a.3)** greedy: always choose a set of variables that, in some sense violate (2) the most at the current iterate (Peng et al., 2013; Facchinei et al., 2014); and,
- **(a.4)** greedy selection using the Gauss-Southwell rule (Tseng and Yun, 2009; Yun et al., 2011).

Step (b) - function approximation. It would be ideal to choose f_p^t to be f itself. However, to make the solution simple and efficient, most methods choose a quadratic approximation that is decoupled at the individual variable level:

$$f_p^t(w_{B_p}^t) = \sum_{j \in B_p} g_j(w^t)(w_j - w_j^t) + \frac{L_j}{2}(w_j - w_j^t)^2 \tag{4}$$

The main advantages of (4) are its simplicity and closed-form minimization when used in (3). Choices for L^j that have been tried are:

- **(b.1)** L_j = a Lipschitz constant for g_j (Bradley et al., 2011; Peng et al., 2013);
- **(b.2)** L_j = a large enough bound on the Lipschitz constant for g_j to suit the sampling in step (a) (Richtárik and Takáč, 2014);
- **(b.3)** adaptive adjustment of L_j (Facchinei et al., 2014); and
- **(b.4)** $L_j = H_{jj}^t$, the j-th diagonal term of the Hessian at w^t (Bian et al., 2013).

Step (c) - step size. The choices are:

- **(c.1)** always fix $\alpha^t = 1$ (Bradley et al., 2011; Richtárik and Takáč, 2014; Peng et al., 2013);
- **(c.2)** use stochastic approximation ideas to choose $\{\alpha^t\}$ so that $\sum_t (\alpha^t)^2 < \infty$ and $\sum_t |\alpha^t| = \infty$ (Facchinei et al., 2014); and
- **(c.3)** choose α^t by line search that is directly tied to the optimization of F in (1) (Bian et al., 2013).

2.3 Discussion of choices for steps (a)-(c)

To understand the role of the various choices better, let us first focus on the use of (4) for f_p^t. Algorithm 1 may not converge to the optimal solution due to one of the following decisions: (i) choosing too many variables ($|S_p^t|$ large) for parallel updating in step (a); (ii) choosing small values for the proximal coefficient L_j in step (b); and (iii) not controlling α^t to be sufficiently small in step (c). This is because each of the above has the potential to cause large step sizes leading to increases in F value and, if this happens uncontrolled at all iterations then convergence to the minimum cannot occur. Different methods control against these by making suitable choices in the steps.

The choice made for step (c) gives a nice delineation of methods. With **(c.1)**, one has to do a suitable mix of large enough L_j and small enough $|S_p^t|$. Choice **(c.2)** is better since the proper control of $\{\alpha^t\} \to 0$ takes care of convergence; however, for good practical performance, L_j and α^t need to be carefully adapted, which is usually messy. Choice **(c.3)** is good in many ways: it leads to monotone decrease in F; it is good theoretically and practically; and, it allows both, small L_j as well as large $|S_p^t|$ without hindering convergence.

3. We will refer to the working subset size, i.e., $|S_p^t|$, as WSS.

Except for Bian et al. (2013), Tseng and Yun (2009) and Yun et al. (2011)[4], **(c.3)** has been unused in other methods because it is considered as 'not-in-line' with a proper parallel approach as it requires a separate α^t determination step requiring distributed computations and also needing F computations for several α^t values within one t. With line search, the actual implementation of Algorithm 1 merges steps **(c)** and **(d)** and so it deviates slightly from the flow of Algorithm 1. Specifically, we compute $\delta y = \sum_p X_{B_p} d_{B_p}^t$ before line search using AllReduce. Then each node can compute f at any α locally using $y + \alpha\,\delta y$. Only a scalar corresponding to the l_1 regularization term needs to be communicated for each α. This means that the communication cost associated with line search is minimal.[5] But truly, the slightly increased computation and communication costs is amply made up by a reduction in the number of iterations to reach sufficient optimality. So we go with the choice **(c.3)** in our method.

The choice of (4) for f_p^t in step (b) (in particular, **(b.4)**) is pretty much unanimously used in all previous works. This is done to make the optimization simple. While this is fine for communication friendly systems such as multicore, it is not the right choice when communication costs are high. Such a setting permits more per-node computation time, and there is much to be gained by using a more complex f_p^t. We propose the use of a function f_p^t that couples the variables in S_p^t. We also advocate an approximate solution of (3) (e.g., a few rounds of coordinate descent within each node) in order to control the computation time.

Crucial gains are also possible via resorting to the greedy choices, **(a.3)** and **(a.4)** for choosing S_p^t. On the other hand, with methods based on **(c.1)**, one has to be careful in using **(a.3)**: apart from difficulties in establishing convergence, practical performance can also be bad, as we show in section 6.

3. Related Work

Our interest is mainly in parallel/distributed computing methods. There are many parallel algorithms targeting a single machine having multi-cores with shared memory (Bradley et al., 2011; Richtárik and Takáč, 2015; Bian et al., 2013; Peng et al., 2013). In contrast, there exist only a few efficient algorithms to solve (1) when the data is distributed (Richtárik and Takáč, 2016; Ravazzi et al., 2013) and communication is an important aspect to consider. In this setting, the problem (1) can be solved in several ways depending on how the data is distributed across machines (Peng et al., 2013; Boyd et al., 2011): (A) example (horizontal) split, (B) feature (vertical) split and (C) combined example and feature split (a block of examples/features per node). While methods such as distributed FISTA (Peng et al., 2013) or ADMM (Boyd et al., 2011) are useful for (A), the block splitting method (Parikh and Boyd, 2013) is useful for (C). We are interested in (B), and the most relevant and important class of methods is parallel/distributed coordinate descent methods, as abstracted in algorithm 1. Most of these methods set f_p^t in step (b) of algorithm 1 to be a quadratic

4. Among these three works, Tseng and Yun (2009) and Yun et al. (2011) mainly focus on general theory and little on distributed implementation.
5. Later, in section 5 when we write costs, we write it to be consistent with Algorithm 1. The total cost of all the steps is the same for the implementation described here for line search. For genericity sake, we keep Algorithm 1 as it is even for the line search case. The actual details of the implementation for the line search case will become clear when we layout the final algorithm in section 7.

approximation that is decoupled at the individual variable level. Table 3 compares these methods along various dimensions.[6]

Most dimensions arise naturally from the steps of algorithm 1, as explained in section 2. Two important points to note are: (i) except Richtárik and Takáč (2016) and our method, none of these methods target and sufficiently discuss distributed setting involving communication and, (ii) from a practical view point, it is difficult to ensure stability and get good speed-up with no line search and non-monotone methods. For example, methods such as Bradley et al. (2011); Richtárik and Takáč (2014, 2015); Peng et al. (2013) that do not do line search are shown to have the monotone property only in expectation and that too only under certain conditions. Furthermore, variable selection rules, proximal coefficients and other method-specific parameter settings play important roles in achieving monotone convergence and improved efficiency. As we show in section 6, our method and the parallel coordinate descent Newton method (Bian et al., 2013) (see below for a discussion) enjoy robustness to various settings and come out as clear winners.

It is beyond the scope of this paper to give a more detailed discussion, beyond Table 3, of the methods from a theoretical convergence perspective on various assumptions and conditions under which results hold. We only briefly describe and comment on them below.

Generic Coordinate Descent Method (Scherrer et al., 2012a,b) Scherrer et al. (2012a) and Scherrer et al. (2012b) presented an abstract framework for coordinate descent methods (GenCD) suitable for parallel computing environments. Several coordinate descent algorithms such as stochastic coordinate descent (Shalev-Shwartz and Tewari, 2011), Shotgun (Bradley et al., 2011) and GROCK (Peng et al., 2013) are covered by GenCD. GROCK is a thread greedy algorithm (Scherrer et al., 2012a) in which the variables are selected greedily using gradient information. One important issue is that algorithms such as Shotgun and GROCK may not converge in practice due to their non-monotone nature with no line search; we faced convergence issues on some datasets in our experiments with GROCK (see section 6). Therefore, the practical utility of such algorithms is limited without ensuring necessary descent property through certain spectral radius conditions on the data matrix.

Distributed Coordinate Descent Method (Richtárik and Takáč, 2016) The multicore parallel coordinate descent method of Richtárik and Takáč (2014) is a much refined version of GenCD with careful choices for steps (a)-(c) of algorithm 1 and a supporting stochastic convergence theory. Richtárik and Takáč (2016) extended this to the distributed setting; so, this method is more relevant to this paper. With no line search, their algorithm HYDRA (Hybrid coordinate descent) has (expected) descent property only for certain sampling types of selecting variables and L_j values. One key issue is setting the right L_j values for good performance. Doing this accurately is a costly operation; on the other hand, inaccurate setting using cheaper computations (e.g., using the number of non-zero elements as suggested in their work) results in slower convergence (see section 6).

Necoara and Clipici (2014) suggest another variant of parallel coordinate descent in which all the variables are updated in each iteration. HYDRA and GROCK can be

6. Although our method will be presented only in section 4, we include our method's properties in the last row of Table 3. This helps to easily compare our method against the rest.

considered as two key, distinct methods that represent the set of methods discussed above. So, in our analysis as well as experimental comparisons in the rest of the paper, we do not consider the methods in this set other than these two.

Flexible Parallel Algorithm (FPA) (Facchinei et al., 2014) This method has some similarities with our method in terms of the approximate function optimized at the nodes. Though Facchinei et al. (2014) suggest several approximations, they use only (4) in its final implementation. More importantly, FPA is a non-monotone method using a stochastic approximation step size rule. Tuning this step size rule along with the proximal parameter L_j to ensure convergence and speed-up is hard. (In section 6 we conduct experiments to show this.) Unlike our method, FPA's inner optimization stopping criterion is unverifiable (for e.g., with (6)); also, FPA does not address the communication cost issue.

Parallel Coordinate Descent Newton (PCD) (Bian et al., 2013) One key difference between other methods discussed above and our DBCD method is the use of line search. Note that the PCD method can be seen as a special case of DBCD (see subsection 5.1). In DBCD, we optimize per-node block variables jointly, and perform line search across the blocks of variables; as shown later in our experimental results, this has the advantage of reducing the number of outer iterations, and overall wall clock time due to reduced communication time (compared to PCD).

Synchronized Parallel Algorithm (Patriksson, 1998b) Patriksson (1998b) proposed a Jacobi type synchronous parallel algorithm with line search using a generic cost approximation (CA) framework for differentiable objective functions (Patriksson, 1998a). Its local linear rate of convergence results hold only for a class of strong monotone CA functions. If we view the approximation function, f_p^t as a mapping that is dependent on w^t, Patriksson (1998b) requires this mapping to be continuous, which is unnecessarily restrictive.

ADMM Methods Alternating direction method of multipliers is a generic and popular distributed computing method. It does not fit into the format of Algorithm 1. This method can be used to solve (1) in different data splitting scenarios (Boyd et al., 2011; Parikh and Boyd, 2013). Several variants of global convergence and rate of convergence (e.g., $O(\frac{1}{k})$) results exist under different weak/strong convexity assumptions on the two terms of the objective function (Deng and Yin, 2016; Deng et al., 2013). Recently, an accelerated version of ADMM (Goldstein et al., 2014) derived using the ideas of Nesterov's accelerated gradient method (Nesterov, 2012) has been proposed; this method has dual objective function convergence rate of $O(\frac{1}{k^2})$ under a strong convexity assumption. ADMM performance is quite good when the augmented Lagrangian parameter is set to the right value; however, getting a reasonably good value comes with computational cost. In section 6 we evaluate our method and find it to be much faster.

Based on the above study of related work, we choose HYDRA, GROCK, PCD and FPA as the main methods for analysis and comparison with our method.[7] Thus, Table 3 gives various dimensions only for these methods.

7. In the experiments of section 6, we also include ADMM.

Method	Is $\mathbf{F}(w^t)$ monotone?	Are limits forced on $\lvert S_p^t \rvert$?	How is $\mathbf{S}_p^t$ chosen?	Basis for choosing $\mathbf{L}_j$	How is α^t chosen	Convergence type	Convergence rate
Existing methods							
HYDRA	No	No, if L_j is varied suitably	Random	Lipschitz bound for g_j suited to S_p^t choice	Fixed	Stochastic	Linear
GROCK	No	Yes	Greedy	Lipschitz bound for g_j	Fixed	Deterministic	Sub-linear
FPA	No	No	Random	Lipschitz bound for g_j	Adaptive	Deterministic	None
PCD	Yes	No	Random	Hessian diagonal	Armijo line search	Stochastic	Sub-linear
Our method							
DBCD	Yes	No	Random/Greedy	Free	Armijo line search	Deterministic	Locally linear

Table 1: Properties of selected methods that fit into the format of Algorithm 1. Methods: HYDRA (Richtárik and Takáč, 2016), GROCK (Peng et al., 2013), FPA (Facchinei et al., 2014), PCD (Bian et al., 2013).

4. DBCD method

The DBCD method that we propose fits into the general format of Algorithm 1. It is actually *a class of algorithms* that allows various possibilities for steps (a), (b) and (c). Below we lay out these possibilities and establish a general convergence theory for the class of algorithms that fall under DBCD. We recommend three specific instantiations of DBCD, analyze their costs in section 5, empirically study them in section 6 and make one final best recommendation in section 7. In this section, we also show the relations of DBCD to other methods on aspects such as variable selection, function approximation, line search, etc. As this section is traversed, it is also useful to re-visit table 3 and compare DBCD against other key methods.

Our goal is to develop an efficient distributed learning method that jointly optimizes the costs involved in the various steps of the algorithm. We observed in the previous section that the methods discussed there lack this careful optimization in one or more steps, resulting in inferior performance. This can be understood better via a cost analysis. To avoid too much deviation, we give the gist of this cost analysis here and postpone the full details to section 5. The cost of Algorithm 1 can be written as $T^P(C^P_{\text{comp}} + C^P_{\text{comm}})$ where P denotes the number of nodes, T^P is the number of outer iterations[8], and, C^P_{comp} and C^P_{comm} respectively denote the computation and communication costs per-iteration. In communication heavy situations, existing algorithms have $C^P_{\text{comp}} \ll C^P_{\text{comm}}$. Our method aims to improve overall efficiency by making each iteration more complex (C^P_{comp} is increased) and, in the process, making T^P much smaller.

4.1 Variable selection

Let us now turn to step (a) of Algorithm 1. We propose two schemes for variable selection, i.e., choosing $S_p^t \subset B_p$.

Gauss-Seidel scheme. In this scheme, we form cycles - each cycle consists of a set of consecutive iterations - while making sure that every variable is touched once in each cycle. We implement a cycle as follows. Let τ denote the iteration where a cycle starts. Choose a positive integer T (T may change with each cycle). For each p, randomly partition B_p into T equal parts: $\{S_p^t\}_{t=\tau}^{\tau+T-1}$. Use these variable selections to do T iterations. *Henceforth, we refer to this scheme as the R-scheme.*

Distributed greedy scheme. This is a greedy scheme which is purely distributed and so more specific than the Gauss-Southwell schemes in Tseng and Yun (2009).[9] In each iteration, our scheme chooses variables based on how badly (2) is violated for various j. For one j, an expression of this violation is as follows. Let g^t and H^t denote, respectively, the gradient and Hessian at w^t. Form the following one variable quadratic approximation:

$$q_j(w_j) = g_j^t(w_j - w_j^t) + \frac{1}{2}(H_{jj}^t + \nu)(w_j - w_j^t)^2 + \lambda|w_j| - \lambda|w_j^t| \quad (5)$$

8. For practical purposes, one can view T^P as the number of outer iterations needed to reach a specified closeness to the optimal objective function value. We will say this more precisely in section 6.
9. Yet, our distributed greedy scheme can be shown to imply the Gauss-Southwell-q rule for a certain parameter setting. See the appendix for details.

where ν is a small positive constant. Let $\bar{q}_j$ denote the optimal objective function value obtained by minimizing $q_j(w_j)$ over all w_j. Since $q_j(w_j^t) = 0$, clearly $\bar{q}_j \leq 0$. The more negative $\bar{q}_j$ is, the better it is to choose j.

Our distributed greedy scheme first chooses a working set size, WSS (the size of S_p^t) and then, in each node p, it chooses the top WSS variables from B_p according to smallness of $\bar{q}_j$, to form S_p^t. *Hereafter, we refer to this scheme as the S-scheme.*

It is worth pointing out that, our distributed greedy scheme requires more computation than the Gauss-Seidel scheme. However, since the increased computation is local, non-heavy and communication is the real bottleneck, it is not a worrisome factor.

4.2 Function approximation

Let us begin with step (b). There are three key items involved: (i) what are some of the choices of approximate functions possible, used by our methods and others? (ii) what is the stopping criterion for the inner optimization (i.e., local problem), and, (iii) what is the method used to solve the inner optimization? We discuss all these details below. We stress the main point that, unlike previous methods, we allow f_p^t to be non-quadratic and also to be a joint function of the variables in w_{B_p}. We first describe a general set of properties that f_p^t must satisfy, and then discuss specific instantiations that satisfy these properties.

Condition 1. $f_p^t \in \mathcal{C}^1$; $g_p^t = \nabla f_p^t$ is Lipschitz continuous, with the Lipschitz constant uniformly bounded over all t; f_p^t is strongly convex (uniformly in t), i.e., $\exists\ \mu > 0$ such that $f_p^t - \frac{\mu}{2}\|w_{B_p}\|^2$ is convex; and, f_p^t is gradient consistent with f at $w_{B_p}^t$, i.e., $g_p^t(w_{B_p}^t) = g_{B_p}(w^t)$.

This assumption is not restrictive. Gradient consistency is essential because it is the property that connects f_p^t to f and ensures that a solution of (3) will make $d_{B_p}^t$ a descent direction for F at $w_{B_p}^t$, thus paving the way for a decrease in F at step (c). Strong convexity is a technical requirement that is needed for establishing sufficient decrease in F in each step of Algorithm 1. Our experiments indicate that it is sufficient to set μ to be a very small positive value. Lipschitz continuity is another technical condition that is needed for ensuring boundedness of various quantities; also, it is easily satisfied by most loss functions.

Choice of f_p^t. Let us now discuss some good ways of choosing f_p^t. *For all these instantiations, a proximal term is added to get the strong convexity required by* Condition 1.

- **Proximal-Jacobi.** We can follow the classical Jacobi method in choosing f_p^t to be the restriction of f to $w_{S_p^t}^t$, with the remaining variables fixed at their values in w^t. Let $\bar{B}_p$ denote the complement of B_p, i.e., the set of variables associated with nodes other than p. Thus we set

$$f_p^t(w_{B_p}) = f(w_{B_p}, w_{\bar{B}_p}^t) + \frac{\mu}{2}\|w_{B_p} - w_{B_p}^t\|^2 \tag{6}$$

  where $\mu > 0$ is the proximal constant. It is worth pointing out that, since each node p keeps a copy of the full classifier output vector y aggregated over all the nodes, the computation of f_p^t and g_p^t due to changes in w_{B_p} can be locally computed in node p. Thus the solution of (3) is local to node p and so step (b) of Algorithm 1 can be executed in parallel for all p.

- **Block GLMNET.** GLMNET (Yuan et al., 2012; Friedman et al., 2010) is a sequential coordinate descent method that has been demonstrated to be very promising for the sequential solution of l_1 regularized problems with logistic loss. At each iteration, GLMNET minimizes the second order Taylor series of f at w^t, followed by line search along the direction generated by this minimizer. We can make a distributed version by choosing f_p^t to be the second order Taylor series approximation of $f(w_{B_p}, w_{\bar{B}_p}^t)$ with respect to w_{B_p} while keeping $w_{\bar{B}_p}$ fixed at $w_{\bar{B}_p}^t$. In other words, we can choose f_p^t as

$$f_p^t(w_{B_p}) = Q^t(w_{B_p}) + \frac{\mu}{2}\|w_{B_p} - w_{B_p}^t\|^2 \tag{7}$$

  where Q^t is the quadratic approximation of $f(w_{B_p}, w_{\bar{B}_p}^t)$ with respect to w_{B_p} at $w_{B_p}^t$ with $w_{\bar{B}_p}^t$ fixed.

- **Block L-BFGS.** One can keep a limited history of $w_{B_p}^t$ and $g_{B_p}^t$ and use an $L-BFGS$ approach to build a second order approximation of f in each iteration to form f_p^t:

$$f_p^t(w_{B_p}) = (g_{B_p}^t)^T(w_{B_p} - w_{B_p}^t) + \frac{1}{2}(w_{B_p} - w_{B_p}^t)^T H_{BFGS}(w_{B_p} - w_{B_p}^t) + \frac{\mu}{2}\|w_{B_p} - w_{B_p}^t\|^2 \tag{8}$$

  where H_{BFGS} is a limited memory BFGS approximation of the Hessian of $f(w_{B_p}, w_{\bar{B}_p}^t)$ with respect to w_{B_p} with $w_{\bar{B}_p}^t$ fixed, formed using $\{g_{B_p}^\tau\}_{\tau \leq t}$.

- **Decoupled quadratic.** Like in existing methods we can also form a quadratic approximation of f that decouples at the variable level - see (4). (An additional proximal term can be added.) If the second order term is based on the diagonal elements of the Hessian at w^t, then the PCDN algorithm given in Bian et al. (2013) can be viewed as a special case of our DBCD method. PCDN (Bian et al., 2013) is based on Gauss-Seidel variable selection. But it can also be used in combination with the distributed greedy scheme that we propose in subsection 4.1 below.

Approximate stopping. In step (b) of Algorithm 1 we mentioned the possibility of approximately solving (3). This is irrelevant for previous methods which solve individual variable level quadratic optimization in closed form, but very relevant to our method. Here we propose an approximate relative stopping criterion and later, in subsection 4.4, also give convergence theory to support it.

Let ∂u_j be the set of subgradients of the regularizer term $u_j = \lambda|w_j|$, i.e.,

$$\partial u_j = [-\lambda, \lambda] \ \text{ if } w_j = 0; \ \ \lambda\,\text{sign}(w_j) \ \text{ if } w_j \neq 0. \tag{9}$$

A point $\bar{w}_{B_p}^t$ is optimal for (3) if, at that point,

$$(g_p^t)_j + \xi_j = 0, \ \ \text{for some } \ \xi_j \in \partial u_j \ \ \forall\ j \in S_p^t. \tag{10}$$

An approximate stopping condition can be derived by choosing a tolerance $\epsilon > 0$ and requiring that, for each $j \in S_p^t$ there exists $\xi_j \in \partial u_j$ such that

$$\delta^j = (g_p^t)_j + \xi_j, \ \ |\delta_j| \leq \epsilon |d_j^t| \ \ \forall\ j \in S_p^t \tag{11}$$

A practical alternative is to replace the stopping condition (11) by simply using a fixed number of cycles of coordinate descent to minimize f_p^t.

Method used for solving (3). Now (3) is an l_1 regularized problem restricted to $w_{S_p^t}$. It has to be solved within node p using a suitable sequential method. Going by the state of the art for sequential solution of such problems (Yuan et al., 2010) we use the coordinate-descent method described in Yuan et al. (2010) for solving (3). For logistic regression loss, it is appropriate to use the new-GLMNET method (Yuan et al., 2012).

4.3 Line search

Line search (step (c) of Algorithm 1) forms an important component for making good decrease in F at each iteration. For non-differentiable optimization, there are several ways of doing line search. For our context, Tseng and Yun (2009) and Patriksson (1998a) give two good ways of doing line search based on Armijo backtracking rule. In this paper we use ideas from the former. Let β and σ be real parameters in the interval $(0, 1)$. (We use the standard choices, $\beta = 0.5$ and $\sigma = 0.01$.) We choose α^t to be the largest element of $\{\beta^k\}_{k=0,1,\ldots}$ satisfying

$$F(w^t + \alpha^t d^t) \leq F(w^t) + \alpha^t \sigma \Delta^t, \tag{12}$$

$$\Delta^t \stackrel{\text{def}}{=} (g^t)^T d^t + \lambda u(w^t + d^t) - \lambda u(w^t). \tag{13}$$

4.4 Convergence

We now establish convergence for the class of algorithmic choices discussed in subsections 4.2-4.3. To do this, we make use of the results of Tseng and Yun (2009). An interesting aspect of this use is that, while the results of Tseng and Yun (2009) are stated only for f_p^t being quadratic, we employ a simple trick that lets us apply the results to our algorithm which involves non-quadratic approximations.

Apart from the conditions in Condition 1 (see subsection 4.2) we need one other technical assumption.

Condition 2. For any given t, w_{B_p} and $\hat{w}_{B_p}$, $\exists$ a positive definite matrix $\hat{H} \geq \mu I$ (note: $\hat{H}$ can depend on t, w_{B_p} and $\hat{w}_{B_p}$) such that

$$\hat{g}^t_{B_p}(w_{B_p}) - \hat{g}^t_{B_p}(\hat{w}_{B_p}) = \hat{H}(w_{B_p} - \hat{w}_{B_p}) \tag{14}$$

In the above, $\hat{g}^t_{B_p}$ is the gradient with respect to w_{B_p} of the approximate function f_p^t formed in step (b) of algorithm 1. Note that $\hat{g}^t_{B_p}(w^t_{B_p}) = g^t_{B_p}$.

Except *Proximal-Jacobi*, the other instantiations of f_p^t mentioned in subsection 4.2 are quadratic functions; for these, g_p^t is a linear function and so (14) holds trivially. Let us turn to *Proximal-Jacobi*. If $f_p^t \in \mathcal{C}^2$, the class of twice continuously differentiable functions, then Condition 2 follows directly from mean value theorem; note that, since $f_p^t - \frac{\mu}{2}\|w\|^2$ is convex, $H_p \geq \mu I$ at any point, where H_p is the Hessian of f_p^t. Thus Condition 2 easily holds for least squares loss and logistic loss. Now consider the SVM squared hinge loss, $\ell(y_i; c_i) = 0.5(\max\{0, 1 - y_i c_i\})^2$, which is not in $\mathcal{C}^2$. Condition 2 holds for it because $g = \sum_i \ell'(y_i; c_i) x_i$ and, for any two real numbers z_1, z_2, $\ell'(z_1; c_i) - \ell'(z_2; c_i) = \kappa(z_1, z_2, c_i)(z_1 - z_2)$ where $0 \leq \kappa(z_1, z_2, c_i) \leq 1$.

The main convergence theorem can now be stated. Its proof is given in the appendix.

Theorem 1. Suppose, in Algorithm 1, the following hold:

(i) step (a) is done via the Gauss-Seidel or distributed greedy schemes of subection 5.2;

(ii) f_p^t in step (b) satisfies Condition 1 and Condition 2;

(iii) (11) is used to terminate (3) with $\epsilon = \mu/2$ (where μ is as in Condition 1); and,

(iv) in step (c), α^t is chosen via Armijo backtracking of subection 5.3.

Then Algorithm 1 is well defined and produces a sequence, $\{w^t\}$ such that any accumulation point of $\{w^t\}$ is a solution of (1). If, in addition, the total loss, f is strongly convex, then $\{F(w^t)\}$ converges Q-linearly and $\{w^t\}$ converges at least R-linearly.[10]

4.5 Specific instantiations of DBCD

For function approximation we found the proximal-Jacobi choice, (6) to be powerful. This choice, combined with variable selection done using the R-scheme and S-scheme leads to the two specific instantiations, DBCD-R and DBCD-S. As we explained in subsection 4.2, the PCDN algorithm given by Bian et al. (2013) (referred to as PCD) is a special case of DBCD using a decoupled quadratic function approximation and the R-scheme for variable selection. We also recommend the use of S-scheme with this method and call that instantiation as PCD-S.

5. DBCD method: Cost analysis

As pointed out earlier, the DBCD method is motivated by an analysis of the costs of various steps of algorithm 1. In this section, we present a detailed cost analysis that explains this more clearly. Based on section 3, we select the following five methods for our study: (1) HYDRA (Richtárik and Takáč, 2016), (2) GROCK (Greedy coordinate-block) (Peng et al., 2013), (3) FPA (Flexible Parallel Algorithm) (Facchinei et al., 2014), (4) PCD (Parallel Coordinate Descent Newton method) (Bian et al., 2013), and (5) DBCD. We will use the above mentioned abbreviations for the methods in the rest of the paper.

Let nz and $|S| = \sum_p |S_p^t|$ denote the number of non-zero entries in the data matrix X and the number of variables updated in each iteration respectively. To keep the analysis simple, we make the homogeneity assumption that the number of non-zero data elements in each node is nz/P. Let $\beta(\gg 1)$ be the relative computation to communication speed in the given distributed system; more precisely, it is the ratio of the times associated with communicating a floating point number and performing one floating point operation. On a cluster with 10 Gbps communication bandwidth where we did our experiments, we found the value of β to be in the range $30 - 100$. Recall that n, m and P denote the number of examples, features and nodes respectively. Table 5 gives cost expressions for different steps of the algorithm in one outer iteration. Here c_1, c_2, c_3, c_4 and c_5 are method dependent parameters. Table 5 gives the cost constants for various methods.We briefly discuss different costs below.

10. See chapter 9 of Ortega and Rheinboldt (1970) for definitions of Q-linear and R-linear convergence.

Cost	Steps of Algorithm 1			
	Step (a) Variable selection	Step (b) Inner optimization	Step (c) Choosing step size	Step (d) Updating output
Computation	$c_1\frac{nz}{P}$	$c_2\frac{nz}{P}\frac{\lvert S\rvert}{m}$	$c_3\lvert S\rvert + c_4 n$	$c_5\frac{nz}{P}\frac{\lvert S\rvert}{m}$
Communication	-	-	≈ 0	$\approx \beta n$[11]

Table 2: Cost of various steps of Algorithm 1. C^P_{comp} and C^P_{comm} are respectively, the sums of costs in the computation and communication rows.

Method	c_1	c_2	c_3	c_4	c_5	Computation cost per iteration	Communication cost per iteration
Existing methods							
HYDRA	0	1	1	0	1	$2\frac{nz}{P}\frac{\lvert S\rvert}{m} + \lvert S\rvert$	βn
GROCK	1	q	1	0	q	$\frac{nz}{P} + 2q\frac{nz}{P}\frac{\lvert S\rvert}{m} + \lvert S\rvert$	βn
FPA	1	q	1	1	q	$\frac{nz}{P} + 2q\frac{nz}{P}\frac{\lvert S\rvert}{m} + \lvert S\rvert + n$	βn
PCD	0	1	τ_{ls}	τ_{ls}	1	$2\frac{nz}{P}\frac{\lvert S\rvert}{m} + \tau_{ls}\lvert S\rvert + \tau_{ls} n$	βn
Variations of our method							
PCD-S	1	q	τ_{ls}	τ_{ls}	q	$\frac{nz}{P} + 2q\frac{nz}{P}\frac{\lvert S\rvert}{m} + \tau_{ls}\lvert S\rvert + \tau_{ls} n$	βn
DBCD-R	0	k	τ_{ls}	τ_{ls}	1	$(k+1)\frac{nz}{P}\frac{\lvert S\rvert}{m} + \tau_{ls}\lvert S\rvert + \tau_{ls} n$	βn
DBCD-S	1	kq	τ_{ls}	τ_{ls}	q	$\frac{nz}{P} + q(k+1)\frac{nz}{P}\frac{\lvert S\rvert}{m} + \tau_{ls}\lvert S\rvert + \tau_{ls} n$	βn

Table 3: Cost parameter values and costs for different methods. q lies in the range: $1 \le q \le \frac{m}{\lvert S\rvert}$. R and S refer to variable selection schemes for step (a); see subsection 4.1. PCD uses the R scheme and so it can also be referred to as PCD-R. Typically τ_{ls}, the number of α values tried in line search, is very small; in our experiments we found that on average it is not more than 10. Therefore all methods have pretty much the same communication cost per iteration.

Step a: Methods like our DBCD-S[12], GROCK, FPA and PCD-S need to calculate the gradient and model update to determine which variables to update. Hence, they need to go through the whole data once ($c_1 = 1$). On the other hand HYDRA, PCD and DBCD-R select variables randomly or in a cyclic order. As a result variable subset selection cost is negligible for them ($c_1 = 0$).

11. Note that the communication latency cost (the time taken to communicate zero bytes) is ignored in the communication cost expressions because it is dominated by the throughput cost for large n. Moreover, as in Agarwal et al. (2013), the broadcast and reduce operators are pipelined over the vector entries. This means that communication cost increases sub-linearly wrt. $logP$. If n is assumed to be large (as in our case), it is almost independent of $logP$ and can be written approximately as βn.
12. The DBCD and PCD methods have two variants, R and S corresponding to different ways of implementing step a; see subsection 4.1.

Step b: All the methods except DBCD-S and DBCD-R use the decoupled quadratic approximation (4). For DBCD-R and DBCD-S, an additional factor of k comes in c_2 since we do k inner cycles of CDN in each iteration. HYDRA, PCD and DBCD-R do a random or cyclic selection of variables. Hence, a factor of $\frac{\lvert S\rvert}{m}$ comes in the cost since only a subset $\lvert S\rvert$ of variables is updated in each iteration. However, methods that do selection of variables based on the magnitude of update or expected objective function decrease (DBCD-S, GROCK, FPA and PCD-S) favour variables with low sparsity. As a result, c_2 for these methods has an additional factor q where $1 \le q \le \frac{m}{\lvert S\rvert}$.

Step c: For methods that do not use line-search, $c_3 = 1$ and $c_4 = 0$[13]. The overall cost is $\lvert S\rvert$ to update the variables. For methods like DBCD-S, DBCD-R, PCD and PCD-S that do line-search, $c_3 = c_4 = \tau_{ls}$ where τ_{ls} is the average number of steps (α values tried) in one line search. For each line search step, we need to recompute the loss function which involves going over n examples once. Moreover, *AllReduce* step needs to be performed to sum over the distributed l_1 regularizer term. Since only one scalar needs to be communicated per line search step, the communication cost is dominated by the communication latency, i.e. the time taken to communicate zero bytes. As pointed out in Bian et al. (2013), τ_{ls} can increase with P; but it is still negligible compared to n. Combined with the fact that n is large in step (d), we will ignore this cost in the subsequent analysis.

Step d: This step involves computing and doing *AllReduce* on updated local predictions to get the global prediction vector for the next iteration and is common for all the methods. Note that because we are dealing with linear models, the updated predictions need to be communicated only once in each iteration even for the methods like ours that require line search, i.e., there is no need to communicate the updated predictions again and again for every line search step in each iteration.

The analysis given above is only for C^P_{comp} and C^P_{comm}, the computation and communication costs in one iteration. If T^P is the number of iterations to reach a certain optimality tolerance, then the total cost of Algorithm 1 is: $C^P = T^P(C^P_{\text{comp}} + C^P_{\text{comm}})$. For P nodes, speed-up is given by C^1/C^P. To illustrate the ill-effects of communication cost, let us take the method of Richtárik and Takáč (2015). For illustration, take the case of $\lvert S\rvert = P$, i.e., one variable is updated per node per iteration. For large P, $C^P \approx T^P C^P_{\text{comm}} = T^P\ \beta n$; both β and n are large in the distributed setting. On the other hand, for $P = 1$, $C^P_{\text{comm}} = 0$ and $C^P = C^P_{\text{comp}} \approx \frac{nz}{m}$. Thus $speedup = \frac{T^1}{T^P}\frac{C^1}{C^P} = \frac{T^1}{T^P}\frac{\frac{nz}{m}}{\beta n}$. Richtárik and Takáč (2015) show that T^1/T^P increases nicely with P. But, the term βn in the denominator of C^1/C^P has a severe detrimental effect. Unless a special distributed system with efficient communication is used, speed up has to necessarily suffer. When the training data is huge and so the data is forced to reside in distributed nodes, *the right question to ask is not whether we get great speed up, but to ask which method is the fastest.* Given this, we ask how various choices in the steps of Algorithm 1 can be made to decrease C^P. Suppose we devise choices such that (a) C^P_{comp} is increased while still remaining in the zone where $C^P_{\text{comp}} \ll C^P_{\text{comm}}$, and (b) in the process, T^P is decreased greatly, then C^P can be decreased. The basic idea of our method is to use a more complex f^t_p than the simple quadratic in (4), due to which, T^P becomes much smaller. The use of line search, **(c.3)** for step **c** aids this further. We see

13. For FPA, $c_4 = 1$ since objective function needs to be computed to automatically set the proximal term parameter.

in table 5 that, DBCD-R and DBCD-S have the maximum computational cost. On the other hand, communication cost is more or less the same for all the methods (except for few scalars in the line search step) and dominates the cost. In section 6, we will see on various datasets how, by doing more computation, our methods reduce T^P substantially over the other methods while incurring a small computation overhead (relative to communication) per iteration. These will become amply clear in section 6; see, for example, table 6.3 in that section.

6. Experimental Evaluation

In this section, we present experimental results on real-world datasets for the training of l_1 regularized linear classifiers using the squared hinge loss. Here training refers to the minimization of the function F in (1). We compare our methods with several state of the art methods, in particular, those analyzed in section 5 (see the methods in the first column of table 5) together with ADMM, the accelerated alternating direction method of multipliers (Goldstein et al., 2014). To the best of our knowledge, such a detailed study has not been done for parallel and distributed l_1 regularized solutions in terms of (a) accuracy and solution optimality performance, (b) variable selection schemes, (c) computation versus communication time and (d) solution sparsity. The results demonstrate the effectiveness of our methods in terms of total (computation + communication) time on both accuracy and objective function measures.

6.1 Experimental Setup

Datasets: We conducted our experiments on four datasets: KDD, URL, ADS and WEBSPAM[14]. The key properties of these datasets are given in table 6.1. These datasets have a large number of features and l_1 regularization is important. The number of examples is large for KDD, URL and ADS. WEBSPAM has a much smaller number of examples and hence communication costs are low for this dataset.

Dataset	n	m	nz	$s = nz/m$
KDD	8.41×10^6	20.21×10^6	0.31×10^9	15.34
URL	2.00×10^6	3.23×10^6	0.22×10^9	68.11
ADS	18.56×10^6	0.20×10^6	5.88×10^9	29966.83
WEBSPAM	0.26×10^6	16.60×10^6	0.98×10^9	58.91

Table 4: Properties of datasets. n is the number of examples, m is the number of features, nz is the number of non-zero elements in the data matrix, and s is the average number of non-zero elements per feature.

Methods and Metrics: We evaluate the performance of all the methods using (a) Area Under Precision-Recall Curve (AUPRC) (Sonnenburg and Franc, 2010; Agarwal et al., 2013)[15] and (b) Relative Function Value Difference (RFVD) as a function of time taken. RFVD is computed as $\frac{F(w^t)-F^*}{F^*}$ where F^* is taken as the best value obtained across the methods after a long duration. We also report per node computation time statistics and sparsity pattern behavior of all the methods.

14. KDD, URL and WEBSPAM are popular benchmark datasets taken from `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. ADS is a proprietary dataset from Microsoft.

Parameter Settings: For each dataset we used cross validation to find the optimal λ value that gave the best AUPRC values. For each dataset we experimented with a range of λ values centred around the optimal value that have good sparsity variations over the optimal solution. Since the relative performance between methods was quite consistent across different λ values, we give details of the performance only for the optimal λ value. With respect to algorithm 1, the working set size (WSS) per node and the number of nodes (P) are common across all the methods. We set WSS in terms of the fraction (r) of the number of features per node, i.e., WSS=rm/P. Note that WSS will change with P for a given fraction r. For all datasets we give results for two r values $(0.01, 0.1)$. Note that r does not play a role in ADMM since all variables are optimized in each node. We experimented with $P = 25, 100$. Only for *ADS* dataset we used $P = 100, 200$ because it has many more examples than others.

Platform: We ran all our experiments on a Hadoop cluster with 379 nodes and 10 Gbit interconnect speed. Each node has Intel (R) Xeon (R) E5-2450L (2 processors) running at 1.8 GHz and 192 GB RAM. (Though the datasets can fit in this memory configuration, our intention is to test the performance in a distributed setting.) All our implementations were done in *C#* including our binary tree *AllReduce* support (Agarwal et al., 2013) on Hadoop. We implemented the pipelined *AllReduce* operation described in Agarwal et al. (2013) that reduces the communication cost from $\beta nlogP$ to βn for large n.

6.2 Method Specific Parameter Settings

We discuss method specific parameter setting used in our experiments and associated practical implications.

Choice of μ and k for DBCD: To get a practical implementation that gives good performance in our method, we deviate slightly from the conditions of Theorem 1. First, we find that the proximal term does not contribute usefully to the progress of the algorithm (see the left side plot in figure 1). So we choose to set μ to a small value, e.g., $\mu = 10^{-12}$. Second, we replace the stopping condition (11) by simply using a fixed number of cycles of coordinate descent to minimize f_p^t. The right side plot in figure 1 shows the effect of number of cycles, k. We found that $k = 5, 10$ are good choices. Since computations are heavier for DBCD-S, we used $k = 5$ for it and used $k = 10$ for DBCD-R.

Let us begin with ADMM. We use the feature partitioning formulation of ADMM described in subsection 8.3 of Boyd et al. (2011). ADMM does not fit into the format of algorithm 1, but the communication cost per outer iteration is comparable to the other methods that fit into algorithm 1. In ADMM, the augmented Lagrangian parameter (ρ) plays an important role in getting good performance. In particular, the number of iterations required by ADMM for convergence is very sensitive with respect to ρ. While many schemes have been discussed in the literature (Boyd et al., 2011) we found that selecting ρ using the objective function value gave a good estimate; we selected ρ^* from a handful of ρ

15. We employed AUPRC instead of AUC because it differentiates methods more finely.

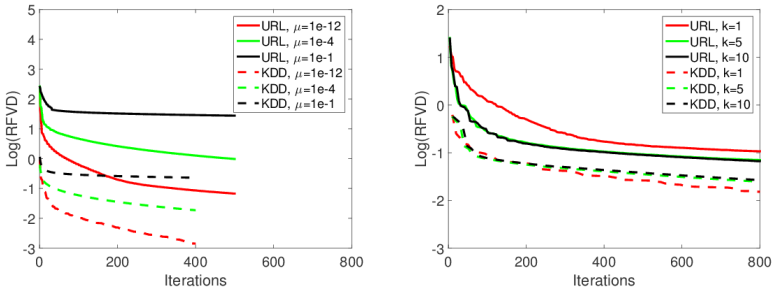


Figure 1: Study of μ and k on *KDD* and *URL*. Left: the effect of μ. Right: the effect of k, the number of cycles to minimize f_p^t. $\mu = 10^{-12}$ and $k = 10$ are good choices. $P = 100$.

values with ADMM run for 10 iterations (i.e., not full training) for each ρ value tried.[16] However, this step incurred some computational/communication time. Note that each ADMM iteration optimizes all variables and involves many inner iterations, thus causing even the ten iterations each for several ρ values to be significantly large. In our time plots shown later, the late start of ADMM results is due to this cost. Note that this minimal number of ten iterations was essential to get a decent ρ^*.

Now consider GROCK, FPA and HYDRA which are based on using Lipschitz constants (L_j). We found GROCK to be either unstable and diverging or extremely slow. The left side plot in figure 2 depicts these behaviors. The solid red line shows the divergence case. FPA requires an additional parameter (γ) setting for the stochastic approximation step size rule. Our experience is that setting right values for these parameters to get good performance can be tricky and highly dataset dependent. The right side plot in figure 2 shows the extremely slow convergence behavior of FPA; its objective function also shows a non-monotone behavior. Therefore, we do not include GROCK and FPA further in our study.

For HYDRA we tuned L_j as follows. We set the first value of L_j to the theoretical default value proposed in Richtárik and Takáč (2016) and decreased it by a factor of $\beta = 2$ each time to create five values for L_j. Then we ran 25 iterations of HYDRA for each of those five values to chose the best value for L_j and then used that value for all remaining iterations. We found that this simple procedure was sufficient to arrive at a near-best single value for L_j. Unlike ADMM, the cost of this tuning step is negligible compared to the overall cost. Richtárik, and Takáč (Richtárik and Takáč, 2016) also showed results with the asynchronous implementation. For fair comparison with other synchronous approaches, we show results with the synchronous implementation only. Extending our work to the asynchronous setting and comparing with the asynchronous variants of other algorithms is an interesting future work.

16. These initial "tuning" iterations are not counted against the limit of 800 we set for the number of iterations. Thus, for ADMM, the total number of iterations can go higher than 800.

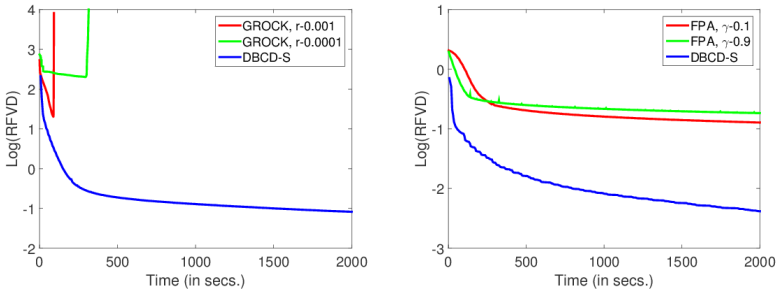


Figure 2: Left: Divergence and slow convergence of GROCK on the URL dataset ($\lambda = 2.4 \times 10^{-6}$ and $P = 25$). Right: Extremely slow convergence of FPA on the KDD dataset ($\lambda = 4.6 \times 10^{-7}$ and $P = 100$).

6.3 Performance Evaluation

We begin by comparing the efficiency of various methods and demonstrating the superiority of the new methods that were developed in section 4 and motivated in section 5. After this we analyze and explain the reasons for the superiority.

Study on AUPRC and RFVD: We have compared the performance of all methods by studying the variation of AUPRC and RFVD as a function of time, for various choices of λ, r (note that r defines the working set size, WSS=rm/P) and the number of nodes (P). To avoid cluttering with too many plots, we provide only representative ones - for each dataset, we choose one value for λ and two values each, for r and P.

Figures 3-6 show the RFVD versus time plots for the four datasets; note the use of log scale for RFVD in those plots. Figures 7-10 show the AUPRC versus time plots. The following observations can be made from these plots.

Superior performance of DBCD-S. In most cases DBCD-S is the best performer. In several of these cases, DBCD-S beats other methods very clearly; for example, on URL with $P = 100$ and $r = 0.01$ (see the bottom left plot of figure 4), the time needed to reach log RFVD=-0.5 is many times smaller than any other method. As another example, with KDD and $r = 0.01$ (see the left side plots in figure 3), if we set the log RFVD value to -2 as the stopping criterion, DBCD-S and PCD-S are faster than all other methods by an order of magnitude. Even in cases where DBCD-S is not the best (e.g., the case in the bottom left plot of figure 5), DBCD-S performs quite close to the best method.

How good is PCD-S? Recall from subsection 4.5 that PCD-S is the variation of the PCDN method Bian et al. (2013) using the S-scheme for variable selection. In some cases such as the last one pointed out in the previous paragraph, PCD-S gives an excellent performance. However, in many other cases, PCD-S does not perform well. This shows that working with quadratic approximations (like PCD-S does) can be quite inferior compared to using the actual nonlinear objective like DBCD-S does.

S-scheme versus R-scheme. In general, the S-scheme of selecting variables (namely, DBCD-S and PCD-S) performs much better than the R-scheme (namely, DBCD-R and

PCD-R), Only on WEBSPAM, PCD-R does slightly better than PCD-S in some cases. One possible reason for this is that WEBSPAM has a small number of examples, causing the communication cost to be much lower than the computation cost; note that the S-scheme requires more computation than the R-scheme.

Effect of r. The choice of r has an effect on the speed of various methods. But the sensitivity is not great. For DBCD-S, a reasonable choice is $r = 0.1$.

Performance of HYDRA. Though HYDRA has a good rate of descent in the objective function during the very early stages, it becomes quite slow soon after, leading to inferior performance. This shows up clearly even in the AUPRC plots.

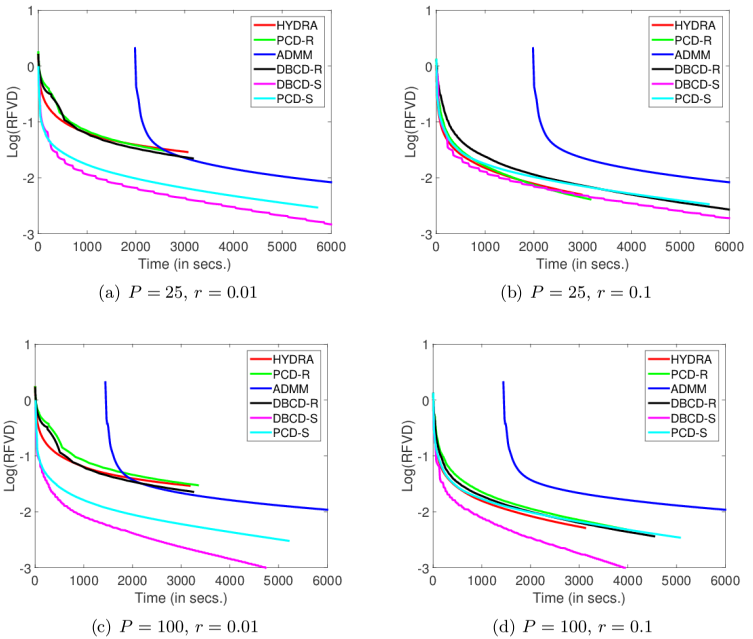
Performance of ADMM. First note that ADMM is independent of r since all the variables are updated. ADMM has a late start due to the time needed for tuning the augmented lagrangian parameter, ρ. (In some cases - see the top two plots in figure 6 - the ADMM curves are not even visible due to the initial tuning cost being relatively large.) Unfortunately, this tuning step is unavoidable; without it, ADMM's performance will be adversely affected. In many cases, DBCD-S reaches excellent solution accuracies even before ADMM begins making any progress.

Consistency between RFVD and AUPRC plots. On KDD, URL and ADS datasets there is good consistency between the two sets of plots. For example, the clear superiority of DBCD-S seen in the top left RFVD plot of figure 4 is also seen in the top left AUPRC plot of figure 8. Only on WEBSPAM (see figure 6 and figure 10) the two sets of plots have some inconsistency; in particular, note that, in figure 6, the initial decrease of the objective function is faster for HYDRA than DBCD-S, while, in figure 10, DBCD-S shows better initial increase in AUPRC than HYDRA. This happens because DBCD-S makes many more variables non-zero and touches many more examples than HYDRA in the initial steps.

Overall, the results point to the choice of DBCD-S as the preferred method as it is highly effective with an order of magnitude improvement over existing methods in many cases. Let us now analyze the reason behind the superior performance of DBCD-S. It is very much along the motivational ideas laid out in section 5: since communication cost dominates computation cost in each outer iteration, DBCD-S reduces overall time by decreasing the number of outer iterations.

Study on the number of outer iterations: We study T^P, the number of outer iterations needed to reach log RFVD$\leq \tau$. Table 6.3 gives T^P values for various methods in various settings. DBCD-S clearly outperforms other methods in terms of having much smaller values for T^P. PCD-S is the second best method, followed by ADMM. The solid reduction of T^P by DBCD-S validates the design that was motivated in section 5. The increased computation associated with DBCD-S is immaterial; because communication cost overshadows computation cost in each iteration for all methods, DBCD-S is also the best in terms of the overall computing time. The next set of results gives the details.

Computation and Communication Time: As emphasized earlier, communication plays an important role in the distributed setting. To study this effect, we measured the computation and communication time separately at each node. Figure 11 shows the computation time per node on the *KDD* dataset. In both cases, ADMM incurs significant computation time compared to other methods. This is because it optimizes over all variables in each node. DBCD-S and DBCD-R come next because our method involves both line search



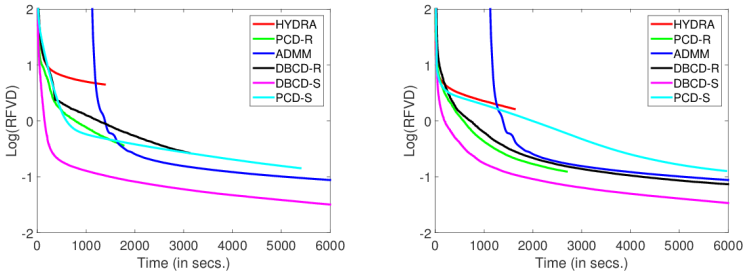
(a) $P = 25$, $r = 0.01$ (b) $P = 25$, $r = 0.1$

(c) $P = 100$, $r = 0.01$ (d) $P = 100$, $r = 0.1$

Figure 3: KDD dataset. Relative function value difference in log scale. $\lambda = 4.6 \times 10^{-7}$

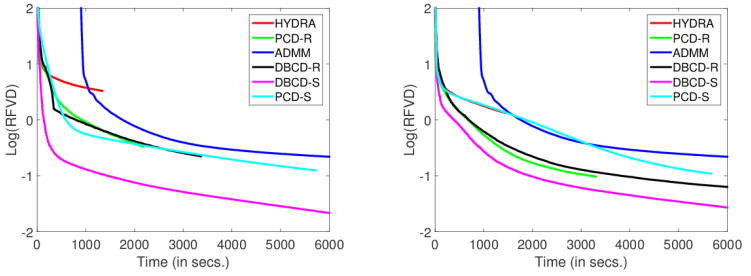
and 10 inner iterations. PCD-R and PCD-S take a little more time than HYDRA because of the line search. As seen in both DBCD and PCD cases, a marginal increase in time is incurred due to the variable selection cost with the S-scheme compared to the R-scheme.

We measured the computation and communication time taken per iteration by each method for different P and r settings. From table 6.3 (which gives representative results for one situation, KDD and $P = 25$), we see that the communication time dominates the cost in HYDRA and PCD-R. DBCD-R takes more computation time than PCD-R and HYDRA since we run through 10 cycles of inner optimization. Note that the methods with S-scheme take more time; however, the increase is not significant compared to the communication cost. DBCD-S takes the maximum computation time and is quite comparable to the communication time. Recall our earlier observation of DBCD-S giving order of magnitude speed-up in the overall time compared to methods such as HYDRA and PCD-R (see figures 3-10). Though the computation times taken by HYDRA, PCD-R and PCD-S are lesser, they need significantly more number of iterations to reach some specified objective function optimality criterion. As a result, these methods become quite inefficient due to extremely large communication cost compared to DBCD. All these observations point to the fact our DBCD method nicely trades-off the computation versus communication cost,



(a) $P = 25$, $r = 0.01$

(b) $P = 25$, $r = 0.1$



(c) $P = 100$, $r = 0.01$

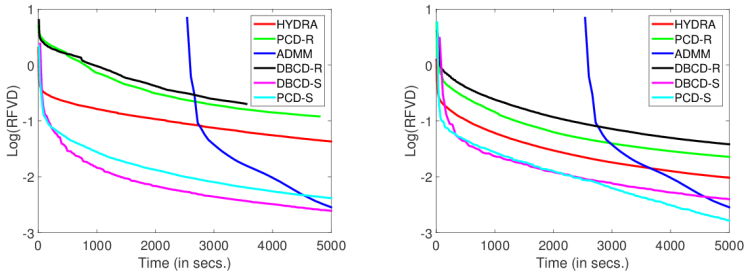
(d) $P = 100$, $r = 0.1$

Figure 4: URL dataset. Relative function value difference in log scale. $\lambda = 9.0 \times 10^{-8}$

and gives an excellent order of magnitude improvement in overall time. With the additional benefit provided by the S-scheme, DBCD-S clearly turns out to be the method of choice for the distributed setting.

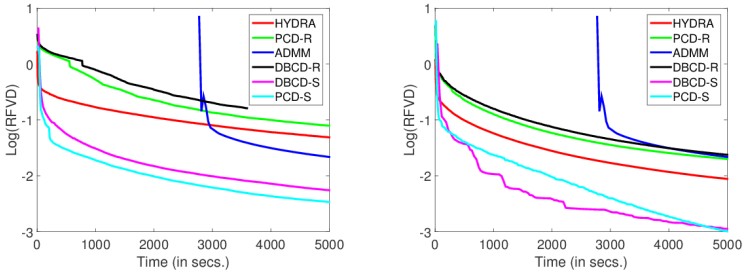
The methods considered in this paper are all synchronous methods. Also, as l_1-regularization gives sparse solutions, load balancing can be prominent and lead to the "curse of last reducer" issue. The increase in waiting time (in our measurement, waiting time is counted as a part of the communication time) is higher for methods that involve greater computation; this is clear from table 6.3 where, roughly, communication time per iteration is higher for methods with higher computation time per iteration. In spite of this, our DBCD-S method, which has the largest computation and communication times per iteration, wins because of the drastically reduced number of iterations compared to other methods.

Sparsity Pattern: To study variable sparsity behaviors of various methods during optimization, we computed the percentage of non-zero variables (ρ) as a function of outer iterations. We set the initial values of the variables to zero. Figure 12 shows similar behaviors for all the random (variable) selection methods. After a few iterations of rise they fall exponentially and remain at the same level. For methods with the S-scheme, many variables remain non-zero for some initial period of time and then ρ falls a lot more sharply.



(a) $P = 100$, $r = 0.01$

(b) $P = 100$, $r = 0.1$



(c) $P = 200$, $r = 0.01$

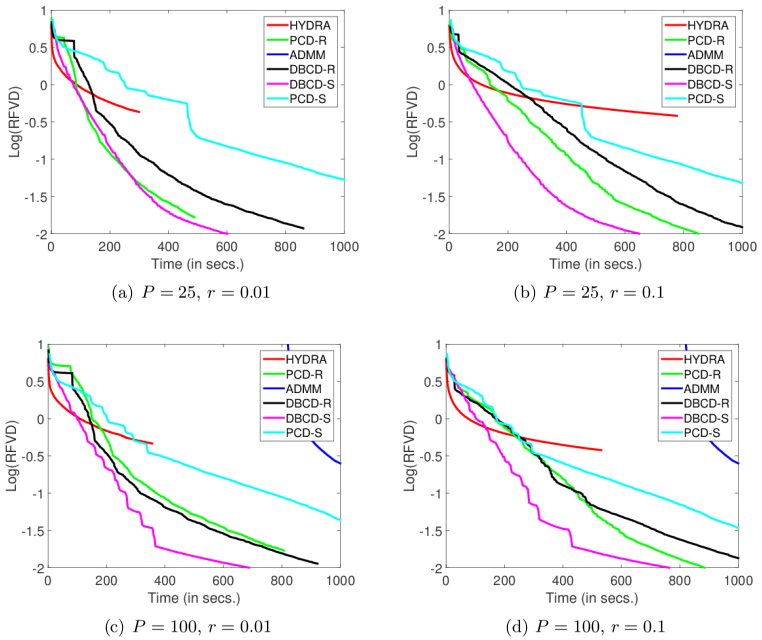
(d) $P = 200$, $r = 0.1$

Figure 5: ADS dataset. Relative function value difference in log scale. $\lambda = 2.65 \times 10^{-6}$.

It is interesting to note that such an initial behavior seems necessary to make good progress in terms of both function value and AUPRC. In all the cases, many variables stay at zero after initial iterations; therefore, shrinking ideas (i.e., do not consider for selection those variables that tend to remain at zero) can be used to improve efficiency.

Remark on Speed up: Let us consider the RFVD plots corresponding to DBCD-S in figures 3 and 4. It can be observed that the times associated with $P = 25$ and $P = 100$ for reaching a certain tolerance, say log RFVD=-2, are close to each other. This means that using 100 nodes gives almost no speed up over 25 nodes, which may prompt the question: *Is a distributed solution really necessary?* There are two answers to this question. First, as we already mentioned, when the training data is huge[17] and so *the data is generated and forced to reside in distributed nodes*, the right question to ask is not whether we get great speed up, but to ask which method is the fastest. Second, for a given dataset, if the time taken to reach a certain optimality tolerance is plotted as a function of P, it may have a minimum at a value different from $P = 1$. In such a case, it is appropriate to choose a P (as well as r) optimally to minimize training time. Many applications involve

17. The KDD and URL datasets are really not huge in the *Big data* sense. In this paper we used them only because of lack of availability of much bigger public datasets.



(a) $P = 25$, $r = 0.01$ (b) $P = 25$, $r = 0.1$

(c) $P = 100$, $r = 0.01$ (d) $P = 100$, $r = 0.1$

Figure 6: WEBSPAM dataset. Relative function value difference in log scale. $\lambda = 3.92 \times 10^{-5}$

periodically repeated model training. For example, in Advertising, logistic regression based click probability models are retrained on a daily basis on incrementally varying datasets. In such scenarios it is worthwhile to spend time to tune parameters such as P and r in an early deployment phase to minimize time, and then use these parameter values for future runs.

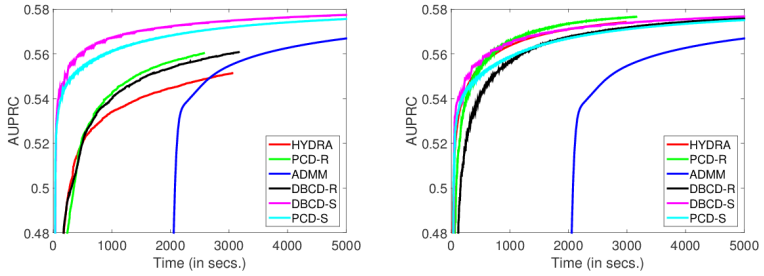
It is also important to point out that the above discussion is relevant to distributed settings in which communication causes a bottleneck. If communication cost is not heavy, e.g., when the number of examples is not large and/or communication is inexpensive such as in multicore solution, then good speed ups are possible; see, for example, the results in Richtárik and Takáč (2015).

7. Recommended DBCD algorithm

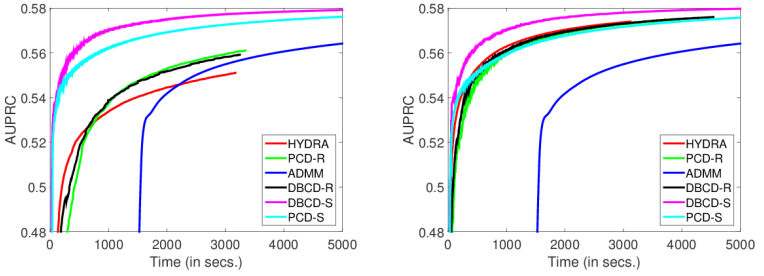
In section 4 we explored various options for the steps of Algorithm 1 looking beyond those considered by existing methods and proposing new ones, and empirically analyzing the various resulting methods in section 6. The experiments clearly show that DBCD-S is the best method. We collect full implementation details of this method in Algorithm 2.

Algorithm 2: Recommended DBCD algorithm

Parameters: Proximal constant $\mu > 0$ (Default: $\mu = 10^{-12}$);
WSS = # variables to choose for updating per node (Default: WSS=$r\,m/P$, $r = 0.1$);
k = # CD iterations to use for solving (3) (Default: $k = 10$);
Line search constants: $\beta, \sigma \in (0, 1)$ (Default: $\beta = 0.5$, $\sigma = 0.01$);
Choose w^0 and compute $y^0 = Xw^0$;
for $t = 0, 1 \ldots$ **do**
  for $p = 1, \ldots, P$ *(in parallel)* **do**
    (a) For each $j \in B_p$, solve (5) to get q_j. Sort $\{q_j : j \in B_p\}$ and choose WSS indices with least q_j values to form S_p^t;
    (b) Form $f_p^t(w_{B_p})$ using (6) and solve (3) using k CD iterations to get $\bar{w}_{B_p}^t$ and set direction: $d_{B_p}^t = \bar{w}_{B_p}^t - w_{B_p}^t$;
    (c) Compute $\delta y^t = \sum_p X_{B_p} d_{B_p}^t$ using AllReduce;
    (d) $\alpha = 1$;
    while *(12-13) are not satisfied* **do**
      $\alpha \leftarrow \alpha\beta$;
      Check (12)-(13) using $y + \alpha\,\delta y$ and aggregating the l_1 regularization value via AllReduce;
    end e
    Set $\alpha^t = \alpha$, $w_{B_p}^{t+1} = w_{B_p}^t + \alpha^t d_{B_p}^t$ and $y^{t+1} = y^t + \alpha^t\,\delta y^t$;
  end f
  Terminate if the optimality conditions (2) hold to the desired approximate level;
end



(a) $P = 25$, $r = 0.01$ (b) $P = 25$, $r = 0.1$



(c) $P = 100$, $r = 0.01$ (d) $P = 100$, $r = 0.1$

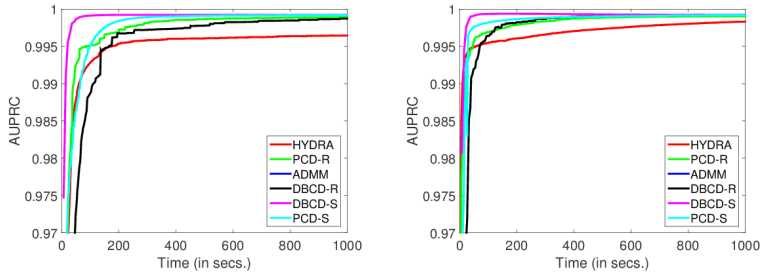
Figure 7: KDD dataset. AUPRC Plots. $\lambda = 4.6 \times 10^{-7}$

8. Conclusion

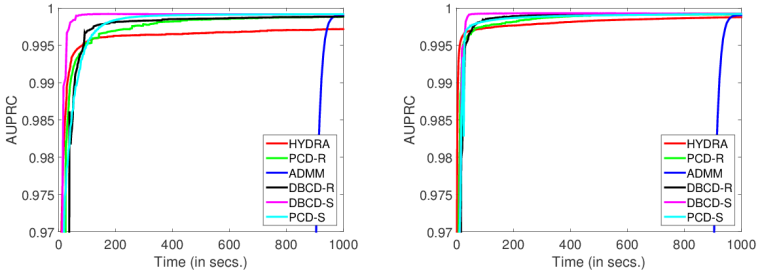
In this paper we have proposed a class of efficient block coordinate methods for the distributed training of l_1 regularized linear classifiers. In particular, the proximal-Jacobi approximation together with a distributed greedy scheme for variable selection came out as a strong performer. There are several useful directions for the future. It would be useful to explore other approximations such as block GLMNET and block L-BFGS suggested in subsection 4.2. Like Richtárik and Takáč (2015), developing a complexity theory for our method that sheds insight on the effect of various parameters (e.g., P) on the number of iterations to reach a specified optimality tolerance is worthwhile. It is possible to extend our method to non-convex problems, e.g., deep net training, which has great value.

Proof of Theorem 1

First let us write δ_j in (11) as $\delta_j = E_{jj} d_j^t$ where $E_{jj} = \delta_j / (d_{B_p}^t)^j$. Note that $|E_{jj}| \leq \mu/2$. Use the condition (14) in Condition 2 with $w_{B_p} = \bar{w}_{B_p}^t$ and $\hat{w}_{B_p} = w_{B_p}$ in (11) together



(a) $P = 25$, $r = 0.01$ (b) $P = 25$, $r = 0.1$



(c) $P = 100$, $r = 0.01$ (d) $P = 100$, $r = 0.1$

Figure 8: URL dataset. AUPRC plots. $\lambda = 9.0 \times 10^{-8}$

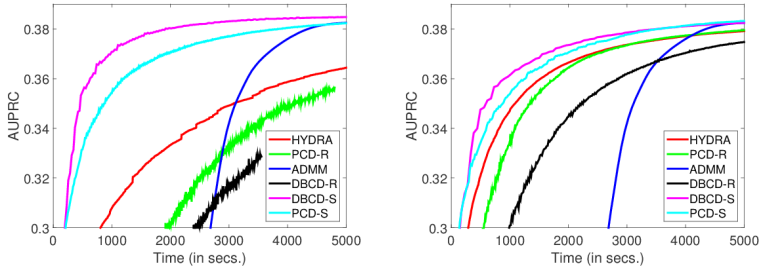
with the gradient consistency property of Condition 1 to get

$$g_{S_p^t}^t + H_{S_p^t}^t d_{S_p^t}^t + \xi_{S_p^t} = 0, \tag{15}$$

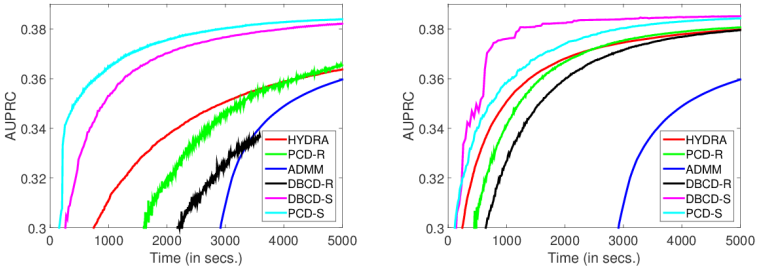
where $H_{S_p^t}^t = \hat{H}_{S_p^t} - E_{S_p^t}$ and $\hat{H}_{S_p^t}$ is the diagonal submatrix of $\hat{H}$ corresponding to S_p^t. Since $\hat{H} \geq \mu I$ and $|E_{jj}| \leq \mu/2$, we get $H_{S_p^t}^t \geq \frac{\mu}{2} I$. Let us extend the diagonal matrix $E_{S_p^t}^t$ to E_{B_p} by defining $E_{jj} = 0 \, \forall j \in B_p \setminus S_p^t$. This lets us extend $H_{S_p^t}^t$ to H_{B_p} via $H_{B_p}^t = \hat{H}_{B_p} - E_{B_p}$.

Now (15) is the optimality condition for the quadratic minimization,

$$d_{B_p}^t = \arg\min_{d_{B_p}} \; (g_{B_p}^t)^T d_{B_p} + \frac{1}{2} (d_{B_p})^T H_{B_p} d_{B_p} + \sum_{j \in B_p} \lambda \, |w_j^t + d_j| \;\; \text{s.t.} \;\; d_j = 0 \; \forall \, j \in B_p \setminus S_p^t \tag{16}$$



(a) $P = 100$, $r = 0.01$ (b) $P = 100$, $r = 0.1$



(c) $P = 200$, $r = 0.01$ (d) $P = 200$, $r = 0.1$

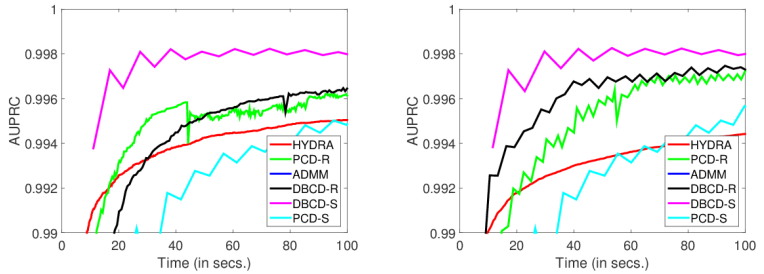
Figure 9: ADS dataset. AUPRC Plots. $\lambda = 2.65 \times 10^{-6}$

Combined over all p,

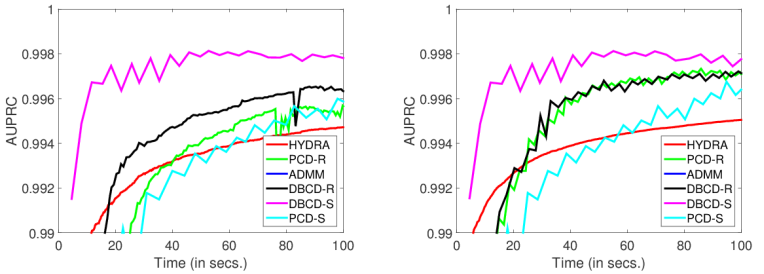
$$d^t = \arg\min_d \ (g^t)^T d + \frac{1}{2} d^T H d + u(w^t + d)$$
$$\text{s.t. } d_j = 0 \ \forall \ j \in \cup_p (B_p \setminus S_p^t) \tag{17}$$

where H is a block diagonal matrix with blocks, $\{H_{B_p}\}$. Thus d^t corresponds to the minimization of a positive definite quadratic form, exactly the type covered by the Tseng-Yun theory (Tseng and Yun, 2009).

The line search condition (12)-(13) is a special case of the line search condition in Tseng and Yun (2009). The Gauss-Seidel scheme of subsection 4.1 is an instance of the Gauss-Seidel scheme of Tseng and Yun (2009). Now consider the distributed greedy scheme in subsection 4.1. Let $j_{\max} = \arg\max_{1 \le j \le m} \bar{q}_j$. By the way the S_p^t are chosen, $j_{\max} \in \cup_p S_p^t$. Therefore, $\sum_{j \in \cup_p S_p^t} \bar{q}_j \le \frac{1}{m} \sum_{j=1}^m \bar{q}_j$, thus satisfying the Gauss-Southwell-q rule condition of Tseng and Yun (2009). Now Theorems 1-4 of Tseng and Yun (2009) can be directly applied to prove our Theorem 1. Note that Theorem 4 of Tseng and Yun (2009) ensures that Assumption 2(a) of that paper holds; Assumption 2(b) of that paper trivially holds because, in our case F is convex.



(a) $P = 25$, $r = 0.01$ (b) $P = 25$, $r = 0.1$



(c) $P = 100$, $r = 0.01$ (d) $P = 100$, $r = 0.1$

Figure 10: WEBSPAM dataset. AUPRC Plots. $\lambda = 3.92 \times 10^{-5}$. Because the initial ρ tuning time for ADMM is large, its curves are not seen in the shown time window of 0-100 secs.

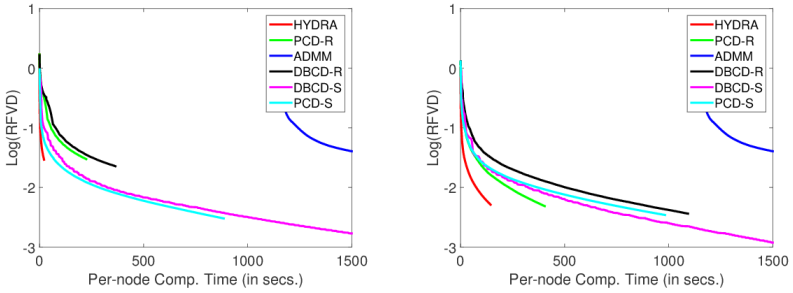
References

A. Agarwal, O. Chapelle, M. Dudik, and J. Langford. A reliable effective terascale linear learning system. *JMLR*, 15:1111–1133, 2013.

Y. Bian, X. Li, and Y. Liu. Parallel coordinate descent Newton for large scale L1 regularized minimization. *arXiv:1306.4080v1*, 2013.

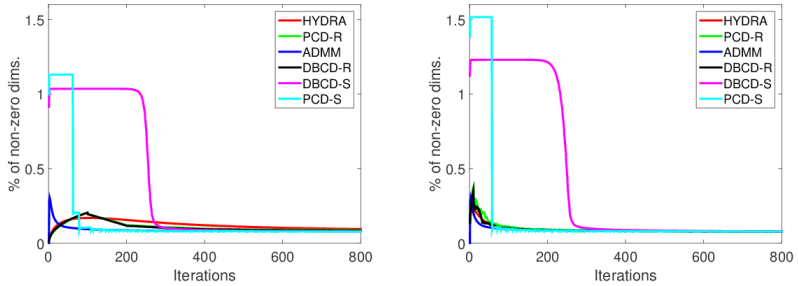
S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, pages 1–122, 2011.

J.K. Bradley, A. Kyrola, D. Bickson, and C. Guestrin. Parallel coordinate descent for l_1-regularized loss minimization. *ICML*, pages 321–328, 2011.



(a) $r = 0.01$ (b) $r = 0.1$

Figure 11: Per-node computation time on the KDD dataset ($\lambda = 4.6 \times 10^{-7}$ and $P = 100$).



(a) $r = 0.01$ (b) $r = 0.1$

Figure 12: KDD dataset: Typical behavior of the percentage of non-zero variables.

W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, pages 889–916, 2016.

W. Deng, M-J. Lai, and W. Yin. On the $o(\frac{1}{k})$ convergence and parallelization of the alternating direction method of multipliers. *arXiv:1312.3040*, 2013.

F. Facchinei, S. Sagratella, and G. Scutari. Flexible parallel algorithms for big data optimization. *ICASSP*, 2014.

J. H. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.

T. Goldstein, O'Donoghue, S. Setzer, and R. Baraniuk. Fast alternating direction optimization methods. *SIAM Journal on Imaging Sciences*, 7:1588–1623, 2014.

M. Jaggi, V. Smith, M Takáč, J. Terhorst, S. Krishnan, T. Hofmann, and M.I. Jordan. Communication-efficient distributed dual coordinate ascent. *NIPS*, 2014.

M. Li, D.G. Andersen, J.W. Park, A.J. Smola, A. Ahmed, J. Josifovski, J. Long, E.J. Shekita, and B. Su. Scaling distributed machine learning with the parameter server. *OSDI*, 2014.

C. Ma, V. Smith, M. Jaggi, M.I. Jordan, M Takáč, and P. Richtárik. Adding vs. averaging in distributed primal-dual optimization. *ICML*, 2015.

I. Necoara and D. Clipici. Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC. *Journal of Process Control*, 23:243–253, 2014.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal of Optimization*, pages 341–362, 2012.

J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, New York, 1970.

N. Parikh and S. Boyd. Block splitting of distributed optimization. *Math. Prog. Comp.*, 2013.

M. Patriksson. Cost approximation: A unified framework of descent algorithms for nonlinear programs. *SIAM J. Optim.*, 8:561–582, 1998a.

M. Patriksson. Decomposition methods for differentiable optimization problems over cartesian product sets. *Comput. Optim. Appl.*, 9:5–42, 1998b.

Z. Peng, M. Yan, and W. Yin. Parallel and distributed sparse optimization. *IEEE Asilomar Conference on Signals, Systems, and Computers*, 2013.

C. Ravazzi, S. M. Fosson, and E. Magli. Distributed soft thresholding for sparse signal recovery. *Proceedings of IEEE Global Communications Conference*, 2013.

P. Richtárik and M Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144:1–38, 2014.

P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 2015.

P. Richtárik and M. Takáč. Distributed coordinate descent method for learning with big data. *JMLR*, 17:1–25, 2016.

C. Scherrer, M. Halappanavar, A. Tewari, and D. Haglin. Scaling up coordinate descent algorithms for large l_1 regularization problems. *ICML*, pages 1407–1414, 2012a.

C. Scherrer, A. Tewari, M. Halappanavar, and D. Haglin. Feature clustering for accelerating parallel coordinate descent. *NIPS*, pages 28–36, 2012b.

S. Shalev-Shwartz and A. Tewari. Stochastic methods for $l1$ regularized loss minimization. *JMLR*, 12:1865–1892, 2011.

S. T. Sonnenburg and V. Franc. COFFIN: a computational framework for linear SVMs. *ICML*, 2010.

P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423, 2009.

G. X. Yuan, K. W. Chang, C. J. Hsieh, and C. J. Lin. A comparison of optimization methods and software for large-scale l_1-regularized linear classification. *JMLR*, pages 3183–3234, 2010.

G. X. Yuan, C. H. Ho, and C. J. Lin. An improved GLMNET for L1-regularized logistic regression and support vector machines. *JMLR*, pages 1999–2030, 2012.

S. Yun, P. Tseng, and K.C. Toh. A coordinate gradient descent method for L1-regularized convex minimization. *Computational Optimization and Applications*, 48:273–307, 2011.

KDD, $\lambda = 4.6 \times 10^{-7}$

		Existing methods		Our methods		
P	τ	HYDRA	PCD-R	PCD-S	DBCD-R	DBCD-S
25	-1	298	294	12	236	**8**
	-2	> 800	> 800	331	> 800	**124**
	-3	> 800	> 800	> 800	> 800	**688**
100	-1	297	299	13	230	**10**
	-2	> 800	> 800	311	> 800	**137**
	-3	> 800	> 800	> 800	> 800	**797**

URL, $\lambda = 9.0 \times 10^{-8}$

		Existing methods		Our methods		
25	0	> 2000	878	201	770	**23**
	-0.5	> 2000	> 2000	929	1772	**42**
	-1	> 2000	> 2000	> 2000	> 2000	**216**
100	0	> 2000	840	197	476	**27**
	-0.5	> 2000	> 2000	858	1488	**55**
	-1	> 2000	> 2000	> 2000	> 2000	**287**

WEBSPAM, $\lambda = 3.9 \times 10^{-5}$

		Existing methods		Our methods		
25	0	521	202	56	169	**13**
	-0.5	> 1500	185	109	128	**25**
	-1.0	> 1500	532	180	439	**39**
100	0	511	203	56	180	**28**
	-0.5	> 1500	308	111	265	**49**
	-1.0	> 1500	525	235	403	**74**

ADS, $\lambda = 2.6 \times 10^{-6}$

		Existing methods		Our methods		
100	-1.0	573	> 600	10	> 600	**7**
	-1.5	> 600	> 600	61	> 600	**18**
	-2.0	> 600	> 600	169	> 600	**53**
200	-1.0	569	> 600	10	> 600	**11**
	-1.5	> 600	> 600	35	> 600	**46**
	-2.0	> 600	> 600	154	> 600	**138**

Table 5: T^P, the number of outer iterations needed to reach log RFVD$\leq \tau$, for various P and τ values. We set $r = 0.01$. Best values are indicated in boldface. For each dataset, the τ values were chosen to cover the region where AUPRC values are in the process of reaching the steady state value. The methods were terminated when the number of iterations exceeded a maximum; this maximum was set differently for different datasets.

KDD, $\lambda = 4.6 \times 10^{-7}$

Method	Comp.	Comm.	Comp.	Comm.
	$r = 0.01$		$r = 0.1$	
HYDRA	0.033	1.665	0.250	1.404
PCD-R	0.125	1.955	0.360	2.093
PCD-S	1.483	3.046	1.598	2.703
DBCD-R	0.311	2.173	1.733	2.396
DBCD-S	4.426	2.263	5.317	2.387

URL, $\lambda = 9.0 \times 10^{-8}$

Method	Comp.	Comm.	Comp.	Comm.
	$r = 0.01$		$r = 0.1$	
HYDRA	0.018	0.443	0.123	0.418
PCD-R	0.033	0.834	0.126	1.157
PCD-S	0.780	1.527	0.915	1.606
DBCD-R	0.124	1.380	0.741	1.956
DBCD-S	3.591	2.438	4.151	1.731

WEBSPAM, $\lambda = 3.9 \times 10^{-5}$

Method	Comp.	Comm.	Comp.	Comm.
	$r = 0.01$		$r = 0.1$	
HYDRA	0.029	0.137	0.284	0.146
PCD-R	0.022	0.374	0.201	0.973
PCD-S	1.283	1.389	1.156	1.498
DBCD-R	0.121	0.535	1.046	0.995
DBCD-S	2.254	1.089	2.559	1.080

ADS, $\lambda = 2.6 \times 10^{-6}$

Method	Comp.	Comm.	Comp.	Comm.
	$r = 0.01$		$r = 0.1$	
HYDRA	0.133	3.652	0.695	4.267
PCD-R	0.198	4.208	0.880	5.784
PCD-S	1.636	6.326	2.220	7.888
DBCD-R	0.197	5.923	0.965	11.05
DBCD-S	1.244	8.285	2.709	13.8

Table 6: Computation and communication costs per iteration (in secs.) for KDD, $P = 25$.

Journal of Machine Learning Research 18 (2017) 1-31 Submitted 11/15; Revised 6/17; Published 9/17

Distributed Learning with Regularized Least Squares

Shao-Bo Lin sblin1983@gmail.com
Department of Mathematics
City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
Xin Guo x.guo@polyu.edu.hk
Department of Applied Mathematics
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
Ding-Xuan Zhou mazhou@cityu.edu.hk
Department of Mathematics
City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong

Editor: Ingo Steinwart

Abstract

We study distributed learning with the least squares regularization scheme in a reproducing kernel Hilbert space (RKHS). By a divide-and-conquer approach, the algorithm partitions a data set into disjoint data subsets, applies the least squares regularization scheme to each data subset to produce an output function, and then takes an average of the individual output functions as a final global estimator or predictor. We show with error bounds and learning rates in expectation in both the L^2-metric and RKHS-metric that the global output function of this distributed learning is a good approximation to the algorithm processing the whole data in one single machine. Our derived learning rates in expectation are optimal and stated in a general setting without any eigenfunction assumption. The analysis is achieved by a novel second order decomposition of operator differences in our integral operator approach. Even for the classical least squares regularization scheme in the RKHS associated with a general kernel, we give the best learning rate in expectation in the literature.

Keywords: Distributed learning, divide-and-conquer, error analysis, integral operator, second order decomposition.

1. Introduction and Distributed Learning Algorithms

In the era of big data, the rapid expansion of computing capacities in automatic data generation and acquisition brings data of unprecedented size and complexity, and raises a series of scientific challenges such as storage bottleneck and algorithmic scalability (Zhou et al., 2014). To overcome the difficulty, some approaches for generating scalable approximate algorithms have been introduced in the literature such as low-rank approximations of kernel matrices for kernel principal component analysis (Schölkopf et al., 1998; Bach, 2013), incomplete Cholesky decomposition (Fine, 2002), early-stopping of iterative optimization algorithms for gradient descent methods (Yao et al., 2007; Raskutti et al., 2014; Lin et al., 2016), and greedy-type algorithms. Another method proposed recently is distributed learning based on a divide-and-conquer approach and a particular learning algorithm implemented in individual machines (Zhang et al., 2015; Shamir and Srebro, 2014). This method produces distributed learning algorithms consisting of three steps: partitioning the data into disjoint subsets, applying a particular learning algorithm implemented in an individual machine to each data subset to produce an individual output (function), and synthesizing a global output by utilizing some average of the individual outputs. This method can significantly reduce computing time and lower single-machine memory requirements. For practical applications in medicine, finance, business and some other areas, the data are often stored naturally across multiple servers in a distributive way and are not combined for reasons of protecting privacy and avoiding high costs. In such situations, the first step of data partitioning is not needed. It has been observed in many practical applications that when the number of data subsets is not too large, the divide-and-conquer approach does not cause noticeable loss of performance, compared with the learning scheme which processes the whole data on a single big machine. Theoretical attempts have been recently made in (Zhang et al., 2013, 2015) to derive learning rates for distributed learning with least squares regularization schemes under certain assumptions.

This paper aims at error analysis of the *distributed learning* with regularized least squares and its approximation to the algorithm processing the whole data in one single machine. Recall (Cristianini and Shawe-Taylor, 2000; Evgeniou et al., 2000) that in a reproducing kernel Hilbert space (RKHS) $(\mathcal{H}_K, \|\cdot\|_K)$ induced by a Mercer kernel K on an input metric space $\mathcal{X}$, with a sample $D = \{(x_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$ where $\mathcal{Y} = \mathbb{R}$ is the output space, the least squares regularization scheme can be stated as

$$f_{D,\lambda} = \arg\min_{f \in \mathcal{H}_K} \left\{ \frac{1}{|D|} \sum_{(x,y)\in D} (f(x) - y)^2 + \lambda \|f\|_K^2 \right\}. \tag{1}$$

Here $\lambda > 0$ is a regularization parameter and $|D| =: N$ is the cardinality of D. This learning algorithm is also called kernel ridge regression in statistics and has been well studied in learning theory. See e.g. (De Vito et al., 2005; Caponnetto and De Vito, 2007; Steinwart et al., 2009; Bauer et al., 2007; Smale and Zhou, 2007; Steinwart and Christmann, 2008). The regularization scheme (1) can be explicitly solved by using a standard matrix inversion technique, which requires costs of $\mathcal{O}(N^3)$ in time and $\mathcal{O}(N^2)$ in memory. However, this matrix inversion technique may not conquer challenges on storages or computations arising from big data.

The distributed learning algorithm studied in this paper starts with partitioning the data set D into m disjoint subsets $\{D_j\}_{j=1}^m$. Then it assigns each data subset D_j to one machine or processor to produce a local estimator $f_{D_j,\lambda}$ by the least squares regularization scheme (1). Finally, these local estimators are communicated to a central processor, and a global estimator $\overline{f}_{D,\lambda}$ is synthesized by taking a weighted average

$$\overline{f}_{D,\lambda} = \sum_{j=1}^m \frac{|D_j|}{|D|} f_{D_j,\lambda} \tag{2}$$

of the local estimators $\{f_{D_j,\lambda}\}_{j=1}^m$. This algorithm has been studied with a matrix analysis approach in (Zhang et al., 2015) where some error analysis has been conducted under some eigenfunction assumptions for the integral operator associated with the kernel, presenting error bounds in expectation.

©2017 Lin, Guo and Zhou.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/15-586.html.

JMLR 18(92):1-31, 2017

In this paper we shall use a novel integral operator approach to prove that $\overline{f}_{D,\lambda}$ is a good approximation of $f_{D,\lambda}$. We present a representation of the difference $\overline{f}_{D,\lambda} - f_{D,\lambda}$ in terms of empirical integral operators, and analyze the error $\overline{f}_{D,\lambda} - f_{D,\lambda}$ in expectation without any eigenfunction assumptions. As a by-product, we present the best learning rates in expectation for the least squares regularization scheme (1) in a general setting, which surprisingly has not been done for a general kernel in the literature (see detailed comparisons below).

2. Main Results

Our analysis is carried out in the standard least squares regression framework with a Borel probability measure ρ on $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, where the input space $\mathcal{X}$ is a compact metric space. The sample D is independently drawn according to ρ. The Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ defines an integral operator L_K on $\mathcal{H}_K$ as

$$L_K(f) = \int_{\mathcal{X}} K_x f(x) d\rho_X, \qquad f \in \mathcal{H}_K, \tag{3}$$

where K_x is the function $K(\cdot, x)$ in $\mathcal{H}_K$ and ρ_X is the marginal distribution of ρ on $\mathcal{X}$.

2.1 Error Bounds for the Distributed Learning Algorithm

Our error bounds in expectation for the distributed learning algorithm (2) require the uniform boundedness condition for the output y, that is, for some constant $M > 0$, there holds $|y| \le M$ almost surely. Our bounds are stated in terms of the approximation error

$$\|f_\lambda - f_\rho\|_\rho, \tag{4}$$

where f_λ is the data-free limit of (1) defined by

$$f_\lambda = \arg\min_{f \in \mathcal{H}_K} \left\{ \int_{\mathcal{Z}} (f(x) - y)^2 d\rho + \lambda \|f\|_K^2 \right\}, \tag{5}$$

$\|\cdot\|_\rho$ denotes the norm of $L^2_{\rho_X}$, the Hilbert space of square integrable functions with respect to ρ_X, and f_ρ is the regression function (conditional mean) of ρ defined by

$$f_\rho(x) = \int_{\mathcal{Y}} y d\rho(y|x), \qquad x \in \mathcal{X},$$

with $\rho(\cdot|x)$ being the conditional distribution of ρ induced at $x \in \mathcal{X}$.

Since K is continuous, symmetric and positive semidefinite, L_K is a compact positive operator of trace class and $L_K + \lambda I$ is invertible. Define a quantity measuring the complexity of $\mathcal{H}_K$ with respect to ρ_X, the *effective dimension* (Zhang, 2005), to be the trace of the operator $(L_K + \lambda I)^{-1} L_K$ as

$$\mathcal{N}(\lambda) = \mathrm{Tr}\left((L_K + \lambda I)^{-1} L_K \right), \qquad \lambda > 0. \tag{6}$$

In Section 6 we shall prove the following first main result of this paper concerning error bounds in expectation of $\overline{f}_{D,\lambda} - f_{D,\lambda}$ in $\mathcal{H}_K$ and in $L^2_{\rho_X}$. Denote $\kappa = \sup_{x \in \mathcal{X}} \sqrt{K(x,x)}$.

Theorem 1 *Assume $|y| \le M$ almost surely. If $|D_j| = \frac{N}{m}$ for $j = 1, \ldots, m$, then we have*

$$E \left\| \overline{f}_{D,\lambda} - f_{D,\lambda} \right\|_\rho \le C_\kappa \left\{ M\sqrt{\lambda m}\mathcal{C}_m \left\{ 1 + \sqrt{m\mathcal{C}_m} \right\} + \frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}} m \left(\sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}} + \sqrt{m}\mathcal{C}_m \right) \right\}$$

and

$$E \left\| \overline{f}_{D,\lambda} - f_{D,\lambda} \right\|_K \le C_\kappa \left\{ M\sqrt{m}\mathcal{C}_m \left\{ 1 + \sqrt{m\mathcal{C}_m} \right\} + \frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}} m \left(\sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}} + \sqrt{m}\mathcal{C}_m \right) \right\},$$

where C_κ is a constant depending only on κ, and $\mathcal{C}_m$ is the quantity given in terms of $m, N, \lambda, \mathcal{N}(\lambda)$ by

$$\mathcal{C}_m := \frac{m}{(N\lambda)^2} + \frac{\mathcal{N}(\lambda)}{N\lambda}.$$

To derive explicit learning rates from the general error bounds in Theorem 1, one can quantify the increment of $\mathcal{N}(\lambda)$ by the following *capacity assumption*, a characteristic of the complexity of the hypothesis space (Caponnetto and De Vito, 2007; Blanchard and Krämer, 2010),

$$\mathcal{N}(\lambda) \le c\lambda^{-\beta}, \qquad \forall \lambda > 0 \tag{7}$$

for some constants $\beta > 0$ and $c > 0$. Let $\{(\lambda_\ell, \varphi_\ell)\}_\ell$ be a set of normalized eigenpairs of L_K on $\mathcal{H}_K$ with $\{\varphi_\ell\}_\ell$ being an orthonormal basis of $\mathcal{H}_K$ and $\{\lambda_\ell\}_\ell$ arranged in a non-increasing order. A sufficient condition for the capacity assumption (7) with $0 < \beta < 1$ is $\lambda_\ell = O(\ell^{-1/\beta})$, which can be found in, e.g. Caponnetto and De Vito (2007).

Remark 2 *The sufficient condition $\lambda_\ell = O(\ell^{-1/\beta})$ with the index $\beta = \frac{d}{2\tau} < 1$ is satisfied by the Sobolev space $W^\tau(B(\mathbb{R}^d))$ with the smoothness index $\tau > d/2$ on a ball $B(\mathbb{R}^d)$ of the Euclidean space $\mathbb{R}^d$ when the marginal distribution ρ_X is the uniform distribution on $B(\mathbb{R}^d)$, see (Steinwart et al., 2009; Edmunds and Triebel, 1996).*

Condition (7) with $\beta = 1$ always holds true with the choice of the constant $c = \kappa^2$. In fact, the eigenvalues of the operator $(L_K + \lambda I)^{-1} L_K$ are $\{\frac{\lambda_\ell}{\lambda_\ell + \lambda}\}_\ell$. So its trace is given by $\mathcal{N}(\lambda) = \sum_\ell \frac{\lambda_\ell}{\lambda_\ell + \lambda} \le \sum_\ell \frac{\lambda_\ell}{\lambda} = \frac{\mathrm{Tr}(L_K)}{\lambda} \le \kappa^2 \lambda^{-1}$.

In the existing literature on learning rates for the classical least squares algorithm (1), the regularization parameter λ is often taken to satisfy the restriction $\frac{\mathcal{N}(\lambda)}{N\lambda} = O(1)$ as in (Caponnetto and De Vito, 2007) up to a logarithmic factor or in (Steinwart et al., 2009) under some assumptions on (K, ρ_X) (see (14) below). Here, to derive learning rates for $E \left\| \overline{f}_{D,\lambda} - f_{D,\lambda} \right\|_\rho$ with $m \ge 1$ corresponding to the distributed learning algorithm (2), we consider λ to satisfy the following restriction with some constant $C_0 > 0$,

$$0 < \lambda \le C_0 \quad \text{and} \quad \frac{m\mathcal{N}(\lambda)}{N\lambda} \le C_0. \tag{8}$$

Corollary 3 *Assume $|y| \le M$ almost surely. If $|D_j| = \frac{N}{m}$ for $j = 1, \ldots, m$, and λ satisfies (8), then we have*

$$E \left\| \overline{f}_{D,\lambda} - f_{D,\lambda} \right\|_\rho \le \widetilde{C}_\kappa \sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}} \left\{ M\sqrt{\lambda} + \frac{m\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}} \right\}$$

and

$$E\left\|\overline{f}_{D,\lambda}-f_{D,\lambda}\right\|_K \le \widetilde{C}_\kappa \sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}}\left\{M+\frac{m\|f_\rho-f_\lambda\|_\rho}{\sqrt{N\lambda}}\right\},$$

where $\widetilde{C}_\kappa$ is a constant depending only on κ, C_0, and the largest eigenvalue of L_K.

In the special case that $f_\rho \in \mathcal{H}_K$, the approximation error can be bounded as $\|f_\lambda - f_\rho\|_\rho \le \|f_\rho\|_K \sqrt{\lambda}$. A more general *regularity condition* can be imposed for the regression function as

$$f_\rho = L_K^r(g_\rho) \quad \text{for some } g_\rho \in L^2_{\rho_X},\ r>0, \tag{9}$$

where the integral operator L_K is regarded as a compact positive operator on $L^2_{\rho_X}$ and its rth power L_K^r is well defined for any $r>0$. The condition (9) means f_ρ lies in the range of L_K^r, and the special case $f_\rho \in \mathcal{H}_K$ corresponds to the choice $r=1/2$. It implies $\|f_\lambda - f_\rho\|_\rho \le \lambda^r \|g_\rho\|_\rho$ by Lemma 21 below. Thus, under condition (9), we obtain from Corollary 3, by choosing λ to minimize the order of the error bound subject to the restriction (8), the following nice convergence rates for the error $\left\|\overline{f}_{D,\lambda}-f_{D,\lambda}\right\|$ of the distributed learning algorithm (2).

Corollary 4 *Assume regularity condition (9) for some $0<r\le 1$, capacity assumption (7) with $\beta=\frac{1}{2\alpha}$ for some $\alpha>0$, and $|y|\le M$ almost surely. If $|D_j|=\frac{N}{m}$ for $j=1,\dots,m$ with*

$$m \le N^{\frac{\alpha\max\{2r,1\}+\frac{1}{2}+2\alpha(r-1)}{2\alpha\max\{2r,1\}+1+2\alpha(r-1)}}, \tag{10}$$

and $\lambda=\left(\frac{m}{N}\right)^{\frac{2\alpha}{2\alpha\max\{2r,1\}+1}}$, then we have

$$E\left\|\overline{f}_{D,\lambda}-f_{D,\lambda}\right\|_\rho \le \widetilde{C}_{\kappa,c,r}\frac{1}{\sqrt{m}}\left(\frac{m}{N}\right)^{\frac{\alpha\max\{2r,1\}}{2\alpha\max\{2r,1\}+1}}$$

and

$$E\left\|\overline{f}_{D,\lambda}-f_{D,\lambda}\right\|_K \le \widetilde{C}_{\kappa,c,r}\frac{1}{\sqrt{m}}\left(\frac{m}{N}\right)^{\frac{\alpha\max\{2r-1,0\}}{2\alpha\max\{2r,1\}+1}},$$

where $\widetilde{C}_{\kappa,c,r}$ is a constant independent of N or m.

In particular, when $f_\rho\in\mathcal{H}_K$ and $m\le N^{\frac{1}{2+2\alpha}}$, taking $\lambda=\left(\frac{m}{N}\right)^{\frac{2\alpha}{2\alpha+1}}$ yields the rates $E\left\|\overline{f}_{D,\lambda}-f_{D,\lambda}\right\|_\rho \le \widetilde{C}_{\kappa,c,r}N^{-\frac{\alpha}{2\alpha+1}}m^{-\frac{1}{4\alpha+2}}$ and $E\left\|\overline{f}_{D,\lambda}-f_{D,\lambda}\right\|_K \le \widetilde{C}_{\kappa,c,r}\frac{1}{\sqrt{m}}$.

Remark 5 *In Corollary 4, we present learning rates in both $\mathcal{H}_K$ and $L^2_{\rho_X}$ norms. The rates with respect to the $L^2_{\rho_X}$ norm provide estimates for the generalization ability of the algorithm for regression. The rates with respect to the $\mathcal{H}_K$ norm give error estimates with respect to the uniform metric due to the relation $\|f\|_\infty \le \kappa\|f\|_K$, and might be used to solve some mismatch problems in learning theory where the generalization ability of learning algorithms is measured with respect to a probability measure μ different from ρ_X.*

Remark 6 *The learning rates in Corollary 4 are stated for the norms of the difference $\overline{f}_{D,\lambda}-f_{D,\lambda}$ which reflects the variance of the distributed learning algorithm (2). These rates decrease as m increases (subject to the restriction (10)) and thereby the regularization parameter λ increases, which is different from the learning rates presented for $E\left\|\overline{f}_{D,\lambda}-f_\rho\right\|_\rho$ in (Zhang et al., 2015). To derive learning rates for $E\left\|\overline{f}_{D,\lambda}-f_\rho\right\|_\rho$ by our analysis, the regularization parameter λ should be independent of m, as shown in Corollary 11 below.*

2.2 Optimal Learning Rates for Least Squares Regularization Scheme

The second main result of this paper is optimal learning rates in expectation for the least squares regularization scheme (1). We can even relax the uniform boundedness to a *moment condition* that for some constant $p\ge 1$,

$$\sigma_\rho^2 \in L^p_{\rho_X}, \tag{11}$$

where σ_ρ^2 is the conditional variance defined by $\sigma_\rho^2(x)=\int_{\mathcal{Y}}(y-f_\rho(x))^2\,d\rho(y|x)$.

The following learning rates in expectation for the least squares regularization scheme (1) will be proved in Section 5. The existence of f_λ is ensured by $E[y^2]<\infty$.

Theorem 7 *Assume $E[y^2]<\infty$ and moment condition (11) for some $1\le p\le\infty$. Then we have*

$$E\left[\|f_{D,\lambda}-f_\rho\|_\rho\right] \le \left(2+56\kappa^4+57\kappa^2\right)(1+\kappa)\left(1+\frac{1}{(N\lambda)^2}+\frac{\mathcal{N}(\lambda)}{N\lambda}\right)$$
$$\left\{\left(1+\frac{1}{\sqrt{N\lambda}}\right)\|f_\lambda-f_\rho\|_\rho+\sqrt{\left\|\sigma_\rho^2\right\|_p}\left(\frac{\mathcal{N}(\lambda)}{N}\right)^{\frac{1}{2}(1-\frac{1}{p})}\left(\frac{1}{N\lambda}\right)^{\frac{1}{2p}}\right\}. \tag{12}$$

If the parameters satisfy $\frac{\mathcal{N}(\lambda)}{N\lambda}=O(1)$, we have the following explicit bound.

Corollary 8 *Assume $E[y^2]<\infty$ and moment condition (11) for some $1\le p\le\infty$. If λ satisfies (8) with $m=1$, then we have*

$$E\left[\|f_{D,\lambda}-f_\rho\|_\rho\right]=O\left(\|f_\lambda-f_\rho\|_\rho+\left(\frac{\mathcal{N}(\lambda)}{N}\right)^{\frac{1}{2}(1-\frac{1}{p})}\left(\frac{1}{N\lambda}\right)^{\frac{1}{2p}}\right).$$

In particular, if $p=\infty$, that is, the conditional variances are uniformly bounded, then

$$E\left[\|f_{D,\lambda}-f_\rho\|_\rho\right]=O\left(\|f_\lambda-f_\rho\|_\rho+\sqrt{\frac{\mathcal{N}(\lambda)}{N}}\right).$$

In particular, when regularity condition (9) is satisfied, we have the following learning rates in expectation.

Corollary 9 *Assume $E[y^2]<\infty$, moment condition (11) for some $1\le p\le\infty$, and regularity condition (9) for some $0<r\le 1$. If the capacity assumption $\mathcal{N}(\lambda)=O(\lambda^{-\frac{1}{2\alpha}})$ holds with some $\alpha>0$, then by taking $\lambda=N^{-\frac{2\alpha}{2\alpha\max\{2r,1\}+1}}$ we have*

$$E\left[\|f_{D,\lambda}-f_\rho\|_\rho\right]=O\left(N^{-\frac{2r\alpha}{2\alpha\max\{2r,1\}+1}+\frac{1}{2p}\frac{2\alpha-1}{2\alpha\max\{2r,1\}+1}}\right).$$

In particular, when $p=\infty$ (the conditional variances are uniformly bounded), we have

$$E\left[\|f_{D,\lambda}-f_\rho\|_\rho\right]=O\left(N^{-\frac{2r\alpha}{2\alpha\max\{2r,1\}+1}}\right).$$

Remark 10 *It was shown in (Caponnetto and De Vito, 2007; Steinwart et al., 2009; Bauer et al., 2007) that under the condition of Corollary 9 with $p = \infty$ and $r \in [\frac{1}{2}, 1]$, the best learning rate, called minimax lower rate of convergence, for learning algorithms with output functions in $\mathcal{H}_K$ is $O\left(N^{-\frac{2r\alpha}{4\alpha r+1}}\right)$. So the convergence rate we obtain in Corollary 9 is optimal.*

Combining bounds for $\left\|\overline{f}_{D,\lambda} - f_{D,\lambda}\right\|_\rho$ and $\|f_{D,\lambda} - f_\rho\|_\rho$, we shall prove in Section 6 the following learning rates in expectation for the distributed learning algorithm (2) for regression.

Corollary 11 *Assume $|y| \le M$ almost surely and regularity condition (9) for some $\frac{1}{2} < r \le 1$. If the capacity assumption $\mathcal{N}(\lambda) = O(\lambda^{-\frac{1}{2\alpha}})$ holds with some $\alpha > 0$, $|D_j| = \frac{N}{m}$ for $j = 1, \ldots, m$, and m satisfies the restriction*

$$m \le N^{\frac{2\alpha(2r-1)}{4\alpha r+1}}, \tag{13}$$

then by taking $\lambda = N^{-\frac{2\alpha}{4\alpha r+1}}$, we have

$$E\left[\left\|\overline{f}_{D,\lambda} - f_\rho\right\|_\rho\right] = O\left(N^{-\frac{2\alpha r}{4\alpha r+1}}\right).$$

Remark 12 *Corollary 11 shows that distributed learning with the least squares regularization scheme in a RKHS can achieve the optimal learning rates in expectation, provided that m satisfies the restriction (13). It should be pointed out that our error analysis is carried out under regularity condition (9) with $1/2 < r \le 1$ while the work in (Zhang et al., 2015) focused on the case with $r = 1/2$. When r approaches $1/2$, the number m of local processors under the restriction (13) reduces to 1, which corresponds to the non-distributed case. In our follow-up work, we will consider to relax the restriction (13) in a semi-supervised learning framework by using additional unlabelled data, as done in (Caponnetto and Yao, 2010). The main contribution of our analysis for distributed learning in this paper is to remove an eigenfunction assumption in (Zhang et al., 2015) by using a novel second order decomposition for a difference of operator inverses.*

Remark 13 *In Corollary 11 and Corollary 9, the choice of the regularization parameter λ is independent of the number m of local processors. This is consistent with the results in (Zhang et al., 2015). There have been several approaches for selecting the regularization parameter λ in regularization schemes in the literature including cross-validation (Györfy et al., 2002; Blanchard and Krämer, 2016) and the balancing principle (De Vito et al., 2010). For practical applications of distributed learning algorithms, how to choose λ and m (except the situations when the data are stored naturally in a distributive way) is crucial. Though we only consider the theoretical topic of error analysis in this paper, it would be interesting to develop parameter selection methods for distributed learning.*

3. Comparisons and Discussion

The least squares regularization scheme (1) is a classical algorithm for regression and has been extensively investigated in statistics and learning theory. There is a vast literature on

this topic. Here for a general kernel beyond those for the Sobolev spaces, we compare our results with the best learning rates in the existing literature.

Under the assumption that the orthogonal projection $f_{\mathcal{H}}$ of f_ρ in $L^2_{\rho_X}$ onto the closure of $\mathcal{H}_K$ satisfies regularity condition (9) for some $\frac{1}{2} \le r \le 1$, and that the eigenvalues $\{\lambda_i\}_i$ of L_K satisfy $\lambda_i \approx i^{-2\alpha}$ with some $\alpha > 1/2$, it was proved in (Caponnetto and De Vito, 2007) that

$$\lim_{\tau\to\infty} \limsup_{N\to\infty} \sup_{\rho\in\mathcal{P}(\alpha)} \text{Prob}\left[\|f_{D,\lambda_N} - f_{\mathcal{H}}\|_\rho^2 > \tau\lambda_N^{2r}\right] = 0,$$

where

$$\lambda_N = \begin{cases} N^{-\frac{2\alpha}{4\alpha r+1}}, & \text{if } \frac{1}{2} < r \le 1, \\ \left(\frac{\log N}{N}\right)^{\frac{2\alpha}{2\alpha+1}}, & \text{if } r = \frac{1}{2}, \end{cases}$$

and $\mathcal{P}(\alpha)$ denotes a set of probability measures ρ satisfying some moment decay condition (which is satisfied when $|y| \le M$). This learning rate in probability is stated with a limit as $\tau \to \infty$ and it has a logarithmic factor in the case $r = \frac{1}{2}$. In particular, to guarantee $\|f_{D,\lambda_N} - f_{\mathcal{H}}\|_\rho^2 \le \tau_\eta \lambda_N^{2r}$ with confidence $1-\eta$ by this result, one needs to restrict $N \ge N_\eta$ to be large enough and has the constant τ_η depending on η to be large enough. Using existing tools for error analysis including those in (Caponnetto and De Vito, 2007), we develop a novel second order decomposition technique for a difference of operator inverses in this paper, and succeed in deriving the optimal learning rate in expectation in Corollary 9 by removing the logarithmic factor in the case $r = \frac{1}{2}$.

Under the assumption that $|y| \le M$ almost surely, the eigenvalues $\{\lambda_i\}_i$ satisfying $\lambda_i \le ai^{-2\alpha}$ with some $\alpha > 1/2$ and $a > 0$, and for some constant $C > 0$, the pair (K, ρ_X) satisfying

$$\|f\|_\infty \le C\|f\|_K^{\frac{1}{2\alpha}}\|f\|_\rho^{1-\frac{1}{2\alpha}} \tag{14}$$

for every $f \in \mathcal{H}_K$, it was proved in (Steinwart et al., 2009) that for some constant $c_{\alpha,C}$ depending only on α and C, with confidence $1-\eta$, for any $0 < \lambda \le 1$,

$$\|\pi_M(f_{D,\lambda}) - f_\rho\|_\rho^2 \le 9\mathcal{A}_2(\lambda) + c_{\alpha,C}\frac{a^{1/(2\alpha)}M^2\log(3/\eta)}{\lambda^{1/(2\alpha)}N}.$$

Here π_M is the projection onto the interval $[-M, M]$ defined (Chen et al., 2004; Wu et al., 2006) by

$$\pi_M(f)(x) = \begin{cases} M, & \text{if } f(x) > M, \\ f(x), & \text{if } |f(x)| \le M, \\ -M, & \text{if } f(x) < -M, \end{cases}$$

and $\mathcal{A}_2(\lambda)$ is the approximation error defined by

$$\mathcal{A}_2(\lambda) = \inf_{f\in\mathcal{H}_K}\left\{\|f - f_\rho\|_\rho^2 + \lambda\|f\|_K^2\right\}.$$

When $f_\rho \in \mathcal{H}_K$, $\mathcal{A}_2(\lambda) = O(\lambda)$ and the choice $\lambda_N = N^{\frac{2\alpha}{2\alpha+1}}$ gives a learning rate of order $\|f_{D,\lambda_N} - f_\rho\|_\rho = O\left(N^{-\frac{\alpha}{2\alpha+1}}\right)$. Here one needs to impose the condition (14) for the functions spaces $L^2_{\rho_X}$ and $\mathcal{H}_K$, and to take the projection onto $[-M, M]$. Our learning rates

in Corollary 9 do not require such a condition for the function spaces, nor do we take the projection. Let us mention the fact from (Steinwart et al., 2009; Mendelson and Neeman, 2010) that condition (14) is satisfied when $\mathcal{H}_K$ is a Sobolev space on a domain of a Euclidean space and ρ_X is the uniform distribution, or when the eigenfunctions $\{\varphi_\ell/\sqrt{\lambda_\ell} : \lambda_\ell > 0\}$ of L_K normalized in $L^2_{\rho_X}$ are uniformly bounded. Recall that $\|\varphi_\ell\|_{L^2_{\rho_X}} = \sqrt{\lambda_\ell}$ since $\lambda_\ell = \lambda_\ell\|\varphi_\ell\|_K^2 = \langle\lambda_\ell\varphi_\ell, \varphi_\ell\rangle_K = \langle L_K\varphi_\ell, \varphi_\ell\rangle_K$ equals

$$\langle\int_{\mathcal{X}} K_x\varphi_\ell(x)d\rho_X(x), \varphi_\ell\rangle_K = \int_{\mathcal{X}} \varphi_\ell(x)\varphi_\ell(x)d\rho_X(x) = \|\varphi_\ell\|^2_{L^2_{\rho_X}}.$$

Learning rates for the least squares regularization scheme (1) in the $\mathcal{H}_K$-metric have been investigated in (Smale and Zhou, 2007).

For the distributed learning algorithm (2) with subsets $\{D_j\}_{j=1}^m$ of equal size, under the assumption that for some constants $2 < k \le \infty$ and $A < \infty$, the eigenfunctions $\{\varphi_i\}_i$ satisfy

$$\begin{cases} \sup_{\lambda_i>0} E\left[\left|\frac{\varphi_i(x)}{\sqrt{\lambda_i}}\right|^{2k}\right] \le A^{2k}, & \text{when } k < \infty, \\ \sup_{\lambda_i>0} \left\|\frac{\varphi_i(x)}{\sqrt{\lambda_i}}\right\|_\infty \le A, & \text{when } k = \infty, \end{cases} \tag{15}$$

and that $f_\rho \in \mathcal{H}_K$ and $\lambda_i \le ai^{-2\alpha}$ for some $\alpha > 1/2$ and $a > 0$, it was proved in (Zhang et al., 2015) that for $\lambda = N^{-\frac{2\alpha}{2\alpha+1}}$ and m satisfying the restriction

$$m \le c_\alpha \begin{cases} \left(\frac{N^{\frac{2(k-4)\alpha-k}{2\alpha+1}}}{A^{4k}\log^k N}\right)^{\frac{1}{k-2}}, & \text{when } k < \infty \\ \frac{N^{\frac{2\alpha-1}{2\alpha+1}}}{A^4\log N}, & \text{when } k = \infty \end{cases} \tag{16}$$

with a constant c_α depending only on α, there holds $E\left[\left\|\overline{f}_{D,\lambda} - f_\rho\right\|_\rho^2\right] = O\left(N^{-\frac{2\alpha}{2\alpha+1}}\right)$. This interesting result was achieved by a matrix analysis approach for which the eigenfunction assumption (15) played an essential role.

It might be challenging to check the eigenfunction assumption (15) involving the integral operator $L_K = L_{K,\rho_X}$ for the pair (K, ρ_X). To our best knowledge, besides the case of finite dimensional RKHSs, there exist in the literature only three classes of pairs (K, ρ_X) proved satisfying this eigenfunction assumption: the first class (Steinwart et al., 2009) is the Sobolev reproducing kernels on Euclidean domains together with the unform measures ρ_X, the second (Williamson et al., 2001) is periodical kernels, and the third class can be constructed by a Mercer type expansion

$$K(x, y) = \sum_i \lambda_i \frac{\varphi_i(x)}{\sqrt{\lambda_i}}\frac{\varphi_i(y)}{\sqrt{\lambda_i}}, \tag{17}$$

where $\{\frac{\varphi_i(x)}{\sqrt{\lambda_i}}\}_i$ is an orthonormal system in $L^2_{\rho_X}$. An example of a C^∞ Mercer kernel was presented in (Zhou, 2002, 2003) to show that only the smoothness of the Mercer kernel does not guarantee the uniform boundedness of the eigenfunctions $\{\frac{\varphi_i(x)}{\sqrt{\lambda_i}}\}_i$. Furthermore, it was shown in (Gittens and Mahoney, 2016) that these normalized eigenfunctions associated with radial basis kernels tend to be localized when the radial basis parameters are made smaller, which raises a practical concern for the uniform boundedness assumption on the eigenfunctions.

Remark 14 *To see how the eigenfunctions $\{\varphi_i(x)\}_i$ change with the marginal distribution, we consider a different measure μ induced by a nonnegative function $P \in L^2_{\rho_X}$ as $d\mu = P(x)d\rho_X$. An eigenpair (λ, φ) of the integral operator $L_{K,\mu}$ associated with the pair (K, μ) with $\lambda > 0$ satisfies*

$$L_{K,\mu}\varphi = \int_{\mathcal{X}} K(\cdot, x)\varphi(x)P(x)d\rho_X = \sum_{\lambda_i>0} \lambda_i \frac{\varphi_i}{\sqrt{\lambda_i}}\int_{\mathcal{X}} \frac{\varphi_i(x)}{\sqrt{\lambda_i}}\varphi(x)P(x)d\rho_X = \lambda\varphi.$$

We introduce an index set $I := \{i : \lambda_i > 0\}$, a possibly infinite matrix

$$\mathbb{K}^P = \left(\lambda_i \int_{\mathcal{X}} \frac{\varphi_i(x)}{\sqrt{\lambda_i}}P(x)\frac{\varphi_j(x)}{\sqrt{\lambda_j}}d\rho_X\right)_{i,j\in I}, \tag{18}$$

and express φ in terms of the orthonormal basis $\{\varphi_i\}_{i\in I}$ as $\varphi = \sum_{i\in I}\frac{c_i}{\sqrt{\lambda_i}}\varphi_i$ where the sequence $c = (c_i)_{i\in I}$ is given by $c_i = \langle\varphi, \frac{\varphi_i}{\sqrt{\lambda_i}}\rangle_{L^2_{\rho_X}} = \sqrt{\lambda_i}\langle\varphi, \varphi_i\rangle_K$. Then the eigenpair (λ, φ) satisfies

$$L_{K,\mu}\varphi = \lambda\varphi \quad\Longleftrightarrow\quad \sum_{i\in I}\frac{\varphi_i}{\sqrt{\lambda_i}}\sum_{j\in I}\left(\mathbb{K}^P\right)_{i,j}c_j = \lambda\sum_{i\in I}\frac{c_i}{\sqrt{\lambda_i}}\varphi_i \quad\Longleftrightarrow\quad \mathbb{K}^P c = \lambda c.$$

Thus the eigenpairs of the integral operator $L_{K,\mu}$ associated with (K, μ) can be characterized by those of the possibly infinite matrix $\mathbb{K}^P$ defined by (18). Finding the eigenpairs of $\mathbb{K}^P$ is an interesting question involving matrix analysis in linear algebra and multiplier operators in harmonic analysis. Note that when $P \equiv 1$ corresponding to $\mu = \rho_X$, $\mathbb{K}^P$ is diagonal with diagonal entries $\{\lambda_i\}_i$ and its eigenvectors yield eigenfunctions $\{\varphi_i\}_i$.

From the above observation, we can see that the marginal distribution ρ_X plays an essential role in the eigenfunction assumption (15) which might be difficult to check for a general marginal distribution ρ_X. For example, it is even unknown whether any of the Gaussian kernels on $\mathcal{X} = [0, 1]^d$ satisfies the eigenfunction assumption (15) when ρ_X is a general Borel measure.

Our learning rates stated in Corollary 4 or Corollary 11 do not require the eigenfunction assumption (15). Moreover, our restriction (10) for the number m of local processors in Corollary 4 is more general than (16) when α is close to $1/2$: with $r = 1/2$ corresponding to the condition $f_\rho \in \mathcal{H}_K$, our restriction (10) in Corollary 4 is $m \le N^{\frac{1}{4+6\alpha}}$ with the power index tending to $\frac{1}{7}$ while the restriction in (16) with $k = \infty$ takes the form $m \le c_\alpha\frac{N^{\frac{2\alpha-1}{2\alpha+1}}}{A^4\log N}$ with the power index tending to 0 as $\alpha \to \frac{1}{2}$. Note that the learning rates stated in Corollary 4 are for the difference $\overline{f}_{D,\lambda} - f_{D,\lambda}$ between the output function of the distributed learning algorithm (2) and that of the algorithm (1) using the whole data. In the special case of $r = \frac{1}{2}$, we can see that $E\left\|\overline{f}_{D,\lambda} - f_{D,\lambda}\right\|_\rho \le \widetilde{C}_{\kappa,c,r}N^{-\frac{\alpha}{2\alpha+1}}m^{-\frac{1}{4\alpha+2}}$, achieved by choosing $\lambda = \left(\frac{m}{N}\right)^{\frac{2\alpha}{2\alpha+1}}$, is smaller as m becomes larger. Here the dependence of λ on m is crucial for achieving this convergence rate of the sample error: if we fix λ and N, the error bounds for $E\left\|\overline{f}_{D,\lambda} - f_{D,\lambda}\right\|$ stated in Theorem 1 and Corollary 3 become larger as m increases. On the other hand, as one expects, increasing the number m of local processors would increase

the approximation error for the regression problem, which can also be seen from the bound with $\lambda = \left(\frac{m}{N}\right)^{\frac{2\alpha}{2\alpha+1}}$ stated in Theorem 7. The result in Corollary 11 with $r > \frac{1}{2}$ compensates and gives the optimal learning rate $E\left[\left\|\overline{f}_{D,\lambda} - f_\rho\right\|_\rho\right] = O\left(N^{-\frac{2r\alpha}{4\alpha r+1}}\right)$ by restricting m as in (13).

Besides the divide-and-conquer technique, there are many other approaches in the literature to reduce the computational complexity in handling big data. These include the localized learning (Meister and Steinwart, 2016), Nyström regularization (Bach, 2013; Rudi et al., 2015), and online learning (Dekel et al., 2012). The divide-and-conquer technique has the advantage of reducing the single-machine complexity of both space and time without a significant lost of prediction power.

It is an important and interesting problem how the big data are decomposed for distributed learning. Here we only study the approach of random decomposition, and the data subsets are regarded as being drawn from the same distribution. There should be better decomposition strategies for some practical applications. For example, intuitively data points should be divided according to their spatial distribution so that the learning process would yield locally optimal predictors, which could then be synthesized to the output function. See (Thomann et al., 2016).

In this paper, we consider the regularized least squares with Mercer kernels. Our results might be extended to more general kernels. A natural class consists of bounded and positive semidefinite kernels as studied in (Steinwart and Scovel, 2012). By means of the Mercer type expansion (17), one needs some assumptions on the system $\{\frac{\varphi_i(x)}{\sqrt{\lambda_i}}\}_i$, the domain $\mathcal{X}$, and the measure ρ_X to relax the continuity of the kernel while keeping compactness of the integral operator. How to minimize the assumptions and to maximize the scope of applications of the framework such as the situation of an input space $\mathcal{X}$ without a metric (Shen et al., 2014; De Vito et al., 2013) is a valuable question to investigate.

Here we only consider distributed learning with the regularized least squares. It would be of great interest and value to develop the theory for distributed learning with other algorithms such as spectral algorithms (Bauer et al., 2007), empirical feature-based learning (Guo and Zhou, 2012; Guo et al., 2017; Shi et al., 2011), the minimum error entropy principle (Hu et al., 2015; Fan et al., 2016), and randomized Kaczmarz algorithms (Lin and Zhou, 2015).

Remark 15 *After the submission of this paper, in our follow-up paper by Z. C. Guo, S. B. Lin, and D. X. Zhou entitled "Learning theory of distributed spectral algorithms" published in Inverse Problems, error analysis and optimal learning rates for distributed learning with spectral algorithms were derived. In late 2016, we learned that similar analysis was carried out for classical (non-distributed) spectral algorithms implemented in one machine by G. Blanchard and N. Mücke in a paper entitled "Optimal rates for regularization of statistical inverse learning problems" (arXiv:1604.04054, April 2016), and by L. H. Dicker, D. P. Foster, and D. Hsu in a paper entitled "Kernel ridge vs. principal component regression: minimax bounds and adaptibility of regularization operators" (arXiv:1605.08839, May 2016). We are indebted to one of the referees for pointing this out to us.*

4. Second Order Decomposition of Operator Differences and Norms

Our error estimates are achieved by a novel second order decomposition of operator differences in our integral operator approach. We approximate the integral operator L_K by the empirical integral operator $L_{K,D(x)}$ on $\mathcal{H}_K$ defined with the input data set $D(x) = \{x_i\}_{i=1}^N = \{x : (x,y) \in D \text{ for some } y \in \mathcal{Y}\}$ as

$$L_{K,D(x)}(f) = \frac{1}{|D|}\sum_{x\in D(x)} f(x)K_x = \frac{1}{|D|}\sum_{x\in D(x)} \langle f, K_x\rangle_K K_x, \qquad f \in \mathcal{H}_K, \tag{19}$$

where the reproducing property $f(x) = \langle f, K_x\rangle_K$ for $f \in \mathcal{H}_K$ is used. Since K is a Mercer kernel, $L_{K,D_j(x)}$ is a finite-rank positive operator and $L_{K,D_j(x)} + \lambda I$ is invertible.

The operator difference in our study is that of the inverses of two operators defined by

$$Q_{D(x)} = \left(L_{K,D(x)} + \lambda I\right)^{-1} - \left(L_K + \lambda I\right)^{-1}. \tag{20}$$

If we denote $A = L_{K,D(x)} + \lambda I$ and $B = L_K + \lambda I$, then our second order decomposition for the operator difference $Q_{D(x)}$ is a special case of the following second order decomposition for the difference $A^{-1} - B^{-1}$.

Lemma 16 *Let A and B be invertible operators on a Banach space. Then we have*

$$A^{-1} - B^{-1} = B^{-1}\{B - A\}B^{-1} + B^{-1}\{B - A\}A^{-1}\{B - A\}B^{-1}. \tag{21}$$

Proof We can decompose the operator $A^{-1} - B^{-1}$ as

$$A^{-1} - B^{-1} = B^{-1}\{B - A\}A^{-1}. \tag{22}$$

This is the first order decomposition.

Write the last term A^{-1} as $B^{-1} + (A^{-1} - B^{-1})$ and apply another first order decomposition similar to (22) as

$$A^{-1} - B^{-1} = A^{-1}\{B - A\}B^{-1}.$$

It follows from (22) that

$$A^{-1} - B^{-1} = B^{-1}\{B - A\}\left\{B^{-1} + A^{-1}\{B - A\}B^{-1}\right\}.$$

Then the desired identity (21) is verified. The lemma is proved. ■

To demonstrate how the second order decomposition leads to tight error bounds for the classical least squares regularization scheme (1) with the output function $f_{D,\lambda}$, we recall the following formula (see e.g. (Caponnetto and De Vito, 2007; Smale and Zhou, 2007))

$$f_{D,\lambda} - f_\lambda = \left(L_{K,D(x)} + \lambda I\right)^{-1}\Delta_D, \quad \Delta_D := \frac{1}{|D|}\sum_{z\in D}\xi_\lambda(z) - E[\xi_\lambda], \tag{23}$$

where Δ_D is induced by the random variables ξ_λ with values in the Hilbert space $\mathcal{H}_K$ defined as

$$\xi_\lambda(z) = (y - f_\lambda(x))K_x, \qquad z = (x,y) \in \mathcal{Z}. \tag{24}$$

Then we can express $f_{D,\lambda} - f_\lambda$ by means of the notation $Q_{D(x)} = \left(L_{K,D(x)} + \lambda I\right)^{-1} - (L_K + \lambda I)^{-1}$ as

$$f_{D,\lambda} - f_\lambda = \left[Q_{D(x)}\right] \Delta_D + (L_K + \lambda I)^{-1} \Delta_D.$$

Due to the identity between the $L^2_{\rho_X}$ norm and the $\mathcal{H}_K$ metric

$$\|g\|_\rho = \|L_K^{\frac{1}{2}} g\|_K, \qquad \forall g \in L^2_{\rho_X}, \tag{25}$$

we can estimate the error $\|f_{D,\lambda} - f_\lambda\|_\rho = \|L_K^{\frac{1}{2}} (f_{D,\lambda} - f_\lambda)\|_K$ by bounding the $\mathcal{H}_K$ norm of the following expression obtained from the second order decomposition (21) with $B - A = L_K - L_{K,D(x)}$

$$\begin{aligned} L_K^{\frac{1}{2}} \left[Q_{D(x)}\right] \Delta_D &= \left\{L_K^{\frac{1}{2}} (L_K + \lambda I)^{-\frac{1}{2}}\right\} \left\{(L_K + \lambda I)^{-\frac{1}{2}} \left\{L_K - L_{K,D(x)}\right\}\right\} (L_K + \lambda I)^{-1} \Delta_D \\ &+ \left\{L_K^{\frac{1}{2}} (L_K + \lambda I)^{-\frac{1}{2}}\right\} \left\{(L_K + \lambda I)^{-\frac{1}{2}} \left\{L_K - L_{K,D(x)}\right\}\right\} \left(L_{K,D(x)} + \lambda I\right)^{-1} \\ &\quad \left\{\left\{L_K - L_{K,D(x)}\right\} (L_K + \lambda I)^{-\frac{1}{2}}\right\} \left\{(L_K + \lambda I)^{-\frac{1}{2}} \Delta_D\right\}. \end{aligned}$$

Combining this expression with the operator norm bound $\left\|L_K^{\frac{1}{2}} (L_K + \lambda I)^{-\frac{1}{2}}\right\| \leq 1$ and the notation

$$\Xi_D = \left\|(L_K + \lambda I)^{-\frac{1}{2}} \left\{L_K - L_{K,D(x)}\right\}\right\| \tag{26}$$

for convenience, we find

$$\begin{aligned} \left\|L_K^{\frac{1}{2}} \left[Q_{D(x)}\right] \Delta_D\right\|_K &\leq \Xi_D \left\|(L_K + \lambda I)^{-1} \Delta_D\right\|_K + \Xi_D \left\|\left(L_{K,D(x)} + \lambda I\right)^{-1}\right\| \\ &\quad \Xi_D \left\|(L_K + \lambda I)^{-\frac{1}{2}} \Delta_D\right\|_K. \end{aligned}$$

By decomposing $(L_K + \lambda I)^{-1} \Delta_D$ as $(L_K + \lambda I)^{-\frac{1}{2}} (L_K + \lambda I)^{-\frac{1}{2}} \Delta_D$ and using the bounds $\left\|\left(L_{K,D(x)} + \lambda I\right)^{-1}\right\| \leq \frac{1}{\lambda}$, $\left\|(L_K + \lambda I)^{-\frac{1}{2}}\right\| \leq 1/\sqrt{\lambda}$, we know that

$$\left\|L_K^{\frac{1}{2}} \left[Q_{D(x)}\right] \Delta_D\right\|_K \leq \left(\frac{\Xi_D}{\sqrt{\lambda}} + \frac{\Xi_D^2}{\lambda}\right) \left\|(L_K + \lambda I)^{-\frac{1}{2}} \Delta_D\right\|_K \tag{27}$$

and

$$\|f_{D,\lambda} - f_\lambda\|_\rho \leq \left(\frac{\Xi_D}{\sqrt{\lambda}} + \frac{\Xi_D^2}{\lambda} + 1\right) \left\|(L_K + \lambda I)^{-\frac{1}{2}} \Delta_D\right\|_K. \tag{28}$$

Thus the classical least squares regularization scheme (1) can be analyzed after estimating the operator norm $\Xi_D = \left\|(L_K + \lambda I)^{-\frac{1}{2}} \left\{L_K - L_{K,D(x)}\right\}\right\|$ and the function norm $\left\|(L_K + \lambda I)^{-\frac{1}{2}} \Delta_D\right\|_K$. In the following two lemmas, to be proved in the appendix, these norms are estimated in terms of effective dimensions by methods in the existing literature (Caponnetto and De Vito, 2007; Bauer et al., 2007; Blanchard and Krämer, 2010).

Lemma 17 *Let D be a sample drawn independently according to ρ. Then the following estimates for the operator norm $\left\|(L_K + \lambda I)^{-\frac{1}{2}} \left\{L_K - L_{K,D(x)}\right\}\right\|$ hold.*

(a) $E\left[\left\|(L_K + \lambda I)^{-\frac{1}{2}} \left\{L_K - L_{K,D(x)}\right\}\right\|^2\right] \leq \frac{\kappa^2 \mathcal{N}(\lambda)}{|D|}$.

(b) For any $0 < \delta < 1$, with confidence at least $1 - \delta$, there holds

$$\left\|(L_K + \lambda I)^{-\frac{1}{2}} \left\{L_K - L_{K,D(x)}\right\}\right\| \leq \mathcal{B}_{|D|,\lambda} \log(2/\delta), \tag{29}$$

where we denote the quantity

$$\mathcal{B}_{|D|,\lambda} = \frac{2\kappa}{\sqrt{|D|}} \left\{\frac{\kappa}{\sqrt{|D|\lambda}} + \sqrt{\mathcal{N}(\lambda)}\right\}. \tag{30}$$

(c) For any $d > 1$, there holds

$$\left\{E\left[\left\|(L_K + \lambda I)^{-\frac{1}{2}} \left\{L_K - L_{K,D(x)}\right\}\right\|^d\right]\right\}^{\frac{1}{d}} \leq (2d\Gamma(d) + 1)^{\frac{1}{d}} \mathcal{B}_{|D|,\lambda},$$

where Γ is the Gamma function defined for $d > 0$ by $\Gamma(d) = \int_0^\infty u^{d-1} \exp\{-u\}\, du$.

Lemma 18 *Let D be a sample drawn independently according to ρ and g be a measurable bounded function on $\mathcal{Z}$ and ξ_g be the random variable with values in $\mathcal{H}_K$ defined by $\xi_g(z) = g(z) K_x$ for $z = (x, y) \in \mathcal{Z}$. Then the following statements hold.*

(a) $E\left[\left\|(L_K + \lambda I)^{-1/2} (K_x)\right\|_K^2\right] = \mathcal{N}(\lambda)$.

(b) For almost every $x \in \mathcal{X}$, there holds $\left\|(L_K + \lambda I)^{-1/2} (K_x)\right\|_K \leq \frac{\kappa}{\sqrt{\lambda}}$.

(c) For any $0 < \delta < 1$, with confidence at least $1 - \delta$, there holds

$$\left\|(L_K + \lambda I)^{-1/2} \left(\frac{1}{|D|} \sum_{z \in D} \xi_g(z) - E\left[\xi_g\right]\right)\right\|_K \leq \frac{2\|g\|_\infty \log(2/\delta)}{\sqrt{|D|}} \left\{\frac{\kappa}{\sqrt{|D|\lambda}} + \sqrt{\mathcal{N}(\lambda)}\right\}.$$

Remark 19 *In the existing literature, the first order decomposition (22) was used. To bound the norm of $\left(L_{K,D(x)} + \lambda I\right)^{-1} \Delta_D$ by this approach, one needs to either use the brute force estimate $\left\|\left(L_{K,D(x)} + \lambda I\right)^{-1}\right\| \leq \frac{1}{\lambda}$ leading to suboptimal learning rates, or applying the approximation of L_K by $L_{K,D(x)}$ which is valid only with confidence and leads to confidence-based estimates with λ depending on the confidence level. In our second order decomposition, we decompose the inverse operator $\left(L_{K,D(x)} + \lambda I\right)^{-1}$ further after the first order decomposition (22). This leads to finer estimates with an additional factor $\|(B - A) B^{-\frac{1}{2}}\|$ in the second term of the bound (21) and gives the refined error bound (28).*

5. Deriving Error Bounds for Least Squares Regularization Scheme

In this section we prove our main result on error bounds for the least squares regularization scheme (1), which illustrates how to apply the second order decomposition (21) for operator differences in our integral operator approach.

Proposition 20 *Assume $E[y^2] < \infty$ and moment condition (11) for some $1 \le p \le \infty$. Then*

$$E\left[\|f_{D,\lambda} - f_\lambda\|_\rho\right] \le \left(2 + 56\kappa^4 + 57\kappa^2\right)\left(1 + \frac{1}{(|D|\lambda)^2} + \frac{\mathcal{N}(\lambda)}{|D|\lambda}\right) \left\{\kappa^{\frac{1}{p}}\sqrt{\left\|\sigma_\rho^2\right\|_p}\left(\frac{\mathcal{N}(\lambda)}{|D|}\right)^{\frac{1}{2}(1-\frac{1}{p})}\left(\frac{1}{|D|\lambda}\right)^{\frac{1}{2p}} + \kappa\frac{\|f_\lambda - f_\rho\|_\rho}{\sqrt{|D|\lambda}}\right\}.$$

Proof We apply the error bound (28) and the Schwarz inequality to get

$$E\left[\|f_{D,\lambda} - f_\lambda\|_\rho\right] \le \left\{E\left[\left(1 + \frac{\Xi_D}{\sqrt{\lambda}} + \frac{\Xi_D^2}{\lambda}\right)^2\right]\right\}^{1/2}\left\{E\left[\left\|(L_K + \lambda I)^{-1/2}\Delta_D\right\|_K^2\right]\right\}^{1/2}. \tag{31}$$

The first factor in the above bound involves $\Xi_D = \left\|(L_K + \lambda I)^{-\frac{1}{2}}\left\{L_K - L_{K,D(x)}\right\}\right\|$ and it can be estimated by applying Lemma 17 as

$$\begin{aligned}\left\{E\left[\left(1 + \frac{\Xi_D}{\sqrt{\lambda}} + \frac{\Xi_D^2}{\lambda}\right)^2\right]\right\}^{1/2} &\le 1 + \left\{E\left[\frac{\Xi_D^2}{\lambda}\right]\right\}^{1/2} + \left\{E\left[\frac{\Xi_D^4}{\lambda^2}\right]\right\}^{1/2} \\ &\le 1 + \left\{\frac{\kappa^2\mathcal{N}(\lambda)}{|D|\lambda}\right\}^{1/2} + \left\{\frac{49\mathcal{B}^4_{|D|,\lambda}}{\lambda^2}\right\}^{1/2} \\ &\le 2 + \frac{56\kappa^4}{(|D|\lambda)^2} + \frac{57\kappa^2\mathcal{N}(\lambda)}{|D|\lambda}. \end{aligned} \tag{32}$$

To deal with the factor in the bound (31), we separate $\Delta_D = \frac{1}{|D|}\sum_{z\in D}\xi_\lambda(z) - E[\xi_\lambda]$ as

$$\Delta_D = \Delta_D' + \Delta_D'',$$

where

$$\Delta_D' := \frac{1}{|D|}\sum_{z\in D}\xi_0(z), \quad \Delta_D'' := \frac{1}{|D|}\sum_{z\in D}\left(\xi_\lambda - \xi_0\right)(z) - E[\xi_\lambda]$$

are induced by another random variable ξ_0 with values in the Hilbert space $\mathcal{H}_K$ defined by

$$\xi_0(z) = (y - f_\rho(x))K_x, \qquad z = (x, y) \in \mathcal{Z}. \tag{33}$$

Then the second factor in (31) can be separated as

$$\begin{aligned}&\left\{E\left[\left\|(L_K + \lambda I)^{-1/2}\Delta_D\right\|_K^2\right]\right\}^{1/2} \\ &\le \left\{E\left[\left\|(L_K + \lambda I)^{-1/2}\Delta_D'\right\|_K^2\right]\right\}^{1/2} + \left\{E\left[\left\|(L_K + \lambda I)^{-1/2}\Delta_D''\right\|_K^2\right]\right\}^{1/2}. \end{aligned} \tag{34}$$

Let us bound the first term of (34). Observe that

$$(L_K + \lambda I)^{-1/2}\Delta_D' = \sum_{z\in D}\frac{1}{|D|}(y - f_\rho(x))\left(L_K + \lambda I\right)^{-1/2}(K_x).$$

Each term in this expression is unbiased because $\int_{\mathcal{Y}}(y - f_\rho(x))\,d\rho(y|x) = 0$. This unbiasedness and the independence tell us that

$$\begin{aligned}\left\{E\left[\left\|(L_K + \lambda I)^{-1/2}\Delta_D'\right\|_K^2\right]\right\}^{1/2} &= \left\{\frac{1}{|D|}E\left[\left\|(y - f_\rho(x))\left[(L_K + \lambda I)^{-1/2}\right](K_x)\right\|_K^2\right]\right\}^{1/2} \\ &= \left\{\frac{1}{|D|}E\left[\sigma_\rho^2(x)\left\|\left[(L_K + \lambda I)^{-1/2}\right](K_x)\right\|_K^2\right]\right\}^{1/2}. \end{aligned} \tag{35}$$

If $\sigma_\rho^2 \in L^\infty$, then $\sigma_\rho^2(x) \le \left\|\sigma_\rho^2\right\|_\infty$ and by Lemma 18 we have

$$\left\{E\left[\left\|(L_K + \lambda I)^{-1/2}\Delta_D'\right\|_K^2\right]\right\}^{1/2} \le \sqrt{\left\|\sigma_\rho^2\right\|_\infty}\sqrt{\mathcal{N}(\lambda)/|D|}.$$

If $\sigma_\rho^2 \in L^p_{\rho_X}$ with $1 \le p < \infty$, we take $q = \frac{p}{p-1}$ ($q = \infty$ for $p = 1$) satisfying $\frac{1}{p} + \frac{1}{q} = 1$ and apply the Hölder inequality $E[|\xi\eta|] \le (E[|\xi|^p])^{1/p}(E[|\eta|^q])^{1/q}$ to $\xi = \sigma_\rho^2$, $\eta = \left\|(L_K + \lambda I)^{-1/2}(K_x)\right\|_K^2$ to find

$$E\left[\sigma_\rho^2(x)\left\|\left[(L_K + \lambda I)^{-1/2}\right](K_x)\right\|_K^2\right] \le \left\|\sigma_\rho^2\right\|_p\left\{E\left[\left\|\left[(L_K + \lambda I)^{-1/2}\right](K_x)\right\|_K^{2q}\right]\right\}^{1/q}.$$

But

$$\left\|\left[(L_K + \lambda I)^{-1/2}\right](K_x)\right\|_K^{2q-2} \le \left(\kappa/\sqrt{\lambda}\right)^{2q-2}$$

and $E\left[\left\|(L_K + \lambda I)^{-1/2}(K_x)\right\|_K^2\right] = \mathcal{N}(\lambda)$ by Lemma 18. So we have

$$\begin{aligned}\left\{E\left[\left\|(L_K + \lambda I)^{-1/2}\Delta_D'\right\|_K^2\right]\right\}^{1/2} &\le \left\{\frac{1}{|D|}\left\|\sigma_\rho^2\right\|_p\left\{\frac{\kappa^{2q-2}}{\lambda^{q-1}}\mathcal{N}(\lambda)\right\}^{1/q}\right\}^{1/2} \\ &= \sqrt{\left\|\sigma_\rho^2\right\|_p}\kappa^{\frac{1}{p}}\left(\frac{\mathcal{N}(\lambda)}{|D|}\right)^{\frac{1}{2}(1-\frac{1}{p})}\left(\frac{1}{|D|\lambda}\right)^{\frac{1}{2p}}. \end{aligned}$$

Combining the above two cases, we know that for either $p = \infty$ or $p < \infty$, the first term of (34) can be bounded as

$$\left\{E\left[\left\|(L_K + \lambda I)^{-1/2}\Delta_D'\right\|_K^2\right]\right\}^{1/2} \le \sqrt{\left\|\sigma_\rho^2\right\|_p}\kappa^{\frac{1}{p}}\left(\frac{\mathcal{N}(\lambda)}{|D|}\right)^{\frac{1}{2}(1-\frac{1}{p})}\left(\frac{1}{|D|\lambda}\right)^{\frac{1}{2p}}.$$

The second term of (34) can be bounded easily as

$$\left\{E\left[\left\|(L_K+\lambda I)^{-1/2}\Delta''_D\right\|_K^2\right]\right\}^{1/2} \leq \frac{1}{\sqrt{|D|}}\left\{E\left[(f_\rho(x)-f_\lambda(x))^2\left\|(L_K+\lambda I)^{-1/2}(K_x)\right\|_K^2\right]\right\}^{1/2}$$
$$\leq \frac{1}{\sqrt{|D|}}\left\{E\left[(f_\rho(x)-f_\lambda(x))^2\frac{\kappa^2}{\lambda}\right]\right\}^{1/2}=\frac{\kappa\|f_\rho-f_\lambda\|_\rho}{\sqrt{|D|\lambda}}.$$

Putting the above estimates for the two terms of (34) into (31) and applying the bound (32) involving Ξ_D, we know that $E\left[\|f_{D,\lambda}-f_\lambda\|_\rho\right]$ is bounded by

$$\left(2+\frac{56\kappa^4}{(|D|\lambda)^2}+\frac{57\kappa^2\mathcal{N}(\lambda)}{|D|\lambda}\right)\left(\sqrt{\left\|\sigma_\rho^2\right\|_p}\kappa^{\frac{1}{p}}\left(\frac{\mathcal{N}(\lambda)}{|D|}\right)^{\frac{1}{2}(1-\frac{1}{p})}\left(\frac{1}{|D|\lambda}\right)^{\frac{1}{2p}}+\frac{\kappa\|f_\rho-f_\lambda\|_\rho}{\sqrt{|D|\lambda}}\right).$$

Then our desired error bound follows. The proof of the proposition is complete. ■

Proof of Theorem 7 Combining Proposition 20 with the triangle inequality $\|f_{D,\lambda}-f_\rho\|_\rho \leq \|f_{D,\lambda}-f_\lambda\|_\rho+\|f_\lambda-f_\rho\|_\rho$, we know that

$$E\left[\|f_{D,\lambda}-f_\rho\|_\rho\right] \leq \|f_\lambda-f_\rho\|_\rho+\left(2+56\kappa^4+57\kappa^2\right)\left(1+\frac{1}{(|D|\lambda)^2}+\frac{\mathcal{N}(\lambda)}{|D|\lambda}\right)$$
$$\left\{\kappa^{\frac{1}{p}}\sqrt{\left\|\sigma_\rho^2\right\|_p}\left(\frac{\mathcal{N}(\lambda)}{|D|}\right)^{\frac{1}{2}(1-\frac{1}{p})}\left(\frac{1}{|D|\lambda}\right)^{\frac{1}{2p}}+\frac{\kappa}{\sqrt{|D|\lambda}}\|f_\lambda-f_\rho\|_\rho\right\}.$$

Then the desired error bound holds true, and the proof of Theorem 7 is complete. ■

Proof of Corollary 8 By the definition of effective dimension,

$$\mathcal{N}(\lambda)=\sum_\ell\frac{\lambda_\ell}{\lambda_\ell+\lambda}\geq\frac{\lambda_1}{\lambda_1+\lambda}.$$

Combining this with the restriction (8) with $m=1$, we find $\mathcal{N}(\lambda)\geq\frac{\lambda_1}{\lambda_1+C_0}$ and $N\lambda\geq\frac{\lambda_1}{(\lambda_1+C_0)C_0}$. Putting these and the restriction (8) with $m=1$ into the error bound (12), we know that

$$E\left[\|f_{D,\lambda}-f_\rho\|_\rho\right]\leq\left(2+56\kappa^4+57\kappa^2\right)(1+\kappa)\left(1+\frac{(\lambda_1+C_0)^2C_0^2}{\lambda_1^2}+C_0\right)$$
$$\left\{\left(1+\sqrt{(\lambda_1+C_0)C_0/\lambda_1}\right)\|f_\lambda-f_\rho\|_\rho+\sqrt{\left\|\sigma_\rho^2\right\|_p}\left(\frac{\mathcal{N}(\lambda)}{N}\right)^{\frac{1}{2}(1-\frac{1}{p})}\left(\frac{1}{N\lambda}\right)^{\frac{1}{2p}}\right\}.$$

Then the desired bound follows. The proof of Corollary 8 is complete. ■

To prove Corollary 9, we need the following bounds (Smale and Zhou, 2007) for $\|f_\lambda-f_\rho\|_\rho$ and $\|f_\lambda-f_\rho\|_K$.

Lemma 21 *Assume regularity condition (9) with $0<r\leq 1$. There holds*

$$\|f_\lambda-f_\rho\|_\rho\leq\lambda^r\|g_\rho\|_\rho. \tag{36}$$

Furthermore, if $1/2\leq r\leq 1$, then we have

$$\|f_\lambda-f_\rho\|_K\leq\lambda^{r-1/2}\|g_\rho\|_\rho. \tag{37}$$

Proof of Corollary 9 By Lemma 21, regularity condition (9) with $0<r\leq 1$ implies

$$\|f_\lambda-f_\rho\|_\rho\leq\lambda^r\|g_\rho\|_\rho.$$

If

$$\mathcal{N}(\lambda)\leq C_0\lambda^{-\frac{1}{2\alpha}},\qquad\forall\lambda>0$$

for some constant $C_0\geq 1$, then the choice $\lambda=N^{-\frac{2\alpha}{2\alpha\max\{2r,1\}+1}}$ yields

$$\frac{\mathcal{N}(\lambda)}{N\lambda}\leq\frac{C_0\lambda^{-\frac{1}{2\alpha}-1}}{N}=C_0N^{\frac{2\alpha+1}{2\alpha\max\{2r,1\}+1}-1}\leq C_0.$$

So (8) with $m=1$ is satisfied. With this choice we also have

$$\left(\frac{\mathcal{N}(\lambda)}{N}\right)^{\frac{1}{2}(1-\frac{1}{p})}\left(\frac{1}{N\lambda}\right)^{\frac{1}{2p}} \leq C_0^{\frac{1}{2}(1-\frac{1}{p})}N^{-\frac{2\alpha\max\{2r,1\}}{2\alpha\max\{2r,1\}+1}\frac{1}{2}(1-\frac{1}{p})}N^{-\frac{2\alpha\max\{2r,1\}+1-2\alpha}{2\alpha\max\{2r,1\}+1}\frac{1}{2p}}$$
$$=C_0^{\frac{1}{2}(1-\frac{1}{p})}N^{-\frac{\alpha\max\{2r,1\}}{2\alpha\max\{2r,1\}+1}+\frac{1}{2p}\frac{2\alpha-1}{2\alpha\max\{2r,1\}+1}}.$$

Putting these estimates into Corollary 8, we know that

$$E\left[\|f_{D,\lambda}-f_\rho\|_\rho\right] = O\left(N^{-\frac{2\alpha r}{2\alpha\max\{2r,1\}+1}}+N^{-\frac{\alpha\max\{2r,1\}}{2\alpha\max\{2r,1\}+1}+\frac{1}{2p}\frac{2\alpha-1}{2\alpha\max\{2r,1\}+1}}\right)$$
$$=O\left(N^{-\frac{\alpha\min\{2r,\max\{2r,1\}\}}{2\alpha\max\{2r,1\}+1}+\frac{1}{2p}\frac{2\alpha-1}{2\alpha\max\{2r,1\}+1}}\right).$$

But we find

$$\min\{2r,\max\{2r,1\}\}=2r$$

by discussing the two different cases $0<r<\frac{1}{2}$ and $\frac{1}{2}\leq r\leq 1$. Then our conclusion follows immediately. The proof of Corollary 9 is complete. ■

6. Proof of Error Bounds for the Distributed Learning Algorithm

To analyze the error $\overline{f}_{D,\lambda}-f_{D,\lambda}$ for distributed learning, we recall the notation $Q_{D(x)}=\left(L_{K,D(x)}+\lambda I\right)^{-1}-(L_K+\lambda I)^{-1}$ for the difference of inverse operators and use the notation $Q_{D_j(x)}$ involving the data subset D_j. The empirical integral operator $L_{K,D_j(x)}$ is defined with D replaced by the data subset D_j. For our error analysis for the distributed learning algorithm (2), we need the following error decomposition for $\overline{f}_{D,\lambda}-f_{D,\lambda}$.

Lemma 22 *Assume $E[y^2] < \infty$. For $\lambda > 0$, we have*

$$\begin{aligned}\overline{f}_{D,\lambda} - f_{D,\lambda} &= \sum_{j=1}^{m} \frac{|D_j|}{|D|} \left[\left(L_{K,D_j(x)} + \lambda I \right)^{-1} - \left(L_{K,D(x)} + \lambda I \right)^{-1} \right] \Delta_j \\ &= \sum_{j=1}^{m} \frac{|D_j|}{|D|} \left[Q_{D_j(x)} \right] \Delta_j' + \sum_{j=1}^{m} \frac{|D_j|}{|D|} \left[Q_{D_j(x)} \right] \Delta_j'' - \left[Q_{D(x)} \right] \Delta_D, \quad (38)\end{aligned}$$

where

$$\Delta_j := \frac{1}{|D_j|} \sum_{z \in D_j} \xi_\lambda(z) - E[\xi_\lambda], \quad \Delta_D := \frac{1}{|D|} \sum_{z \in D} \xi_\lambda(z) - E[\xi_\lambda],$$

and

$$\Delta_j' := \frac{1}{|D_j|} \sum_{z \in D_j} \xi_0(z), \quad \Delta_j'' := \frac{1}{|D_j|} \sum_{z \in D_j} \left(\xi_\lambda - \xi_0 \right)(z) - E[\xi_\lambda].$$

Proof Recall the expression (23) for $f_{D,\lambda} - f_\lambda$. When the data subset D_j is used, we have

$$f_{D_j,\lambda} - f_\lambda = \left(L_{K,D_j(x)} + \lambda I \right)^{-1} \Delta_j.$$

So we know that

$$\overline{f}_{D,\lambda} - f_\lambda = \sum_{j=1}^{m} \frac{|D_j|}{|D|} \left\{ f_{D_j,\lambda} - f_\lambda \right\} = \sum_{j=1}^{m} \frac{|D_j|}{|D|} \left(L_{K,D_j(x)} + \lambda I \right)^{-1} \Delta_j.$$

We can decompose Δ_D as

$$\Delta_D = \frac{1}{|D|} \sum_{z \in D} \xi_\lambda(z) - E[\xi_\lambda] = \sum_{j=1}^{m} \frac{|D_j|}{|D|} \left\{ \frac{1}{|D_j|} \sum_{z \in D_j} \xi_\lambda(z) - E[\xi_\lambda] \right\} = \sum_{j=1}^{m} \frac{|D_j|}{|D|} \Delta_j$$

in the expression (23) for $f_{D,\lambda} - f_\lambda$ and find

$$f_{D,\lambda} - f_\lambda = \sum_{j=1}^{m} \frac{|D_j|}{|D|} \left(L_{K,D(x)} + \lambda I \right)^{-1} \Delta_j.$$

Then the first desired expression for $\overline{f}_{D,\lambda} - f_{D,\lambda}$ follows.

By adding and subtracting the operator $(L_K + \lambda I)^{-1}$, writing $\Delta_j = \Delta_j' + \Delta_j''$, and noting $E[\xi_0] = 0$, we know that the first expression implies (38). This proves Lemma 22. ∎

Before proving our first main result on the error $\overline{f}_{D,\lambda} - f_{D,\lambda}$ in the $\mathcal{H}_K$ metric and L_ρ^2 metric, we state the following more general result which allows different sizes for data subsets $\{D_j\}$.

Theorem 23 *Assume that for some constant $M > 0$, $|y| \le M$ almost surely. Then we have*

$$E\left[\left\| \overline{f}_{D,\lambda} - f_{D,\lambda} \right\|_\rho \right] \le C_\kappa' M \sqrt{\lambda} \left\{ \sum_{j=1}^{m} \left(\frac{|D_j|}{|D|} \right)^2 \left(\mathcal{S}_{|D_j|,\lambda}^2 + \mathcal{S}_{|D_j|,\lambda}^3 \right) \right\}^{\frac{1}{2}} + C_\kappa' \sum_{j=1}^{m} \frac{|D_j|}{|D|}$$
$$\frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{|D_j|\lambda}} \left(\sqrt{\frac{\mathcal{N}(\lambda)}{|D_j|\lambda}} + \mathcal{S}_{|D_j|,\lambda} \right) + C_\kappa' \left(\sqrt{\frac{\mathcal{N}(\lambda)}{|D|\lambda}} + \mathcal{S}_{|D|,\lambda} \right) \left(M \sqrt{\frac{\mathcal{N}(\lambda)}{|D|}} + \frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{|D|\lambda}} \right)$$

and

$$E\left[\left\| \overline{f}_{D,\lambda} - f_{D,\lambda} \right\|_K \right] \le C_\kappa' M \left\{ \sum_{j=1}^{m} \left(\frac{|D_j|}{|D|} \right)^2 \left(\mathcal{S}_{|D_j|,\lambda}^2 + \mathcal{S}_{|D_j|,\lambda}^3 \right) \right\}^{\frac{1}{2}} + C_\kappa' \sum_{j=1}^{m} \frac{|D_j|}{|D|}$$
$$\frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{|D_j|\lambda}} \left(\sqrt{\frac{\mathcal{N}(\lambda)}{|D_j|\lambda}} + \mathcal{S}_{|D_j|,\lambda} \right) + C_\kappa' \left(\sqrt{\frac{\mathcal{N}(\lambda)}{|D|\lambda}} + \mathcal{S}_{|D|,\lambda} \right) \left(M \sqrt{\frac{\mathcal{N}(\lambda)}{|D|\lambda}} + \frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{|D|\lambda}} \right),$$

where C_κ' is a constant depending only on κ, and $\mathcal{S}_{|D_j|,\lambda}$ is the quantity given by

$$\mathcal{S}_{|D_j|,\lambda} = \frac{1}{|D_j|^2 \lambda^2} + \frac{\mathcal{N}(\lambda)}{|D_j|\lambda}.$$

Proof Recall the expression (38) for $\overline{f}_{D,\lambda} - f_{D,\lambda}$ in Lemma 22. It enables us to express

$$L_K^{1/2} \left\{ \overline{f}_{D,\lambda} - f_{D,\lambda} \right\} = J_1 + J_2 + J_3, \quad (39)$$

where the terms J_1, J_2, J_3 are given by

$$J_1 = \sum_{j=1}^{m} \frac{|D_j|}{|D|} \left[L_K^{1/2} Q_{D_j(x)} \right] \Delta_j', \; J_2 = \sum_{j=1}^{m} \frac{|D_j|}{|D|} \left[L_K^{1/2} Q_{D_j(x)} \right] \Delta_j'', \; J_3 = - \left[L_K^{1/2} Q_{D(x)} \right] \Delta_D.$$

These three terms will be dealt with separately in the following.

For the first term J_1 of (39), each summand with $j \in \{1, \ldots, m\}$ can be expressed as $\sum_{z \in D_j} \frac{1}{|D|} (y - f_\rho(x)) \left[L_K^{1/2} Q_{D_j(x)} \right] (K_x)$, and is unbiased because $\int_{\mathcal{Y}} y - f_\rho(x) d\rho(y|x) = 0$. The unbiasedness and the independence tell us that

$$E\left[\|J_1\|_K \right] \le \left\{ E\left[\|J_1\|_K^2 \right] \right\}^{1/2} \le \left\{ \sum_{j=1}^{m} \left(\frac{|D_j|}{|D|} \right)^2 E\left[\left\| \left[L_K^{1/2} Q_{D_j(x)} \right] \Delta_j' \right\|_K^2 \right] \right\}^{1/2}. \quad (40)$$

Let $j \in \{1, \ldots, m\}$. The estimate (27) derived from the second order decomposition (21) yields

$$\left\| \left[L_K^{1/2} Q_{D_j(x)} \right] \Delta_j' \right\|_K^2 \le \left(\frac{\Xi_{D_j}}{\sqrt{\lambda}} + \frac{\Xi_{D_j}^2}{\lambda} \right)^2 \left\| (L_K + \lambda I)^{-1/2} \Delta_j' \right\|_K^2. \quad (41)$$

Now we apply the formula

$$E[\xi] = \int_0^\infty \text{Prob}\,[\xi > t]\,dt \tag{42}$$

to estimate the expected value of (41). By Part (b) of Lemma 17, for $0 < \delta < 1$, there exists a subset $\mathcal{Z}_{\delta,1}^{|D_j|}$ of $\mathcal{Z}^{|D_j|}$ of measure at least $1-\delta$ such that

$$\Xi_{D_j} \le \mathcal{B}_{|D_j|,\lambda} \log(2/\delta), \qquad \forall D_j \in \mathcal{Z}_{\delta,1}^{|D_j|}. \tag{43}$$

Applying Part (c) of Lemma 18 to $g(z) = y - f_\rho(x)$ with $\|g\|_\infty \le 2M$ and the data subset D_j, we know that there exists another subset $\mathcal{Z}_{\delta,2}^{|D_j|}$ of $\mathcal{Z}^{|D_j|}$ of measure at least $1-\delta$ such that

$$\left\|(L_K+\lambda I)^{-1/2}\,\Delta_j'\right\|_K \le \frac{2M}{\kappa}\mathcal{B}_{|D_j|,\lambda}\log(2/\delta), \qquad \forall D_j \in \mathcal{Z}_{\delta,2}^{|D_j|}.$$

Combining this with (43) and (41), we know that for $D_j \in \mathcal{Z}_{\delta,1}^{|D_j|} \cap \mathcal{Z}_{\delta,2}^{|D_j|}$,

$$\left\|\left[L_K^{1/2}Q_{D_j(x)}\right]\Delta_j'\right\|_K^2 \le \left(\frac{\mathcal{B}_{|D_j|,\lambda}^2}{\lambda} + \frac{\mathcal{B}_{|D_j|,\lambda}^4}{\lambda^2}\right)\left(\frac{M}{\kappa}\right)^2 \mathcal{B}_{|D_j|,\lambda}^2 \left(2\log(2/\delta)\right)^6.$$

Since the measure of the set $\mathcal{Z}_{\delta,1}^{|D_j|} \cap \mathcal{Z}_{\delta,2}^{|D_j|}$ is at least $1-2\delta$, by denoting

$$\mathcal{C}_{|D_j|,\lambda} = 64\left(\frac{\mathcal{B}_{|D_j|,\lambda}^2}{\lambda} + \frac{\mathcal{B}_{|D_j|,\lambda}^4}{\lambda^2}\right)\left(\frac{M}{\kappa}\right)^2 \mathcal{B}_{|D_j|,\lambda}^2,$$

we see that

$$\text{Prob}\left[\left\|\left[L_K^{1/2}Q_{D_j(x)}\right]\Delta_j'\right\|_K^2 > \mathcal{C}_{|D_j|,\lambda}\left(\log(2/\delta)\right)^6\right] \le 2\delta.$$

For $0 < t < \infty$, the equation $\mathcal{C}_{|D_j|,\lambda}\left(\log(2/\delta)\right)^6 = t$ has the solution

$$\delta_t = 2\exp\left\{-\left(t/\mathcal{C}_{|D_j|,\lambda}\right)^{1/6}\right\}.$$

When $\delta_t < 1$, we have

$$\text{Prob}\left[\left\|\left[L_K^{1/2}Q_{D_j(x)}\right]\Delta_j'\right\|_K^2 > t\right] \le 2\delta_t = 4\exp\left\{-\left(t/\mathcal{C}_{|D_j|,\lambda}\right)^{1/6}\right\}.$$

This inequality holds trivially when $\delta_t \ge 1$ since the probability is at most 1. Thus we can apply the formula (42) to the nonnegative random variable $\xi = \left\|\left[L_K^{1/2}Q_{D_j(x)}\right]\Delta_j'\right\|_K^2$ and obtain

$$E\left[\left\|\left[L_K^{1/2}Q_{D_j(x)}\right]\Delta_j'\right\|_K^2\right] = \int_0^\infty \text{Prob}\,[\xi > t]\,dt \le \int_0^\infty 4\exp\left\{-\left(t/\mathcal{C}_{|D_j|,\lambda}\right)^{1/6}\right\}dt$$

which equals $24\Gamma(6)\mathcal{C}_{|D_j|,\lambda}$. Therefore,

$$E\left[\|J_1\|_K\right] \le \left\{\sum_{j=1}^m \left(\frac{|D_j|}{|D|}\right)^2 24\Gamma(6)\mathcal{C}_{|D_j|,\lambda}\right\}^{1/2}$$
$$\le 1536\sqrt{5}\kappa M\left\{\sum_{j=1}^m \left(\frac{|D_j|}{|D|}\right)^2 \lambda\left(\frac{\kappa^2}{|D_j|^2\lambda^2} + \frac{\mathcal{N}(\lambda)}{|D_j|\lambda}\right)^2\left\{1 + 8\kappa^2\left(\frac{\kappa^2}{|D_j|^2\lambda^2} + \frac{\mathcal{N}(\lambda)}{|D_j|\lambda}\right)\right\}\right\}^{1/2}.$$

For the second term J_2 of (39), we apply (27) again and obtain

$$\left\|\left[L_K^{1/2}Q_{D_j(x)}\right]\Delta_j''\right\|_K \le \left(\frac{\Xi_{D_j}}{\sqrt{\lambda}} + \frac{\Xi_{D_j}^2}{\lambda}\right)\left\|(L_K+\lambda I)^{-1/2}\,\Delta_j''\right\|_K.$$

Applying the Schwarz inequality and Lemmas 17 and 18, we get

$$\begin{aligned}
E\left[\left\|\left[L_K^{1/2}Q_{D_j(x)}\right]\Delta_j''\right\|_K\right] &\le \left\{E\left[\left(\frac{\Xi_{D_j}}{\sqrt{\lambda}} + \frac{\Xi_{D_j}^2}{\lambda}\right)^2\right]\right\}^{1/2} \\
&\quad \frac{1}{\sqrt{|D_j|}}\left\{E\left[(f_\rho(x) - f_\lambda(x))^2\left\|(L_K+\lambda I)^{-1/2}(K_x)\right\|_K^2\right]\right\}^{1/2} \\
&\le \left(\left\{\frac{\kappa^2\mathcal{N}(\lambda)}{|D_j|\lambda}\right\}^{1/2} + \left\{\frac{49\mathcal{B}_{|D_j|,\lambda}^4}{\lambda^2}\right\}^{1/2}\right)\frac{\kappa\|f_\rho - f_\lambda\|_\rho}{\sqrt{|D_j|\lambda}} \\
&\le \left(\kappa\sqrt{\frac{\mathcal{N}(\lambda)}{|D_j|\lambda}} + 56\kappa^2\left(\frac{\kappa^2}{(|D_j|\lambda)^2} + \frac{\mathcal{N}(\lambda)}{|D_j|\lambda}\right)\right)\frac{\kappa\|f_\rho - f_\lambda\|_\rho}{\sqrt{|D_j|\lambda}}.
\end{aligned}$$

It follows that

$$E\left[\|J_2\|_K\right] \le \sum_{j=1}^m \frac{|D_j|}{|D|}\frac{\kappa\|f_\rho - f_\lambda\|_\rho}{\sqrt{|D_j|\lambda}}\left(\kappa\sqrt{\frac{\mathcal{N}(\lambda)}{|D_j|\lambda}} + 56\kappa^2\left(\frac{\kappa^2}{(|D_j|\lambda)^2} + \frac{\mathcal{N}(\lambda)}{|D_j|\lambda}\right)\right).$$

The last term J_3 of (39) has been handled by (27) and in the proof of Proposition 20 by ignoring the summand 1 in the bound (31), and we find from the trivial bound $\left\|\sigma_\rho^2\right\|_\infty \le 4M^2$ with $p = \infty$ that

$$E\left[\|J_3\|_K\right] \le \left(\kappa\sqrt{\frac{\mathcal{N}(\lambda)}{|D|\lambda}} + 56\kappa^2\left(\frac{\kappa^2}{(|D|\lambda)^2} + \frac{\mathcal{N}(\lambda)}{|D|\lambda}\right)\right)\left(2M\left(\frac{\mathcal{N}(\lambda)}{|D|}\right)^{\frac{1}{2}} + \frac{\kappa\|f_\rho - f_\lambda\|_\rho}{\sqrt{|D|\lambda}}\right).$$

Combining the above estimates for the three terms of (39), we see that the desired error bound in the $L^2_{\rho_X}$ metric holds true.

The estimate in the $\mathcal{H}_K$ metric follows from the steps in deriving the error bound in the $L^2_{\rho_X}$ metric except that in the representation (39) the operator $L_K^{1/2}$ in the front disappears. This change gives an additional factor $1/\sqrt{\lambda}$, the bound for the operator $(L_K+\lambda I)^{-1/2}$, and proves the desired error bound in the $\mathcal{H}_K$ metric. ■

Remark 24 *A crucial step in the above error analysis for the distributed learning algorithm is to use the unbiasedness and independence to get (40) where the norm is squared in the expected value. Thus we can obtain the optimal learning rates in expectation for the distributed learning algorithm (2) but have difficulty in getting rates in probability. It would be interesting to derive error bounds in probability by combining our second order decomposition technique with some analysis in the literature (Caponnetto and De Vito, 2007; Blanchard and Krämer, 2010; Steinwart and Christmann, 2008; Wu and Zhou, 2008).*

Proof of Theorem 1 Since $|D_j| = \frac{N}{m}$ for $j = 1, \ldots, m$, the bound in Theorem 23 in the $L^2_{\rho_X}$ metric can be simplified as

$$\begin{aligned}
E\left[\left\|\overline{f}_{D,\lambda} - f_{D,\lambda}\right\|_\rho\right] &\le C'_\kappa M\sqrt{\lambda}\sqrt{m}\left(\frac{m}{(N\lambda)^2} + \frac{\mathcal{N}(\lambda)}{N\lambda}\right)\left\{1 + \sqrt{m}\left(\frac{m}{(N\lambda)^2} + \frac{\mathcal{N}(\lambda)}{N\lambda}\right)^{\frac{1}{2}}\right\} \\
&\quad + C'_\kappa \frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}} m\left(\sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}} + \frac{m\sqrt{m}}{(N\lambda)^2} + \frac{\sqrt{m}\mathcal{N}(\lambda)}{N\lambda}\right) \\
&\quad + C'_\kappa\left(\frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}} + M\sqrt{\frac{\mathcal{N}(\lambda)}{N}}\right)\left(\sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}} + \frac{1}{(N\lambda)^2} + \frac{\mathcal{N}(\lambda)}{N\lambda}\right) \\
&\le C'_\kappa M(\sqrt{m}+1)\sqrt{\lambda}\left(\frac{m}{(N\lambda)^2} + \frac{\mathcal{N}(\lambda)}{N\lambda}\right)\left\{1 + \sqrt{m}\left(\frac{m}{(N\lambda)^2} + \frac{\mathcal{N}(\lambda)}{N\lambda}\right)^{\frac{1}{2}}\right\} \\
&\quad + C'_\kappa \frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}}(m+1)\left(\sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}} + \frac{m\sqrt{m}}{(N\lambda)^2} + \frac{\sqrt{m}\mathcal{N}(\lambda)}{N\lambda}\right) \\
&\le 2C'_\kappa M\sqrt{\lambda}\mathcal{C}_m\left\{\sqrt{m} + m\sqrt{\mathcal{C}_m}\right\} + 2C'_\kappa\frac{\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}} m\left(\sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}} + \sqrt{m}\mathcal{C}_m\right).
\end{aligned}$$

Then the desired error bound in the $L^2_{\rho_X}$ metric with $C_\kappa = 2C'_\kappa$ follows. The proof for the error bound in the $\mathcal{H}_K$ metric is similar. The proof of Theorem 1 is complete. ■

Proof of Corollary 3 As in the proof of Corollary 8, the restriction (8) implies $\mathcal{N}(\lambda) \ge \frac{\lambda_1}{\lambda_1 + C_0}$ and $N\lambda \ge \frac{m\lambda_1}{(\lambda_1 + C_0)C_0}$. It follows that

$$\frac{m}{(N\lambda)^2} \le \frac{(\lambda_1 + C_0)C_0}{\lambda_1}\frac{1}{N\lambda} \le \frac{(\lambda_1 + C_0)^2 C_0}{\lambda_1^2}\frac{\mathcal{N}(\lambda)}{N\lambda}$$

and with $\widetilde{C}'_\kappa := \frac{(\lambda_1 + C_0)^2 C_0}{\lambda_1^2} + 1$,

$$\mathcal{C}_m \le \widetilde{C}'_\kappa \frac{\mathcal{N}(\lambda)}{N\lambda}.$$

Putting these bounds into Theorem 1 and applying the restriction $\frac{m\mathcal{N}(\lambda)}{N\lambda} \le C_0$, we know that

$$\begin{aligned}
E\left\|\overline{f}_{D,\lambda} - f_{D,\lambda}\right\|_\rho &\le C_\kappa\left(\widetilde{C}'_\kappa\right)^{\frac{3}{2}}\sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}}\left(1 + \sqrt{\frac{m\mathcal{N}(\lambda)}{N\lambda}}\right)\left\{M\sqrt{\frac{m\mathcal{N}(\lambda)}{N}} + \frac{m\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}}\right\} \\
&\le C_\kappa\left(\widetilde{C}'_\kappa\right)^{\frac{3}{2}}\left(1 + \sqrt{C_0}\right)\sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}}\left\{M\sqrt{C_0}\sqrt{\lambda} + \frac{m\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}}\right\}
\end{aligned}$$

and

$$E\left\|\overline{f}_{D,\lambda} - f_{D,\lambda}\right\|_K \le C_\kappa\left(\widetilde{C}'_\kappa\right)^{\frac{3}{2}}\left(1 + \sqrt{C_0}\right)\sqrt{\frac{\mathcal{N}(\lambda)}{N\lambda}}\left\{M\sqrt{C_0} + \frac{m\|f_\rho - f_\lambda\|_\rho}{\sqrt{N\lambda}}\right\}.$$

Then the desired error bounds hold by taking the constant $\widetilde{C}_\kappa = C_\kappa\left(\widetilde{C}'_\kappa\right)^{\frac{3}{2}}\left(1 + \sqrt{C_0}\right)^2$. This proves Corollary 3. ■

Proof of Corollary 4 If

$$\mathcal{N}(\lambda) \le C_0\lambda^{-\frac{1}{2\alpha}}, \qquad \forall \lambda > 0$$

for some constant $C_0 \ge 1$, then the choice $\lambda = \left(\frac{m}{N}\right)^{\frac{2\alpha}{2\alpha\max\{2r,1\}+1}}$ satisfies (8). With this choice we also have

$$\frac{m\mathcal{N}(\lambda)}{N\lambda} \le C_0\left(\frac{m}{N}\right)^{\frac{2\alpha(\max\{2r,1\}-1)}{2\alpha\max\{2r,1\}+1}}.$$

Since regularity condition (9) yields $\|f_\lambda - f_\rho\|_\rho \le \|g_\rho\|_\rho\lambda^r$ by Lemma 21, we have by Corollary 3,

$$\begin{aligned}
E\left\|\overline{f}_{D,\lambda} - f_{D,\lambda}\right\|_\rho &\le \widetilde{C}_\kappa\sqrt{C_0}\frac{1}{\sqrt{m}}\left(\frac{m}{N}\right)^{\frac{\alpha(\max\{2r,1\}-1)}{2\alpha\max\{2r,1\}+1}}\left(\|g_\rho\|_\rho\left(\frac{m}{N}\right)^{\frac{2\alpha}{2\alpha\max\{2r,1\}+1}(r-\frac{1}{2})}\frac{m}{\sqrt{N}}\right. \\
&\quad \left. + M\left(\frac{m}{N}\right)^{\frac{\alpha}{2\alpha\max\{2r,1\}+1}}\right).
\end{aligned}$$

The inequality $\left(\frac{m}{N}\right)^{\frac{2\alpha}{2\alpha\max\{2r,1\}+1}(r-\frac{1}{2})}\frac{m}{\sqrt{N}} \le \left(\frac{m}{N}\right)^{\frac{\alpha}{2\alpha\max\{2r,1\}+1}}$ is equivalent to

$$m^{1+\frac{2\alpha}{2\alpha\max\{2r,1\}+1}(r-1)} \le N^{\frac{1}{2}+\frac{2\alpha}{2\alpha\max\{2r,1\}+1}(r-1)}$$

and it can be expressed as (10). Since (10) is valid, we have

$$E\left\|\overline{f}_{D,\lambda} - f_{D,\lambda}\right\|_\rho \le \widetilde{C}_\kappa\sqrt{C_0}\left(\|g_\rho\|_\rho + M\right)\frac{1}{\sqrt{m}}\left(\frac{m}{N}\right)^{\frac{\alpha\max\{2r,1\}}{2\alpha\max\{2r,1\}+1}}.$$

This proves the first desired convergence rate. The second rate follows easily. This proves Corollary 4. ■

Proof of Corollary 11 By Corollary 9, with the choice $\lambda = N^{-\frac{2\alpha}{4\alpha r+1}}$, we can immediately bound $\|f_{D,\lambda} - f_\rho\|_\rho$ as

$$E\left[\|f_{D,\lambda} - f_\rho\|_\rho\right] = O\left(N^{-\frac{2\alpha r}{4\alpha r+1}}\right).$$

The assumption $\mathcal{N}(\lambda) = O(\lambda^{-\frac{1}{2\alpha}})$ tells us that for some constant $C_0 \ge 1$,

$$\mathcal{N}(\lambda) \le C_0\lambda^{-\frac{1}{2\alpha}}, \qquad \forall \lambda > 0.$$

So the choice $\lambda = N^{-\frac{2\alpha}{4\alpha r+1}}$ yields

$$\frac{m\mathcal{N}(\lambda)}{N\lambda} \le C_0\frac{m\lambda^{-\frac{1+2\alpha}{2\alpha}}}{N} = C_0 m N^{\frac{1+2\alpha}{4\alpha r+1}-1} = C_0 m N^{\frac{2\alpha(1-2r)}{4\alpha r+1}}. \tag{44}$$

If m satisfies (13), then (8) is valid, and by Corollary 3,

$$\begin{aligned} E\left\|\overline{f}_{D,\lambda}-f_{D,\lambda}\right\|_\rho &\le \widetilde{C}_\kappa\sqrt{C_0}N^{\frac{\alpha(1-2r)}{4\alpha r+1}}\left(\lambda^r\|g_\lambda\|_\rho\frac{m}{\sqrt{N\lambda}}+M\sqrt{\lambda}\right)\\ &\le \widetilde{C}_\kappa\sqrt{C_0}\Big(\|g_\lambda\|_\rho+M\Big)\lambda^r\left(\frac{mN^{\frac{\alpha(1-2r)}{4\alpha r+1}}}{\sqrt{N\lambda}}+N^{\frac{\alpha(1-2r)}{4\alpha r+1}}\lambda^{\frac12-r}\right)\\ &= \widetilde{C}_\kappa\sqrt{C_0}\Big(\|g_\lambda\|_\rho+M\Big)N^{-\frac{2\alpha r}{4\alpha r+1}}\left(mN^{-\frac{2\alpha(2r-1)+\frac12}{4\alpha r+1}}+1\right). \end{aligned}$$

Since (13) is satisfied, we have

$$E\left\|\overline{f}_{D,\lambda}-f_{D,\lambda}\right\|_\rho \le 2\widetilde{C}_\kappa\sqrt{C_0}\Big(\|g_\lambda\|_\rho+M\Big)N^{-\frac{2\alpha r}{4\alpha r+1}},$$

and thereby

$$E\left[\left\|\overline{f}_{D,\lambda}-f_\rho\right\|_\rho\right]=O\left(N^{-\frac{2\alpha r}{4\alpha r+1}}\right).$$

This proves Corollary 11. ∎

Acknowledgments

We thank the anonymous referees for their constructive suggestions. The work described in this paper is supported partially by the Research Grants Council of Hong Kong [Project No. CityU 11304114]. The corresponding author is Ding-Xuan Zhou.

Appendix

This appendix provides detailed proofs of the norm estimates stated in Lemmas 17 and 18 involving the approximation of L_K by $L_{K,D(x)}$. To this end, we need the following probability inequality for vector-valued random variables in (Pinelis, 1994).

Lemma 25 *For a random variable ξ on $(\mathcal{Z},\rho)$ with values in a separable Hilbert space $(H,\|\cdot\|)$ satisfying $\|\xi\|\le\widetilde{M}<\infty$ almost surely, and a random sample $\{z_i\}_{i=1}^s$ independent drawn according to ρ, there holds with confidence $1-\widetilde{\delta}$,*

$$\left\|\frac1s\sum_{i=1}^s[\xi(z_i)-E(\xi)]\right\|\le\frac{2\widetilde{M}\log(2/\widetilde{\delta})}{s}+\sqrt{\frac{2E(\|\xi\|^2)\log(2/\widetilde{\delta})}{s}}. \tag{45}$$

Proof of Lemma 17 We apply Lemma 25 to the random variable η_1 defined by

$$\eta_1(x)=(L_K+\lambda I)^{-1/2}\langle\cdot,K_x\rangle_K K_x,\qquad x\in\mathcal{X} \tag{46}$$

It takes values in $HS(\mathcal{H}_K)$, the Hilbert space of Hilbert-Schmidt operators on $\mathcal{H}_K$, with inner product $\langle A,B\rangle_{HS}=\mathrm{Tr}(B^TA)$. Here Tr denotes the trace of a (trace-class) linear operator. The norm is given by $\|A\|_{HS}^2=\sum_i\|Ae_i\|_K^2$ where $\{e_i\}$ is an orthonormal basis

of $\mathcal{H}_K$. The space $HS(\mathcal{H}_K)$ is a subspace of the space of bounded linear operators on $\mathcal{H}_K$, denoted as $(L(\mathcal{H}_K),\|\cdot\|)$, with the norm relations

$$\|A\|\le\|A\|_{HS},\qquad \|AB\|_{HS}\le\|A\|_{HS}\|B\|. \tag{47}$$

Now we use effective dimensions to estimate norms involving η_1. The random variable η_1 defined by (46) has mean $E(\eta_1)=(L_K+\lambda I)^{-1/2}L_K$ and sample mean $(L_K+\lambda I)^{-1/2}L_{K,D(x)}$. Recall the set of normalized (in $\mathcal{H}_K$) eigenfunctions $\{\varphi_i\}_i$ of L_K. It is an orthonormal basis of $\mathcal{H}_K$. If we regard L_K as an operator on $L^2_{\rho_X}$, the normalized eigenfunctions in $L^2_{\rho_X}$ are $\{\frac{1}{\sqrt{\lambda_i}}\varphi_i\}_i$ and they form an orthonormal basis of the orthogonal complement of the eigenspace associated with eigenvalue 0. By the Mercer Theorem, we have the following uniform convergent Mercer expansion

$$K(x,y)=\sum_i\lambda_i\frac{1}{\sqrt{\lambda_i}}\varphi_i(x)\frac{1}{\sqrt{\lambda_i}}\varphi_i(y)=\sum_i\varphi_i(x)\varphi_i(y). \tag{48}$$

Take the orthonormal basis $\{\varphi_i\}_i$ of $\mathcal{H}_K$. By the definition of the HS norm, we have

$$\|\eta_1(x)\|_{HS}^2=\sum_i\left\|(L_K+\lambda I)^{-1/2}\langle\cdot,K_x\rangle_K K_x\varphi_i\right\|_K^2.$$

For a fixed i,

$$\langle\cdot,K_x\rangle_K K_x\varphi_i=\varphi_i(x)K_x,$$

and $K_x\in\mathcal{H}_K$ can be expended by the orthonormal basis $\{\varphi_\ell\}_\ell$ as

$$K_x=\sum_\ell\langle\varphi_\ell,K_x\rangle_K\varphi_\ell=\sum_\ell\varphi_\ell(x)\varphi_\ell. \tag{49}$$

Hence

$$\begin{aligned}\|\eta_1(x)\|_{HS}^2 &= \sum_i\left\|\varphi_i(x)\sum_\ell\varphi_\ell(x)\left(L_K+\lambda I\right)^{-1/2}\varphi_\ell\right\|_K^2\\ &= \sum_i\left\|\varphi_i(x)\sum_\ell\varphi_\ell(x)\frac{1}{\sqrt{\lambda_\ell+\lambda}}\varphi_\ell\right\|_K^2=\sum_i\left(\varphi_i(x)\right)^2\sum_\ell\frac{\left(\varphi_\ell(x)\right)^2}{\lambda_\ell+\lambda}.\end{aligned}$$

Combining this with (48), we see that

$$\|\eta_1(x)\|_{HS}^2=K(x,x)\sum_\ell\frac{\left(\varphi_\ell(x)\right)^2}{\lambda_\ell+\lambda},\qquad\forall x\in\mathcal{X} \tag{50}$$

and

$$E\left[\|\eta_1(x)\|_{HS}^2\right]\le\kappa^2E\left[\sum_\ell\frac{\left(\varphi_\ell(x)\right)^2}{\lambda_\ell+\lambda}\right]=\kappa^2\sum_\ell\frac{\int_{\mathcal{X}}\left(\varphi_\ell(x)\right)^2d\rho_X}{\lambda_\ell+\lambda}.$$

But

$$\int_{\mathcal{X}}\left(\varphi_\ell(x)\right)^2d\rho_X=\|\varphi_\ell\|_{L^2_{\rho_X}}^2=\left\|\sqrt{\lambda_\ell}\frac{1}{\sqrt{\lambda_\ell}}\varphi_\ell\right\|_{L^2_{\rho_X}}^2=\lambda_\ell. \tag{51}$$

So we have

$$E\left[\|\eta_1\|_{HS}^2\right] \leq \kappa^2 \sum_\ell \frac{\lambda_\ell}{\lambda_\ell+\lambda} = \kappa^2 \mathrm{Tr}\left(\left(L_K+\lambda I\right)^{-1} L_K\right) = \kappa^2 \mathcal{N}(\lambda) \tag{52}$$

and

$$E\left\|\frac{1}{|D|}\sum_{x\in D(x)} \eta_1(x) - E[\eta_1]\right\|_{HS}^2 = E\left[\left\|\left(L_K+\lambda I\right)^{-1/2}\left\{L_K - L_{K,D(x)}\right\}\right\|_{HS}^2\right] \leq \frac{\kappa^2\mathcal{N}(\lambda)}{|D|}.$$

Then our desired inequality in Part (a) follows from the first inequality of (47).

From (49) and (50), we find a bound for η_1 as

$$\|\eta_1(x)\|_{HS} \leq \kappa \frac{1}{\sqrt{\lambda}} \sqrt{\sum_\ell \left(\varphi_\ell(x)\right)^2} \leq \frac{\kappa}{\sqrt{\lambda}}\sqrt{K(x,x)} \leq \frac{\kappa^2}{\sqrt{\lambda}}, \qquad \forall x \in \mathcal{X}.$$

Applying Lemma 25 to the random variable η_1 with $\widetilde{M} = \frac{\kappa^2}{\sqrt{\lambda}}$, we know by (47) that with confidence at least $1-\delta$,

$$\begin{aligned}\left\|E[\eta_1] - \frac{1}{|D|}\sum_{x\in D(x)}\eta_1(x)\right\| &\leq \left\|E[\eta_1] - \frac{1}{|D|}\sum_{x\in D(x)}\eta_1(x)\right\|_{HS} \\ &\leq \frac{2\kappa^2\log(2/\delta)}{|D|\sqrt{\lambda}} + \sqrt{\frac{2\kappa^2\mathcal{N}(\lambda)\log(2/\delta)}{|D|}}.\end{aligned}$$

Writing the above bound by taking a factor $\frac{2\kappa\log(2/\delta)}{\sqrt{|D|}}$, we get the desired bound (29).

Recall $\mathcal{B}_{|D|,\lambda}$ defined by (30). Apply the formula (42) for nonnegative random variables to $\xi = \left\|\left(L_K+\lambda I\right)^{-1/2}\left\{L_K - L_{K,D(x)}\right\}\right\|^d$ and use the bound

$$\mathrm{Prob}\left[\xi > t\right] = \mathrm{Prob}\left[\xi^{\frac{1}{d}} > t^{\frac{1}{d}}\right] \leq 2\exp\left\{-\frac{t^{\frac{1}{d}}}{\mathcal{B}_{|D|,\lambda}}\right\}$$

derived from (29) for $t \geq \log^d 2\mathcal{B}_{|D|,\lambda}$. We find

$$E\left[\left\|\left(L_K+\lambda I\right)^{-1/2}\left\{L_K - L_{K,D(x)}\right\}\right\|^d\right] \leq \log^d 2\mathcal{B}_{|D|,\lambda} + \int_0^\infty 2\exp\left\{-\frac{t^{\frac{1}{d}}}{\mathcal{B}_{|D|,\lambda}}\right\} dt.$$

The second term on the right-hand side of above equation equals $2d\mathcal{B}_{|D|,\lambda}^d \int_0^\infty u^{d-1}\exp\{-u\}\,du$. Then the desired bound in Part (c) follows from $\int_0^\infty u^{d-1}\exp\{-u\}\,du = \Gamma(d)$ and the lemma is proved. ■

Proof of Lemma 18 Consider the random variable η_2 defined by

$$\eta_2(z) = \left(L_K+\lambda I\right)^{-1/2}(K_x), \qquad z=(x,y)\in\mathcal{Z}. \tag{53}$$

It takes values in $\mathcal{H}_K$. By (49), it satisfies

$$\|\eta_2(z)\|_K = \left\|\left(L_K+\lambda I\right)^{-1/2}\left(\sum_\ell \varphi_\ell(x)\varphi_\ell\right)\right\|_K = \left(\sum_\ell \frac{(\varphi_\ell(x))^2}{\lambda_\ell+\lambda}\right)^{1/2}.$$

So

$$E\left[\left\|\left(L_K+\lambda I\right)^{-1/2}(K_x)\right\|_K^2\right] = E\left[\sum_\ell \frac{(\varphi_\ell(x))^2}{\lambda_\ell+\lambda}\right] = \mathcal{N}(\lambda).$$

This is the statement of Part (a).

We also see from (49) that for $x\in\mathcal{X}$,

$$\|\eta_2(z)\|_K = \left(\sum_\ell \frac{(\varphi_\ell(x))^2}{\lambda_\ell+\lambda}\right)^{1/2} \leq \frac{1}{\sqrt{\lambda}}\left(\sum_\ell (\varphi_\ell(x))^2\right)^{1/2} = \frac{1}{\sqrt{\lambda}}\sqrt{K(x,x)} \leq \frac{\kappa}{\sqrt{\lambda}}.$$

This verifies the statement of Part (b).

For Part (c), we consider another random variable η_3 defined by

$$\eta_3(z) = \left(L_K+\lambda I\right)^{-1/2}(g(z)K_x), \qquad z=(x,y)\in\mathcal{Z}. \tag{54}$$

It takes values in $\mathcal{H}_K$ and satisfies

$$\|\eta_3(z)\|_K = |g(z)|\left\|\left(L_K+\lambda I\right)^{-1/2}(K_x)\right\|_K = |g(z)|\left(\sum_\ell \frac{(\varphi_\ell(x))^2}{\lambda_\ell+\lambda}\right)^{1/2}.$$

So

$$\|\eta_3(z)\|_K \leq \frac{\kappa\|g\|_\infty}{\sqrt{\lambda}}, \qquad z\in\mathcal{Z}$$

and

$$E\left[\|\eta_3\|_K^2\right] \leq \|g\|_\infty^2 E\left[\sum_\ell \frac{(\varphi_\ell(x))^2}{\lambda_\ell+\lambda}\right] = \|g\|_\infty^2 \mathcal{N}(\lambda).$$

Applying Lemma 25 verifies the statement in Part (c). The proof of the lemma is complete. ■

References

F. Bach. Sharp analysis of low-rank kernel matrix approximations. *Proceedings of the 26th Annual Conference on Learning Theory*, 185-209, 2013.

F. Bauer, S. Pereverzev, and L. Rosasco. On regularization algorithms in learning theory. *Journal of Complexity*, 23:52-72, 2007.

G. Blanchard and N. Krämer. Optimal learning rates for kernel conjugate gradient regression. *Advances in Neural Information Processing Systems*, 226-234, 2010.

G. Blanchard and N. Krämer. Convergence rates for kernel conjugate gradient for random design regression. *Analysis and Applications*, 14:763-794, 2016.

A. Caponnetto and E. De Vito. Optimal rates for the regularized least squares algorithm. *Foundations of Computational Mathematics*, 7:331-368, 2007.

A. Caponnetto and Y. Yao. Cross-validation based adaptation for regularization operators in learning theory. *Analysis and Applications*, 8:161-183, 2010.

D. R. Chen, Q. Wu, Y. Ying, and D. X. Zhou. Support vector machine soft margin classifiers: error analysis. *Journal of Machine Learning Research*, 5:1143-1175, 2004.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

O. Dekel, R. Gilad-Bachrach, O. Shamir, and X. Lin. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13:165-202, 2012.

E. De Vito, A. Caponnetto, and L. Rosasco. Model selection for regularized least-squares algorithm in learning theory. *Foundations of Computational Mathematics*, 5:59-85, 2005.

E. De Vito, S. Pereverzyev, and L. Rosasco. Adaptive kernel methods using the balancing principle. *Foundations of Computational Mathematics*, 10:455-479, 2010.

E. De Vito, V. Umanità, and S. Villa. An extension of Mercer theorem to vector-valued measurable kernels. *Applied and Computational Harmonic Analysis*, 34:339-351, 2013.

D.E. Edmunds and H. Triebel. *Function Spaces, Entropy Numbers, Differential Operators.* Cambridge University Press, Cambridge, 1996.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advance in Computional Mathematics*, 13:1-50, 2000.

J. Fan, T. Hu, Q. Wu, and D.X. Zhou. Consistency analysis of an empirical minimum error entropy algorithm. *Applied and Computational Harmonic Analysis*, 41:164-189, 2016.

S. Fine and K. Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243-264, 2002.

A. Gittens and M. Mahoney. Revisiting the Nyström method for improved large-scale machine learning. *Journal of Machine Learning Research*, 17:1-65, 2016.

X. Guo and D. X. Zhou. An empirical feature-based learning algorithm producing sparse approximations. *Applied and Computational Harmonic Analysis*, 32:389-400, 2012.

Z.C. Guo, D.H. Xiang, X. Guo, and D.X. Zhou. Thresholded spectral algorithms for sparse approximations. *Analysis and Applications*, 15:433-455, 2017.

L. Györfy, M. Kohler, A. Krzyzak, H. Walk. *A Distribution-Free Theory of Nonparametric Regression.* Springer-Verlag, Berlin, 2002.

T. Hu, J. Fan, Q. Wu, and D. X. Zhou. Regularization schemes for minimum error entropy principle. *Analysis and Applications*, 13:437-455, 2015.

J. H. Lin, L. Rosasco, and D. X. Zhou. Iterative regularization for learning with convex loss functions. *Journal of Machine Learning Research*, 17(77): 1-38, 2016.

J. H. Lin and D. X. Zhou. Learning theory of randomized Kaczmarz algorithm. *Journal of Machine Learning Research*, 16:3341-3365, 2015.

S. Mendelson and J. Neeman. Regularization in kernel learning. *The Annals of Statistics*, 38(1):526-565, 2010.

M. Meister and I. Steinwart. Optimal learning rates for localized SVMs. *Journal of Machine Learning Research*, 17: 1-44, 2016.

I. Pinelis. Optimum bounds for the distributions of martingales in Banach spaces. *The Annals of Probability*, 22:1679-1706, 1994.

G. Raskutti, M. Wainwright, and B. Yu. Early stopping and non-parametric regression: an optimal data-dependent stopping rule. *Journal of Machine Learning Research*, 15:335-366, 2014.

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. *Advances in Neural Information Processing Systems*, 1657-1665, 2015.

B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *IEEE Transactions on Information Theory*, 10:1299-1319, 1998.

O. Shamir and N. Srebro. Distributed stochastic optimization and learning. *In the 52nd Annual Allerton Conference on Communication, Control and Computing*, 2014.

W.J. Shen, H.S. Wong, Q.W. Xiao, X. Guo, and S. Smale. Introduction to the peptide binding problem of computational immunology: new results. *Foundations of Computational Mathematics*, 14:951-984, 2014.

L. Shi, Y.L. Feng, and D.X. Zhou. Concentration estimates for learning with ℓ^1-regularizer and data dependent hypothesis spaces. *Applied and Computational Harmonic Analysis*, 31:286-302, 2011.

S. Smale and D.X. Zhou. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26:153-172, 2007.

I. Steinwart and A. Christmann, *Support Vector Machines*. Springer, New York, 2008.

I. Steinwart, D. Hush, and C. Scovel. Optimal rates for regularized least squares regression. *In Proceedings of the 22nd Annual Conference on Learning Theory* (S. Dasgupta and A. Klivans, eds.), pp. 79-93, 2009.

I. Steinwart and C. Scovel. Mercer's theorem on general domains: On the interaction between measures, kernels, and RKHSs. *Constructive Approximation*, 35(3):363-417, 2012.

P. Thomann, I. Steinwart, I. Blaschzyk, and M. Meister. Spatial decompositions for large scale SVMs. ArXiv:1612.00374v1, 2016.

R. C. Williamson, A. J. Smola, and B. Schölkopf. Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transactions on Information Theory*, 47:2516-2532, 2001.

Q. Wu, Y. Ying, and D. X. Zhou. Learning rates of least-square regularized regression. *Foundations of Computational Mathematics*, 6:171-192, 2006.

Q. Wu and D. X. Zhou. Learning with sample dependent hypothesis spaces. *Computers and Mathematics with Applications*, 56:2896-2907, 2008.

Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26:289-315, 2007.

T. Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17:2077-2098, 2005.

Y. C. Zhang, J. Duchi, and M. Wainwright. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14:3321-3363, 2013.

Y. C. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16:3299-3340, 2015.

D. X. Zhou. The covering number in learning theory. *Journal of Complexity*, 18:739-767, 2002.

D. X. Zhou. Capacity of reproducing kernel spaces in learning theory. *IEEE Transactions on Information Theory*, 49:1743-1752, 2003.

Z. H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams. Big data opportunities and challenges: Discussions from data analytics perspectives. *IEEE Computational Intelligence Magazine*, 9:62-74, 2014.

Journal of Machine Learning Research 18 (2017) 1-67 Submitted 12/15; Revised 12/16; Published 9/17

Identifying Unreliable and Adversarial Workers in Crowdsourced Labeling Tasks

Srikanth Jagabathula sjagabat@stern.nyu.edu
Department of Information, Operations, and Management Sciences
Leonard N. Stern School of Business
44 West Fourth Street
New York University, NY 10012, USA

Lakshminarayanan Subramanian lakshmi@cs.nyu.edu
Ashwin Venkataraman ashwin@cs.nyu.edu
Department of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York University, NY 10012, USA

Editor: Le Song

Abstract

We study the problem of identifying *unreliable* and *adversarial* workers in crowdsourcing systems where workers (or users) provide labels for tasks (or items). Most existing studies assume that worker responses follow specific probabilistic models; however, recent evidence shows the presence of workers adopting non-random or even malicious strategies. To account for such workers, we suppose that workers comprise a mixture of honest and adversarial workers. *Honest* workers may be reliable or unreliable, and they provide labels according to an unknown but explicit probabilistic model. *Adversaries* adopt labeling strategies different from those of honest workers, whether probabilistic or not. We propose two reputation algorithms to identify unreliable honest workers and adversarial workers from *only* their responses. Our algorithms assume that honest workers are in the majority, and they classify workers with outlier label patterns as adversaries. Theoretically, we show that our algorithms successfully identify unreliable honest workers, workers adopting deterministic strategies, and worst-case *sophisticated* adversaries who can adopt arbitrary labeling strategies to degrade the accuracy of the inferred task labels. Empirically, we show that filtering out outliers using our algorithms can significantly improve the accuracy of several state-of-the-art label aggregation algorithms in real-world crowdsourcing datasets.

Keywords: crowdsourcing, reputation, adversary, outliers

1. Introduction

The growing popularity of online crowdsourcing services like Amazon Mechanical Turk and CrowdFlower has made it easy to collect low-cost labels from the crowd to generate training datasets for machine learning applications. Unfortunately, these labels are typically of low quality because of unintentional or intentional inaccuracies introduced by unreliable and malicious workers (Kittur et al., 2008; Le et al., 2010). Determining correct labels of tasks from such noisy labels is challenging because the reliabilities or qualities of workers are often unknown. While one may use "gold standard" tasks—whose true label is already known—to identify low-reliability workers (Snow et al., 2008; Downs et al., 2010; Le et al., 2010), accessing true labels for a sufficient number of tasks can be difficult and expensive. To address these challenges, a common solution is to use *redundancy* (Sheng et al., 2008), that is, collecting multiple labels for each task and assigning multiple tasks to each worker. Given the redundant labels, most existing studies make specific probabilistic assumptions on how individual workers provide labels and propose techniques to infer true task labels given workers' responses. Common probabilistic models include the one-coin model (Zhang et al., 2014), two-coin model (Raykar and Yu, 2012), and the general Dawid-Skene model (Dawid and Skene, 1979). For example, the "one-coin" model assumes that each worker w provides the correct label to an assigned task with probability p_w and (an) incorrect label with probability $1 - p_w$. The parameter p_w thus measures the reliability of worker w.

While most existing work relies on explicit probabilistic models, recent studies (Vuurens et al., 2011; Difallah et al., 2012) and anecdotal evidence show that worker labeling strategies may not be probabilistic in practice. For instance, for binary classification tasks, workers may adopt strategies that: (a) uniformly label all tasks $+1$ if it is known that $+1$ labels are more prevalent than -1 among true labels in the corpus of tasks; [1] (b) provide accurate labels to the first few tasks and random labels to remaining ones; and (c) systematically provide $+1$ labels to certain types of tasks and -1 to other types. Vuurens et al. (2011) recently provided real-world empirical evidence of similar worker strategies. In addition, workers may be malicious and may adopt sophisticated strategies for explicitly altering inferred labels of tasks. For instance, Wang et al. (2014) showed that malicious crowdsourcing campaigns, called "crowdturfing", are increasing in popularity in both dedicated and generic crowdsourcing websites (Motoyama et al., 2011; Wang et al., 2012). Further, evidence indicates the presence of malicious users in online content rating systems (such as Digg, Amazon, and Yelp) in which users can choose items to rate and the ratings are public. Specifically, users have been observed to explicitly alter the popularity of advertisements and phishing articles (Tran et al., 2009) and collaboratively target products on Amazon (Mukherjee et al., 2012).

This paper focuses on the problem of explicitly identifying *unreliable* and *adversarial* workers in crowdsourced labeling tasks by using only the labels provided by workers as an input. We consider a general crowdsourcing setting in which users/workers provide labels to items/tasks. The setting may be a crowdsourced classification application (such as Mechanical Turk) in which labels are collected for tasks by assigning [2] them to workers; or a public crowdsourcing system (such as Digg, Amazon, or Yelp) in which users provide labels/ratings to a collection of items they choose. For brevity, we use the generic terms "worker" and "task" for both types of applications. We make the following assumptions. The tasks have binary true labels in the set $\{-1, +1\}$; for cases in which the notion of true task labels is subjective, we consider it to be the population's majority opinion. We distinguish between two types of workers: *honest* and *adversarial*. The population of workers is mostly honest, with adversaries comprising a "small" fraction. Honest workers provide responses according

1. Note that the one-coin model cannot capture this strategy, but the more general two-coin model (Raykar and Yu, 2012) can
2. Workers can still choose from among assigned tasks; however, the assignment can be done to ensure that the graph representing workers' assignment to tasks has particular structures—see Section 4.1.

©2017 Srikanth Jagabathula, Lakshminarayanan Subramanian and Ashwin Venkataraman.
License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/15-650.html.

to a well-defined probabilistic model, say, M (e.g. the one-coin model introduced above), and therefore, they *can make mistakes*. However, the model M is not known. Motivated by the presence of non-random worker strategies, we go beyond standard probabilistic models and consider a much broader class of *adversarial* worker strategies. Specifically, adversaries adopt strategies different from those of honest workers, whether probabilistic or not, that is, their responses are not generated according to model M. Further, different adversaries may adopt distinct strategies.

Our work differs from most prior literature (Dawid and Skene, 1979; Smyth et al., 1995; Whitehill et al., 2009; Raykar et al., 2010; Welinder et al., 2010; Ghosh et al., 2011; Karger et al., 2011; Liu et al., 2012; Zhou et al., 2012; Dalvi et al., 2013; Zhang et al., 2014) that primarily focused on designing label aggregation algorithms to maximize the accuracy of recovered task labels. Only a few studies have explicitly focused on identifying unreliable workers based on their responses; these either relied on access to true task labels (Snow et al., 2008; Downs et al., 2010; Le et al., 2010) or did not assume any form of adversarial behavior (Vuurens et al., 2011; Raykar and Yu, 2012; Hovy et al., 2013). We aim to identify unreliable and adversarial workers using *only* their responses and do not assume access to any true task labels. Furthermore, to the best of our knowledge, we are unaware of previous studies that have addressed the problem of identifying adversarial workers who can adopt *arbitrary* strategies in crowdsourcing systems. Refer to Section 1.2 for further discussion on this aspect.

For the above setting, we design a scoring algorithm that computes "reputation scores" for workers to indicate the degree to which their labeling patterns are adversarial. We base our algorithms on the intuition that as the population is mostly honest and adversaries' labeling patterns differ from those of honest workers, adversary labeling patterns should be statistical outliers. The reputation score then indicates the degree to which a worker's labeling pattern is a statistical outlier. The adversaries identified by our algorithms may be discarded or processed separately depending on the application. Section 5 shows that discarding adversary labels can enable standard label aggregation algorithms to infer true task labels more accurately.

We note that the problem of identifying unreliable honest workers and adversaries from only their responses is nontrivial, especially because we cannot access true labels for any of the tasks. In particular, even in the simple and commonly observed case where adversaries always provide the label $+1$, the natural approach of classifying workers who only give $+1$ labels as adversaries can have arbitrarily bad performance (see Lemma 2). Furthermore, because we do not restrict the adversary strategy in any way, differentiating them from honest workers can be difficult. In fact, we show (see Theorem 14) that by carefully choosing their responses, *sophisticated* adversarial workers can render themselves *indistinguishable* from honest ones and ensure that the true labels of some fraction of tasks cannot be inferred better than a random guess.

1.1 Main contributions

Our work makes algorithmic, theoretical, and empirical contributions: [3]

3. This work expands on the results described in a previous version (Jagabathula et al., 2014)

Algorithmic contributions. Our main algorithmic contribution is a reputation algorithm that is designed to identify outlier labeling patterns. This algorithm takes as inputs the set of workers, set of tasks, and binary labels provided by each worker. Each worker may label only a subset of the tasks. Because the algorithm makes no specific assumptions on the worker labeling strategies, it identifies outliers by *penalizing* workers for the number of "conflicts" they are involved in. Specifically, suppose each task receives both $+1$ and -1 labels. For each task t_j, the algorithm maintains the number d_j^+ of $+1$ labels, number d_j^- of -1 labels, and a penalty budget of 2. Intuitively, if $d_j^+ > d_j^-$, then the worker assigning -1 to t_j is "more" of an outlier than a worker assigning label $+1$. Accordingly, the algorithm makes the following decisions: (a) how much of the penalty budget to allocate to each worker for each task and (b) how to aggregate the penalties allocated to each worker to arrive at the final reputation score.

We propose two algorithms that differ in how they make these two decisions. The first algorithm (Algorithm 1) performs a "soft" assignment: it allocates a penalty of $1/d_j^+$ (resp. $1/d_j^-$) to every worker who has provided the label $+1$ (resp. -1) to task t_j and computes the net penalty as the average of the penalties allocated across all tasks assigned to a worker. [4] The second algorithm (Algorithm 2) performs a "hard" assignment: instead of spreading the penalty score of 1 over all workers who provide the label $+1$ (resp. -1), it identifies one "representative" worker among the workers who provide the label $+1$ (resp. -1) to task t_j and allocates the entire penalty of 1 to the representative worker. The net penalty for the worker is computed by summing the accrued penalties across all assigned tasks. If the representative worker is chosen uniformly at random, then a worker who agrees with the majority is less likely to be chosen and therefore less likely to receive the penalty. However, we show that it is more appropriate to choose the representative worker in a "load-balanced" fashion by using the concept of *optimal semi-matching* (Harvey et al., 2003), where the semi-matching is defined on the bipartite graph between the workers and tasks, in which every worker is connected to the tasks that it has labeled. Both penalty assignments are based on the intuition that when there are more honest workers than adversaries, the former are more likely to agree with the majority and therefore receive lower penalties.

Theoretical contributions. We analyze our algorithms under three settings: (a) there are no adversaries and honest workers adopt the one-coin model; (b) honest workers adopt the one-coin model and adversaries, the Uniform strategy in which they label all assigned tasks $+1$; and (c) honest workers adopt the spammer-hammer model (Karger et al., 2011) and the adversaries are *sophisticated*, having infinite computational capacity and knowledge of honest workers' labels. For the first two settings, we derive guarantees for the soft-penalty algorithm and show that these guarantees extend to a random, normalized approximation of the hard-penalty algorithm; we analyze this approximation because the hard-penalty algorithm is not analytically tractable in these settings. For the last setting, we show that the hard-penalty algorithm is robust to sophisticated adversary strategies but the soft-penalty algorithm is vulnerable. Specifically, the example at the end of Section 3.1 shows an instance of an adversary strategy for which the soft-penalty algorithm misclassifies

4. We do not explicitly define the reputation score. It can be interpreted as the inverse of the net penalty—higher the penalty, the lower is the reputation, and vice-versa.

all adversaries as honest. We also show (see Lemma 13) that existing label aggregation algorithms can perform very poorly in the presence of sophisticated adversaries.

For the first two settings, we analyze our algorithms under a standard probabilistic model for crowdsourcing in which there are n workers; the worker-task assignment graph is (l, r)-regular, that is, each worker labels l tasks and each task is labeled by r workers; and the population is described by three parameters: fraction q of honest workers, fraction γ of tasks with $+1$ true labels, and average reliability μ of honest workers under the one-coin model. Recall that the reliability of an honest worker is defined as the probability of providing the correct response for any given task. This setup best represents the setting of crowdsourced classification in which tasks may be assigned according to an (l, r)-regular graph. An (l, r)-regular graph is analytically tractable, and Karger et al. (2014) showed that it achieves order-optimal performance (with respect to the best combination of task assignment and inference algorithm) when given a task assignment budget. We derive the expected penalties received by honest and adversarial workers and bound the asymptotic *error rate* or *misclassification rate* of a threshold classifier: given a penalty threshold θ, the threshold classifier classifies a worker as honest if its penalty is less than or equal to θ and as adversarial otherwise. Then, the misclassification rate is defined as the expected fraction of errors made by this classifier. This is non-trivial, especially because the natural approach that classifies all workers that provide only $+1$ labels as adversaries can have a misclassification rate arbitrarily close to 1 (see Lemma 2).

For the last setting, we make no assumptions on the structure of the worker-task assignment graph and derive performance guarantees as a function of the graph structure. This setup is reflective of public crowdsourced settings in which workers choose which tasks to label and the assumption of sophisticated adversaries is most relevant.

Formally, we establish the following results:

1. *No adversaries.* We show that the penalties assigned by the soft-penalty algorithm are consistent with worker reliabilities in the one-coin model—the higher the reliability, the lower is the penalty (see Theorem 3). When $l = \log n$ and r is constant, we show that the misclassification rate scales as $O(1/n^{2\varepsilon^2})$ for some small enough $\varepsilon \in (0, \frac{1}{\sqrt{2}})$ as $n \to \infty$, where a worker is said to be misclassified if it has high reliability (exceeding a threshold) but is classified as adversarial (see Theorem 6). In other words, our algorithm classifies all highly reliable workers as honest as $n \to \infty$. For this, the number of labels collected from each worker must tend to infinity, but only logarithmically in n.

2. Uniform *adversaries.* We derive necessary and sufficient conditions under which the soft-penalty algorithm assigns lower expected penalties to honest workers than to adversaries (see Theorem 4). Our conditions essentially require a sufficient majority of honest workers; specifically, for fixed μ and γ, we need the fraction of honest workers $q > \max\left\{1/2, h_\mu^{-1}\left(\gamma/(1-\gamma)\right)\right\}$, where $h_\mu^{-1}(\cdot)$ is an increasing function (see Theorem 4). This result shows that as γ increases, it becomes harder to separate honest workers from adversaries. This is because as the proportion γ of tasks with true labels $+1$ increases, the adversaries (who always provide the label $+1$) become more accurate, making it harder to distinguish them from honest workers. Further, when $l = \log n$ and r is constant, we show that the misclassification rate scales as $O(1/n^{2\varepsilon^2})$ for adversaries and as $F(\tilde{\mu}) + O(1/n^{2\varepsilon^2})$ for honest workers, for some $\tilde{\mu} < \mu$ and small enough $\varepsilon > 0$ as $n \to \infty$ (see Theorem 10). Here, $F(\cdot)$ is the cumulative distribution function (CDF) of honest workers' reliabilities. Our result essentially shows that asymptotically, we correctly classify all adversaries and honest workers with above average reliabilities. We argue that the presence of the term $F(\tilde{\mu})$ is necessary because it is difficult to distinguish low-reliability honest workers from adversaries.

3. *Sophisticated adversaries.* We suppose that the goal of sophisticated adversaries is to maximize the number of tasks they *affect*, that is, cause to receive labels different from those they would have received otherwise. When honest workers are perfectly reliable, we show that k sophisticated adversaries can render themselves indistinguishable from k honest workers, so that a random guess misclassifies workers with probability (1/2). We use this to provide a lower bound on the minimum number of tasks that k adversaries can affect (the true label cannot be inferred better than a random guess) *irrespective* of the label aggregation algorithm used to aggregate the worker labels (as long as it is agnostic to worker/task identities). The bound depends on the graph structure between honest workers and tasks (see Theorem 14 for details). Our result is valid across different labeling patterns and a large class of label aggregation algorithms, and therefore, it provides *fundamental* limits on the damage that k adversaries can cause. Furthermore, we propose a label aggregation algorithm (Algorithm 3) utilizing worker reputations computed by the hard-penalty algorithm and prove the existence of an upper bound on the worst-case number of affected tasks (see Theorem 17), under the assumption that honest workers adopt the popular spammer-hammer model. This combined with the result of Theorem 14 shows that our proposed label aggregation algorithm is optimal (up to a constant factor) in recovering the true labels of tasks.

Empirical contributions. We conducted two numerical studies to demonstrate the practical value of our methods (see Section 5). The first study illustrates a concrete application of our methods. With five real-world crowdsourcing datasets, it shows that *discarding* the labels of adversaries identified by our methods allows standard label aggregation algorithms to infer true task labels more accurately. These improvements are demonstrated for the following six label aggregation algorithms: (a) simple majority, (b) expectation-maximization (EM) for the two-coin model (Raykar and Yu, 2012), (c) KOS (Karger et al., 2011), (d) a normalized variant of KOS [5], (e) spectral EM (Zhang et al., 2014), and (f) regularized minimax conditional entropy (Zhou et al., 2015). Our results show that by removing up to 10 workers, our methods can improve predictive accuracy by 9.8% on average. These improvements suggest that the label patterns of discarded workers do not conform to standard probabilistic models.

The second study is designed to complement our theoretical analysis. By using synthetic data, we show that both soft- and hard-penalty algorithms successfully identify Uniform adversaries who label all assigned tasks $+1$ and low-reliability honest workers when the worker-task assignment graph has a power-law degree distribution. These degree distributions commonly arise in crowdsourcing systems when workers organically choose the

5. This variant was designed to capture non-uniform worker and task degrees because the standard KOS algorithm is designed to operate on (l, r)-regular worker-task assignment graphs.

tasks to label, with some workers labeling many tasks and some tasks receiving many labels (Franklin et al., 2011). This study also offers insights into the settings under which the soft- or hard-penalty algorithm is appropriate; specifically, these algorithms are more appropriate when the adversaries have lower and higher degrees, respectively.

1.2 Related Work

Our work is part of the literature on crowdsourcing that proposes statistical techniques to exploit the redundancy in collected labels to simultaneously infer the latent reliabilities of workers and true task labels. In particular, our work is related to three broad streams. The first stream focuses on "crowdsourced classification", namely, inferring underlying true labels of tasks when workers adopt specific probabilistic labeling strategies. Our reputation algorithm can work in conjunction with any of these methods, possibly by filtering out low reputation workers. The second stream proposes methods to explicitly filter out low-reliability workers; it is similar in spirit to our approach. The third stream focuses on methods to address sophisticated attacks in online settings; it is related to our treatment of sophisticated adversaries.

Crowdsourced classification. The literature on crowdsourced classification is vast. Most studies are based on the worker model proposed by Dawid and Skene (1979), which is a generalization of the one-coin model to tasks with more than two categories. The standard solution is to use the expectation-maximization (EM) algorithm (or its variants) to estimate worker reliability parameters and true task labels (Smyth et al., 1995; Raykar et al., 2010). The methods proposed in Liu et al. (2012) and Chen et al. (2013) take a Bayesian approach by assuming different priors over the worker reliability parameters. Whitehill et al. (2009) included task difficulty as an additional parameter in the model, and Welinder et al. (2010) studied a model with multi-dimensional latent variables for each worker, such as competence, expertise, and bias. Zhou et al. (2012, 2015) introduced a natural generalization of the Dawid-Skene model that captures tasks with differing difficulties and proposed a minimax entropy based approach which works well in real datasets. Although most of these approaches show improved performance on real-world datasets, they offer no theoretical guarantees on the resulting estimates of true task labels and model parameters.

From a theoretical perspective, the crowdsourced classification problem has been studied in two distinct regimes: *dense* and *sparse*. In the *dense* regime, it is assumed that each worker has a certain probability of labeling each task. As the problem size (or number of workers) increases, each task receives an increasing number of responses; therefore, the true labels of all tasks are eventually identified correctly (with high probability). The theoretical analysis therefore focuses on identifying the rate at which the algorithm estimates converge to the true task labels under different settings. The dense regime was first studied by Ghosh et al. (2011), who proposed a spectral method to infer task labels. More recently, Gao and Zhou (2016) studied the minimax optimal error rate of a projected EM algorithm under the one-coin model, and Li and Yu (2014) provided upper bounds on the error rate of weighted majority voting algorithms for the Dawid-Skene model. Zhang et al. (2014) showed that the EM algorithm for the Dawid-Skene model achieves the optimal convergence rate when initialized using a spectral method.

In the *sparse* regime, each task is assigned a "small" number of workers, that is, of size $O(1)$, so that the accuracy does not increase with problem size. Karger et al. (2014) were the first to analyze this scenario; they proposed an iterative message-passing algorithm for estimating true task labels as well as a task assignment scheme that minimizes the total price that must be paid to achieve an overall target accuracy. They showed that their algorithm is optimal by comparing it against an oracle estimator that knows the reliability of every worker. Dalvi et al. (2013) proposed methods based on singular-value decomposition (SVD) and analyzed the consistency of their estimators for the one-coin model. Recently, Khetan and Oh (2016) analyzed the "Generalized Dawid-Skene Model" introduced by Zhou et al. and showed that spectral approaches achieve near-optimal performance, whereas Ok et al. (2016) proved that belief propagation (BP) is optimal, that is, it matches the performance of the MAP estimator, under the one-coin model when each worker is assigned at most two tasks. In our theoretical analysis, we focus on this regime. The key distinction of our work from previous studies is that we focus on characterizing the misclassification rate of our algorithm in classifying workers, as opposed to the accuracy of recovering true task labels.

Detecting Unreliable/Adversarial workers. Some studies have aimed to explicitly detect and/or remove unreliable workers based on observed labels. One approach is to use "gold standard" tasks, that is, tasks whose true labels are already known, to identify low-reliability workers (Snow et al., 2008; Downs et al., 2010; Le et al., 2010). However, accessing true task labels can be difficult and might involve additional payment. In this work, we do not assume access to any gold standard tasks and identify adversarial workers based only on the provided labels. Vuurens et al. (2011) defined scores customized to specific adversary strategies to identify and remove them. Similarly, Hovy et al. (2013) modeled the worker population as consisting of two types—those who always provide the correct label and spammers who provide uniformly random labels—and estimated each worker's trustworthiness by using the observed labels. In contrast to these studies, we allow adversaries to adopt arbitrary strategies. Ipeirotis et al. (2010) proposed a method for quantifying worker quality by transforming the observed labels into soft posterior labels based on the estimated *confusion matrix* (Dawid and Skene, 1979). Similar to our work, their approach computes an expected cost for each worker, where the higher the cost, the lower is the worker's quality. Raykar and Yu (2012) proposed an empirical Bayesian algorithm to eliminate workers whose labels are not correlated with the true label (called *spammers*), and estimated consensus labels from the remaining workers. Both of these works rely on the Dawid-Skene model, whereas our algorithms do not assume knowledge of the probabilistic model used by workers. Some studies have tried to quantify the price of having adversarial workers under some restricted settings. Ghosh et al. (2011) (in the dense regime) and Karger et al. (2013) (in the sparse regime) considered malicious workers who can collude and provide arbitrary responses to degrade the performance of the aggregation algorithms and showed that their approaches are robust to manipulation by a small constant fraction of such adversaries. However, both these works assume specific structures on the worker-task assignment graph and do not consider adversaries who can adapt their responses based on the labels submitted by honest workers. Our analysis considers *arbitrary* worker-task assignment graphs, and we allow adversaries to choose their labels based on observed honest workers' responses.

Sybil attacks. Finally, our work is also broadly related to the rich literature on identifying *Sybil* identities in online social networks. Most such schemes (Yu et al., 2006, 2008; Danezis and Mittal, 2009; Tran et al., 2011; Viswanath et al., 2012) use the graph (or trust) structure between users to limit the corruptive influences of *Sybil attacks* (see Viswanath et al., 2010 for a nice overview). In our context, there is no information about the network structure or trust relationships between workers, and because most crowdsourcing tasks involve some form of payment, it is harder to launch Sybil attacks by forging financial credentials like credit cards or bank accounts.

2. Setup

We consider the following broad setting. There is a set $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ of m tasks such that each task t_j is associated with a latent ground-truth binary label $y_j \in \{-1, +1\}$. We elicit binary labels for these tasks from a set $W = \{w_1, w_2, \ldots, w_n\}$ of n workers. Each worker typically labels only a subset of the tasks, and we generically say that the subset of tasks is *assigned* to the worker. We represent this assignment by using a bipartite graph $\mathcal{B} = (W \cup \mathcal{T}, E)$ with workers on one side, tasks on the other side, and an edge $(w_i, t_j) \in E$ indicating that task t_j is assigned to worker w_i. We call $\mathcal{B}$ the *worker-task assignment graph* and suppose that the assignment is pre-specified.

Each worker w_i provides a binary response [6] $w_i(t_j) \in \{-1, +1\}$ for each task t_j assigned to it. We encode the responses as the *response matrix* $\mathcal{L} \in \{-1, 0, +1\}^{|W| \times |\mathcal{T}|}$ such that $\mathcal{L}_{ij} = w_i(t_j)$, for all $1 \leq i \leq n$ and $1 \leq j \leq m$, where we set $w_i(t_j) = 0$ for any task t_j not assigned to worker w_i. $W_j \subseteq W$ denotes the set of workers who labeled task t_j and $\mathcal{T}_i \subseteq \mathcal{T}$, the set of tasks assigned to worker w_i. Let d_j^+ (resp. d_j^-) denote the number of workers labeling task t_j as $+1$ (resp. -1).

Worker model. We assume that the population of workers comprises two disjoint classes: *honest* and *adversarial*. That is, $W = H \cup A$ with $H \cap A = \emptyset$, where H is the set of honest workers and A, the set of adversarial workers. The class memberships of the workers are latent, so we do not know whether a worker is honest. Honest workers provide (noisy) labels according to an unknown but explicit probabilistic model (such as the one-coin model). Adversarial workers are those whose labeling strategy does not conform to this probabilistic model; they can adopt arbitrary (deterministic or probabilistic) strategies. If honest workers adopt the one-coin model, example adversary strategies include (a) the *uniform strategy*, in which the worker arbitrarily provides uniform responses $+1$ or -1 to all assigned tasks irrespective of the true label; (b) *smart strategy*, in which the adversary is smart and chooses the uniform label in accordance with the population prevalence of the true labels so that if more than 50% of tasks are a priori known to have label -1, then the worker chooses -1 as the uniform label and vice-versa; or (c) *sophisticated strategy*, in which the worker adopts strategies specifically designed to cause the maximum "damage" (see Section 4.2 for details).

Most existing works only focus on honest workers, whereas our approach also considers adversarial workers. Furthermore, our definition of "adversary" is intentionally broader than common definitions to accommodate a wide range of labeling strategies. In fact, some of the abovementioned example adversary strategies may be accommodated by extensions of the one-coin model; for example, the uniform strategy is captured by the two-coin model (Raykar and Yu, 2012), and the smart strategy can be accommodated by allowing worker reliability parameters to depend on the population prevalence of the task labels. While such case-by-case extensions are feasible in theory, they do not extend to general adversary strategies, including sophisticated strategies specifically designed to inflict the maximum "damage".

6. We use the terms "label" and "response" interchangeably.

Given the broad definition of adversaries, our approach is to design a general algorithm to *identify* adversarial workers. The adversaries identified using our algorithm may be filtered out or investigated further, depending on the application. Specifically, our objective is to solve the following problem:

Problem 1 *Given a set of workers $W = H \cup A$, tasks $\mathcal{T}$, and response matrix $\mathcal{L}$, identify the subset of adversarial workers A.*

We describe a reputation-based algorithm that only relies on the response matrix $\mathcal{L}$ to detect adversaries. The algorithm relies on detecting workers whose labeling patterns are statistical *outliers* among the population of workers.

3. Reputation Algorithms

We now describe the proposed algorithm for identifying adversarial workers given the response matrix $\mathcal{L}$. We suppose that no side information is available on the workers' identities (e.g., a social network or worker-level demographic information), and therefore, the algorithm must solely rely on the response patterns given by workers. Our approach is to compute a "reputation" or "trust" score for each worker as a measure of the degree to which their response pattern is a statistical outlier or an anomaly. Workers with low reputation scores are significant outliers and are identified as adversaries.

To compute a worker's reputation, the algorithm relies on the number of *conflicts* the worker is involved in. A worker is involved in a conflict if its response to an assigned task is in disagreement with those of other workers. Note that tasks with a consensus opinion, that is, those having all $+1$ or -1 labels, do not provide any *discriminative* information about the workers who labeled the task. In other words, we cannot distinguish between honest and adversarial workers from just this specific task. Therefore, we focus on tasks that lack consensus, that is, those having a mix of $+1$ and -1 labels. We call this subset of tasks the *conflict set* $\mathcal{T}_{cs}$, and workers who respond to tasks in this set are all involved in conflicts. A conflict typically indicates the presence of low-reliability honest workers (who tend to make mistakes) or adversaries. In the ideal case, when all honest workers are perfectly reliable, a conflict necessarily means the presence of an adversarial worker. In this case, the number of conflicts a worker is involved in can serve as a rough indicator of the possibility of this worker being an adversary. However, when honest workers are not perfect and make mistakes, a conflict indicates only a *chance* of the presence of an adversary. Then, simply counting the number of conflicts may over-penalize honest workers who label a large number of tasks.

To overcome this issue, we propose two penalty allocation techniques, resulting in two variants of our algorithm: (a) *soft-penalty* and (b) *hard-penalty*.

3.1 Soft Penalty

In the *soft-penalty* algorithm (see Algorithm 1), for any task t_j in the conflict set, we allocate a penalty of $1/d_j^+$ and $1/d_j^-$ to all workers who provide the label $+1$ and -1 for t_j, respectively. Then, for each worker, we compute the *net penalty* by averaging the penalties across all assigned (conflict) tasks.

The above allocation of penalties implicitly rewards agreements among worker responses by making the penalty inversely proportional to the number of other workers that agree with a worker. In particular, if a worker agrees with the majority opinion on some task, then it is allocated a lower penalty than a worker who disagrees with the majority. Further, averaging normalizes for the number of tasks labeled by any worker. The algorithm relies on the following intuition for allocating penalties: assuming the average reliability of honest workers to be $> \frac{1}{2}$, we expect that honest workers provide the correct response to the assigned tasks on average. Furthermore, because there are more honest workers than adversaries, we expect the majority response to be the same as the true label of the task for most tasks. Therefore, we expect that the above allocation of penalties assigns lower penalties to high-reliability honest workers and higher penalties to low-reliability honest *and* adversarial workers. We formalize this intuition in Section 4, where we prove theoretical guarantees for the soft-penalty algorithm. We show that the soft-penalty algorithm performs well in identifying low-reliability honest workers as well as adversarial workers employing deterministic strategies (see Theorems 3, 4, 6, and 10). Our results demonstrate the asymptotic consistency of the soft-penalty algorithm in identifying adversaries under standard assumptions on the structure of the worker-task assignment graph.

Although the soft-penalty algorithm can successfully identify adversarial workers adopting certain types of strategies, its performance depends on the complexity of these strategies. If adversarial workers are non-colluding and adopt non-deliberate strategies, then the soft-penalty algorithm can identify them from the observed responses. However, this algorithm could be manipulated by more sophisticated adversaries who can collude together and adapt their labeling strategy to target certain tasks to lower their penalty scores. In particular, the soft-penalty algorithm treats each task in isolation when assigning penalties; therefore, it is susceptible to attack by determined adversaries who can cleverly decide their responses based on honest workers' labels and the structure of the worker-task assignment graph to cause maximum "damage". For example, suppose that the subgraph of $\mathcal{B}$ between honest workers and tasks is r-right regular, that is, each task receives labels from exactly r honest workers (such graphs are commonly used in practice, see Karger et al., 2014 and Ok et al., 2016), and all honest workers are perfectly reliable. Now, suppose that there are $k > r$ adversaries and that each adversary provides the incorrect response to *all* tasks. Then, every task has r correct responses, all provided by honest workers, and k incorrect responses, all provided by adversaries, resulting in net penalties of $1/r$ for each honest worker and $1/k$ for each adversary (note that the degree of the workers does not affect the net penalty because the penalty received from each task is the same). Because $k > r$, adversaries receive lower penalties than do honest workers. Therefore, filtering out k workers with the highest penalties will always filter out honest workers. Furthermore, natural aggregation algorithms (such as simple majority or weighted majority with penalties as weights) result

in incorrect labels for all tasks (see Lemma 13). In fact, for such worst-case adversaries, we can establish the following result:

Theoretical Result (Refer to Theorem 14). *Given any collection of honest worker responses, there exists an adversary strategy that can achieve a lower bound on the fraction of tasks whose true labels cannot be inferred correctly (better than a random guess) by* ***any*** *label aggregation algorithm that is agnostic to the worker and task identities.*

Section 4.2.2 provides a formal description of the result. The proof relies on the fact that sophisticated adversaries can render themselves *indistinguishable* from honest workers by carefully choosing their responses. This shows that identifying such worst-case adversarial workers can be difficult. To deal with such adversarial behavior, we use the *hard-penalty* algorithm.

Algorithm 1 soft penalty

1: **Input:** W, $\mathcal{T}$, and $\mathcal{L}$
2: For every task $t_j \in \mathcal{T}_{cs}$, allocate penalty s_{ij} to each worker $w_i \in W_j$ as follows:

$$s_{ij} = \begin{cases} \frac{1}{d_j^+}, & \text{if } \mathcal{L}_{ij} = +1 \\ \frac{1}{d_j^-}, & \text{if } \mathcal{L}_{ij} = -1 \end{cases}$$

3: **Output:** Net penalty of worker w_i:

$$pen(w_i) = \frac{\sum_{t_j \in \mathcal{T}_i \cap \mathcal{T}_{cs}} s_{ij}}{|\mathcal{T}_i \cap \mathcal{T}_{cs}|}$$

Algorithm 2 hard penalty

1: **Input:** W, $\mathcal{T}$, and $\mathcal{L}$
2: Create a bipartite graph $\mathcal{B}^{cs}$ as follows: (i) Each worker $w_i \in W$ is represented by a node on the left (ii) Each task $t_j \in \mathcal{T}_{cs}$ is represented by two nodes on the right, t_j^+ and t_j^- (iii) Add the edge (w_i, t_j^+) if $\mathcal{L}_{ij} = +1$ or edge (w_i, t_j^-) if $\mathcal{L}_{ij} = -1$
3: Compute an optimal semi-matching $\mathcal{M}$ on $\mathcal{B}^{cs}$
4: **Output:** Net penalty of worker w_i: $pen(w_i) = deg_{\mathcal{M}}(w_i)$

3.2 Hard Penalty

To deal with sophisticated adversaries, we propose a *hard* penalty allocation scheme (Algorithm 2) in which the penalty allocation for a particular task takes into account the structure of the worker-task assignment graph and the responses of other workers on *all* other tasks. In particular, instead of distributing the penalty evenly across all workers that respond to a given task, this algorithm chooses two "representative" workers to penalize for each conflict task: one each among those who provide the label $+1$ and -1. The representative workers are chosen in a load-balanced manner to "spread" the penalty across all workers and thereby avoid over-penalizing workers who provide labels for a large number of tasks. The net penalty of each worker is the sum of penalties accrued across all (conflict) tasks assigned to this worker. Intuitively, such a hard allocation of penalties will penalize workers with higher degrees (i.e. large number of assigned tasks) and many conflicts (who are potential worst-case adversaries), thereby leading to a low reputation.

To choose representative workers in a load-balanced fashion, we use the concept of *optimal semi-matchings* (Harvey et al., 2003) in bipartite graphs. For a bipartite graph $G = (V_1 \cup V_2, E)$, a *semi-matching* in G is a set of edges $M \subseteq E$ such that each vertex in V_2 is incident to exactly one edge in M (note that vertices in V_1 could be incident to

multiple edges in M). A semi-matching generalizes the notion of matchings on bipartite graphs. The optimal semi-matching is the semi-matching with the minimum *cost*. We use the common degree-based cost function defined as follows: for each $u \in V_1$, let $deg_M(u)$ denote the *degree* of u, that is, the number of edges in M that are incident to u, and let $cost_M(u)$ be defined as

$$cost_M(u) := \sum_{i=1}^{deg_M(u)} i = \frac{deg_M(u) \cdot (deg_M(u)+1)}{2}$$

Then, an *optimal semi-matching* is one that minimizes $\sum_{u \in V_1} cost_M(u)$. Intuitively, an optimal semi-matching *fairly* matches V_2 vertices across V_1 vertices such that the "load" on any V_1 vertex is minimized. The above notion of cost is motivated by the load balancing problem for scheduling tasks on machines. Specifically, consider a set of unit-time tasks T and a set of machines P. Suppose that each task t can be processed on a subset of machines; this can be specified as a bipartite graph between T and P. On any given machine, the tasks are executed one after the other in series. An optimal semi-matching can be thought of as an assignment of tasks to the machines such that the *flow-time*, that is, average completion time of a task, is minimized. See (Harvey et al., 2003) for more details.

To determine the representative workers for each task, we compute the optimal semi-matching in the following augmented worker-task assignment graph: we split each task t_j into two copies, t_j^+ and t_j^-, and connect worker w_i to t_j^+ or t_j^- depending on whether this worker labeled the task $+1$ or -1, respectively. By definition, the optimal semi-matching yields two representative workers for each task t_j—one connected to t_j^+ and the other connected to t_j^-. As in the soft-penalty algorithm, we only consider conflict tasks when creating this augmented bipartite graph. The worker degrees in the optimal semi-matching then constitute their net penalties. Algorithm 2 describes the hard-penalty algorithm.

3.3 Connection between soft-penalty and hard-penalty algorithms

Although the hard- and soft-penalty algorithms appear different, the latter can be interpreted as a random, normalized variant of the former. Specifically, suppose we choose a random semi-matching M in the augmented worker-task assignment graph $\mathcal{B}^{cs}$, defined in Algorithm 2, and assign the penalty $deg_M(w_i)/deg_{\mathcal{B}^{cs}}(w_i)$ to worker w_i, where $deg_{\mathcal{B}^{cs}}(w_i)$ is the degree of worker w_i in $\mathcal{B}^{cs}$. When the random semi-matching is constructed by mapping each copy t_j^+ (or t_j^-) of task t_j uniformly at random to a worker connected to it, the probability that it will be mapped to worker $w_i \in W_j$ is equal to $1/deg_{\mathcal{B}^{cs}}(t_j^+)$ (or $1/deg_{\mathcal{B}^{cs}}(t_j^-)$), or equivalently, $1/d_j^+$ (or $1/d_j^-$). Therefore, the expected degree $\mathbb{E}[deg_M(w_i)]$ of worker w_i is equal to $\sum_{t_j \in \mathcal{T}_i \cap \mathcal{T}_{cs}} s_{ij}$, where $s_{ij} = 1/d_j^+$ if $\mathcal{L}_{ij} = +1$ and $1/d_j^-$ if $\mathcal{L}_{ij} = -1$. Because the degree $deg_{\mathcal{B}^{cs}}(w_i)$ of worker w_i is equal to $|\mathcal{T}_i \cap \mathcal{T}_{cs}|$, it follows that the expected penalty of worker w_i is equal to $\mathbb{E}[deg_M(w_i)]/deg_{\mathcal{B}^{cs}}(w_i) = \sum_{t_j \in \mathcal{T}_i \cap \mathcal{T}_{cs}} s_{ij} / |\mathcal{T}_i \cap \mathcal{T}_{cs}|$, which is exactly the penalty allocated by the soft-penalty algorithm. It thus follows that the expected penalties under the above random, normalized variant of the hard-penalty algorithm are equal to the penalties allocated by the soft-penalty algorithm. When all workers are assigned the same number of tasks, the expected penalty assigned by the random hard-penalty algorithm is equal to the penalty assigned by the soft-penalty algorithm, but scaled by a constant factor.

With the above interpretation of the soft-penalty algorithm, it follows that the hard-penalty algorithm differs from the soft-penalty algorithm in two key aspects: it (a) does not normalize the penalties by degrees and (b) uses optimal semi-matchings as opposed to random semi-matchings. The absence of degree-based normalization of the penalties results in significant penalization of high-degree workers. The use of the optimal semi-matching results in a more balanced allocation of penalties by optimizing a global objective function. Both of these effects make the hard-penalty algorithm conservative and robust to sophisticated adversary strategies, as established theoretically in Section 4. The above connection also suggests that the random, normalized variant of the hard-penalty algorithm should have performance similar to that of the soft-penalty algorithm. We explore this aspect theoretically at the end of Section 4.1.

4. Theoretical Results

Our reputation algorithms are analytically tractable, and we establish their theoretical properties below. Our analysis is aimed at deriving the conditions under which our algorithms separate adversaries from honest workers. We use uppercase boldface letters (say, $\boldsymbol{X}$) to denote random variables, unless it is clear from the context. The proofs of all results are given in Appendix A.

4.1 Soft-penalty algorithm: common adversary strategies

First, we analyze the performance of the soft-penalty algorithm. We assume that honest workers adopt the one-coin model, where each worker w is characterized by parameter $\mu_w \in [0,1]$ that specifies the probability that w provides the correct response for any task. We choose this model because it is the simplest non-trivial model to analyze, and it has been widely studied in existing literature (Ghosh et al., 2011; Dalvi et al., 2013; Karger et al., 2014; Zhang et al., 2014; Gao and Zhou, 2016). We focus on two settings: (a) the classical setting in which there are no adversaries ($A = \emptyset$) and (b) the setting in which adversaries adopt the Uniform strategy.

Definition 1 (Uniform strategy) *Every adversarial worker provides the same $+1$ response to all tasks assigned to it.*

We consider the Uniform strategy because of its ubiquity and simplicity. It is commonly observed in practice (Vuurens et al., 2011), and our analysis of real-world datasets (discussed in detail in Section 5) reveals that a significant fraction of workers ($\approx 30\%, 17\%$, and 9% of workers in the `stage2, task2`, and `temp` datasets, respectively) provide uniform labels for all assigned tasks. Note that it is not captured by the standard one-coin model, in which workers assign the label y_j or $-y_j$ to task t_j, where y_j is the true label. It can be adopted by both "lazy" and "smart" adversaries. Lazy workers adopt this strategy to maximize the number of tasks they label and the corresponding payment they obtain. Smart adversaries adopt this strategy if it is known a priori that the prevalence (or proportion) γ of tasks with true labels $+1$ is large, say, above 90%. For instance, medical images showing tumors contain a large proportion of benign ones and a correspondingly small proportion of malignant ones,

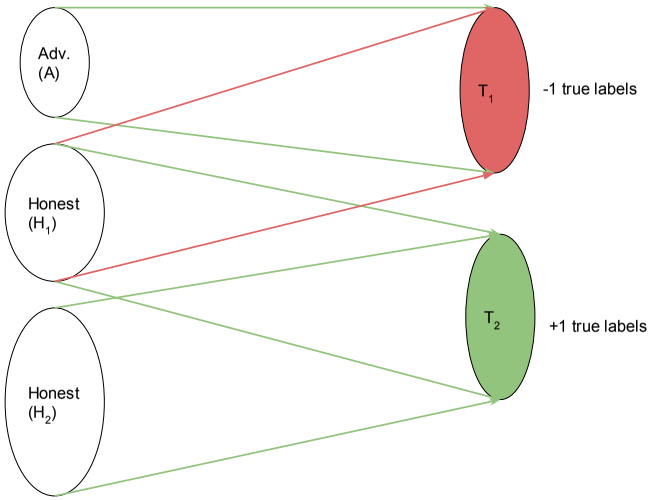


Figure 1: Example where filtering workers who give only +1 labels performs poorly

leading a "smart" worker to label all assigned images as benign without carefully considering each image. Therefore, the Uniform strategy actually comprises a spectrum of strategies of varying degrees of "smartness", with higher values of γ indicating smarter strategies.

A natural way to identify adversaries who adopt the Uniform strategy is to classify all workers who have labeled *all* assigned tasks +1 as adversaries. However, we show that this approach can have arbitrarily large *misclassification rate*, because it may misclassify almost all (asymptotic fraction approaching 1) perfectly reliable honest workers as adversarial.

Lemma 2 (Hardness of identifying Uniform adversarial workers) *Consider a simple binary classifier $\hat{\boldsymbol{I}}_{\text{natural}}(\cdot)$ as follows:*

$$\hat{\boldsymbol{I}}_{\text{natural}}(w_i) = \begin{cases} \textit{adversarial}, & \textit{if } w_i \textit{ gives all } +1 \textit{ labels} \\ \textit{honest}, & \textit{o.w.} \end{cases}$$

Then, the misclassification rate, that is, fraction of incorrectly classified workers, of the above classifier can be arbitrarily close to 1.

Proof Suppose the collection of tasks partitions into two sets, T_1 and T_2, consisting of tasks that have true labels -1 and $+1$, respectively (see Figure 1). The adversary A follows the Uniform strategy and labels all tasks in T_1 as $+1$. There are two groups of honest workers—H_1 and H_2—who are perfectly reliable, that is, they always provide the correct label. The workers in H_1 label all tasks in $T_1 \cup T_2$, and workers in H_2 label only the tasks in T_2. Now, because honest workers H_2 give only +1 labels, the classifier $\hat{\boldsymbol{I}}_{\text{natural}}$ misclassifies

all honest workers in H_2 as adversaries. In other words, we have

$$\frac{1}{|H_1 \cup H_2|} \sum_{h \in H_1 \cup H_2} \mathbf{1}\left[\hat{\boldsymbol{I}}_{\text{natural}}(h) \neq \textit{honest}\right] = \frac{|H_2|}{|H_1 \cup H_2|}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. Suppose H_2 comprises a large fraction of honest workers, that is, $|H_2| = (1-\rho)\,|H_1 \cup H_2|$ for some "small" $\rho > 0$. Then, it follows from the above equation that the misclassification rate of the natural classifier $\hat{\boldsymbol{I}}_{\text{natural}}(\cdot)$ is $1-\rho$, which can be arbitrarily close to 1. ■

The above lemma shows that the problem of identifying adversaries is nontrivial even for the Uniform strategy case, and our analysis below provides precise conditions under which we can identify such adversarial workers. In particular, for the the scenario outlined in the proof above, our soft-penalty algorithm assigns a penalty of $\frac{1}{|A|}$ to adversaries, $\frac{1}{|H_1|}$ to honest workers in H_1, and zero penalty to honest workers in H_2 (because they are not part of any conflicts). Consequently, as long as $|H_1| > |A|$, our algorithm correctly separates out honest workers from adversaries by choosing a threshold $\theta \in \left(\frac{1}{|H_1|}, \frac{1}{|A|}\right)$ and classifying all workers with penalty $> \theta$ as adversarial.

Because the performance of the algorithm depends on the specific crowdsourced classification instance (worker-task assignment graph, true task labels, reliabilities of honest workers), we conduct a probabilistic analysis under a natural generative model. For our analysis, we focus on worker-task assignment graphs $\mathcal{B}$ that are (l, r)-regular, in which each worker is assigned l tasks and each task is labeled by r workers. These assignment graphs are analytically tractable and have been shown by Karger et al. (2014) to achieve order-optimal performance (with respect to the best combination of task assignment and inference algorithm) when given a certain budget for task assignment. To generate the crowdsourcing instance, we use the probabilistic model of crowdsourced labeling proposed by Karger et al. (2014) but extended to incorporate adversarial workers:

Generative model. Suppose the fraction $q \in (0, 1]$ of honest workers, number of tasks m, number of workers n, worker degree $l > 1$, and task degree $r > 1$ are fixed. Let $\gamma \in [0, 1]$ denote the prevalence (or proportion) of tasks with true labels $+1$ and $F(\cdot)$, the cumulative distribution function (CDF) of the honest worker reliabilities under the one-coin model, with $\mu \in [0, 1]$ denoting the mean. Sample a crowdsourced classification instance as follows:

1. *Worker-task assignment graph:* Assign m tasks to n workers using the *configuration model*—take $n \cdot l$ half-edges for worker nodes and $m \cdot r$ half-edges for the task nodes, pick a random permutation of worker half-edges, and map them to task half-edges.

2. *True task labels:* For each task t_j, sample the true label $\boldsymbol{Y}_j \in \{-1, +1\}$ independently according to the Bernoulli distribution with $\Pr[\boldsymbol{Y}_j = +1] = \gamma$.

3. *Worker identities:* For each worker w_i, set its identity to *honest* with probability q and *adversarial* with probability $1 - q$.

4. *Honest worker reliabilities and responses:* If w_i is honest, sample its reliability $\boldsymbol{M}_i = \mu_i$ from the distribution $F(\cdot)$. For each task t_j assigned to w_i, set the response $w_i(t_j)$ to $\boldsymbol{Y}_j$ with probability μ_i and $-\boldsymbol{Y}_j$ with probability $1 - \mu_i$.

5. *Adversarial worker responses:* If w_i is adversarial, set the response $w_i(t_j) = +1$ for all tasks t_j assigned to w_i.

The above generative model may be justified as follows. First, the *configuration model* is a simple random construction to generate graphs that is popular in random graph literature (Bollobás, 2001). It may result in a graph with multi-edges (where two nodes are connected by more than one edge); however, the number of double-edges converges to a Poisson distribution with mean $(l-1)(r-1)/2$ (see Bollobás, 2001). Therefore, the proportion of nodes with multi-edges is $\approx lr/n$, which tends to zero as $n \to \infty$ as long as $l = o(n)$ and r is constant.

The model for true task labels, worker identities, and reliabilities may be justified by supposing that the m tasks $\mathcal{T} = \{t_1, \ldots, t_m\}$ are drawn from a "large" population of tasks with a prevalence γ of $+1$ tasks and workers are drawn from a "large" population with a proportion $1-q$ of adversaries and a proportion q of honest workers, whose reliabilities have the distribution $F(\cdot)$. Then, the distributional assumptions for the true task labels, worker identities, and honest worker reliabilities will be met when the task assignment is randomized and there is no distinction between sampling with and without replacement because of the large population sizes. The model for generating honest worker responses is the standard one-coin model. See Karger et al. (2014) for a detailed discussion of the settings under which the above probabilistic model is reasonable.

For our theoretical analysis, we assume that non-conflict tasks (with all $+1$ or -1 labels) are *not* ignored/dropped for the purposes of penalty computation. This assumption makes the analysis less cumbersome and may be justified by noting that for a large enough r, the probability that a task will be non-conflict is low. Even if a task is non-conflict, we expect little impact from its inclusion because the penalty from this task will be $1/r$, which is negligible for large values of r. We also tested this assertion numerically and observed negligible differences in the performances of the two variants (with and without dropping high-degree non-conflict tasks) of the soft-penalty algorithm (see Section 5).

4.1.1 Analysis of expected penalties

We first analyze the expected penalties received by honest and adversarial workers under the abovementioned generative model and identify the conditions for population parameters q, μ, and γ under which honest workers receive lower expected penalties. Let $\mathbf{PEN}_i$ denote the penalty assigned by the soft-penalty algorithm to worker w_i; note that it is a random variable under the abovementioned generative model.

First, we focus on the classical setting in which there are no adversarial workers; therefore, $A = \emptyset$. We obtain the following result:

Theorem 3 (Reputations consistent with reliabilities) *When $q = 1$ (i.e., there are no adversarial workers) and $\mu > \frac{1}{2}$, we have*

$$\mathbb{E}\left[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_1\right] < \mathbb{E}\left[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_2\right] \iff \mu_1 > \mu_2$$

for any worker w_i.

Theorem 3 shows that the expected reputation scores are consistent with honest workers' reliabilities: as the honest worker's reliability decreases, the expected penalty increases,

that is, the expected reputation score decreases. As honest worker reliabilities capture their propensities to make mistakes, our algorithm flags workers who are prone to making mistakes, as desired. Consequently, filtering out low-reputation workers filters out workers with low reliabilities (we make this claim precise in Section 4.1.2 below).

Next, we consider the case in which $A \neq \emptyset$ and there is a fraction $1-q$ of workers who are adversarial and adopt the Uniform strategy. Let p_h and p_a denote the expected penalties that a worker receives conditioned on being honest and adversarial, respectively, that is,

$$p_h = \mathbb{E}\left[\mathbf{PEN}_i \mid w_i \text{ is honest }\right] \quad \text{and} \quad p_a = \mathbb{E}\left[\mathbf{PEN}_i \mid w_i \text{ is adversarial }\right]$$

Because of symmetry, these expectations do not depend on worker indices. Then, we obtain the following result:

Theorem 4 (Penalties under Uniform strategy) *When $q < 1$, $\mu > \frac{1}{2}$, and the adversaries adopt the* Uniform *strategy, we obtain*

$$p_h < p_a \iff \left(q\mu > \frac{1}{2} \quad \text{and} \quad \frac{\mu}{1-\mu} \cdot h(\mu, q) > \frac{\gamma}{1-\gamma}\right)$$

where $h(\mu, q)$ is a strictly increasing function in q for a fixed μ, and it is defined as

$$h(\mu, q) = \frac{g(1-Q) - g(Q)}{g(P) - g(1-P)} \text{ where } P := q\mu + (1-q), Q := 1 - q\mu, \text{ and } g(x) := \frac{1-x^r}{r \cdot (1-x)}.$$

The above result reveals the conditions for the parameters q, μ, and γ under which the soft-penalty algorithm is successful in assigning lower penalties to honest workers than to adversaries.

To understand this, we first focus on the condition $q\mu > 1/2$. Note that because $\mu \leq 1$, this condition implies that the population must consist of more honest workers than adversaries ($q > 1/2$). This is because our algorithm is designed to identify "outlier" response patterns—those which deviate from the majority—and for adversaries to be declared outliers, they must necessarily be in the minority.

Furthermore, note that the necessary condition $\mu > 1/(2q)$ implies that for our algorithm to be successful, the average reliability μ of the honest workers must be "large enough"; specifically, it must exceed $\frac{1}{2q}$ (for a fixed q). To obtain an intuitive understanding of this condition (the proof is given in Appendix A.1.2), note the following. Consider a task with true label $+1$. Then, in expectation, there are $rq\mu$ honest workers and $(1-q) \cdot r$ adversaries who provide the response $+1$ and $rq \cdot (1-\mu)$ honest workers who provide the response -1. Now, the adversaries will agree with the majority if and only if $r \cdot (q\mu + (1-q)) \geq rq \cdot (1-\mu)$, that is, $\mu \geq 1 - 1/(2q)$. Similarly, when the true label of a task is -1, then in expectation, there are $r \cdot (q \cdot (1-\mu) + (1-q))$ workers providing $+1$ label and $rq\mu$ workers providing -1 label. Again, the adversaries will be in the majority in expectation if and only if $\mu \leq 1/(2q)$. It thus follows that if $\mu \in [1 - \frac{1}{2q}, \frac{1}{2q}]$, then the adversaries are always in majority in expectation, and therefore, they will receive a lower penalty. Because $\mu > 1/2$ and $1 - \frac{1}{2q} \leq 1/2$, a necessary condition for the worker to receive a lower penalty when being honest is therefore $\mu > \frac{1}{2q}$, that is, $q\mu > \frac{1}{2}$.

Now assuming that the first condition is met, that is, $q\mu > 1/2$, we focus on the second condition: $\frac{\mu}{1-\mu} \cdot h(\mu, q) > \frac{\gamma}{1-\gamma}$. When $\gamma = 1$, this condition is not met (unless $\mu = 1$), and therefore, honest workers receive higher expected penalties than adversaries. This is because if all tasks have a true label $+1$, then in expectation, there is a fraction $q\mu + 1 - q > 1/2$ (because $q\mu > 1/2$) of workers providing the label $+1$ to each task, implying that the adversaries always agree with the majority label for each task. As a result, our algorithm filters out honest workers; however, it must be noted that adversaries actually have higher (perfect) reliabilities in this special case. Similarly, when $\mu = 1$, that is, honest workers are perfectly reliable, the condition is always met because honest workers are in the majority at each task (in expectation); specifically, when the true label is $+1$, all responses are $+1$, and when the true label is -1, all honest workers (a fraction $q > 1/2$) provide the label -1.

Next, we investigate the performance of our algorithm as the adversary strategies become "smarter". As noted above, the Uniform strategy comprises a spectrum of strategies of varying degrees of "smartness", with higher values of γ indicating smarter strategies.

Corollary 5 (Penalties under smarter adversary strategies) *Suppose $\mu > \frac{1}{2}$ is fixed and $q\mu > \frac{1}{2}$. Then, it follows that*

$$p_h < p_a \iff q > h_\mu^{-1}\left(\frac{\gamma}{1-\gamma}\right)$$

where $h_\mu(q) := \frac{\mu}{1-\mu} \cdot h(\mu, q)$ and $h_\mu^{-1}(\cdot)$ is the inverse of $h_\mu(\cdot)$. In other words, for fixed $\mu > \frac{1}{2}$, we require a minimum fraction of honest workers to ensure that adversaries receive a higher penalty than honest workers, and this fraction increases as γ increases.

The above result shows that as the adversary strategies become smarter, it becomes more difficult to distinguish them from honest workers. Specifically, because $h_\mu^{-1}(\cdot)$ is a strictly increasing function, we require honest workers to have a larger majority as γ increases to ensure that they receive lower expected penalties than adversaries.

4.1.2 Asymptotic identification of adversaries and honest workers

Assuming that the expected penalties of adversaries and honest workers are separated, we now derive the asymptotic error rates, defined as the expected fraction of errors, of the soft-penalty algorithm as $n \to \infty$ when (1) there are no adversaries and (2) adversaries adopt the Uniform strategy.

To analyze the error rates, we consider the following threshold-classifier $\hat{\boldsymbol{I}}_\theta(\cdot)$ based on the soft-penalty algorithm: given a penalty threshold $\theta \in \mathbb{R}$, define the binary classifier

$$\hat{\boldsymbol{I}}_\theta(w_i) = \begin{cases} \text{honest}, & \text{if } \mathbf{PEN}_i \le \theta \\ \text{adversarial}, & \text{o.w.} \end{cases}$$

Let $\boldsymbol{I}(w_i)$ denote the latent true identity of worker w_i. Note that both $\boldsymbol{I}(w_i)$ and $\hat{\boldsymbol{I}}_\theta(w_i)$ are random variables under our generative model.

As above, we first consider the case in which there are no adversaries, that is, $q = 1$, so that $\boldsymbol{I}(w_i) = \textit{honest}$ for all workers w_i. In this case, Theorem 3 shows that workers with

higher reliabilities receive lower expected penalties. Furthermore, we now show that the threshold classifier correctly classifies high-reliability workers as honest with high probability as $n \to \infty$:

Theorem 6 (Identification of high-reliability workers) *Suppose $q = 1$ and $\mu > \frac{1}{2}$. Given $\theta \in (0,1)$ and $\varepsilon \in (0, \frac{1}{\sqrt{2}})$ such that $\theta - \varepsilon \in (g(1-\mu), g(\mu))$, define $\hat{\mu}(\theta) := \frac{g(\mu)+\varepsilon-\theta}{g(\mu)-g(1-\mu)}$, where the function $g(\cdot)$ was defined in Theorem 4. Then, under the generative model, we obtain*

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \text{ and } \boldsymbol{M}_i > \hat{\mu}(\theta)\right) \le \frac{l^2 r^2}{n-1} + \exp\left(-\frac{2l\varepsilon^2}{(1-1/r)^2}\right)$$

When $l = \log n$ and r is fixed, we obtain

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \text{ and } \boldsymbol{M}_i > \hat{\mu}(\theta)\right) = O\left(\frac{1}{n^{2\varepsilon^2}}\right) \text{ as } n \to \infty$$

Theorem 6 provides an upper bound for the *error rate* or *misclassification rate* of the threshold classifier. We say that the classifier makes an *error* if it classifies a worker whose reliability is higher than $\hat{\mu}(\theta)$ as adversarial, and the misclassification rate is defined as the expected fraction of errors. As $n \to \infty$, the first term, $l^2 r^2/(n-1)$, in the error bound goes to 0 as long as $l = o(\sqrt{n})$. With the task degree r fixed, the second term goes to zero as long as $l \to \infty$ when $n \to \infty$. Upon combining these two observations, it follows that taking $l = \log n$ yields the error bound of $O(1/n^{2\varepsilon^2})$ for a fixed $\varepsilon \in (0, \frac{1}{\sqrt{2}})$ and r, as $n \to \infty$. In other words, our result shows that as long as we collect $\log n$ labels from each worker and a fixed number of labels for each task, we will classify workers with reliabilities higher than $\hat{\mu}(\theta)$ as honest with a high probability as $n \to \infty$. Although the number of labels collected from each worker must tend to infinity, it must only grow logarithmically in the total number of workers n. Finally, if the population reliability μ is known, we can determine the value of threshold θ for a given reliability threshold $\hat{\mu}(\theta)$.

The proof of Theorem 6 relies on establishing that the worker penalties are concentrated around their respective expectations, for which we need the worker-task assignment graph $\mathcal{B}$ to be locally tree-like:

Definition 7 (Locally tree-like assignment graphs) *An (l, r)-regular worker-task assignment graph $\mathcal{B}$ is said to be D-locally tree-like at a worker node w_i if the subgraph $\mathcal{B}_{w_i,D}$, consisting of nodes at a distance of at most D from w_i, is a tree.*

For our purposes, it suffices to have $\mathcal{B}_{w_i,2}$ be a tree. Note that the subgraph $\mathcal{B}_{w_i,2}$ consists of the worker node w_i; tasks labeled by w_i, that is, the set $\mathcal{T}_i$; and workers $\bigcup_{j\in\mathcal{T}_i} W_j$ who labeled the tasks in $\mathcal{T}_i$. Karger et al. (2014) show that a random construction of the assignment graph using the configuration model ensures that $\mathcal{B}_{w,2}$ is a tree with a high probability as $n \to \infty$ for a randomly chosen worker w.

Lemma 8 (Random construction ensures local tree-structure) *If $\mathcal{B}$ is random (l, r)-regular constructed according to the configuration model, then for a randomly chosen worker $\boldsymbol{w}$*

$$\Pr\left(\mathcal{B}_{\boldsymbol{w},2} \text{ is not a tree}\right) \le \frac{l^2 r^2}{n-1}$$

The proof of the above lemma is in Appendix A.2.1 and it follows the arguments in Karger et al. (2014). Based on the result of Lemma 8, the proof of Theorem 6 proceeds in two steps. First, whenever the configuration model generates an assignment graph $\mathcal{B}$ that is not locally tree-like, we immediately declare an error, incurring an error probability that is bounded above by $l^2r^2/(n-1)$; this yields the first term in the error bound. Second, when $\mathcal{B}_{w_i,2}$ is indeed a tree, we obtain the second term in the error bound by invoking the following concentration result:

Lemma 9 (Concentration of honest worker penalties) *Suppose that $q = 1$ and $\mathcal{B}_{w_i,2}$ is a tree. Under the generative model and for a fixed reliability $\boldsymbol{M}_i = \mu_i$, given any $\varepsilon > 0$, the penalty assigned to honest worker w_i concentrates as*

$$\Pr\left(\mathbf{PEN}_i \geq \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] + \varepsilon \,\middle|\, \boldsymbol{M}_i = \mu_i\right) \leq \exp\left(\frac{-2l\epsilon^2}{(1-1/r)^2}\right)$$

Note that the above lemma holds for *any* fixed value of reliability μ_i. The proof of the above result relies on expressing the penalty scores as an average of l random variables and then invoking Hoeffding's concentration bound. The local tree-like property of the assignment graph $\mathcal{B}$ at worker node w_i ensures that the l random variables are mutually independent (which is required for Hoeffding's inequality).

Next, we consider the case in which there is a fraction $1-q$ of workers who are adversaries and adopt the Uniform strategy. Theorem 4 above provides the necessary and sufficient conditions for an honest worker to receive a lower expected penalty than an adversary, that is, for $p_h < p_a$. Under these conditions, we obtain the following result:

Theorem 10 (Identification of honest and adversarial workers) *Suppose $p_h < p_a$ and let $\theta \in (p_h + \varepsilon, p_a - \varepsilon)$ for some ε small enough such that $0 < \epsilon < (p_a - p_h)/2$. Then, under the generative model we obtain*

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \,\middle|\, \boldsymbol{I}(w_i) = \textit{honest}\right) \leq \frac{l^2r^2}{n-1} + \exp\left(-\frac{2l\varepsilon^2}{(1-1/r)^2}\right) + F\left(\hat{\mu}(q,\theta)\right)$$

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \,\middle|\, \boldsymbol{I}(w_i) = \textit{adversarial}\right) \leq \frac{l^2r^2}{n-1} + \exp\left(-\frac{2l\varepsilon^2}{(1-1/r)^2}\right)$$

where $\hat{\mu}(q,\theta)$ is such that $\hat{\mu}(1,\theta) = \hat{\mu}(\theta)$ and $\hat{\mu}(q,\theta) < \mu$ for all $q \in (0,1]$ and $\theta \in (p_h + \varepsilon, p_a - \varepsilon)$.

When $l = \log n$ and r is fixed, we obtain

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \,\middle|\, \boldsymbol{I}(w_i) = \textit{honest}\right) = O\left(\frac{1}{n^{2\varepsilon^2}}\right) + F\left(\hat{\mu}(q,\theta)\right)$$

$$\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\hat{\boldsymbol{I}}_\theta(w_i) \neq \boldsymbol{I}(w_i) \,\middle|\, \boldsymbol{I}(w_i) = \textit{adversarial}\right) = O\left(\frac{1}{n^{2\varepsilon^2}}\right)$$

The precise expression for $\hat{\mu}(q,\theta)$ is involved and is given in Appendix A.2.4. Theorem 10 provides the misclassification rate of our algorithm when the population parameters q, μ, and γ satisfy the conditions of Theorem 4, ensuring that honest workers receive a lower expected penalty than adversaries. Following the arguments from the discussion under Theorem 6 above, it can be seen that when $l = \log n$ and r is fixed, the fraction of adversaries that is misclassified is $O(1/n^{2\varepsilon^2})$. On the other hand, the fraction of honest workers that is misclassified scales as $O(1/n^{2\varepsilon^2}) + F(\hat{\mu}(q,\theta))$. The first term tends to zero as $n \to \infty$. The second term denotes the probability that honest worker reliability is less than or equal to $\hat{\mu}(q,\theta)$. In other words, our algorithm misclassifies low-reliability workers as adversaries. In the special case when all honest workers have the same reliability μ, it immediately follows that the probability density function is a point mass at μ, from which it follows that $F(\hat{\mu}(q,\theta)) = 0$ because $\hat{\mu}(q,\theta) < \mu$. In this case, the misclassification error for the honest workers also tends to zero as $n \to \infty$.

We note that the dependence of the honest worker misclassification rate on $F\left(\hat{\mu}(q,\theta)\right)$ is fundamental to our algorithm. As an example, consider the case when the reliability distribution is a two-point distribution with probability mass μ at 1 and the remaining $1-\mu$ mass at 0. This distribution results in two types of honest workers: workers who always provide the correct response and those that always provide the incorrect response. Note that the average reliability under this distribution is μ. Let p_0 and p_1 denote the expected penalties under our generative model for a worker conditioned on being honest with reliabilities 0 and 1, respectively. Then, it follows from Lemma 19 in the Appendix that

$$\begin{aligned} p_1 &= \gamma \cdot g(1-P) + (1-\gamma) \cdot g(Q) \\ p_a &= \gamma \cdot g(1-P) + (1-\gamma) \cdot g(1-Q) \\ p_0 &= \gamma \cdot g(P) + (1-\gamma) \cdot g(1-Q) \end{aligned}$$

From Theorem 4, it follows that $P > 1/2$ and $Q < 1/2$ is necessary to ensure $p_h < p_a$. Combined with the fact that $g(\cdot)$ is increasing, this implies that $g(Q) < g(1-Q)$ and $g(1-P) < g(P)$. As a result, we obtain $p_1 < p_a < p_0$. It now follows that when $n \to \infty$, the penalties of honest workers with reliability 0 concentrate around p_0, and consequently, they are classified as adversarial whenever the threshold $\theta < p_a$, resulting in a misclassification error of $1-\mu$. However, it should be noted that honest workers classified as adversaries indeed have low reliabilities.

Similar to the proof of Theorem 6 above, the proof of Theorem 10 proceeds in two steps. The first term in the error bound of Theorem 10 comes from Lemma 8 because we immediately declare an error whenever the assignment graph is not locally tree-like. The second term comes from the case when $\mathcal{B}_{w_i,2}$ is indeed a tree by invoking the following concentration result:

Lemma 11 (Concentration of worker penalties) *Suppose that $q < 1$ and $\mathcal{B}_{w_i,2}$ is a tree. Under the generative model, given any reliability value $\hat{\mu} \in (0,1)$ and $\varepsilon > 0$, the penalty assigned to worker w_i concentrates as*

$$\Pr\left(\mathbf{PEN}_i \geq \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \hat{\mu}] + \varepsilon \,\middle|\, \boldsymbol{I}(w_i) = \textit{honest}\right) \leq \exp\left(\frac{-2l\epsilon^2}{(1-1/r)^2}\right) + F(\hat{\mu})$$

and

$$\Pr\left(\mathbf{PEN}_i \leq p_a - \varepsilon \;\middle|\; \boldsymbol{I}(w_i) = \textit{adversarial}\right) \leq \exp\left(\frac{-2l\epsilon^2}{(1-1/r)^2}\right)$$

The proof of the above result is similar to that of Lemma 9. For the case of adversarial workers, we use Hoeffding's argument to establish the concentration. For the case of honest workers, the first term follows directly from Lemma 9 when w_i has reliability $\mu_i > \hat{\mu}$ and the second term is the probability that the reliability $\mu_i \leq \hat{\mu}$.

The above results establish that the soft-penalty algorithm successfully identifies low-reliability honest workers and adversaries adopting the Uniform strategy asymptotically with high probability. Note that all results also extend to the random, normalized variant of the hard-penalty algorithm mentioned in Section 3.3, where the expectation is taken over the generative model *and* the randomized hard-penalty algorithm (see Appendix A.3).

We would like to note that similar results can be derived if it is assumed that honest workers employ the two-coin model (instead of the one-coin model assumed in the preceding analysis). However, the precise error bounds require significantly more notation to explain, without adding too much in terms of insights. Instead, we evaluate the performance of our algorithm for the two-coin model in the numerical experiments and show that it is still able to identify uniform adversaries and low-reliability honest workers (see Section 5.1).

4.2 Hard-penalty algorithm: sophisticated adversary strategies

In the preceding analysis, we focused on common adversary strategies in which adversaries were not intentionally malicious. However, existing studies provide ample evidence for the presence of workers with malicious intent in public crowdsourcing systems, where workers choose which tasks to label and the worker labels are public. These workers are usually hired online by an attacker (Wang et al., 2012) to create fake accounts and manipulate ratings/reviews to alter the aggregate ratings or rankings received by tasks. Specific examples include workers on Digg altering the "popularity" of advertisements and phishing articles (Tran et al., 2009), fake review groups collaboratively targeting products on Amazon (Mukherjee et al., 2012), workers providing fake ratings and reviews to alter the aggregate ratings of restaurants on Yelp (Molavi Kakhki et al., 2013), and malicious crowd behavior in online surveys (Gadiraju et al., 2015). Refer to the recent work by Wang et al. (2014) for more examples. Motivated by these examples, we study settings with *sophisticated adversaries*, who are defined as follows:

Definition 12 (Sophisticated adversaries) *Sophisticated adversaries provide responses with the objective of maximizing the number of tasks whose inferred labels are different from the labels they would otherwise have received from any label aggregation algorithm. They are computationally unbounded and colluding, and they possess knowledge about the labels provided by honest workers; therefore, they can adopt arbitrary response strategies.*

Our definition allows sophisticated adversaries to not just be malicious but also be capable of executing the most complex strategies. In practice, an adversary may adopt feasible strategies with varying complexities depending on the application context and their objectives. Focusing on the most sophisticated adversary makes our analysis broadly applicable,

independent of the application context. Further, we do not restrict the structure of the worker-task assignment graph because unlike in a crowdsourced-classification task, we have no control on which tasks each worker labels.

We first note that existing label aggregation algorithms can have arbitrarily bad performance in the presence of sophisticated adversaries, even if we assume that all honest workers are perfectly reliable:

Lemma 13 (Hardness of recovering true task labels) *Suppose honest workers are perfectly reliable, that is, they always provide the correct response. Let the assignment graph between honest workers and tasks be such that each task receives responses from at most $r > 1$ honest workers. Suppose there are $k > r$ sophisticated adversaries, and each adversary provides the incorrect label on all tasks. Then, both the simple majority and EM (initialized by majority estimates) label aggregation algorithms output the incorrect label for all tasks.*

Note that the above lemma characterizes the performance of label aggregation algorithms, which are designed to infer true task labels as opposed to identifying adversarial workers. Because sophisticated adversaries aim to maximize the number of tasks whose inferred labels are incorrect, the lemma shows that for standard label aggregation algorithms like simple majority and EM, the adversaries can cause arbitrarily bad "damage". Consequently, in our theoretical analysis below, we focus on the accuracy of label aggregation algorithms in the presence of sophisticated adversaries, as opposed to the misclassification rate, as was done in Section 4.1.

Lemma 13 holds for r-right regular graphs, that is, each task receives exactly r honest worker labels, which are commonly used in practice (see the discussion in Section 4.1). Therefore, standard label aggregation algorithms can have very poor accuracy in recovering true task labels in the presence of sophisticated adversaries. In fact, we can actually establish something much stronger—in the presence of sophisticated adversaries, there exists a lower bound on the number of tasks whose true label cannot be inferred correctly (better than random guess) *irrespective* of the label aggregation algorithm used to aggregate the worker responses.

4.2.1 Lower bound on number of tasks that receive incorrect labels

To state our result, we need the following additional notation. We represent any label aggregation algorithm as a *decision rule* $\boldsymbol{R} : \mathcal{L} \to \{-1, +1\}^m$, which maps the observed labeling matrix $\mathcal{L}$ to a set of output labels for each task. Because of the absence of any auxiliary information about the workers or the tasks, the class of decision rules, say $\mathcal{C}$, is invariant to permutations of the identities of workers and/or tasks. More precisely, $\mathcal{C}$ denotes the class of decision rules that satisfy $\boldsymbol{R}(P\mathcal{L}Q) = \boldsymbol{R}(\mathcal{L})Q$ for any $n \times n$ permutation matrix P and $m \times m$ permutation matrix Q. We say that a task is *affected* if a decision rule outputs the incorrect label for the task, and we define the *quality* of a decision rule $\boldsymbol{R}(\cdot)$ as the *worst-case* number of affected tasks over all possible true labelings of the tasks and adversary strategies given a *fixed* set of honest worker responses. Fixing the responses provided by honest workers allows isolation of the effect of the adversary strategy on the accuracy of the decision rule. Considering the worst-case over all possible true task labelings makes the quality metric robust to ground-truth assignments, which are typically application specific.

To formally define the quality, let $\mathcal{B}_H$ denote the subgraph of the worker-task assignment graph restricted to honest workers H and $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ denote the vector of true labels for the tasks. Because the number of affected tasks depends on the actual honest worker responses, we focus on the case when all the honest workers are perfectly reliable, that is, they always provide the correct response. Focusing on completely reliable honest workers allows us to isolate the impact of adversaries because any misidentification is caused by the presence of adversaries. Finally, let $\mathcal{S}_k$ denote the strategy space of $k < |H|$ adversaries, where each strategy $\sigma \in \mathcal{S}_k$ specifies the $k \times m$ response matrix given by the adversaries. Because we do not restrict the adversary strategy in any way, it follows that $\mathcal{S}_k = \{-1, 0, +1\}^{k \times m}$. The *quality* of a decision rule $\boldsymbol{R} \in \mathcal{C}$ is then defined as

$$\mathrm{Aff}(\boldsymbol{R}, \mathcal{B}_H, k) = \max_{\sigma \in \mathcal{S}_k, \boldsymbol{y} \in \{-1,+1\}^m} \left| \left\{ t_j \in \mathcal{T} \colon R_{t_j}^{\boldsymbol{y},\sigma} \neq y_j \right\} \right|$$

where $R_t^{\boldsymbol{y},\sigma} \in \{-1, +1\}$ is the label output by the decision rule $\boldsymbol{R}$ for task t when the true label vector is $\boldsymbol{y}$ and the adversary strategy is σ. Note that our notation $\mathrm{Aff}(\boldsymbol{R}, \mathcal{B}_H, k)$ makes the dependence of the quality measure on the honest worker subgraph $\mathcal{B}_H$ and the number of adversaries k explicit.

We obtain the following result, which establishes a lower bound on the quality of any decision rule:

Theorem 14 (Lower bound on number of affected tasks) *Suppose that $|A| = k$ and all honest workers provide correct responses. Let $\mathrm{PreIm}(\mathcal{T}')$ denote the set of honest workers who label at least one task in $\mathcal{T}' \subseteq \mathcal{T}$. Then, given any honest worker-task assignment graph $\mathcal{B}_H$, there exists an adversary strategy $\sigma^* \in \mathcal{S}_k$ that is independent of any decision rule $\boldsymbol{R} \in \mathcal{C}$, such that*

$$\max_{\boldsymbol{y} \in \{-1,+1\}^m} \mathrm{Aff}(\boldsymbol{R}, \sigma^*, \boldsymbol{y}) \geq L \quad \forall \boldsymbol{R} \in \mathcal{C}, \quad \textit{where}$$

$$L := \frac{1}{2} \cdot \left(\max_{\substack{\mathcal{T}' \subseteq \mathcal{T}: \\ |\mathrm{PreIm}(\mathcal{T}')| \leq k}} |\mathcal{T}'| \right)$$

and $\mathrm{Aff}(\boldsymbol{R}, \sigma^*, \boldsymbol{y})$ denotes the number of affected tasks under adversary strategy σ^*, decision rule $\boldsymbol{R}$, and true label vector $\boldsymbol{y}$ (with the assumption that the maximum over an empty set is zero). In particular, this means that $\mathrm{Aff}(\boldsymbol{R}, \mathcal{B}_H, k) \geq L$ for all decision rules $\boldsymbol{R} \in \mathcal{C}$. In addition, there exist honest worker-task assignment graphs such that σ^* renders the adversaries A indistinguishable from a set of k honest workers, so that a random guess misclassifies $2k$ workers with probability $(1/2)$.

We describe the main idea of the proof. The proof proceeds in two steps: (i) we provide an explicit construction of adversary strategy σ^* that depends only on $\mathcal{B}_H$ and (ii) we show the existence of two possible true labelings $\tilde{\boldsymbol{y}} \neq \boldsymbol{y}$ such that $\boldsymbol{R}$ outputs exactly the same labels in both scenarios. The adversary labeling strategy we construct uses the idea of *indistinguishability*, which captures the fact that by carefully choosing their responses, adversaries can render themselves indistinguishable from honest workers. In the simple case

when there is only one honest worker, the adversary simply flips the response provided by the honest worker, so that each task will have two labels of opposite parity. It can be argued that because there is no other discriminatory information, it is impossible for any decision rule $\boldsymbol{R} \in \mathcal{C}$ to distinguish the honest worker from the adversary and to therefore identify the true label of any task (better than a random guess). We extend this to the general case, where the adversary "targets" at most k honest workers and derives a strategy based on the subgraph of $\mathcal{B}_H$ restricted to the targeted workers. The resultant strategy can be shown to result in incorrectly identified labels for at least L tasks for some ground-truth label assignment.

Note that Theorem 14 holds for *any* honest worker-task assignment graph $\mathcal{B}_H$. This is particularly remarkable given that the analysis of aggregation algorithms becomes extremely complicated for general graphs (a fact observed in previous studies; see Dalvi et al., 2013). The bound L itself depends on the structure of $\mathcal{B}_H$, and therefore, it can be difficult to interpret in general. However, it becomes interpretable for (r, γ, α)-bipartite expanders, as defined next.

Definition 15 (Expanders) *An honest worker-task assignment graph $\mathcal{B}_H = (H \cup \mathcal{T}; E)$, with edges E between the honest workers H and tasks $\mathcal{T}$, is (r, γ, α)-bipartite expander if (i) $\mathcal{B}_H$ is r-right regular, that is, each task is labeled by r honest workers and (ii) for all $\mathcal{T}' \subseteq \mathcal{T}$ such that $|\mathcal{T}'| \leq \gamma |\mathcal{T}|$, the pre-image of $\mathcal{T}'$ satisfies $|\mathrm{PreIm}(\mathcal{T}')| \geq \alpha |\mathcal{T}'|$, where $\mathrm{PreIm}(\mathcal{T}')$ is the set of all honest workers who label at least one task in $\mathcal{T}'$.*

Note that the definition entails that $\alpha \leq r$. We have the following corollary of Theorem 14 when $\mathcal{B}_H$ is (r, γ, α)-bipartite expander.

Corollary 16 (Lower bound for expanders) *Suppose $\mathcal{B}_H$ is (r, γ, α)-bipartite expander. Then, k adversary identities can affect at least L tasks such that $\lfloor \frac{k}{r} \rfloor \leq 2L \leq \lceil \frac{k}{\alpha} \rceil$, provided $\lceil \frac{k}{\alpha} \rceil + 1 < \gamma \cdot |\mathcal{T}|$. Furthermore, given any constant r, there exists $\gamma > 0$ such that a uniformly random $\mathcal{B}_H$ is $(r, \gamma, r-2)$-bipartite expander with probability at least $1/2$, in which case the lower bound $L = \frac{1}{2} \lceil \frac{k}{r-2} \rceil$.*

The proof is provided in Appendix A.5. The above statement says that if the honest worker-task assignment graph $\mathcal{B}_H$ is constructed randomly, then k adversary identities can affect at least $\frac{1}{2} \lfloor \frac{k}{r} \rfloor$ tasks. The bound implies that the ability of the adversaries to affect tasks increases linearly as the number of identities k increases. Further, the damage that k adversaries can do decreases inversely with the number of honest workers r who provide labels for each task. Both implications are intuitive. As can be seen from the proof, the lower bound $\frac{1}{2} \lfloor \frac{k}{r} \rfloor$ on L in Corollary 16 holds for all r-right regular graphs, even if they are not expanders.

Having established the above lower bound and in light of Lemma 13, the natural question to ask is as follows: does there exist a label aggregation algorithm for which we can prove an upper bound on the number of affected tasks, irrespective of the strategy employed by the sophisticated adversaries? Below, we show such a label aggregation algorithm that is a natural extension of the hard-penalty algorithm.

4.2.2 Accuracy of hard-penalty algorithm

We introduce the *penalty-based label aggregation* algorithm (see Algorithm 3) for our analysis, which is a natural extension of the hard-penalty algorithm to also perform label aggregation:

Algorithm 3 penalty-based label aggregation

1: **Input:** W, $\mathcal{T}$, and $\mathcal{L}$
2: Perform steps 2 and 3 of the hard-penalty algorithm
3: For each task t_j, let $w_{t_j^+}, w_{t_j^-}$ be worker nodes that task nodes t_j^+, t_j^- are respectively mapped to in optimal semi-matching $\mathcal{M}$ in Step 2
4: **Output**

$$\hat{y}_j = \begin{cases} +1 & \text{if } deg_{\mathcal{M}}(w_{t_j^+}) < deg_{\mathcal{M}}(w_{t_j^-}) \\ -1 & \text{if } deg_{\mathcal{M}}(w_{t_j^+}) > deg_{\mathcal{M}}(w_{t_j^-}) \\ \leftarrow \{-1,+1\} & \text{otherwise} \end{cases}$$

(here $\hat{y}_j$ refers to the output label for task t_j and $\leftarrow \{-1,+1\}$ means that $\hat{y}_j$ is drawn uniformly at random from $\{-1,+1\}$)

For our analysis, we consider the following model for honest worker responses that is based on the *spammer-hammer* model popularly used in previous studies (Karger et al., 2011; Liu et al., 2012; Ok et al., 2016; Khetan and Oh, 2016): $0 \leq \varepsilon < 1$ fraction of honest workers are "spammers," that is, they make mistakes in their responses, and the remaining $1-\varepsilon$ fraction are "hammers," that is, they are perfectly reliable and always provide the correct response. In the theorem below, $\mathcal{B}_H^{cs}$ refers to the bipartite graph created as follows: (i) each honest worker h is represented by a node on the left; (ii) each task $t_j \in \mathcal{T}$ is represented by (at most) two nodes on the right, t_j^+ and t_j^-; and (iii) add the edge (h, t_j^+) (resp. (h, t_j^-)) if honest worker h labels task t_j as $+1$ (resp. -1). Then, we can show the following:

Theorem 17 (Accuracy of penalty-based label aggregation) *Suppose that $|A| = k$ and let $\varepsilon \in [0,1)$ denote the fraction of honest workers who make mistakes in their responses. Furthermore, let $d_1 \geq d_2 \geq \cdots \geq d_{|H|}$ denote the degrees of honest workers in the optimal semi-matching on $\mathcal{B}_H^{cs}$. For any true labeling $\boldsymbol{y}$ of the tasks and under Algorithm 3 (with the convention that $d_i = 0$ for $i > |H|$):*

1. *If $\varepsilon = 0$, there exists an adversary strategy σ^* such that the number of affected tasks is at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$*
2. *Assuming that each task receives at least one correct response from the honest workers, no adversary strategy can affect more than U tasks where*
   (a) $U = \sum_{i=1}^{k+\varepsilon\cdot|H|} d_i$, *when at most one adversary provides correct responses*
   (b) $U = \sum_{i=1}^{2k+\varepsilon\cdot|H|} d_i$, *in the general case*

A few remarks are in order. First, it can be shown that for optimal semi-matchings, the degree sequence $d_1, d_2, \ldots, d_{|H|}$ is unique (see the proof in Appendix A.7), and therefore, the bounds in the theorem above are uniquely defined given $\mathcal{B}_H^{cs}$. The result of Theorem 17 provides both a lower and upper bound for the number of tasks that can be affected by k adversaries under the penalty-based label aggregation algorithm, irrespective of the adversary strategy. This is remarkable, especially because we established that existing label aggregation algorithms can be arbitrarily bad (Lemma 13). Assuming honest workers are always correct, that is, $\varepsilon = 0$, our characterization is reasonably tight when all but (at most) one adversary provide incorrect responses. In this case, the gap between the upper and a constant factor of the lower bound is d_k, which can be "small" for large enough k. However, our characterization is loose in the general case when adversaries can provide arbitrary responses. Here, the gap is $\sum_{i=k}^{2k} d_i$; we attribute this to our proof technique and conjecture that the upper bound of $\sum_{i=1}^{k} d_i$ also applies to the more general case. When $\varepsilon > 0$, the upper bound U increases because there are more incorrect responses and, in turn, the scope for adversaries to affect a larger number of tasks.

One might wonder whether we could perform a similar analysis for the case when honest workers follow the one-coin model. Given the complexity of analyzing optimal semi-matchings, our current proof technique does not readily extend to this scenario, and therefore, there is an opportunity to improve the analysis in future work.

Optimality of penalty-based label aggregation. We now compare the upper bound U in Theorem 17 to the lower bound L in Theorem 14 in the case when honest workers are perfectly reliable, that is, $\varepsilon = 0$. We show that (see Appendix A.8) when the degrees $d_1, d_2, \ldots, d_{|H|}$ are all distinct, $L \geq \frac{1}{2}\sum_{i=1}^{k-1} d_i$. Combined with Theorem 14, this shows that k adversaries can affect at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$ tasks *irrespective* of the label aggregation algorithm used to aggregate the worker responses. From Theorem 17, we also have that under the penalty-based label aggregation algorithm, k adversaries can affect at most $U = \sum_{i=1}^{2k} d_i \leq 3(\sum_{i=1}^{k-1} d_i)$ tasks (as long as $k \geq 2$). Therefore, our algorithm achieves constant factor optimality in recovering the true labels of tasks *irrespective* of the adversary's strategy.

5. Numerical analysis

We conducted two empirical studies to demonstrate the practical value of our methods. The first study illustrates a concrete real-world application of our methodology. By using standard crowdsourcing datasets (as described below), it shows that filtering out the adversaries identified by our methods allows existing label aggregation algorithms to infer true task labels more accurately. Such improvements in accuracy by *discarding* labels from certain workers suggests that their label patterns do not conform to standard probabilistic assumptions. Although not illustrated in our study, instead of being filtered out, the adversary labels may also be used by fitting a model different from that of honest workers in applications where such assumptions are reasonable. The second study is designed to assess the ability of our methods to successfully identify adversaries and low-reliability honest workers. It is a simulation study in which we injected a standard crowdsourcing task with "spammers" (workers who label a task $+1$ and -1 with probability $1/2$ each irrespective of the true label) and workers adopting the Uniform strategy. It demonstrates that both soft- and hard-penalty algorithms successfully identify adversaries and low-reliability

honest workers when the worker-task assignment graph has a power-law degree distribution for workers and tasks, thus complementing the results in Section 4.1 which focused on (l, r)-regular worker-task assignment graphs.

For the purposes of our studies, we focused on the following six label aggregation algorithms: (a) simple majority algorithm MV; (b) EM algorithm for the two-coin model (Raykar and Yu, 2012); (c) KOS algorithm (Karger et al., 2014); (d) KOS(NORM), a normalized variant of KOS in which messages are scaled by the corresponding worker and task degrees to account for non-regular node degrees in the assignment graph (see Appendix B for the precise description); (e) SPEC-EM, the spectral EM algorithm of Zhang et al. (2014); and (f) MMCE, the regularized minimax conditional entropy approach of Zhou et al. (2015). We implemented both variants of our reputation algorithm: (a) soft-penalty (SOFT) and (b) hard-penalty (HARD). As removing workers alters the penalties of the remaining workers, we filtered the workers iteratively. In each iteration, we recomputed the penalties of the remaining workers, removed the worker with the highest penalty, and repeated until a prespecified number of workers was removed. [7]

Finally, as mentioned in Section 4.1, we implemented two variants of the soft-penalty algorithm: one in which non-conflict tasks are dropped for assigning worker penalties and another in which they are retained. The results were essentially the same, so we only report the results for the variant in which the non-conflict tasks were dropped.

Dataset	Workers	Tasks	Responses
rte	164	800	8000
temp	76	462	4620
stage2	68	711	2035
task2	386	2269	12435
tweets	66	1000	4977

Table 1: Summary of real datasets used in the experiments

5.1 Accuracy improvements on real-world crowdsourcing datasets

We focused on the following standard datasets:

- `stage2` and `task2`: consisting of a collection of topic-document pairs labeled as relevant or non-relevant by workers on Amazon Mechanical Turk (see Tang and Lease, 2011). These datasets were collected as part of the TREC 2011 crowdsourcing track.
- `rte` and `temp`: consisting of annotations by Amazon Mechanical Turk workers for different natural language processing (NLP) tasks. `rte` consists of binary judgments for textual entailment (whether one sentence can be inferred from another) and `temp` for temporal ordering of events (see Snow et al., 2008).
- `tweets`: consisting of sentiment (positive or negative) labels for 1000 tweets (see Mozafari et al., 2014).

7. The performance was similar for the non-iterative variant discussed in Section 3—we report the results for the iterative version because it had marginally better performance.

As inferring the reliabilities of workers who labeled very few tasks [8] is difficult, we preprocessed the datasets to remove all workers who labeled less than three tasks. Table 1 shows summary statistics of the datasets after our preprocessing.

Table 2 reports the accuracies of various label aggregation algorithms for inferring true task labels. For each benchmark label aggregation algorithm, the column BASE reports the accuracy of the algorithm in isolation. The columns SOFT and HARD report the best accuracy of the algorithms for filtering $k = 1, 2, \ldots, 10$ workers using the soft- and hard-penalty algorithms, respectively; the numbers in parentheses indicate the k value for which we observed the best performance. [9]

Algorithm	rte			temp			stage2			task2			tweets		
	BASE	SOFT	HARD	BASE	SOFT	HARD	BASE	SOFT	HARD	BASE	SOFT	HARD	BASE	SOFT	HARD
MV	91.9	92.1(7)	**92.5(3)**	93.9	93.9	**94.4(5)**	74.3	75.4(1)	**80.5(2)*****	64.2	64.2	**67.8(10)*****	69.6	69.8(4)	**73.3(1)*****
EM	93.0	93.0	**93.3(5)**	94.1	94.1	94.1	70.2	76.8(4)	**81.2(6)*****	67.0	67.1(6)	**68.6(9)*****	71.2	71.2	**71.7(1)**
KOS	49.7	89.0(10)	**91.6(10)*****	56.9	69.3(4)	**93.7(3)*****	74.5	74.7(7)	**75.1(2)**	57.4	57.4	**65.6(10)*****	65.8	66.0(4)	**70.5(1)*****
KOS(NORM)	91.3	92.6(5)	**93.1(6)****	93.9	94.4(7)	**94.4(1)**	75.5	75.5	**78.2(2)***	58.3	58.9(8)	**67.7(9)*****	68.7	68.7	**71.0(2)*****
SPEC-EM	90.1	92.4(8)	**92.8(6)**	56.1	94.2(7)	**94.4(5)**	82.8	82.8	82.8	56.1	56.1	**57.6(10)*****	67.0	67.0	**69.0(1)*****
MMCE	92.5	92.9(9)	**93.2(3)**	94.4	94.4	94.4	57.0	58.5(5)	**73.3(10)*****	66.1	66.1	**66.9(10)**	72.7	72.7	72.7

Table 2: **Percentage accuracy** in recovering true labels of benchmark algorithms in isolation and when combined with our modified reputation algorithms. For each benchmark, the best-performing combination is highlighted in bold. The number in parentheses represents the number of workers filtered by our reputation algorithm (an absence indicates that no performance improvement was achieved while removing upto 10 workers with the highest penalties). The p-values, according to a two-sided paired t-test, are denoted as * $p < 0.1$, ** $p < 0.05$, and *** $p < 0.01$.

The key conclusion we draw is that filtering out workers flagged by our algorithms as adversaries boosts the predictive accuracy of state-of-the-art aggregation algorithms significantly across the datasets: the average improvement in accuracy for MV is 3.7%, EM is 3.4%, KOS is 30.2%, KOS(NORM) is 4.4%, SPEC-EM is 12.6%, and MMCE is 4.7% when using the hard-penalty algorithm. The improvement is large for KOS because it is designed for regular graphs (with all workers having the same degree and all tasks having the same degree) and suffers in performance on real-world graphs that are not regular. Second, we note that our methods can boost the performance of the MV and KOS algorithms to the level of the popular EM algorithm. The MV algorithm is simple to implement, and the KOS algorithm is designed for scenarios where the underlying assignment graph is random (l, r)-regular and has strong theoretical guarantees and robustness to different initializations (Karger et al., 2014). Our results suggest that implementing the MV and KOS algorithms in conjunction with our reputation algorithms can allow us to obtain their respective simplicity and robustness along with strong practical performance (even for irregular graphs) comparable

8. The datasets `stage2`, `task2`, and `tweets` contain several workers who provided responses for only a single task.
9. The performance was robust to the choice of k. We matched or improved the accuracy of the underlying label aggregation algorithm in 66% and 70% of cases on average for the soft- and hard-penalty algorithm, respectively.

to that of the EM algorithm. [10] Finally, because discarding the labels of certain workers improves the predictive accuracy, our results suggest that standard probabilistic models (including the two-coin model) are insufficient to capture the labeling patterns of workers in real-world datasets.

To gain insights into the types of workers identified by our algorithms, we conducted a qualitative analysis of the labeling patterns of the workers that were filtered out. We observed the following key types:

1. Workers who labeled at least 10 tasks, of which more than 95% had the same label. For instance, our algorithms detected six such workers in the `temp` dataset, five in the `task2` dataset, and one in the `tweets` dataset. For the `stage2` dataset, we detected two workers who gave all $+1$ labels and one worker who gave all but a single response as -1; it should be noted that the empirically calculated prevalence γ of $+1$ tasks in the `stage2` dataset was 0.83, potentially suggesting that the two workers who gave all $+1$ labels were adopting "smart" strategies.

2. Workers who provide labels independent of true task labels. For instance, we detected four such workers in the `rte` dataset, seven workers in the `temp` dataset, seven in the `task2` dataset, three in the `stage2` dataset, and one in the `tweets` dataset, whose label patterns are such that the empirical fractions $\hat{\alpha}$ and $\hat{\beta}$ of correct responses among the tasks with true labels $+1$ and -1, respectively, satisfy $|\hat{\alpha}+\hat{\beta}-1| \leq 0.05$. Raykar and Yu (2012) showed that such workers effectively assign a label of $+1$ with probability $\hat{\alpha}$ and -1 with probability $1-\hat{\alpha}$ independent of the true task label.

3. Workers with skewed reliabilities. Such workers were accurate on tasks with one type of true label, say, $+1$, but not on others, say, tasks with true label -1. Such label patterns of workers may be indicative of tasks that require subjective assessments. For instance, we found four workers in the `tweets` dataset and two workers in `stage2` dataset that had skewed reliabilities. In the `tweets` dataset, workers were asked to rate the sentiment of a tweet as being positive or negative. As the notion of tweet sentiment can be subjective, workers with biased views of the sentiment make systematic errors on one type of task. A similar explanation applies to the `stage2` dataset, in which workers were asked to label a topic-document pair as relevant or not, requiring a potentially subjective assessment. See Kamar et al. (2015) for more examples.

In summary, our reputation algorithms are successful in identifying adversarial workers adopting a broad set of strategies. Furthermore, although not reported, the empirical reliabilities of the workers filtered out using our algorithms were on average *lower* than those of unfiltered workers. This suggests that the labels our algorithms discard are from low-reliability workers who provide little to no information about the true task labels; this explains the improved accuracy we obtain.

10. Note that the EM algorithm can also have theoretical guarantees with appropriate initializations (see Gao and Zhou, 2016 and Zhang et al. 2014).

5.2 Identifying low-reliability honest workers and adversaries

We use a simulation study to show that our reputation algorithms successfully identify low-reliability honest workers and adversaries when the worker-task assignment graphs have power-law degree distributions for the worker and task nodes. Such graph structures are common in many real-world crowdsourcing scenarios (Franklin et al., 2011). The results of the simulation study complement the theoretical results presented in Section 4.1 for (l, r)-regular assignment graphs, which show that the soft-penalty algorithm successfully identifies low-reliability honest workers and adversaries adopting the Uniform strategy.

For our study, we used the following broad procedure: (a) generate a random crowdsourcing instance from the ground-truth model, (b) generate synthetic responses from the workers for a sample of tasks, (c) filter out workers with the highest penalties according to our reputation algorithms, and (d) compute the *precision*, that is, the fraction of filtered-out workers who are adversarial and who have low empirical reliabilities.

Setup of study. We considered a total of $n = 100$ workers. The probability q that a worker is honest was set to 0.7; therefore, on average, there are 30 adversaries among the 100 workers. The prevalence γ of $+1$ tasks was set to 0.5. We sampled worker degrees according to a power-law distribution (with exponent $a = 2.5$) with the minimum degree equal to 5, and then, we used the Python networkx library (Hagberg et al., 2008) to generate the worker-task assignment graph.[11] Note that the number of tasks m is determined from the total number of workers and the sampled worker degrees.

As the worker degrees are skewed, the performance of the algorithms is influenced by the adversary degrees. To capture this, we considered two scenarios: (a) adversaries have high degrees and (b) adversaries have low degrees. To ensure that adversaries on average have high degrees, we set worker w in the sampled worker-task assignment graph to be honest with probability $q_w = q \cdot (d_{\max} - d_w)/(d_{\max} - d_{\text{avg}})$ and to be adversarial with probability $1 - q_w$, where d_w is the degree of worker w and $d_{\max}$, $d_{\min}$, and d_{avg} are the maximum, minimum, and average degrees, respectively. Similarly, to ensure that adversaries on average have lower degrees, we set $q_w = q \cdot (d_w - d_{\min})/(d_{\text{avg}} - d_{\min})$. See Appendix B for details.

For each scenario, after the worker-task assignment graph and worker identities were sampled, we generated the crowdsourcing instances as follows: (a) set the true label y_j of each task t_j to be $+1$ with probability $1/2$ and -1 with probability $1/2$; (b) for each honest worker w, sample its reliability μ_w u.a.r from the interval $[0.8, 1.0)$, and set its response to each assigned task t_j to be the true label y_j with probability μ_w and $-y_j$ with probability $1-\mu_w$; and (c) generate responses from adversarial workers according to the chosen strategy. We focused on two adversary strategies: (a) Spammer—label each task $+1$ or -1 with prob. $1/2$ and (b) Uniform—label every assigned task $+1$. The first strategy reflects the setting of Theorem 3 because it is captured by the one-coin model, and the second strategy reflects the setting of Theorem 4.

Results. Table 3 reports the *precisions* of the SOFT and HARD algorithms when (iteratively) removing 10 workers. The precision is defined as the fraction of filtered-out workers who are either adversarial or honest with empirical reliabilities less than 0.85 (which is less

11. Specifically, we used the following function: `https://networkx.github.io/documentation/networkx-1.10/reference/generated/networkx.algorithms.bipartite.generators.preferential_attachment_graph.html`.

	Spammer		Uniform	
	Low Deg	High Deg	Low Deg	High Deg
SOFT	**90.03**	88.70	**77.13**	69.57
HARD	89.00	**93.83**	73.43	**76.73**

Table 3: The *precisions* of the soft- and hard-penalty algorithms in identifying low-reliability honest workers and adversaries when filtering 10 workers. The rows SOFT and HARD correspond to the soft- and hard-penalty algorithms, respectively. The columns Spammer and Uniform correspond to the two adversary strategies, and for each adversary strategy, the columns "Low Deg" and "High Deg" refer to the scenarios in which low- and high-degree workers, respectively, are more likely to be adversaries. Refer to the text for more details.

than the mean $\mu = 0.9$ of the honest worker reliability distribution), where the empirical reliability of an honest worker is equal to the fraction of its responses that were correct. The table reports the precision of the algorithms for the two adversary strategies, Spammer and Uniform, under the two scenarios: Low Deg, in which low-degree workers are more likely to be adversaries, and High Deg, in which high-degree workers are more likely to be adversaries. For each combination of adversary strategy and scenario, the reported numbers are the precision values averaged over 300 randomly generated instances.

We draw the following key conclusions. First, our algorithms have precision values $> 69\%$ in all scenarios, indicating that they are successful in identifying adversaries and low-reliability honest workers when the worker and task degrees have a power-law distribution. This finding complements the results of Theorem 6 and Theorem 10, which establish a similar result when the worker-task assignment graph is (l, r)-regular. Second, our results offer insights into the settings under which the soft- or hard-penalty algorithm is appropriate. Generally, we observe that the hard-penalty algorithm is more appropriate when the adversaries have higher degrees, whereas the soft-penalty algorithm is more appropriate when the adversaries have lower degrees. We also note that when the adversaries have labeling patterns (such as Spammer) that are probabilistically similar to those of honest workers, the soft-penalty algorithm has a performance comparable to that of the hard-penalty algorithm even when the adversaries have high degrees.

6. Conclusions

This study investigated the problem of identifying a broad class of adversarial workers in crowdsourced systems, when the population of workers consists of a mixture of honest workers and adversaries. Honest workers may be reliable or unreliable, and they provide labels according to a well-defined probabilistic model. The adversaries adopt strategies different from those of honest workers, whether probabilistic or not. Under this setting, we make algorithmic, theoretical, and empirical contributions. The key algorithmic contribution is the design of two reputation-based algorithms—soft-penalty and hard-penalty algorithms—that analyze workers' label patterns and assign a reputation score to each worker that indicate

the degree to which the labeling pattern is a statistical outlier. Under standard probabilistic assumptions, we show that the reputation scores assigned by the soft-penalty algorithm are consistent with worker reliabilities (probability that a worker provides the true task label) when there are no adversaries. We also show that under appropriate conditions, the soft-penalty algorithm can asymptotically separate adversaries from honest workers when adversaries adopt deterministic strategies. When adversaries are sophisticated, we derive a lower bound for the number of tasks that k adversaries can affect (true label cannot be inferred better than a random guess) and a corresponding upper bound for the number of affected tasks under the hard-penalty algorithm. Empirically, we show that our algorithm can be used to significantly enhance the accuracy of existing label aggregation algorithms in real-world crowdsourcing datasets.

To the best of our knowledge, our work is the first to consider general worker strategies in crowdsourced labeling tasks. Our work opens the doors for several exciting future directions. Both of our penalty-based algorithms assume that the task labels are binary; analyzing natural extensions of our algorithms to multi-class settings is an interesting direction. Because our algorithm allows the identification of adversarial workers, it could be combined with adaptive techniques to recruit more workers (Ramesh et al., 2012) and identify the right workers (Li et al., 2014). Finally, applying our outlier detection technique to ensemble learning approaches, in which outputs from multiple learning algorithms are combined, could be a promising future direction; indeed there has already been some work in this space (Wang and Yeung, 2014).

Acknowledgments

Ashwin Venkataraman is partially supported by NSF CAREER grant CMMI 1454310. We gratefully acknowledge the thoughtful and detailed feedback of the anonymous referees which led to a substantial improvement in the quality, organization, and completeness of the paper. The authors would also like to thank Sewoong Oh and Kyunghyun Cho for insightful discussions that helped improve the paper.

Appendix A. Proofs of the theorems

In this section, we describe the proofs of all our theoretical results, starting with the results stated in section 4.1.1.

A.1 Analysis of Expected Penalties under the soft-penalty algorithm

First, we introduce some additional notation. Let $\mathbb{P}_{ih}[E]$ (resp. $\mathbb{P}_{ia}[E]$) denote the probability of some event E conditioned on the fact that worker w_i is honest (resp. adversarial). Similarly, $\mathbb{E}_{ih}[\boldsymbol{X}]$ (resp. $\mathbb{E}_{ia}[\boldsymbol{X}]$) denotes the conditional expectation of the random variable $\boldsymbol{X}$ given the event that worker w_i is honest (resp. adversarial). Also, let $f(\cdot)$ denote the probability density function (PDF) of the worker reliability distribution. Recall the definitions $P := q\mu + (1-q)$, $Q := 1 - q\mu$, and the function $g(x) := \frac{1-x^r}{r(1-x)}$, and note that $g(\cdot)$ is *strictly increasing* on $(0,1)$. Let $\mathbf{1}[A]$ denote the indicator variable taking value 1 if A is true and 0 otherwise. Let $\boldsymbol{D}_j^+$ (resp. $\boldsymbol{D}_j^-$) denote the number of workers who label task t_j as $+1$ (resp -1). In other words, $\boldsymbol{D}_j^+ = \sum_{w_i \in W_j} \mathbf{1}[\boldsymbol{W}_{ij} = +1]$ and $\boldsymbol{D}_j^- = \sum_{w_i \in W_j} \mathbf{1}[\boldsymbol{W}_{ij} = -1]$, where recall that W_j denotes the set of workers who labeled task t_j. Here, $\boldsymbol{W}_{ij}$ represents the label assigned by worker w_i to task t_j. Finally, let $\mathrm{Bin}(n,p)$ denote the Binomial distribution with parameters n and p.

We begin by proving some important lemmas that will be repeatedly used in the proofs below.

Lemma 18 *Under the generative model in section 4.1, the probability that worker w_i provides response $+1$ for a task t_j is given by*

$$\Pr[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = +1] = P$$
$$\Pr[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = -1] = Q$$

Furthermore, conditioned on $\boldsymbol{Y}_j = +1$ or $\boldsymbol{Y}_j = -1$, the random variables $\mathbf{1}[\boldsymbol{W}_{ij} = +1]$ are i.i.d. for all $w_i \in W_j$. As a result, it follows that $\boldsymbol{D}_j^+ \,\big|\, \boldsymbol{Y}_j = +1 \sim \mathrm{Bin}(r,P)$ and $\boldsymbol{D}_j^+ \,\big|\, \boldsymbol{Y}_j = -1 \sim \mathrm{Bin}(r,Q)$.

Proof Consider the case when $\boldsymbol{Y}_j = +1$:

$$\begin{aligned}
&\Pr[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = +1] \\
&= \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = +1] \cdot \Pr[w_i \text{ is honest} \,\big|\, \boldsymbol{Y}_j = +1] \\
&+ \mathbb{P}_{ia}[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = +1] \cdot \Pr[w_i \text{ is adversarial} \,\big|\, \boldsymbol{Y}_j = +1] \\
&= \left(\int_0^1 \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = +1] \cdot f(\mu_i) d\mu_i \right) q + 1 \cdot (1-q) \\
&= \left(\int_0^1 \mu_i f(\mu_i) d\mu_i \right) q + (1-q) \\
&= q\mu + (1-q) = P,
\end{aligned}$$

where the first term of the second equality follows from the law of total expectation, the second term because the adversary always labels a task $+1$, and the third equality follows from the definition of honest worker reliability $\boldsymbol{M}_i$.

Furthermore, it follows from our generative process that conditioned on $\boldsymbol{Y}_j = +1$, the labels from any two workers $w_i \neq w_{i'}$ for task t_j are generated independently. Therefore, we have shown that, conditioned on $\boldsymbol{Y}_j = +1$, the random variables $\mathbf{1}[\boldsymbol{W}_{ij} = +1]$ are independently and identically distributed with probability P of taking the value 1. Because $\boldsymbol{D}_j^+ = \sum_{w_i \in W_j} \mathbf{1}[\boldsymbol{W}_{ij} = +1]$ and $|W_j| = r$, it follows that $\boldsymbol{D}_j^+ \,\big|\, \boldsymbol{Y}_j = +1$ is a sum of r i.i.d. Bernoulli random variables and, hence, $\boldsymbol{D}_j^+ \,\big|\, \boldsymbol{Y}_j = +1 \sim \mathrm{Bin}(r,P)$.

A similar argument shows that $\Pr[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = -1] = Q$ and consequently $\boldsymbol{D}_j^+ \,\big|\, \boldsymbol{Y}_j = -1 \sim \mathrm{Bin}(r,Q)$. ■

Lemma 19 *Under the generative model in section 4.1, suppose that worker w_i is honest with a sampled reliability μ_i and let $\boldsymbol{S}_{ij}$ denote the penalty received by w_i from task $t_j \in \mathcal{T}_i$ in the* soft-penalty *algorithm. Then, we can show*

$$\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i] = \gamma\Big(\mu_i \cdot g(1-P) + (1-\mu_i) \cdot g(P)\Big) + (1-\gamma)\Big(\mu_i \cdot g(Q) + (1-\mu_i) \cdot g(1-Q)\Big)$$

Similarly, if worker w_i is adversarial, then we have:

$$\mathbb{E}_{ia}[\boldsymbol{S}_{ij}] = \gamma \cdot g(1-P) + (1-\gamma) \cdot g(1-Q)$$

Proof We begin with the case when w_i is honest. Using the law of total expectation, we have:

$$\begin{aligned}
&\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i] \\
&= \sum_{v_1, v_2 \in \{-1,+1\}} \mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = v_1; \boldsymbol{Y}_j = v_2] \cdot \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = v_1; \boldsymbol{Y}_j = v_2 \,\big|\, \boldsymbol{M}_i = \mu_i]
\end{aligned}$$

We first consider the case when $v_1 = +1$ and $v_2 = +1$. In this case, because worker w_i assigns label $+1$ to task t_j (i.e., $\boldsymbol{W}_{ij} = +1$), the penalty $\boldsymbol{S}_{ij}$ assigned by the task to the worker is equal to $1/\boldsymbol{D}_j^+$. Furthermore, because $\boldsymbol{Y}_j = +1$ and $\mathbf{1}[\boldsymbol{W}_{ij} = +1] = 1$, it follows from the arguments in Lemma 18 that $\boldsymbol{D}_j^+ - 1$ is distributed as $\mathrm{Bin}(r-1,P)$. We can now

write

$$\begin{aligned}
&\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1] \\
=&\mathbb{E}_{ih}\left[1/\boldsymbol{D}_j^+ \,\middle|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1\right] \\
=&\sum_{k=0}^{r-1} \frac{1}{1+k}\binom{r-1}{k} \cdot P^k(1-P)^{r-1-k} \\
=&\frac{1}{r} \cdot \sum_{k=0}^{r-1} \binom{r}{k+1} \cdot P^k(1-P)^{r-1-k} \\
=&\frac{1}{rP} \cdot \sum_{k'=1}^{r} \binom{r}{k'} \cdot P^{k'}(1-P)^{r-k'} \\
=&\frac{1-(1-P)^r}{rP} = g(1-P),
\end{aligned}$$

where the third equality follows because $\frac{1}{k+1} \cdot \binom{r-1}{k} = \frac{1}{r} \cdot \binom{r}{k+1}$, the fifth equality follows because $\sum_{k'=0}^{r} \binom{r}{k'} \cdot P^{k'}(1-P)^{r-k'} = 1$, and the last equality follows from the definition of the function $g(\cdot)$.

Furthermore, $\mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1, \boldsymbol{Y}_j = +1 \,\big|\, \boldsymbol{M}_i = \mu_i] = \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{Y}_j = +1; \boldsymbol{M}_i = \mu_i] \cdot \mathbb{P}_{ih}[\boldsymbol{Y}_j = +1 \,\big|\, \boldsymbol{M}_i = \mu_i] = \mu_i \cdot \gamma$. We have thus shown that

$$\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1] \cdot \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1 \,\big|\, \boldsymbol{M}_i = \mu_i] = g(1-P) \cdot \mu_i\gamma$$

The case when $v_1 = -1$ and $v_2 = +1$ (i.e., $\boldsymbol{W}_{ij} = -1$ and $\boldsymbol{Y}_j = +1$) follows a symmetric argument. In particular, because worker w_i assigns the label -1 to task t_j, the penalty $\boldsymbol{S}_{ij}$ that is assigned is equal to $1/\boldsymbol{D}_j^-$. Furthermore, it follows from the arguments in Lemma 18 that $\boldsymbol{D}_j^- - 1$ is distributed as $\mathrm{Bin}(r-1, 1-P)$. It now follows from a symmetric argument (by replacing P by $1-P$ above) that $\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = -1; \boldsymbol{Y}_j = +1] = g(P)$ and $\mathbb{P}_{ih}[\boldsymbol{Y}_j = +1; \boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{M}_i = \mu_i] = \gamma \cdot (1-\mu_i)$. We have thus shown that

$$\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = -1; \boldsymbol{Y}_j = +1] \cdot \mathbb{P}_{ih}[\boldsymbol{W}_{ij} = -1; \boldsymbol{Y}_j = +1 \,\big|\, \boldsymbol{M}_i = \mu_i] = g(P) \cdot (1-\mu_i)\gamma$$

Replacing P by Q, μ_i by $1-\mu_i$, and γ by $1-\gamma$ yields the expressions for the other two cases. In particular,

$$\begin{aligned}
&\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = -1] \cdot \mathbb{P}_{ih}[\boldsymbol{Y}_j = -1; \boldsymbol{W}_{ij} = +1 \,\big|\, \boldsymbol{M}_i = \mu_i] \\
&= g(1-Q) \cdot (1-\gamma) \cdot (1-\mu_i) \\
&\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{W}_{ij} = -1; \boldsymbol{Y}_j = -1] \cdot \mathbb{P}_{ih}[\boldsymbol{Y}_j = -1; \boldsymbol{W}_{ij} = -1 \,\big|\, \boldsymbol{M}_i = \mu_i] \\
&= g(Q) \cdot (1-\gamma) \cdot \mu_i
\end{aligned}$$

Combining the above, we obtain

$$\mathbb{E}_{ih}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i] = \gamma\Big(\mu_i \cdot g(1-P) + (1-\mu_i) \cdot g(P)\Big) + (1-\gamma)\Big(\mu_i \cdot g(Q) + (1-\mu_i) \cdot g(1-Q)\Big)$$

We now consider the case when worker w_i is an adversary. Because adversaries always assign the label $+1$, we need to consider only two cases: $\boldsymbol{Y}_j = +1$ and $\boldsymbol{Y}_j = -1$. The conditional expectations of the penalties in both cases are identical to those above: $\mathbb{E}_{ia}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1] = g(1-P)$ and $\mathbb{E}_{ia}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = -1] = g(1-Q)$. Further, we have $\mathbb{P}_{ia}[\boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = +1] = \gamma$ and $\mathbb{P}_{ia}[\boldsymbol{W}_{ij} = +1; \boldsymbol{Y}_j = -1] = 1-\gamma$. Combining these, we obtain

$$\mathbb{E}_{ia}[\boldsymbol{S}_{ij}] = \gamma \cdot g(1-P) + (1-\gamma) \cdot g(1-Q)$$

The result of the lemma now follows. ■

We are now ready to prove the theorems.

A.1.1 Proof of Theorem 3

First, note that if $q = 1$ then $P = \mu$ and $Q = 1-\mu$. Also, since all workers are honest, we remove the explicit conditioning on worker w_i being honest. Then the expected penalty allocated to a worker w_i with a reliability μ_i:

$$\begin{aligned}
&\mathbb{E}[\mathbf{PEN}_i \,\big|\, \boldsymbol{M}_i = \mu_i] = \frac{1}{l}\sum_{j \in \mathcal{T}_i} \mathbb{E}[\boldsymbol{S}_{ij} \,\big|\, \boldsymbol{M}_i = \mu_i] \\
&= \gamma\Big(\mu_i \cdot g(1-P) + (1-\mu_i) \cdot g(P)\Big) + (1-\gamma)\Big(\mu_i \cdot g(Q) + (1-\mu_i) \cdot g(1-Q)\Big) \\
&\text{(using lemma 19)} \\
&= \gamma\Big(\mu_i \cdot g(1-\mu) + (1-\mu_i) \cdot g(\mu)\Big) + (1-\gamma)\Big(\mu_i \cdot g(1-\mu) + (1-\mu_i) \cdot g(\mu)\Big) \\
&= g(\mu) - \mu_i \cdot (g(\mu) - g(1-\mu))
\end{aligned}$$

Since $g(\cdot)$ is strictly increasing on $(0,1)$ and $\mu > \frac{1}{2}$, we have that $\mu > 1-\mu$ and consequently $g(\mu) - g(1-\mu) > 0$. The claim then follows.

A.1.2 Proof of Theorem 4

For an honest worker w_i, we have that

$$
\begin{aligned}
&p_h := \mathbb{E}_{ih}[\mathbf{PEN}_i] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\mathbb{E}_{ih}[\boldsymbol{S}_{ij}] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\int_0^1 \mathbb{E}_{ih}[\boldsymbol{S}_{ij}\,\Big|\,\boldsymbol{M}_i=\mu_i]f(\mu_i)d\mu_i \;\text{ (law of total expectation)} \\
&= \int_0^1 \Big(\gamma\cdot(\mu_i\cdot g(1-P)+(1-\mu_i)\cdot g(P))+(1-\gamma)\cdot(\mu_i\cdot g(Q)+(1-\mu_i)\cdot g(1-Q))\Big)f(\mu_i)d\mu_i \\
&\text{(using lemma 19)} \\
&= \gamma\Big(\mu\cdot g(1-P)+(1-\mu)\cdot g(P)\Big)+(1-\gamma)\Big(\mu\cdot g(Q)+(1-\mu)\cdot g(1-Q)\Big) \\
&\text{(since } \int_0^1 \mu_i f(\mu_i)d\mu_i=\mu\text{)}
\end{aligned}
$$

Similarly, when worker w_i is adversarial,

$$
\begin{aligned}
p_a &:= \mathbb{E}_{ia}[\mathbf{PEN}_i] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\mathbb{E}_{ia}[\boldsymbol{S}_{ij}] \\
&= \frac{1}{l}\sum_{j\in\mathcal{T}_i}\gamma\cdot g(1-P)+(1-\gamma)\cdot g(1-Q) \quad \text{(using lemma 19)} \\
&= \gamma\cdot g(1-P)+(1-\gamma)\cdot g(1-Q)
\end{aligned}
$$

Suppose $q\mu \le 1/2$, then we have that $Q = 1-q\mu \ge 1/2$ and $P > Q \ge 1/2$. Further, because $g(\cdot)$ is strictly increasing, it follows that $g(1-P)-g(P) < 0$ and $g(1-Q)-g(Q) \le 0$. Given this, and using the expressions for the expected penalty computed above, we have that:

$$
\begin{aligned}
&p_a - p_h \\
&= \gamma\cdot(1-\mu)\cdot\underbrace{(g(1-P)-g(P))}_{<0}+(1-\gamma)\cdot\mu\cdot\underbrace{(g(1-Q)-g(Q))}_{\le 0} \\
&\le 0
\end{aligned}
$$

Therefore, $q\mu > \frac{1}{2}$ is a *necessary* condition for $p_h < p_a$. Assuming this condition is met, we derive the second condition:

$$
\begin{aligned}
p_a > p_h &\iff p_a - p_h > 0 \\
&\iff \gamma\cdot(1-\mu)\cdot\Big(g(1-P)-g(P)\Big)+(1-\gamma)\cdot\mu\Big(g(1-Q)-g(Q)\Big) > 0 \\
&\iff (1-\gamma)\cdot\mu\Big(g(1-Q)-g(Q)\Big) > \gamma\cdot(1-\mu)\cdot\Big(g(P)-g(1-P)\Big) \\
&\iff \frac{\mu}{1-\mu}\frac{\Big(g(1-Q)-g(Q)\Big)}{\Big(g(P)-g(1-P)\Big)} > \frac{\gamma}{1-\gamma} \\
&\text{(since } P > q\mu > \frac{1}{2} \text{ and } Q = 1-q\mu < \frac{1}{2}\text{)}
\end{aligned}
$$

Consider the function $h(\mu,q) = \frac{g(1-Q)-g(Q)}{g(P)-g(1-P)}$ in the regime $q\mu > \frac{1}{2}$. Note that as q increases, Q decreases (since $\frac{\partial Q}{\partial q} = -\mu < 0$) and therefore $g(1-Q)-g(Q)$ (strictly) increases. Similarly, P decreases as q increases (since $\frac{\partial P}{\partial q} = \mu - 1 \le 0$) and therefore $g(P)-g(1-P)$ decreases. It follows that $h(\mu,q)$ is a *strictly* increasing function of q. The result of the theorem now follows.

A.1.3 Proof of Corollary 5

Since $q\mu > \frac{1}{2}$, we have that $P > \frac{1}{2}$ and $Q < \frac{1}{2}$. From the proof of theorem 4 above, we have

$$
\begin{aligned}
p_a > p_h &\iff \frac{\mu}{1-\mu}h(\mu,q) > \frac{\gamma}{1-\gamma} \\
&\iff h_\mu(q) > \frac{\gamma}{1-\gamma} \\
&\iff q > h_\mu^{-1}\Big(\frac{\gamma}{1-\gamma}\Big) \\
&\text{(since } h_\mu(q) \text{ is strictly increasing)}
\end{aligned}
$$

A.2 Asymptotic identification of honest and adversarial workers

We begin with the proof of the lemma establishing the locally-tree like property of the worker-task assignment graph.

A.2.1 Proof of Lemma 8

We adapt the proof from Karger et al. Consider the following (discrete time) random process that generates the random graph $\mathcal{B}_{\boldsymbol{w},2}$ starting from the root node $\boldsymbol{w}$. In the first step, we connect l task nodes to node $\boldsymbol{w}$ according to the configuration model, where l half-edges are matched to a randomly chosen subset of mr task half-edges of size l. Let α_1 denote the probability that the resulting graph is *not* a tree, that is, at least one pair

of edges are connected to the same task node. Since there are $\binom{l}{2}$ pairs and each pair of half-edges is connected to the same task node with probability $\frac{r-1}{mr-1}$, we have that:

$$\alpha_1 \leq \binom{l}{2}\frac{r-1}{mr-1} \leq \frac{l^2}{2m} = \frac{lr}{2n}$$

where we use the fact that $(a-1)/(b-1) \leq a/b$ for all $a \leq b$ and the relation $mr = nl$. Next, define $\beta_2 \equiv \Pr[\mathcal{B}_{\boldsymbol{w},2}$ is not a tree $\big|\ \mathcal{B}_{\boldsymbol{w},1}$ is a tree$]$ so that we have:

$$\Pr\left(\mathcal{B}_{\boldsymbol{w},2} \text{ is not a tree}\right) \leq \alpha_1 + \beta_2$$

We can similarly bound β_2. For generating $\mathcal{B}_{\boldsymbol{w},2}$ conditioned on $\mathcal{B}_{\boldsymbol{w},1}$ being a tree, there are $l\hat{r}$ half-edges where $\hat{r} = r-1$. Among the $\binom{l\hat{r}}{2}$ pairs of these half-edges, each pair will be connected to the same worker with probability $\frac{l-1}{l(n-1)-1}$ and therefore:

$$\begin{aligned}\beta_2 &\leq \frac{l^2\hat{r}^2}{2}\frac{l-1}{l(n-1)-1} \\ &\leq \frac{l^2\hat{r}^2}{2}\frac{l}{l(n-1)} = \frac{l^2\hat{r}^2}{2(n-1)}\end{aligned}$$

Combining this with the above, we get that

$$\Pr[\mathcal{B}_{\boldsymbol{w},2} \text{ is not a tree}] \leq \alpha_1 + \beta_2 \leq \frac{lr}{2n} + \frac{l^2\hat{r}^2}{2(n-1)} \leq \frac{l^2r^2}{n-1}$$

A.2.2 Proof of Lemma 9

Consider an honest worker w_i with a given reliability μ_i. Recall that the penalty assigned to w_i is of the form:

$$\mathbf{PEN}_i = \frac{1}{l}\sum_{j\in\mathcal{T}_i} \boldsymbol{S}_{ij}$$

where $\boldsymbol{S}_{ij} = \frac{1}{\boldsymbol{D}_j^+}$ if $\boldsymbol{W}_{ij} = +1$ and $\boldsymbol{S}_{ij} = \frac{1}{\boldsymbol{D}_j^-}$ when $\boldsymbol{W}_{ij} = -1$. For any two tasks $t_j \neq t_{j'} \in \mathcal{T}_i$ note that $\boldsymbol{W}_{ij}$ and $\boldsymbol{W}_{ij'}$ are independent, conditioned on the reliability $\boldsymbol{M}_i$. This is because

for $(v_1, v_2) \in \{-1, +1\}$:

$$\begin{aligned}&\Pr[\boldsymbol{W}_{ij} = v_1; \boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i] \\ &= \sum_{(x_1,x_2)\in\{-1,+1\}} \Pr[\boldsymbol{W}_{ij} = v_1; \boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = x_1; \boldsymbol{Y}_{j'} = x_2] \Pr[\boldsymbol{Y}_j = x_1; \boldsymbol{Y}_{j'} = x_2] \\ &(\text{ since the true task labels are independent of } \boldsymbol{M}_i) \\ &= \Big(\Pr[\boldsymbol{W}_{ij} = v_1 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = +1]\Pr[\boldsymbol{Y}_j = +1] \\ &+ \Pr[\boldsymbol{W}_{ij} = v_1 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_j = -1]\Pr[\boldsymbol{Y}_j = -1]\Big)\times \\ &\Big(\Pr[\boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_{j'} = +1]\Pr[\boldsymbol{Y}_{j'} = +1] \\ &+ \Pr[\boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i; \boldsymbol{Y}_{j'} = -1]\Pr[\boldsymbol{Y}_{j'} = -1]\Big) \\ &= \Pr[\boldsymbol{W}_{ij} = v_1 \,|\, \boldsymbol{M}_i = \mu_i] \cdot \Pr[\boldsymbol{W}_{ij'} = v_2 \,|\, \boldsymbol{M}_i = \mu_i]\end{aligned}$$

Note that third equality makes use of the fact that $\boldsymbol{W}_{ij}$ (resp. $\boldsymbol{W}_{ij'}$) is independent of all other random variables conditioned on the reliability $\boldsymbol{M}_i$ and the true label $\boldsymbol{Y}_j$ (resp. $\boldsymbol{Y}_{j'}$). The argument above can be extended for any subset of random variables $\boldsymbol{W}_{ij_1}, \boldsymbol{W}_{ij_2}, \ldots, \boldsymbol{W}_{ij_i}$. Further, if the worker-task assignment graph is 2-locally tree-like at w_i, there is no other overlap in the set of workers labeling the tasks $t_j, t_{j'}$ apart from w_i. This combined with the above claim shows that the random variables $\{\boldsymbol{S}_{ij} : j \in \mathcal{T}_i\}$ are *mutually independent* under our generative model. Further, note that $\frac{1}{r} \leq \boldsymbol{S}_{ij} \leq 1$ for any task $t_j \in \mathcal{T}_i$, i.e the random variables $\boldsymbol{S}_{ij}$ are bounded. Then, we can apply Hoeffding's inequality to bound the difference between $\mathbf{PEN}_i$ and $\mathbb{E}[\mathbf{PEN}_i \,\big|\, \boldsymbol{M}_i = \mu_i]$ for any $\varepsilon > 0$:

$$\Pr\left(\mathbf{PEN}_i \geq \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] + \varepsilon \,\Big|\, \boldsymbol{M}_i = \mu_i\right) \leq \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right)$$

A.2.3 Proof of Theorem 6

Suppose we draw a random worker $\boldsymbol{w}$ uniformly from the set of workers W. Then we want to compute the average probability of error, which is the probability that we misclassify a randomly chosen worker:

$$\frac{1}{n}\sum_{i=1}^{n}\Pr\left(\boldsymbol{I}(w_i) \neq \hat{\boldsymbol{I}}_\theta(w_i) \text{ and } \boldsymbol{M}_i > \hat{\mu}(\theta)\right) = \Pr\left(\boldsymbol{I}(\boldsymbol{w}) \neq \hat{\boldsymbol{I}}_\theta(\boldsymbol{w}) \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)\right)$$

Let $\mathbf{PEN}_{\boldsymbol{w}}$ denote the penalty received by the worker $\boldsymbol{w}$. Recall the expression for the expected penalty received by an honest worker from the proof of theorem 3 above, $\mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \,\big|\, \boldsymbol{M}_{\boldsymbol{w}} = \mu_{\boldsymbol{w}}] = g(\mu) - \mu_{\boldsymbol{w}} \cdot (g(\mu) - g(1-\mu))$. Based on the definition of $\hat{\mu}(\theta)$ in the theorem it follows that $p_{\hat{\mu}(\theta)} := \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \,\big|\, \boldsymbol{M}_{\boldsymbol{w}} = \hat{\mu}(\theta)] = \theta - \varepsilon$.

We upper bound the probability $\Pr\left(\boldsymbol{I}(\boldsymbol{w}) \neq \hat{\boldsymbol{I}}_\theta(\boldsymbol{w}) \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)\right)$ in two steps. First, if $\mathcal{B}_{\boldsymbol{w},2}$ is not a tree, we suppose that we always declare w_i as adversarial, thereby making an error. So supposing $\mathcal{B}_{\boldsymbol{w},2}$ is a tree, the probability that we misclassify $\boldsymbol{w}$ is given by

$$
\begin{aligned}
&\Pr(\mathbf{PEN}_{\boldsymbol{w}} > \theta \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)) \\
&= \Pr(\mathbf{PEN}_{\boldsymbol{w}} > p_{\hat{\mu}(\theta)} + \varepsilon \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)) \\
&= \int_{\hat{\mu}(\theta)}^{1} \Pr\left(\mathbf{PEN}_{\boldsymbol{w}} > p_{\hat{\mu}(\theta)} + \varepsilon \,\middle|\, \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}\right) f(\tilde{\mu}) d\tilde{\mu} \\
&\leq \int_{\hat{\mu}(\theta)}^{1} \Pr\left(\mathbf{PEN}_{\boldsymbol{w}} > \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \mid \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}] + \varepsilon \,\middle|\, \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}\right) f(\tilde{\mu}) d\tilde{\mu} \\
&(\text{since } \tilde{\mu} > \hat{\mu}(\theta) \implies \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \mid \boldsymbol{M}_{\boldsymbol{w}} = \tilde{\mu}] < p_{\hat{\mu}(\theta)}) \\
&\leq \int_{\hat{\mu}(\theta)}^{1} \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right) f(\tilde{\mu}) d\tilde{\mu} \\
&(\text{from lemma 9 above}) \\
&\leq \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right)
\end{aligned}
$$

Finally, combining all the claims above, we have

$$
\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} \Pr\left(\boldsymbol{I}(w_i) \neq \hat{\boldsymbol{I}}_\theta(w_i) \text{ and } \boldsymbol{M}_i > \hat{\mu}(\theta)\right) \\
&= \Pr\left(\boldsymbol{I}(\boldsymbol{w}) \neq \hat{\boldsymbol{I}}_\theta(\boldsymbol{w}) \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta)\right) \\
&\leq \Pr(\mathcal{B}_{\boldsymbol{w},2} \text{ is not a tree }) \cdot 1 + \Pr\left(\boldsymbol{I}(\boldsymbol{w}) \neq \hat{\boldsymbol{I}}_\theta(\boldsymbol{w}) \text{ and } \boldsymbol{M}_{\boldsymbol{w}} > \hat{\mu}(\theta) \,\middle|\, \mathcal{B}_{\boldsymbol{w},2} \text{ is a tree }\right) \cdot 1 \\
&\leq \frac{l^2 r^2}{n-1} + \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right)
\end{aligned}
$$

and the first part of the theorem follows. For the second part, when $l = \log n$ and r is fixed, note that

$$
\exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right) = \exp\left(\frac{-2\log n\varepsilon^2}{(1-1/r)^2}\right) = \exp\left(\log\left(\frac{1}{n}\right)^{\frac{2\varepsilon^2}{(1-1/r)^2}}\right) = O\left(\frac{1}{n^{2\varepsilon^2}}\right)
$$

In addition, because $\varepsilon < \frac{1}{\sqrt{2}} \implies 2\varepsilon^2 < 1$, it follows that $\frac{\log^2 n \cdot r^2}{n-1} = O\left(\frac{1}{n^{2\varepsilon^2}}\right)$ and the claim follows.

A.2.4 Proof of Theorem 10

We follow a similar line of reasoning to that of theorem 6. As before, if $\mathcal{B}_{\boldsymbol{w},2}$ is not a tree, we suppose that the worker is always misclassified.

First, we focus on **honest** workers. Given a threshold θ, choose the reliability threshold such that $\mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \hat{\mu}(q,\theta)] = \theta - \varepsilon$ where the expected penalty for an honest worker

conditioned on reliability was computed in lemma 19. Specifically, we have the following expression for $\hat{\mu}(q,\theta)$:

$$
\begin{aligned}
&\gamma\Big(\hat{\mu}(q,\theta)\cdot g(1-P) + (1-\hat{\mu}(q,\theta))\cdot g(P)\Big) + (1-\gamma)\Big(\hat{\mu}(q,\theta)\cdot g(Q) + (1-\hat{\mu}(q,\theta))\cdot g(1-Q)\Big) \\
&= \theta - \varepsilon \implies \\
&\Big(\gamma g(1-P) - \gamma g(P) + (1-\gamma)g(Q) - (1-\gamma)g(1-Q)\Big)\hat{\mu}(q,\theta) \\
&= \theta - \varepsilon - \gamma g(P) - (1-\gamma)g(1-Q) \implies \\
&\hat{\mu}(q,\theta) = \frac{\gamma g(P) + (1-\gamma)g(1-Q) + \varepsilon - \theta}{\gamma\cdot(g(P)-g(1-P)) + (1-\gamma)\cdot(g(1-Q)-g(Q))}
\end{aligned}
$$

Note that such a threshold always exists since (1) $\theta - \varepsilon > p_h = \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu]$ (2) $\mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i]$ (strictly) increases as μ_i decreases and (3) $\mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = 0] \geq p_a > \theta > \theta - \varepsilon$. In particular, this means that $\hat{\mu}(q,\theta) < \mu$. Further, observe that if $q = 1$, then we have $\hat{\mu}(1,\theta) = \hat{\mu}(\theta)$.

Then, the probability that we misclassify a randomly chosen honest worker is given by

$$
\Pr(\mathbf{PEN}_{\boldsymbol{w}} > \theta \,\Big|\, \boldsymbol{w} \text{ is honest }) \leq \Pr\left(\mathbf{PEN}_{\boldsymbol{w}} \geq \mathbb{E}[\mathbf{PEN}_{\boldsymbol{w}} \mid \boldsymbol{M}_{\boldsymbol{w}} = \hat{\mu}(q,\theta)] + \varepsilon \,\middle|\, \boldsymbol{w} \text{ is honest }\right)
$$

The claim then follows from the result of lemma 11 below.

When $\boldsymbol{w}$ is **adversarial**, the probability that we misclassify $\boldsymbol{w}$ is given by

$$
\Pr(\mathbf{PEN}_{\boldsymbol{w}} \leq \theta \mid \boldsymbol{w} \text{ is adversarial}) \leq \Pr(\mathbf{PEN}_{\boldsymbol{w}} \leq p_a - \varepsilon \mid \boldsymbol{w} \text{ is adversarial}) \leq \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right)
$$

where the first inequality follows since $\theta < p_a - \varepsilon$ and the second follows from the result of lemma 11 below.

Coming to the second part of the theorem, first observe that the expected penalties p_h and p_a lie between 0 and 1. As a result, we have that $\varepsilon < (p_a - p_h)/2 \leq \frac{1}{2} \implies 2\varepsilon^2 < 1$ and the claim follows from the sequence of arguments in the proof of theorem 6 above.

A.2.5 Proof of Lemma 11

Suppose that worker w_i is **honest** and let $p_{\hat{\mu}} = \mathbb{E}[\mathbf{PEN}_i \,\big|\, \boldsymbol{M}_i = \hat{\mu}]$. We have the following (conditioned on $\mathcal{B}_{w_i,2}$ being a tree):

$$
\begin{aligned}
&\Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \,\big|\, w_i \text{ is honest }) \\
&= \int_0^1 \Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \,\big|\, w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&= \int_0^{\hat{\mu}} \Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \,\big|\, w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&+ \int_{\hat{\mu}}^1 \Pr(\mathbf{PEN}_i \geq p_{\hat{\mu}} + \epsilon \,\big|\, w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&\leq \int_0^{\hat{\mu}} f(\mu_i) d\mu_i \\
&+ \int_{\hat{\mu}}^1 \Pr(\mathbf{PEN}_i \geq \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] + \epsilon \,\big|\, w_i \text{ is honest and } \boldsymbol{M}_i = \mu_i) f(\mu_i) d\mu_i \\
&(\text{ since } \mu_i \geq \hat{\mu} \implies \mathbb{E}[\mathbf{PEN}_i \mid \boldsymbol{M}_i = \mu_i] \leq p_{\hat{\mu}}) \\
&\leq F(\hat{\mu}) + \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right) \\
&(\text{ from lemma 9})
\end{aligned}
$$

When w_i is an **adversary**, note that her responses $\{\boldsymbol{W}_{ij} \,:\, j \in \mathcal{T}_i\}$ are trivially independent since adversarial workers always respond with $+1$ on all assigned tasks. Further, since $\mathcal{B}_{w_i,2}$ is locally tree-like, there is no other overlap in the set of workers labeling any two tasks $t_j, t_{j'} \in \mathcal{T}_i$ apart from w_i. As a result, the assigned penalties $\{\boldsymbol{S}_{ij} \,:\, j \in \mathcal{T}_i\}$ are *mutually independent* and using the Hoeffding's inequality we can establish the concentration bound for adversarial workers:

$$
\Pr[\mathbf{PEN}_i \leq p_a - \varepsilon \,\big|\, w_i \text{ is adversarial}] \leq \exp\left(\frac{-2l^2\varepsilon^2}{\sum_{j=1}^{l}(1-1/r)^2}\right) = \exp\left(\frac{-2l\varepsilon^2}{(1-1/r)^2}\right)
$$

A.3 Extension to random, normalized hard-penalty

Here, we show that the theoretical results proved above for the soft-penalty algorithm extend to the random, normalized variant of the hard-penalty algorithm mentioned in section 3.3. First, we focus on the expected penalties. Since the penalty algorithm is randomized, the expectation also takes into account the randomness in the algorithm. As above, if $\boldsymbol{S}_{ij}$ denotes the penalty received by worker w_i from task t_j, then we have that $\boldsymbol{S}_{ij} \in \{0, 1\}$. Further, conditioned on the fact that $\boldsymbol{W}_{ij} = +1$, we have that $\mathbb{E}[\boldsymbol{S}_{ij} \mid \boldsymbol{W}_{ij} = +1] = \mathbb{E}[\frac{1}{\boldsymbol{D}_j^+} \mid \boldsymbol{W}_{ij} = +1]$ using the law of iterated expectations. Similarly, for the case when $\boldsymbol{W}_{ij} = -1$. Then, using the arguments above it is easy to see that the expressions for the expected penalties are the same.

Moving on to the concentration results, observe that $\boldsymbol{S}_{ij}$ depends *only* on $\boldsymbol{W}_{ij}$ and $\boldsymbol{D}_j^+$ (and consequently, $\boldsymbol{D}_j^-$). This is because when $\boldsymbol{W}_{ij} = +1$, we have

$$
(\boldsymbol{S}_{ij} \mid \boldsymbol{W}_{ij} = +1) = \begin{cases} 1 & \text{w.p. } 1/\boldsymbol{D}_j^+ \\ 0 & \text{w.p. } 1 - 1/\boldsymbol{D}_j^+ \end{cases}
$$

Similarly, for the case when $\boldsymbol{W}_{ij} = -1$. Now, when $\mathcal{B}$ is locally tree-like at worker node w_i, the arguments in lemma 9 and 11 establish that the random variables $\{\boldsymbol{S}_{ij} \,:\, t_j \in \mathcal{T}_i\}$ are still mutually independent conditioned on the identity of worker w_i (this also relies on the fact that each task is treated independently when computing the random semi-matching). Therefore, we can apply the Hoeffding's bound to establish the concentration of the penalties under the random, normalized variant of the hard-penalty algorithm around the expected values.

A.4 Proof of Lemma 13

For the simple majority algorithm, it is easy to see that each task has k incorrect responses and at most $r < k$ correct responses and consequently, it outputs the incorrect label for all tasks.

For the expectation-maximization (EM) algorithm, recall that the generative model for worker ratings was as follows: the true task labels are sampled from a population with $\gamma \in [0, 1]$ fraction of tasks having $+1$ true label. Each worker w_i has reliability parameter $\mu_i \in [0, 1]$, which specifies the probability that w_i provides the correct label on any task. Then, given the response matrix $\mathcal{L}$, the log-likelihood of the parameters, $\boldsymbol{\Theta} = (\mu_1, \mu_2, \cdots, \mu_n, \gamma)$ under this generative model can be written as:

$$
\log \Pr[\mathcal{L} \mid \boldsymbol{\Theta}] = \sum_{j=1}^{m} \log\left(a_j \cdot \gamma + b_j \cdot (1-\gamma)\right)
$$

where $M(j)$ is the set of workers who label task t_j and

$$
\begin{aligned}
a_j &= \prod_{i \in M(j)} {\mu_i}^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot (1-\mu_i)^{\mathbf{1}[\mathcal{L}_{ij}=-1]} \\
b_j &= \prod_{i \in M(j)} (1-\mu_i)^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot {\mu_i}^{\mathbf{1}[\mathcal{L}_{ij}=-1]}
\end{aligned}
$$

The objective is to find the model parameters $\boldsymbol{\Theta}^*$ that maximize the log-likelihood, i.e. $\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \log \Pr[\mathcal{L} \mid \Theta]$. The EM algorithm computes the maximum likelihood estimates (MLE) of the model parameters by introducing the latent true label of each task, denoted by the vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]$. The complete data log-likelihood can then be written as:

$$
\log \Pr[\mathcal{L}, \boldsymbol{y} \mid \boldsymbol{\Theta}] = \sum_{j=1}^{m} \left(y_j \log(a_j\gamma) + (1-y_j)\log(1-\gamma)b_j\right) \tag{1}
$$

Each iteration x of the EM algorithm consists of two steps:

- **E-step:** Given the response matrix $\mathcal{L}$ and the current estimate of the model parameters $\mathbf{\Theta}^{(x)}$, the conditional expectation of the complete data log-likelihood is computed as

$$\mathbb{E}\left\{\log \Pr[\mathcal{L}, \boldsymbol{y} \mid \mathbf{\Theta}^{(x)}]\right\} = \sum_{j=1}^{m} \left(\gamma_j^{(x)} \log(a_j^{(x)}\gamma^{(x)}) + (1-\gamma_j^{(x)}) \log(1-\gamma^{(x)}) b_j^{(x)}\right)$$

where the expectation is w.r.t to $\Pr[\boldsymbol{y} \mid \mathcal{L}; \mathbf{\Theta}^{(x)}]$ and $\gamma_j^{(x)} = \Pr[y_j = +1 \mid \mathcal{L}; \mathbf{\Theta}^{(x)}]$. Using Bayes theorem, we can compute

$$\begin{aligned}\gamma_j^{(x)} &= \frac{\Pr[\{\mathcal{L}_{ij} \quad : \quad i \in M(j)\} \mid y_j = +1; \mathbf{\Theta}^{(x)}] \cdot \Pr[y_j = +1 \mid \mathbf{\Theta}^{(x)}]}{\Pr[\{\mathcal{L}_{ij} \quad : \quad i \in M(j)\} \mid \mathbf{\Theta}^{(x)}]} \\ &= \frac{a_j^{(x)}\gamma^{(x)}}{a_j^{(x)}\gamma^{(x)} + b_j^{(x)}(1-\gamma^{(x)})}\end{aligned} \tag{2}$$

where the first equality follows from the fact that labels for different tasks t_j are independent of each other and the second equality follows from the law of total probability.

- **M-step:** Based on the current posterior estimates of the true labels $\gamma_j^{(x)}$ and the response matrix $\mathcal{L}$, the model parameters are updated by maximizing $\mathbb{E}\left\{\log \Pr[\mathcal{L}, \boldsymbol{y} \mid \mathbf{\Theta}^{(x)}]\right\}$, which can be shown to be a lower bound on the data log-likelihood (equation (1)). The prevalence of positive tasks γ is updated as:

$$\gamma^{(x+1)} = \frac{\sum_{j=1}^{m} \gamma_j^{(x)}}{m} \tag{3}$$

Similarly, the parameter μ_i is updated as:

$$\mu_i^{(x+1)} = \frac{\sum_{j \in N(i)} \left(\mathbf{1}[\mathcal{L}_{ij} = +1]\gamma_j^{(x)} + \mathbf{1}[\mathcal{L}_{ij} = -1](1-\gamma_j^{(x)})\right)}{|N(i)|} \tag{4}$$

where $N(i)$ is the set of tasks assigned to worker w_i.

These two steps are iterated until convergence of the log-likelihood $\log \Pr[\mathcal{L} \mid \mathbf{\Theta}]$. After convergence, the true labels of the tasks are estimated as:

$$\hat{y}_j = 2 \cdot \mathbf{1}[\gamma_j^{(K)} \geq 0.5] - 1 \tag{5}$$

where K is the number of iterations to convergence.

Since the original problem is non-convex, the EM algorithm converges to a local optimum in general. Its performance critically depends on the initialization of the posterior estimates $\gamma_j^{(0)}$ for each task t_j. A popular approach is to use the majority estimate (see Raykar and Yu (2012)), given by $\gamma_j^{(0)} := \frac{\sum_{i \in M(j)} \mathbf{1}[\mathcal{L}_{ij}=+1]}{|M(j)|}$. We show that this initialization results in incorrect labels for all the tasks.

Suppose that the true labels for all tasks are $+1$, i.e $y_j = +1$ for all $1 \leq j \leq m$. Then all honest worker labels are $+1$ and adversary labels are -1. Further, since each task has more adversary than honest labels, we have that $\gamma_j^{(0)} < 0.5$ for all $1 \leq j \leq m$. Given this, it easy to see from equation (4) that $\mu_i^{(1)} < 0.5$ whenever w_i is honest and $\mu_i^{(1)} > 0.5$ for all adversarial workers w_i. In addition, we have $\gamma^{(1)} < 0.5$ which follows from equation (3). With these estimates, we update the posterior probabilities γ_j for each task t_j. Again, it follows that $\gamma_j^{(1)} < 0.5$ for all $1 \leq j \leq m$, using the update rule in equation (2). The above sequence of claims shows that $\gamma_j^{(x)} < 0.5$ for all iterations $x \geq 1$ and consequently, $\hat{y}_j = -1$ for all tasks t_j (using equation (5)). Therefore, the EM algorithm outputs incorrect labels for all tasks.

Now, let us consider the general case. If we initialize using the majority estimate, we obtain that $\gamma_j^{(0)} > 0.5$ for all tasks with true label $y_j = -1$ and $\gamma_j^{(0)} < 0.5$ for all tasks with $y_j = +1$. Then, it follows from equation (4) that $\mu_i^{(1)} < 0.5$ for all honest workers w_i and $\mu_i^{(1)} > 0.5$ for adversarial workers w_i. In addition, if we look at the update equations (3) and (4) together, we also get that $\mu_i^{(1)} \geq \max(1-\gamma^{(1)}, \gamma^{(1)})$ for all adversaries w_i. To see this, first note that adversaries label on all the tasks, so that $|N(i)| = m$ for an adversarial worker w_i. Next, let $N_+(i)$ and $N_-(i)$ denote the set of tasks for which adversary w_i labels $+1$ and -1 respectively. If $N_+(i)$ is empty, then we have $\mu_i^{(1)} = 1-\gamma^{(1)}$. Also, since we have all adversary labels as -1, we have $\gamma_j^{(0)} < 0.5$ for all $1 \leq j \leq m$ and therefore $\gamma^{(1)} < 1-\gamma^{(1)}$. Therefore, in this case we have $\mu_i^{(1)} \geq \max(1-\gamma^{(1)}, \gamma^{(1)})$. A symmetric argument can be applied when $N_-(i)$ is empty. So suppose that both $N_+(i)$ and $N_-(i)$ are non-empty, then we have from equation (4):

$$\begin{aligned}\mu_i^{(1)} &= \frac{\sum_{j \in N(i)} \left(\mathbf{1}[\mathcal{L}_{ij} = +1]\gamma_j^{(0)} + \mathbf{1}[\mathcal{L}_{ij} = -1](1-\gamma_j^{(0)})\right)}{|N(i)|} \\ &= \frac{\sum_{j \in N_+(i)} \gamma_j^{(0)} + \sum_{j \in N_-(i)} (1-\gamma_j^{(0)})}{m} \\ &> \frac{\sum_{j \in N_+(i)} \gamma_j^{(0)} + \sum_{j \in N_-(i)} \gamma_j^{(0)}}{m} \\ &\quad \text{(since } \gamma_j^{(0)} < 0.5 \text{ if adversaries label } t_j \text{ as } -1\text{)} \\ &= \frac{\sum_{j \in N(i)} \gamma_j^{(0)}}{m} \\ &= \gamma^{(1)}\end{aligned}$$

where the last equality follows from equation (3). Similarly, we can show that $\mu_i^{(1)} > 1-\gamma^{(1)}$ and the claim follows.

Now, let $M_h(j)$ and $M_a(j)$ denote the set of honest and adversarial workers labeling task t_j. Consider a task t_j for which the true label $y_j = -1$. Now, we have two cases:

Case 1: $\gamma^{(1)} \geq 0.5$. First observe that since $\mu_i^{(1)} < 0.5$ for all honest workers w_i, we have $\mu_i^{(1)} < 1 - \mu_i^{(1)}$ for all honest workers w_i. Similarly, we have $1 - \mu_i^{(1)} < \mu_i^{(1)}$ for all

adversarial workers w_i. Then, we have

$$\begin{aligned} b_j^{(1)} \cdot (1-\gamma^{(1)}) &= \left(\prod_{i \in M(j)} (1-\mu_i^{(1)})^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot (\mu_i^{(1)})^{\mathbf{1}[\mathcal{L}_{ij}=-1]} \right) \cdot (1-\gamma^{(1)}) \\ &= \prod_{i \in M_h(j)} \mu_i^{(1)} \cdot \prod_{i \in M_a(j)} (1-\mu_i^{(1)}) \cdot (1-\gamma^{(1)}) \\ &\quad \text{(since honest worker response is } -1 \text{ and adversary response is } +1) \\ &< \prod_{i \in M_h(j)} (1-\mu_i^{(1)}) \cdot \prod_{i \in M_a(j)} \mu_i^{(1)} \cdot \gamma^{(1)} \\ &= a_j^{(1)} \cdot \gamma^{(1)} \end{aligned}$$

where the inequality follows from the observation above and the fact that $\gamma^{(1)} \geq 0.5$. Using equation (2), it follows that $\gamma_j^{(1)} > 0.5$.

Case 2: $\gamma^{(1)} < 0.5$. In this scenario, we have:

$$\begin{aligned} & b_j^{(1)} \cdot (1-\gamma^{(1)}) \\ &= \left(\prod_{i \in M(j)} (1-\mu_i^{(1)})^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot (\mu_i^{(1)})^{\mathbf{1}[\mathcal{L}_{ij}=-1]} \right) \cdot (1-\gamma^{(1)}) \\ &= \prod_{i \in M_h(j)} \mu_i^{(1)} \cdot \prod_{i \in M_a(j)} (1-\mu_i^{(1)}) \cdot (1-\gamma^{(1)}) \\ &= \prod_{i \in M_h(j)} \mu_i^{(1)} \cdot (1-\mu_{w_{i_1}}^{(1)}) \cdot (1-\gamma^{(1)}) \cdot \prod_{i \in M_a(j) \setminus \{i_1\}} (1-\mu_i^{(1)}) \\ &\quad \text{(where } w_{i_1} \text{ is some adversarial worker)} \\ &\leq \prod_{i \in M_h(j)} \mu_i^{(1)} \cdot \mu_{w_{i_1}}^{(1)} \cdot \gamma^{(1)} \cdot \prod_{i \in M_a(j) \setminus \{i_1\}} (1-\mu_i^{(1)}) \\ &\quad \Big(\text{since } 1-\gamma^{(1)} \leq \mu_{w_{i_1}}^{(1)} \text{ because } w_{i_1} \text{ is adversarial; see discussion before Case 1} \Big) \\ &< \prod_{i \in M_h(j)} (1-\mu_i^{(1)}) \cdot \mu_{w_{i_1}}^{(1)} \cdot \gamma^{(1)} \cdot \prod_{i \in M_a(j) \setminus \{i_1\}} \mu_i^{(1)} \\ &\quad \text{(from Case 1 above)} \\ &= a_j^{(1)} \cdot \gamma^{(1)} \end{aligned}$$

Consequently, in both cases, the posterior probability $\gamma_j^{(1)} > 0.5$ given that the true label was $y_j = -1$. A symmetric argument shows that $\gamma_j^{(1)} < 0.5$ if $y_j = +1$. Then, the above sequence of claims establishes that this remains true for all iterations in the future. Finally, step (5) implies that the output label is incorrect for all tasks.

A.5 Proof of Theorem 14

We prove the result for the the case when there exists at least one subset $\mathcal{T}' \subseteq \mathcal{T}$ such that $\text{PreIm}(\mathcal{T}') \leq k$. Otherwise, the lower bound $L = 0$ by definition and the result of the theorem is trivially true.

Let H^* denote the set $\text{PreIm}(\mathcal{T}^*)$ where

$$\mathcal{T}^* \stackrel{\text{def}}{=} \underset{\mathcal{T}' \subseteq \mathcal{T}:\ |\text{PreIm}(\mathcal{T}')| \leq k}{\arg\max} \left| \mathcal{T}' \right|.$$

We construct an adversary strategy σ^* under which at least L tasks are affected for some true labeling of the tasks, for any decision rule $R \in \mathcal{C}$. Specifically for a fixed honest worker-task assignment graph $\mathcal{B}_H$ and ground-truth labeling $\boldsymbol{y}$ of the tasks, consider the following adversary strategy (that depends on the obtained honest worker responses): letting $H^* = \{h_1, h_2, \ldots, h_{|H^*|}\}$ and the set of adversaries $A = \{a_1, a_2, \ldots, a_k\}$, we have (recall the notation in Section 2)

$$a_i(t) = \begin{cases} -h_i(t) & \text{if } t \in \mathcal{T}^* \\ h_i(t) & \text{otherwise} \end{cases} \qquad \forall i = 1, 2, \ldots, |H^*| \tag{6}$$

In other words, the adversaries label *flip* the labels of the honest workers H^* for tasks in $\mathcal{T}^*$ and *copy* their responses for all other tasks. Note that since $|H^*| \leq k$ by construction, the above strategy is feasible. In addition, if $|H^*| < k$, then we only use $|H^*|$ of the k adversary identities. Let $\mathcal{L}(\boldsymbol{y})$ denote the $n \times m$ labeling matrix obtained for this adversary strategy, where we explicitly denote the dependence on the true label vector $\boldsymbol{y}$. Here n denotes the total number of workers.

Now consider the scenario in which the true labels of all tasks in $\mathcal{T}^*$ were reversed, let this ground-truth be denoted as $\tilde{\boldsymbol{y}}$. Let $\tilde{h}(t)$ denote the response of honest worker h for task t in the new scenario. Since, honest workers always respond correctly, we have that:

$$\tilde{h}(t) = \begin{cases} -h(t) & \text{if } t \in \mathcal{T}^* \\ h(t) & \text{otherwise} \end{cases} \qquad \forall\ h \in H \tag{7}$$

Correspondingly, according to the adversary labeling strategy σ^* described above, the adversary responses would also change. In particular, using $\tilde{a}_i(t)$ to denote the adversary response in the new scenario, we have

$$\tilde{a}_i(t) = \begin{cases} -\tilde{h}_i(t) & \text{if } t \in \mathcal{T}^* \\ \tilde{h}_i(t) & \text{otherwise} \end{cases} \qquad \forall i = 1, 2, \ldots, |H^*| \tag{8}$$

Finally, let $\mathcal{L}(\tilde{\boldsymbol{y}})$ denote the labeling matrix in this new scenario. We now argue that $\mathcal{L}(\tilde{\boldsymbol{y}}) = P\mathcal{L}(\boldsymbol{y})$ for some $n \times n$ permutation matrix P. In order to see this, for any worker w (honest or adversary), let $r(w)$ and $\tilde{r}(w)$ respectively denote the row vectors in matrices $\mathcal{L}(\boldsymbol{y})$ and $\mathcal{L}(\tilde{\boldsymbol{y}})$. We show that $\mathcal{L}(\tilde{\boldsymbol{y}})$ can be obtained from $\mathcal{L}(\boldsymbol{y})$ through a permutation of the rows.

First observe that for any honest worker $h \notin H^*$, we must have by definition of PreIm that $h(t) = 0$ for any $t \in \mathcal{T}^*$. Thus, it follows from equation (7) that $\tilde{h}(t) = -h(t) = 0 = h(t)$

for any $t \in \mathcal{T}^*$. Furthermore, $\tilde{h}(t) = h(t)$ for any $t \notin \mathcal{T}^*$ by (7). Therefore, we have that $\tilde{r}(h) = r(h)$ for any $h \notin H^*$. Next, consider an honest worker $h_i \in H^*$ for some i. It can be argued that $\tilde{r}(h_i) = r(a_i)$. To see this, for any task $t \notin \mathcal{T}^*$, we have by equation (7) that $\tilde{h}_i(t) = h_i(t) = a_i(t)$, where the second equality follows from equation (6). Similarly, for any $t \in \mathcal{T}^*$, we have $\tilde{h}_i(t) = -h_i(t) = a_i(t)$ again using equation (6). Thus, we have shown that the rows $\tilde{r}(h_i) = r(a_i)$ for any $1 \leq i \leq |H^*|$. A symmetric argument shows that $\tilde{r}(a_i) = r(h_i)$ for all $1 \leq i \leq |H^*|$. Consequently, $\mathcal{L}(\tilde{\boldsymbol{y}})$ is obtained from $\mathcal{L}(\boldsymbol{y})$ by swapping rows corresponding to h_i with a_i for all $i = 1, 2, \ldots, |H^*|$.

Now that we have shown $\mathcal{L}(\tilde{\boldsymbol{y}}) = P\mathcal{L}(\boldsymbol{y})$ for some permutation matrix P, it follows from the fact that $R \in \mathcal{C}$ that $R(\mathcal{L}(\tilde{\boldsymbol{y}})) = R(\mathcal{L}(\boldsymbol{y}))$. Thus, the labels output by R for all tasks in $\mathcal{T}^*$ is the same under both scenarios. As a result, it follows that $\text{Aff}(R, \sigma^*, \boldsymbol{y}) + \text{Aff}(R, \sigma^*, \tilde{\boldsymbol{y}}) = |\mathcal{T}^*| = 2 * L$ and therefore, either $\text{Aff}(R, \sigma^*, \boldsymbol{y}) \geq L$ or $\text{Aff}(R, \sigma^*, \tilde{\boldsymbol{y}}) \geq L$.

In other words, there exists a ground-truth task labeling for which the number of affected tasks is at least L, and since we take a maximum over all possible ground-truth labelings, the result of the theorem follows. Further, any decision rule that outputs labels randomly in case of ties (i.e equal number of $+1$ and -1 responses) will achieve the lower bound L, including the simple majority decision rule.

Misclassification of workers. Since we allow the sophisticated adversaries to adopt arbitrary strategies, we need to assume some characteristic property that helps to identify honest workers. In the theorem we assumed that all honest workers are perfectly reliable, so that they agree with each other on their labels for all tasks. Given this, any ML algorithm that is trying to separate honest workers from the sophisticated adversaries will output two groups of workers, say W_1, W_2 such that $W_1 \cup W_2 = H \cup A$ and $W_1 \cap W_2 = \phi$. Further, let us suppose that the algorithm has knowledge of k—the number of sophisticated adversaries, so that the output satisfies $|W_1| = n - k$ and $|W_2| = k$ where $|H \cup A| = n$. We call an output (W_1, W_2) *valid* if all workers in W_1 agree with each other on their labels for all tasks. Note that the "true" output $W_1 = H$ and $W_2 = A$ is valid. We argue that for the above adversary strategy, the output $W_1 = (H \setminus H^*) \cup A$ and $W_2 = H^*$ is also valid, where H^* is as defined in the proof above.

To see this, note that the algorithm can identify the set $\mathcal{T}^*$ of tasks for which there are conflicts, i.e. both $+1$ and -1 labels, and partition the set of workers labeling these tasks into two groups where workers in each group agree on their label for all tasks. Now, one of these groups consists of adversarial workers A and the other honest workers, corresponding to the set H^* above. The remaining workers correspond to the set $H \setminus H^*$. Now, if we consider the output $W_1 = (H \setminus H^*) \cup A$, any two workers in W_1 indeed agree with each other on their labels for all tasks. This is because, by the definition of PreIm, workers in $H \setminus H^*$ *only* label tasks in the set $\mathcal{T} \setminus \mathcal{T}^*$. Further, since adversaries agree with honest workers H^* on these tasks (refer to the strategy above) and honest workers are perfectly reliable, this means that adversaries A and honest workers $H \setminus H^*$ also agree with each other on their labels for all tasks in $\mathcal{T} \setminus \mathcal{T}^*$. Consequently, we have that the output $((H \setminus H^*) \cup A, H^*)$ is also *valid*. Further note that this is the only other valid output, since honest workers H^* and adversaries A do *not* agree with each other on their labels for tasks in $\mathcal{T}^*$ and therefore, cannot be placed together in the set W_1.

Finally, since the ML algorithm does not have any other information, it cannot distinguish between the two valid outputs described above, so that a random guess will misclassify $2k$ workers with probability $(1/2)$.

A.5.1 Proof of Corollary 16

We first prove that $2L \geq \lfloor \frac{k}{r} \rfloor$. Consider $\mathcal{T}' \subseteq \mathcal{T}$ such that $|\mathcal{T}'| = \lfloor \frac{k}{r} \rfloor$, note that we can always choose such a $\mathcal{T}'$ since $k < |H| \leq r \cdot |\mathcal{T}| \implies k/r < |\mathcal{T}|$. Since $\mathcal{B}_H$ is r-right regular, the pre-image of $\mathcal{T}'$ in $\mathcal{B}_H$ satisfies $|\text{PreIm}(\mathcal{T}')| \leq r|\mathcal{T}'| = r\lfloor \frac{k}{r} \rfloor \leq k$. In other words, any subset of tasks of size $\lfloor \frac{k}{r} \rfloor$ has a pre-image of size at most k. By the definition of L, we have that $2L \geq \lfloor \frac{k}{r} \rfloor$.

For the upper-bound, consider any $\mathcal{T}' \subset \mathcal{T}$ such that $|\mathcal{T}'| = e$ where $e := \lceil \frac{k}{\alpha} \rceil + 1$. Since $e < \gamma|\mathcal{T}|$, by the expander property we have that $|PreIm(\mathcal{T}')| \geq \alpha|\mathcal{T}'| = \alpha \cdot e \geq \alpha \cdot (\frac{k}{\alpha} + 1) > k$. This means that any subset $\mathcal{T}'$ of size *at least* e has a pre-image of size strictly greater than k, since the size of the pre-image can only increase with the addition of more tasks. Therefore, this implies that $2L \leq \lceil \frac{k}{\alpha} \rceil$.

For the second part of the corollary, refer to Theorem 4.4 in Chapter 4 of Vadhan et al. (2012).

A.6 Proof of Theorem 17

Before we can prove the theorem, we need the following definitions and lemmas.

Definition 20 *A bipartite graph $G = (V_1, V_2, E)$ is termed* **degenerate** *if the following condition is satisfied:*

$$|V_1| > |V_2|$$

Definition 21 *A bipartite graph $G = (V_1, V_2, E)$ is termed* **growth** *if the following condition is satisfied:*

$$\forall \ V \subseteq V_1, \ \ |V| \leq |\text{Img}(V)|$$

where $\text{Img}(V) = \{v_2 \in V_2 \mid \exists v \in V \ \text{ s.t. } (\text{v}, \text{v}_2) \in \text{E}\}$*, i.e. the set of neighboring nodes of V.*

Lemma 22 *Any bipartite graph can be decomposed into degenerate and growth sub-graphs where there are cross-edges only between the left nodes of the growth component and the right nodes of the degenerate component.*

Proof Let $G = (V_1, V_2, E)$ be a given bipartite graph. Define V^* to be the largest subset of V_1 such that $|V^*| > |\text{Img}_G(V^*)|$ where Img_G denotes the image in the graph G. If no such V^* exists then the graph is already growth and we are done. If $V^* = V_1$ then the graph is degenerate and again we are done. Else, we claim that the sub-graph J of G restricted to $V_1 \setminus V^*$ on the left and $V_2 \setminus \text{Img}_G(V^*)$ on the right is growth. Suppose not, then there exists a subset V' of nodes on the left such that $|V'| > |\text{Img}_J(V')|$ where $\text{Img}_J(V') \subseteq V_2 \setminus \text{Img}_G(V^*)$ denotes the image of V' in the sub-graph J. But then, we can add V' to V^* to get a larger degenerate sub-graph in G which contradicts our choice of

V^*. To see this, consider the set $V^* \cup V'$ on the left and $\text{Img}_G(V^*) \cup \text{Img}_J(V')$ on the right. We have $|V^* \cup V'| = |V^*| + |V'| > |\text{Img}_G(V^*)| + |\text{Img}_J(V')| = |\text{Img}_G(V^* \cup V')|$. Also, note that the only cross-edges are between $V_1 \setminus V^*$ and $\text{Img}_G(V^*)$. The claim then follows. ■

Lemma 23 *Let $G = (V_1, V_2, E)$ be a bipartite graph and suppose that M is* any *semi-matching on G. Further, let $J = (V_1, V_2', E')$ be the subgraph of G restricted to only the nodes $V_2' \subseteq V_2$ on the right. Starting with $M' \subseteq M$, we can use algorithm $\mathcal{A}_{SM2}$ in Harvey et al. (2003) to obtain an optimal semi-matching $\mathcal{N}$ for the subgraph J. Let the nodes in V_1 be indexed such that $deg_M(1) \geq deg_M(2) \geq \ldots \geq deg_M(|V_1|)$ and indexed again such that $deg_{\mathcal{N}}(1) \geq deg_{\mathcal{N}}(2) \geq \ldots \geq deg_{\mathcal{N}}(|V_1|)$. Then for any $1 \leq s \leq |V_1|$, we have $\sum_{i=1}^{s} deg_{\mathcal{N}}(i) \leq \sum_{i=1}^{s} deg_M(i)$, i.e. the sum of the top s-degrees can only decrease as we go from M to $\mathcal{N}$.*

Proof Note that if we restrict M to just the nodes V_2', we get a feasible semi-matching M' on the subgraph J. Algorithm $\mathcal{A}_{SM2}$ proceeds by the iterated removal of cost-reducing paths. Note that when a cost-reducing path is removed, load is transferred from a node with larger degree (in the current semi-matching) to a node with strictly smaller degree. To see this, let $P = (v_1^{(1)}, v_2^{(1)}, v_1^{(2)}, \ldots, v_1^{(d)})$ be a cost-reducing path (see section 2.1 in Harvey et al. (2003)) in some semi-matching $\bar{M}$ on J. This means that $deg_{\bar{M}}(v_1^{(1)}) > deg_{\bar{M}}(v_1^{(d)}) + 1$. When we eliminate the cost-reducing path P, the degree of $v_1^{(1)}$ decreases by 1 and that of $v_1^{(d)}$ increases by 1, but still the new degree of $v_1^{(d)}$ is strictly lower than the old degree of $v_1^{(1)}$. In other words, if $d_1^{\text{bef}} \geq d_2^{\text{bef}} \geq \ldots \geq d_{|V_1|}^{\text{bef}}$ and $d_1^{\text{aft}} \geq d_2^{\text{aft}} \geq \ldots \geq d_{|V_1|}^{\text{aft}}$ be the degree sequence (in the current semi-matching) before and after the removal of a cost-reducing path, then $\sum_{i=1}^{s} d_i^{\text{aft}} \leq \sum_{i=1}^{s} d_i^{\text{bef}}$ for any $1 \leq s \leq |V_1|$. Since this invariant is satisfied after every iteration of algorithm $\mathcal{A}_{SM2}$, it holds at the beginning and the end, and we have

$$\sum_{i=1}^{s} deg_{\mathcal{N}}(i) \leq \sum_{i=1}^{s} deg_{M'}(i) \tag{9}$$

Finally, observe that when we restrict M to only the set V_2', the sum of the top s-degrees can only *decrease*, i.e.

$$\sum_{i=1}^{s} deg_{M'}(i) \leq \sum_{i=1}^{s} deg_M(i) \tag{10}$$

Combining equations (9) and (10), the result follows. ■

Lemma 24 *Let $G = (V_1, V_2, E)$ be a bipartite graph and suppose that M is* any *semi-matching on G. Further, let $J = (V_1 \cup V_3, V_2, E')$ be the supergraph of G obtained by adding nodes V_3 on the left and edges $E' \setminus E$ to G. Starting with M, we can use algorithm $\mathcal{A}_{SM2}$ in Harvey et al. (2003) to obtain an optimal semi-matching $\mathcal{N}$ for the supergraph J. Let the nodes in V_1 be indexed such that $deg_M(1) \geq deg_M(2) \geq \ldots \geq deg_M(|V_1|)$ and the nodes in $V_1 \cup V_3$ indexed again such that $deg_{\mathcal{N}}(1) \geq deg_{\mathcal{N}}(2) \geq \ldots \geq deg_{\mathcal{N}}(|V_1 \cup V_3|)$. Then for any $1 \leq s \leq |V_1 \cup V_3|$, we have $\sum_{i=1}^{s} deg_{\mathcal{N}}(i) \leq \sum_{i=1}^{s} deg_M(i)$, i.e. the sum of the top s-degrees can only decrease as we go from M to $\mathcal{N}$.*

Proof Note that M is a feasible semi-matching for the graph J, with all nodes in V_3 having degree 0. We can repeat the argument from the previous lemma to show that if $d_1^{\text{bef}} \geq d_2^{\text{bef}} \geq \ldots \geq d_{|V_1 \cup V_3|}^{\text{bef}}$ and $d_1^{\text{aft}} \geq d_2^{\text{aft}} \geq \ldots \geq d_{|V_1 \cup V_3|}^{\text{aft}}$ be the degree sequence (in the current semi-matching) before and after the removal of any cost-reducing path, then $\sum_{i=1}^{s} d_i^{\text{aft}} \leq \sum_{i=1}^{s} d_i^{\text{bef}}$ for any $1 \leq s \leq |V_1 \cup V_3|$. The result then follows. ■

Notation for the proofs. Let $\mathcal{T}^+$ denote the set $\{t^+ : t \in \mathcal{T}\}$, and similarly $\mathcal{T}^-$ denote the set $\{t^- : t \in \mathcal{T}\}$, these are "task copies". Now partition the set of task copies $\mathcal{T}^+ \cup \mathcal{T}^-$ as $E \cup F$ such that for any task t, if the **true** label is $+1$, we put t^+ in E and t^- in F, otherwise, we put t^- in E and t^+ in F. Thus, E contains task copies with *true* labels while F contains task copies with *incorrect* labels. Let $F' \subseteq F$ denote the set of tasks for which honest workers provide incorrect responses, and denote the optimal semi-matching on the graph $\mathcal{B}_H^{cs}$ by $\mathcal{M}_H$. Without loss of generality, suppose that honest workers are indexed such that $d_1 \geq d_2 \geq \cdots \geq d_{|H|}$ where d_h denotes the degree of honest worker h in semi-matching $\mathcal{M}_H$.

Part 1. Adversary strategy that affects at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$ tasks

If $\varepsilon = 0$, there are no conflicts in the responses provided by honest workers, and therefore the bipartite graph $\mathcal{B}_H^{cs}$ contains just "true" task copies E on the right.

Given this, the adversaries *target* honest workers $\{1, 2, \ldots, k-1\}$: for each i, adversary a_i labels opposite to worker h_i (i.e. provides the incorrect response) on every task that h_i is mapped to in the semi-matching $\mathcal{M}_H$. Furthermore, the adversary uses its last identity a_k to provide the incorrect response on every task $t \in \mathcal{T}$ for which one of the first $k-1$ adversaries have not already labeled on. We argue that under the penalty-based aggregation algorithm, this adversary strategy results in incorrect labels for at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$ tasks. To see this, first note that the conflict set $\mathcal{T}_{cs} = \mathcal{T}$. Further, the bipartite graph $\mathcal{B}^{cs}$ decomposes into two disjoint subgraphs: bipartite graph $\mathcal{B}^{cs}(E)$ between H and E and semi-matching $M(F)$ between A and F that represents the adversary labeling strategy (it is a semi-matching because there is exactly one adversary that labels each task). Since the bipartite graph $\mathcal{B}^{cs}$ decomposes into two disjoint bipartite graphs, computing the optimal semi-matching on $\mathcal{B}^{cs}$ is equivalent to separately computing optimal semi-matchings on $\mathcal{B}^{cs}(E)$ and M_F. The argument above says that $\mathcal{B}^{cs}(E)$ is nothing but the graph $\mathcal{B}_H^{cs}$ defined in the theorem statement and therefore $\mathcal{M}_H$ is the optimal semi-matching on $\mathcal{B}^{cs}(E)$. Given that $M(F)$ is already a semi-matching by construction, the optimal semi-matching on $\mathcal{B}^{cs}$ is the disjoint union of $\mathcal{M}_H$ and $M(F)$. It is easy to see that in the resultant semi-matching, honest worker h_i and adversary a_i have the same degrees for $i = 1, 2, \ldots, k-1$. Hence, for every task mapped to honest worker h_i for $i = 1, 2, \ldots, k-1$ in the optimal semi-matching, the algorithm outputs a random label, and therefore outputs the correct label for at most half of these tasks. Thus, the above adversary strategy results in incorrect labels for at least $\frac{1}{2}\sum_{i=1}^{k-1} d_i$ tasks.

Part 2. Upper Bound on number of affected tasks

To simplify the exposition, we assume in the arguments below that the optimal semi-matching in the HARD PENALTY algorithm is computed for the entire task set and not just

the conflict set $\mathcal{T}_{cs}$. However, the bounds provided still hold as a result of lemma 23 above. Consequently, we abuse notation and use $\mathcal{B}^{cs}$ to denote the following bipartite graph in the remainder of the discussion: each worker w is represented by a node on the left, each task t is represented by at most two nodes on the right – t^+ and t^- – and we add an edge (w, t^+) if worker w labels task t as $+1$ and edge (w, t^-) if w labels t as -1. Then, it is easy to see that the subgraph of $\mathcal{B}^{cs}$ restricted to just the honest workers H on the left and task copies $E \cup F'$ on the right, is nothing but the graph $\mathcal{B}^{cs}_H$ defined in the theorem statement. Consequently, $\mathcal{M}_H$ is a feasible semi-matching for the subgraph $\mathcal{B}^{cs}(E \cup F')$, which is obtained by restricting $\mathcal{B}^{cs}$ to nodes $E \cup F'$ on the right. Further, we can assume that the adversary labeling strategy is always a semi-matching, i.e. there is at most one adversary response for any task. If the adversary labeling strategy is not a semi-matching, they can replace it with an alternate strategy where they only label for tasks to which they will be mapped in the optimal semi-matching (the adversaries can compute this since they have knowledge of the honest workers' responses). The optimal semi-matching doesn't change (otherwise it contradicts the optimality of the original semi-matching) and hence neither does the number of affected tasks.

We first state the following important lemma:

Lemma 25 *For any adversary labeling strategy, let $\mathcal{B}^{cs}(E \cup F')$ denote the bipartite graph $\mathcal{B}^{cs}$ restricted to all the workers $W = H \cup A$ on the left and, "true" task copies E as well as subset F' of the "incorrect" task copies F on the right. Let $\mathcal{M}$ be the optimal semi-matching on the bipartite graph $\mathcal{B}^{cs}$ and $\mathcal{M}(E \cup F') \subset M$ be the semi-matching $\mathcal{M}$ restricted to only the task copies $E \cup F'$. Then, $\mathcal{M}(E \cup F')$ is an optimal semi-matching for the sub-graph $\mathcal{B}^{cs}(E \cup F')$.*

Proof First observe that if $F' = F$, then the statement is trivially true. So we can assume $F' \subset F$. Suppose the statement is not true and let $\mathcal{N}(E \cup F')$ denote the optimal semi-matching on $\mathcal{B}^{cs}(E \cup F')$. We use $d_w(K)$ to denote the degree of worker w in a semi-matching K. Note that, $d_a(\mathcal{N}(E \cup F')) \leq d_a(\mathcal{M}(E \cup F')) \leq d_a(\mathcal{M})$ for all adversaries $a \in A$, where the first inequality follows from the fact that the adversary strategy is a semi-matching and the second inequality is true because $\mathcal{M}(E \cup F') \subset \mathcal{M}$. The adversaries who do not provide any responses for the task copies $E \cup F'$ will have degrees 0 in the semi-matchings $\mathcal{N}(E \cup F')$ and $\mathcal{M}(E \cup F')$ but the inequality is still satisfied. Now, since $\mathcal{N}(E \cup F')$ is an optimal semi-matching and $\mathcal{M}(E \cup F')$ is not, we have that

$$cost(\mathcal{N}(E \cup F')) < cost(\mathcal{M}(E \cup F')) \Rightarrow$$
$$\sum_{h \in H} d_h^2(\mathcal{N}(E \cup F')) + \sum_{a \in A} d_a^2(\mathcal{N}(E \cup F')) < \sum_{h \in H} d_h^2(\mathcal{M}(E \cup F')) + \sum_{a \in A} d_a^2(\mathcal{M}(E \cup F'))$$

Now, consider the semi-matching $\mathcal{N}$ on $\mathcal{B}^{cs}$ where we start with the semi-matching $\mathcal{N}(E \cup F')$ and then map the remaining task copies in $\mathcal{B}^{cs}$ (which belong to the set $F \setminus F'$) to the adversaries which they were assigned to in $\mathcal{M}$. We claim that $cost(\mathcal{N}) < cost(\mathcal{M})$ which is

a contradiction since $\mathcal{M}$ was assumed to be an optimal semi-matching on $\mathcal{B}^{cs}$. To see this:

$$\begin{aligned}
&cost(\mathcal{M}) - cost(\mathcal{N}) \\
&= \sum_{h \in H} d_h^2(\mathcal{M}) + \sum_{a \in A} d_a^2(\mathcal{M}) - \left(\sum_{h \in H} d_h^2(\mathcal{N}) + \sum_{a \in A} d_a^2(\mathcal{N}) \right) \\
&= \sum_{h \in H} d_h^2(\mathcal{M}(E \cup F')) + \sum_{a \in A} \left(d_a(\mathcal{M}(E \cup F')) + \Delta_a \right)^2 \\
&- \left(\sum_{h \in H} d_h^2(\mathcal{N}(E \cup F')) + \sum_{a \in A} \left(d_a(\mathcal{N}(E \cup F')) + \Delta_a \right)^2 \right) \\
&\Big(\text{where } \Delta_a \stackrel{\text{def}}{=} d_a(\mathcal{M}) - d_a(\mathcal{M}(E \cup F')) \geq 0 \Big) \\
&= \left(\sum_{h \in H} d_h^2(\mathcal{M}(E \cup F')) + \sum_{a \in A} d_a^2(\mathcal{M}(E \cup F')) - \sum_{h \in H} d_h^2(\mathcal{N}(E \cup F')) - \sum_{a \in A} d_a^2(\mathcal{N}(E \cup F')) \right) \\
&+ 2 \sum_{a \in A} \left(d_a(\mathcal{M}(E \cup F')) - d_a(\mathcal{N}(E \cup F')) \right) * \Delta_a > 0 \\
&\big(\text{since } d_a(\mathcal{M}(E \cup F')) \geq d_a(\mathcal{N}(E \cup F')) \text{ as stated above} \big)
\end{aligned}$$

Therefore, $\mathcal{M}(E \cup F')$ is an optimal semi-matching for the sub-graph $\mathcal{B}^{cs}(E \cup F')$. ■

Part 2a. Adversaries only provide incorrect responses or one adversary provides correct responses

Adversaries only provide incorrect responses. Let us begin with the case when all adversaries only provide incorrect responses.

Lemma 26 *Suppose that $\varepsilon = 0$ and adversaries only provide incorrect responses. Let M be an* arbitrary *semi-matching on the bipartite graph $\mathcal{B}^{cs}$ and suppose that this semi-matching is used in the* penalty-based aggregation *Algorithm to compute the true labels of the tasks. Further, let $b_1 \geq b_2 \geq \cdots \geq b_{|H|}$ denote the degrees of the honest workers in this semi-matching where b_i is the degree of honest worker h_i. Then, the number of affected tasks is at most $\sum_{i=1}^{k} b_i$.*

Proof It follows from the assumption that $\varepsilon = 0$ and that adversaries only provide incorrect responses, that there are no cross-edges between nodes H and F as well as A and E in the bipartite graph $\mathcal{B}^{cs}$. Thus, for any adversary labeling strategy, we can decompose $\mathcal{B}^{cs}$ into *disjoint* bipartite graphs $\mathcal{B}^{cs}(E)$ and $\mathcal{B}^{cs}(F)$, where $\mathcal{B}^{cs}(E)$ is the subgraph consisting of honest workers H and task copies E and $\mathcal{B}^{cs}(F)$ is the subgraph between the adversaries A and the task copies F. This further means that the semi-matching M is a disjoint union of semi-matchings on $\mathcal{B}^{cs}(E)$ and $\mathcal{B}^{cs}(F)$. Let the semi-matchings on the subgraphs be termed as $M(E)$ and $M(F)$ respectively. Further, let $T_{\text{aff}} \subseteq \mathcal{T}$ denote the set of tasks that are affected under this strategy of the adversaries and when the semi-matching M is used to compute the penalties of the workers in the penalty-based aggregation algorithm.

We claim that $|T_{\text{aff}}| \leq \sum_{i=1}^{k} b_i$. To see this, for each adversary $a \in A$, let $H(a) \subset H$ denote the set of honest workers who have "lost" to a i.e., for each worker $h \in H(a)$ there exists some task $t \in \mathcal{T}_{cs}$ such that h is mapped to the *true* copy of t in $M(E)$, a is mapped to the *incorrect* copy of t in $M(F)$, and the degree of h in $M(E)$ is greater than or equal to the degree of a in $M(F)$. Of course, $H(a)$ may be empty. Let $\bar{A}$ denote the set of adversaries $\{a \in A \colon H(a) \neq \emptyset\}$ and let $\bar{H}$ denote the set of honest workers $\bigcup_{a \in \bar{A}} H(a)$. Now define a bipartite graph between the nodes $\bar{A}$ and $\bar{H}$ with an edge between $a \in \bar{A}$ and $h \in \bar{H}$ if and only if $h \in H(a)$. This bipartite graph can be decomposed into degenerate and growth subgraphs by lemma 22 above. In the growth subgraph, by Hall's condition, we can find a perfect matching from adversaries to honest workers. Let (A_1, H_1) with $A_1 \subseteq \bar{A}$ and $H_1 = \text{Img}(A_1)$ be the degenerate component. The number of tasks that adversaries in A_1 affect is bounded above by $\sum_{h \in \text{Img}(A_1)} b_h$. Similarly, for $A_2 = \bar{A} \setminus A_1$, we can match each adversary to a *distinct* honest worker whose degree in $M(E)$ is greater than or equal to the degree of the adversary in $M(F)$. We can bound the number of affected tasks caused due to the adversaries in A_2 by the sum of their degrees, which in turn is bounded above by the sum of the degrees of honest workers that the adversaries are matched to. Let H_2 denote the set of honest workers matched to adversaries in the perfect matching. Thus, we have upper bounded the number of affected tasks by $\sum_{h \in H_1 \cup H_2} b_h$. It is easy to see that $|H_1 \cup H_2| \leq k$. Therefore, $\sum_{h \in H_1 \cup H_2} b_h \leq \sum_{i=1}^{k} b_i$. Therefore, the number of affected tasks $|T_{\text{aff}}|$ is at most $\sum_{i=1}^{k} b_i$. ■

We now extend the above lemma to the case when $\varepsilon \neq 0$.

Lemma 27 *Suppose that adversaries only provide incorrect responses. Let M be an* arbitrary *semi-matching on the bipartite graph $\mathcal{B}^{cs}$ and suppose that this semi-matching is used in the* penalty-based aggregation *Algorithm to compute the true labels of the tasks. Further, let $b_1 \geq b_2 \geq \cdots \geq b_{|H|}$ denote the degrees of the honest workers in this semi-matching where b_i is the degree of honest worker h_i. Then, the number of affected tasks is at most $\sum_{i=1}^{k+\varepsilon \cdot |H|} b_i$.*

Proof Since honest workers can make mistakes, there are cross-edges between H and F in the bipartite graph $\mathcal{B}^{cs}$. Observe that there are two kinds of affected tasks in this case: (1) tasks that adversaries "win" against honest workers, say $T_{\text{aff}}(A)$, similar to those in the previous lemma and (2) tasks that are affected when 2 honest workers are compared in the final step of the penalty-based aggregation algorithm, say $T_{\text{aff}}(H)$. We can repeat the argument from lemma 26 above to bound $|T_{\text{aff}}(A)|$ by the sum of the degrees of (some) k honest workers, say H_k, in semi-matching M. Further, we can bound $|T_{\text{aff}}(H)|$ by the sum of the degrees of honest workers who make mistakes, in the semi-matching M. By assumption, there are at most $\varepsilon \cdot |H|$ such workers. Now if any of these workers belong to H_k, then we have already accounted for their degree. Consequently, we have that $|T_{\text{aff}}(H)| + |T_{\text{aff}}(A)|$ is upper bounded by the sum of the degrees of the top $k + \varepsilon \cdot |H|$ honest workers in the semi-matching M, which establishes the result. ■

Since the above lemmas are true for *any* choice of semi-matching M, they hold in particular for the optimal semi-matching on $\mathcal{B}^{cs}$. Therefore, it gives us an upper bound on the number of affected tasks when the adversaries only provide incorrect responses.

One adversary provides correct responses. Next consider the case when there is exactly 1 adversary that provides correct responses and all other adversaries only provide incorrect responses. Let $\mathcal{M}$ be the optimal semi-matching on the bipartite graph $\mathcal{B}^{cs}$ resulting from such an adversary strategy and let $\bar{a}$ denote the adversary who provides correct responses. Observe that we can repeat the argument from lemma 27 above to get an upper bound on the number of affected tasks that the adversaries "win" against honest workers as well as those that honest workers who make mistakes "win" against other honest workers. Let T_1 be the collection of such tasks. In the proof of lemma 26, there are two possible scenarios: either we obtain a perfect matching between the k adversaries and some k honest workers in which case we have accounted for all of the affected tasks that come from adversaries winning (against honest workers or $\bar{a}$). In the other scenario, when the degenerate component is non-empty, we have at most $k-1$ honest workers on the right and we bound the number of tasks that adversaries "win" by the sum of the degrees of these honest workers. Note however that we may be missing out on some of the affected tasks, namely those that the adversary $\bar{a}$ "loses" against *other adversaries* (the losses against honest workers who make mistakes are already accounted for). The tasks that we might be missing out on correspond exactly to the task copies in E that the adversary $\bar{a}$ is mapped to in the optimal semi-matching $\mathcal{M}$.

Next observe that in both scenarios above—all adversaries provide incorrect responses and exactly one adversary provides correct responses—we can upper bound the total number of affected tasks by the sum of the degrees of some $k + \varepsilon \cdot |H|$ workers in the optimal semi-matching $\mathcal{M}$ restricted to just the task copies $E \cup F'$ on the right (since honest workers provide responses only on these tasks), which we denote as $\mathcal{M}(E \cup F')$. Lemma 25 tells us that $\mathcal{M}(E \cup F')$ is, in fact, the optimal semi-matching on the subgraph $\mathcal{B}^{cs}(E \cup F')$ between workers W and the task copies $E \cup F'$. In addition, lemma 24 tells us that this sum is at most $\sum_{i=1}^{k+\varepsilon \cdot |H|} d_i$ (by starting with $\mathcal{M}_H$ as a feasible semi-matching) and the bound follows.

Part 2b. Adversaries can provide arbitrary responses

Consider the general case when all adversaries can provide arbitrary responses. First recall that lemma 27 was applicable to any semi-matching and in fact, we can use the argument even when adversaries provide correct responses. Formally, consider an arbitrary adversary strategy resulting in an optimal semi-matching $\mathcal{M}$ on $\mathcal{B}^{cs}$. Let $\mathcal{M}(E \cup F')$ denote the semi-matching M restricted to just the task copies $E \cup F'$. Suppose that the set of affected tasks T_{aff} under this adversary strategy is such that $T_{\text{aff}} = T_{\text{aff}}(A_1) \cup T_{\text{aff}}(H) \cup T_{\text{aff}}(A_2)$ where $T_{\text{aff}}(A_1)$ are the tasks that the adversaries "win" against honest workers, $T_{\text{aff}}(H)$ are the tasks that honest workers who make mistakes "win" (against other honest workers and/or adversaries) and $T_{\text{aff}}(A_2)$ are the tasks that are affected when 2 adversaries are compared against each other in the final step of the penalty-based aggregation algorithm. We can then utilize the argument in lemma 27 to bound $|T_{\text{aff}}(H)| + |T_{\text{aff}}(A_1)|$ by the sum of the degrees of (some) $k + \varepsilon \cdot |H|$ honest workers in the optimal semi-matching $\mathcal{M}$ (the

tasks affected when honest workers who make mistakes win against adversaries are also accounted). Further, we can bound $|T_{\text{aff}}(A_2)|$ by the sum of the degrees of the adversaries in the semi-matching $\mathcal{M}(E)$, which is the semi-matching $\mathcal{M}$ restricted to task copies E.

Let $A(H) \subseteq A$ denote the adversaries that have non-zero degrees in semi-matching $\mathcal{M}(E)$, i.e. they are mapped to some task copy in E in semi-matching $\mathcal{M}$. The above sequence of claims implies that we can bound the number of affected tasks $|T_{\text{aff}}|$ by the sum of the degrees of the top $s = k + \varepsilon \cdot |H| + |A(H)|$ workers in the semi-matching $\mathcal{M}(E \cup F')$, which is $\mathcal{M}$ restricted to the task copies $E \cup F'$ (because honest workers, by assumption, only provide responses on task copies $E \cup F'$ and the semi-matching $\mathcal{M}(E) \subseteq \mathcal{M}(E \cup F')$). Now, we claim that this itself is upper bounded by the sum of the degrees of the top s honest workers in the optimal semi-matching $\mathcal{M}_H$ on the bipartite graph $\mathcal{B}^{cs}_H$. To see this, start with $\mathcal{M}_H$ as a feasible semi-matching from workers W to task copies $E \cup F'$ (recall that it is feasible since we assume that each task has at least one correct response from honest workers). Then, lemma 24 tells us that the sum of the degrees of the top s workers in the optimal semi-matching on $\mathcal{B}^{cs}(E \cup F')$ is at most the sum of the degrees of the top s honest workers in $\mathcal{M}_H$. Further, lemma 25 tells us that the optimal semi-matching on the subgraph $\mathcal{B}^{cs}(E \cup F')$ is precisely the semi-matching $\mathcal{M}(E \cup F')$. This shows that we can bound the number of affected tasks by $\sum_{i=1}^{s} d_i$. Finally, note that $|A(H)| \leq k \Rightarrow s \leq 2k + \varepsilon \cdot |H|$ and hence, we can bound the total number of affected tasks by $\sum_{i=1}^{2k+\varepsilon\cdot|H|} d_i$.

A.7 Uniqueness of degree-sequence in optimal semi-matchings

In the arguments above, we have implicitly assumed some sort of uniqueness for the optimal semi-matching on any bipartite graph. Clearly its possible to have multiple optimal semi-matchings for a given bipartite graph. However, we prove below that the degree sequence of the vertices is unique across all optimal semi-matchings and hence our bounds still hold without ambiguity.

Lemma 28 *Let M and M' be two optimal semi-matchings on a bipartite graph $G = (V_1, V_2, E)$ with $|V_1| = n$ and let $d_1 \geq d_2 \cdots \geq d_n$ and $d'_1 \geq d'_2 \geq \cdots d'_n$ be the degree sequence for the V_1-vertices in M and M' respectively. Then, $d_i = d'_i \ \forall\ 1 \leq i \leq n$, or in other words, any two optimal semi-matchings have the same degree sequence.*

Proof Let s be the smallest index such that $d_s \neq d'_s$, note that we must have $s < n$ since we have that $\sum_{j=1}^{n} d'_j = \sum_{j=1}^{n} d_j$. This means that we have $d_j = d'_j\ \forall j < s$. Without loss of generality, assume that $d'_s > d_s$. Now, $\exists q \in \mathbb{N}$ such that $d'^q_s > d^q_s + \sum_{j=s+1}^{n} d^q_j$ and since $d_j = d'_j\ \forall j < s$, we have that $\sum_{j=1}^{n} d'^q_j \geq \sum_{j=1}^{l} d'^q_j > \sum_{j=1}^{n} d^q_j$. But, this is a contradiction since an optimal semi-matching minimizes the $\mathcal{L}_p$ norm of the degree-vector of V_1-vertices for any $p \geq 1$ (Section 3.4 in Harvey et al. (2003)). Hence, we have that $d_i = d'_i\ \forall i$. ■

A.8 Relation between Lower bound L and Optimal semi-matching degrees

We prove here the relationship between the lower bound L in theorem 14 and the honest worker degrees $d_1, d_2, \ldots, d_{|H|}$ in the optimal semi-matching on the bipartite graph $\mathcal{B}^{cs}_H$.

Lemma 29 *Suppose that honest workers are perfectly reliable and let $d_1 > d_2 > \cdots > d_{|H|}$ denote the degrees of the honest workers in the optimal semi-matching $\mathcal{M}_H$ on $\mathcal{B}^{cs}_H$. Then the lower bound L in theorem 14 is such that $L \geq \sum_{i=1}^{k-1} d_i$.*

Proof Let $T_1, T_2, \ldots, T_{k-1}$ denote the set of tasks that are mapped to honest workers $h_1, h_2, \ldots, h_{k-1}$ in $\mathcal{M}_H$ and $T := \bigcup_{j=1}^{k-1} T_j$. Now, we claim that for any $t \in T$, the only honest workers that provide responses for t are amongst $h_1, h_2, \ldots, h_k$. In other words, $\text{PreIm}(T) \subseteq \{h_1, h_2, \ldots, h_k\}$. Suppose not, so that there exists $h_i \in \text{PreIm}(T)$ such that $i > k$. This would contradict the fact that $\mathcal{M}_H$ is an optimal semi-matching. Specifically, Theorem 3.1 in Harvey et al. (2003) shows that a semi-matching M is optimal if and only if there is no *cost-reducing path* relative to M. A cost-reducing path $P = (h^{(1)}, t^{(1)}, h^{(2)}, \ldots h^{(d)})$ for a semi-matching M on $\mathcal{B}^{cs}_H$ is an alternating sequence of honest workers and tasks such that $t^{(x)}$ is mapped to $h^{(x)}$ in the semi-matching M for all $1 \leq x \leq d-1$ and $deg_M(h^{(1)}) > deg_M(h^{(d)}) + 1$. Here $deg_M()$ denotes the degree of a node in the semi-matching M (see section 2.1 in Harvey et al. (2003) for a precise definition). Since $i > k$, we have that $d_s > d_i + 1$ for all $s \in \{1, 2, \ldots, k-1\}$, which introduces a cost-reducing path. Therefore, we have that $\text{PreIm}(T) \subseteq \{h_1, h_2, \ldots, h_k\}$. Now

$$|T| = |\bigcup_{j=1}^{k-1} T_j| = \sum_{j=1}^{k-1} |T_j| = \sum_{j=1}^{k-1} d_j$$

where we have used the property of a semi-matching that a given task is mapped to only one worker. Using the definition of the lower bound L, it follows that $L \geq |T| = \sum_{i=1}^{k-1} d_i$. ■

Appendix B. Experimental Details

In this section we describe the details of our experimental setup discussed in section 5. We start with the benchmark algorithms.

EM algorithm. We consider the EM algorithm proposed by Raykar and Yu (2012). The worker model they consider is as follows: for each worker w_i, her accuracy is modeled separately for positive and negative tasks (referred to as the "two-coin" model). For a task t_j with true label $+1$, the *sensitivity* (true positive rate) for worker w_i is defined as:

$$\alpha_i := \Pr[w_i(t_j) = +1 \,\Big|\, y_j = +1]$$

Similarly, the *specificity* (1- false positive rate) is defined as:

$$\beta_i := \Pr[w_i(t_j) = -1 \,\Big|\, y_j = -1]$$

Let $\boldsymbol{\Theta} = [\left\{(\alpha_i, \beta_i) \,\Big|\, i \in [n]\right\}, \gamma]$ denote the set of all parameters. For ease of exposition, we assume that the worker-task assignment graph is complete, i.e. all workers provide responses for all items, but the algorithm can be immediately extended to the case of

incomplete graphs. Given the response matrix $\mathcal{L}$, the log-likelihood of the parameters Θ can be written as:

$$\log Pr[\mathcal{L} \Big| \boldsymbol{\Theta}] = \sum_{j=1}^{m} \log \left(\prod_{i=1}^{n} \alpha_i^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot (1-\alpha_i)^{\mathbf{1}[\mathcal{L}_{ij}=-1]} \cdot \gamma + \prod_{i=1}^{n} (1-\beta_i)^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot \beta_i^{\mathbf{1}[\mathcal{L}_{ij}=-1]} \cdot (1-\gamma) \right)$$

The MLE of the parameters can be computed by introducing the latent true label of each task, denoted by the vector $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]$. The complete data log-likelihood can then be written as:

$$\log Pr[\mathcal{L}, \boldsymbol{y} \Big| \boldsymbol{\Theta}] = \sum_{j=1}^{m} \Big(y_j \log(a_j \gamma) + (1-y_j) \log(1-\gamma) b_j \Big) \tag{11}$$

where

$$a_j = \prod_{i=1}^{n} \alpha_i^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot (1-\alpha_i)^{\mathbf{1}[\mathcal{L}_{ij}=-1]}$$
$$b_j = \prod_{i=1}^{n} (1-\beta_i)^{\mathbf{1}[\mathcal{L}_{ij}=+1]} \cdot \beta_i^{\mathbf{1}[\mathcal{L}_{ij}=-1]}$$

Each iteration of the EM algorithm consists of two steps:

- **E-step:** Given the response matrix $\mathcal{L}$ and the current estimate of the model parameters $\boldsymbol{\Theta}^{(k)}$, the conditional expectation of the complete data log-likelihood is computed as

  $$\mathbb{E}\left\{ \log \Pr[\mathcal{L}, \boldsymbol{y} \Big| \boldsymbol{\Theta}^{(k)}] \right\} = \sum_{j=1}^{m} \Big(\gamma_j^{(k)} \log(a_j^{(k)} \gamma^{(k)}) + (1-\gamma_j^{(k)}) \log(1-\gamma^{(k)}) b_j^{(k)} \Big)$$

  where the expectation is w.r.t to $\Pr[\boldsymbol{y} \Big| \mathcal{L}; \boldsymbol{\Theta}^{(k)}]$ and $\gamma_j^{(k)} = \Pr[y_j = +1 \Big| \mathcal{L}; \boldsymbol{\Theta}^{(k)}]$. Using Bayes theorem, we can compute

  $$\gamma_j^{(k)} \propto \Pr[\mathcal{L}_{1j}, \mathcal{L}_{2j}, \ldots, \mathcal{L}_{nj} \Big| y_j = +1; \boldsymbol{\Theta}^{(k)}] \cdot \Pr[y_j = +1 \Big| \boldsymbol{\Theta}^{(k)}] = \frac{a_j^{(k)} \gamma^{(k)}}{a_j^{(k)} \gamma^{(k)} + b_j^{(k)} (1-\gamma^{(k)})}$$

- **M-step:** Based on the current posterior estimates of the true labels $\gamma_j^{(k)}$ and the response matrix $\mathcal{L}$, the model parameters are updated by maximizing $\mathbb{E}\left\{ \log \Pr[\mathcal{L}, \boldsymbol{y} \Big| \boldsymbol{\Theta}^{(k)}] \right\}$, which can be shown to be a lower bound on the data log-likelihood (eq 11). The prevalence of positive tasks γ is updated as:

  $$\gamma^{(k+1)} = \frac{\sum_{j=1}^{m} \gamma_j^{(k)}}{m}$$

Similarly, the parameters α_i, β_i are updated as:

$$\alpha_i^{(k+1)} = \frac{\sum_{j=1}^{m} \mathbf{1}[\mathcal{L}_{ij} = +1] \gamma_j^{(k)}}{\sum_{j=1}^{m} \gamma_j^{(k)}}$$
$$\beta_i^{(k+1)} = \frac{\sum_{j=1}^{m} \mathbf{1}[\mathcal{L}_{ij} = -1] (1-\gamma_j^{(k)})}{\sum_{j=1}^{m} (1-\gamma_j^{(k)})}$$

These two steps are iterated until convergence of the log-likelihood $\Pr[\mathcal{L} \Big| \boldsymbol{\Theta}]$. To initialize the EM algorithm, we use the majority estimate $\gamma_j^{(0)} = \frac{\sum_{i=1}^{n} \mathbf{1}[\mathcal{L}_{ij}=+1]}{n}$.

We implement the above algorithm in our experiments. Note that the model we consider in the theoretical analysis is the simpler "one-coin" model where every worker is characterized by only a single parameter μ_i - the probability that she labels an assigned task correctly. The EM algorithm for that case can be derived in a similar manner to the one described above; see the proof of Lemma 13 above.

KOS algorithms. We implemented the iterative algorithm presented in Karger et al. (2014) which we replicate below in our notation.

Algorithm 4 KOS algorithm

1: **Input:** $\mathcal{L}$,$\mathcal{B} = (W, \mathcal{T}, E)$, $k_{\max}$.
2: For all $(w_i, t_j) \in E$, initialize $m_{i \to j}^{(0)}$ with random $Z_{ij} \sim \mathcal{N}(1,1)$
3: For $k = 1, 2, \ldots, k_{\max}$,

- For all $(w_i, t_j) \in E$, update $m_{j \to i}^{(k)} = \sum_{i' \neq i} \mathcal{L}_{ij} m_{i \to j}^{(k-1)}$
- For all $(w_i, t_j) \in E$, update $m_{i \to j}^{(k)} = \sum_{j' \neq j} \mathcal{L}_{ij} m_{j \to i}^{(k)}$

4: For all t_j, compute $m_j = \sum_{i=1}^{n} \mathcal{L}_{ij} m_{i \to j}^{(k_{\max}-1)}$
5: **Output:** label for task t_j as $\hat{y}_j = \text{sign}(\text{m}_\text{j})$

This algorithm was proposed for random regular graphs in the paper and we modified it in the following way for use in non-regular graphs:

Algorithm 5 KOS(NORM) algorithm

1: **Input:** $\mathcal{L}$,$\mathcal{B} = (W, \mathcal{T}, E)$, $k_{\max}$.
2: For all $(w_i, t_j) \in E$, initialize $m_{i \to j}^{(0)}$ with random $Z_{ij} \sim \mathcal{N}(1,1)$
3: For $k = 1, 2, \ldots, k_{\max}$,

- For all $(w_i, t_j) \in E$, update $m_{j \to i}^{(k)} = \frac{1}{deg_{\mathcal{B}}(t_j)} \sum_{i' \neq i} \mathcal{L}_{ij} m_{i \to j}^{(k-1)}$
- For all $(w_i, t_j) \in E$, update $m_{i \to j}^{(k)} = \frac{1}{deg_{\mathcal{B}}(w_i)} \sum_{j' \neq j} \mathcal{L}_{ij} m_{j \to i}^{(k)}$

4: For all t_j, compute $m_j = \sum_{i=1}^{n} \mathcal{L}_{ij} m_{i \to j}^{(k_{\max}-1)}$
5: **Output:** label for task t_j as $\hat{y}_j = \text{sign}(\text{m}_\text{j})$

We chose $k_{\max} = 100$ in our experiments.

Simulation Details. Here we discuss how we imposed the degree bias on adversaries in our simulation study. Given a worker-task assignment graph, let d_w denote the degree of worker w and $d_{\min}, d_{\text{avg}}, d_{\max}$ denote resp. the minimum, average and maximum worker degrees.

- *Adversaries have high degrees*. For each worker w, define $q_w = q \cdot \frac{d_{\max} - d_w}{d_{\max} - d_{\text{avg}}}$. Then, each worker w is an adversary with probability $1 - q_w$. First, note that the expected number of honest workers is given by

$$\sum_w q_w = \frac{q}{d_{\max} - d_{\text{avg}}} \sum_w (d_{\max} - d_w) = q \cdot n$$

  Next, we can see that workers with higher degrees have a smaller q_w, which implies that they have a greater chance of being an adversary. In an analogous manner, we deal with the case of low degrees.

- *Adversaries have low degrees*. For each worker w, define $q_w = q \cdot \frac{d_w - d_{\min}}{d_{\text{avg}} - d_{\min}}$. Then, each worker w is an adversary with probability $1 - q_w$. Again, we have that the expected number of honest workers is given by

$$\sum_w q_w = \frac{q}{d_{\text{avg}} - d_{\min}} \sum_w (d_w - d_{\min}) = q \cdot n$$

  In this case, lower the degree d_w, higher the chance $(1 - q_w)$ that a worker is chosen as an adversary.

References

Béla Bollobás. *Random Graphs*. Cambridge University Press, 2001.

Xi Chen, Qihang Lin, and Dengyong Zhou. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 64–72, 2013.

Nilesh Dalvi, Anirban Dasgupta, Ravi Kumar, and Vibhor Rastogi. Aggregating crowdsourced binary ratings. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 285–294. ACM, 2013.

George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *Proceedings of the Network and Distributed System Security Symposium*, 2009.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, pages 20–28, 1979.

Djellel Eddine Difallah, Gianluca Demartini, and Philippe Cudré-Mauroux. Mechanical cheat: Spamming schemes and adversarial techniques on crowdsourcing platforms. In *Proceedings of the First International Workshop on Crowdsourcing Web Search*, pages 26–30, 2012.

Julie S Downs, Mandy B Holbrook, Steve Sheng, and Lorrie Faith Cranor. Are your participants gaming the system?: Screening mechanical turk workers. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, pages 2399–2402. ACM, 2010.

Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. Crowddb: Answering queries with crowdsourcing. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, pages 61–72. ACM, 2011.

Ujwal Gadiraju, Ricardo Kawase, Stefan Dietze, and Gianluca Demartini. Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 1631–1640. ACM, 2015.

Chao Gao and Dengyong Zhou. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. *arXiv preprint arXiv:1310.5764v6*, 2016.

Arpita Ghosh, Satyen Kale, and Preston McAfee. Who moderates the moderators? crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pages 167–176. ACM, 2011.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, August 2008.

Nicholas JA Harvey, Richard E Ladner, László Lovász, and Tami Tamir. Semi-matchings for bipartite graphs and load balancing. In *Algorithms and Data Structures*, pages 294–306. Springer, 2003.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. Learning whom to trust with mace. In *Proceedings of NAACL-HLT*, pages 1120–1130, 2013.

Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 64–67. ACM, 2010.

Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Reputation-based worker filtering in crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 2492–2500, 2014.

Ece Kamar, Ashish Kapoor, and Eric Horvitz. Identifying and accounting for task-dependent bias in crowdsourcing. In *Third AAAI Conference on Human Computation and Crowdsourcing*, pages 92–101, 2015.

David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in Neural Information Processing Systems*, pages 1953–1961, 2011.

David R Karger, Sewoong Oh, and Devavrat Shah. Efficient crowdsourcing for multi-class labeling. In *Proceedings of ACM SIGMETRICS*, pages 81–92. ACM, 2013.

David R Karger, Sewoong Oh, and Devavrat Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2014.

Ashish Khetan and Sewoong Oh. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 4844–4852, 2016.

Aniket Kittur, Ed H Chi, and Bongwon Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 453–456. ACM, 2008.

John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *Proc. ACM SIGIR Workshop on Crowdsourcing for Search Evaluation*, 2010.

Hongwei Li and Bin Yu. Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086*, 2014.

Hongwei Li, Bo Zhao, and Ariel Fuxman. The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 165–176. ACM, 2014.

Qiang Liu, Jian Peng, and Alex Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 701–709, 2012.

Arash Molavi Kakhki, Chloe Kliman-Silver, and Alan Mislove. Iolaus: Securing online content rating systems. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 919–930. ACM, 2013.

Marti Motoyama, Damon McCoy, Kirill Levchenko, Stefan Savage, and Geoffrey M Voelker. Dirty jobs: The role of freelance labor in web service abuse. In *Proceedings of the 20th USENIX conference on Security*, pages 14–29. USENIX Association, 2011.

Barzan Mozafari, Purna Sarkar, Michael Franklin, Michael Jordan, and Samuel Madden. Scaling up crowd-sourcing to very large datasets: A case for active learning. *Proceedings of the VLDB Endowment*, 8(2):125–136, 2014.

Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st International Conference on World Wide Web*, pages 191–200. ACM, 2012.

Jungseul Ok, Sewoong Oh, Jinwoo Shin, and Yung Yi. Optimality of belief propagation for crowdsourced classification. In *Proceedings of the 33rd International Conference on Machine Learning (ICML-16)*, pages 535–544, 2016.

Aditya Ramesh, Aditya Parameswaran, Hector Garcia-Molina, and Neoklis Polyzotis. Identifying reliable workers swiftly. 2012.

Vikas C Raykar and Shipeng Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, 13:491–518, 2012.

Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 614–622. ACM, 2008.

Padhraic Smyth, Usama Fayyad, Michael Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of venus images. In *Advances in Neural Information Processing Systems*, pages 1085–1092, 1995.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: Evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263. Association for Computational Linguistics, 2008.

Wei Tang and Matthew Lease. Semi-supervised consensus labeling for crowdsourcing. In *SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR)*, pages 1–6, 2011.

Nguyen Tran, Bonan Min, Jinyang Li, and Lakshminarayanan Subramanian. Sybil-resilient online content voting. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, pages 15–28. USENIX Association, 2009.

Nguyen Tran, Jinyang Li, Lakshminarayanan Subramanian, and Sherman SM Chow. Optimal sybil-resilient node admission control. In *Proceedings of IEEE INFOCOM*, pages 3218–3226. IEEE, 2011.

Salil P Vadhan et al. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review*, 40:363–374, 2010.

Bimal Viswanath, Mainack Mondal, Krishna P Gummadi, Alan Mislove, and Ansley Post. Canal: Scaling social network-based sybil tolerance schemes. In *Proceedings of the 7th ACM European Conference on Computer Systems*, pages 309–322. ACM, 2012.

Jeroen Vuurens, Arjen P de Vries, and Carsten Eickhoff. How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *Proc. ACM SIGIR Workshop on Crowdsourcing for Information Retrieval (CIR11)*, pages 21–26, 2011.

Gang Wang, Christo Wilson, Xiaohan Zhao, Yibo Zhu, Manish Mohanlal, Haitao Zheng, and Ben Y Zhao. Serf and turf: Crowdturfing for fun and profit. In *Proceedings of the 21st International Conference on World Wide Web*, pages 679–688. ACM, 2012.

Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y Zhao. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *23rd USENIX Security Symposium*. USENIX Association, 2014.

Naiyan Wang and Dit-Yan Yeung. Ensemble-based tracking: Aggregating crowdsourced structured time series data. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1107–1115, 2014.

Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems*, pages 2424–2432, 2010.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems*, pages 2035–2043, 2009.

Haifeng Yu, Michael Kaminsky, Phillip B Gibbons, and Abraham Flaxman. Sybilguard: Defending against sybil attacks via social networks. *ACM SIGCOMM Computer Communication Review*, 36:267–278, 2006.

Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, pages 3–17, 2008.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems*, pages 1260–1268, 2014.

Dengyong Zhou, Sumit Basu, Yi Mao, and John C Platt. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems*, pages 2195–2203, 2012.

Dengyong Zhou, Qiang Liu, John C Platt, Christopher Meek, and Nihar B Shah. Regularized minimax conditional entropy for crowdsourcing. *arXiv preprint arXiv:1503.07240*, 2015.

Journal of Machine Learning Research 18 (2017) 1-38 Submitted 4/16; Revised 7/17; Published 9/17

An Easy-to-hard Learning Paradigm for Multiple Classes and Multiple Labels

Weiwei Liu liuweiwei863@gmail.com
School of Computer Science and Engineering
The University of New South Wales, Sydney
NSW 2052, Australia
and
Centre for Artificial Intelligence
FEIT, University of Technology Sydney
NSW 2007, Australia

Ivor W. Tsang ivor.tsang@uts.edu.au
Centre for Artificial Intelligence
FEIT, University of Technology Sydney
NSW 2007, Australia

Klaus-Robert Müller klaus-robert.mueller@tu-berlin.de
Machine Learning Group, Computer Science
Berlin Institute of Technology (TU Berlin)
Marchstr 23, 10587 Berlin, Germany
and
Max Planck Institute for Informatics
Stuhlsatzenhausweg, 66123, Saarbrcken
and
Department of Brain and Cognitive Engineering
Korea University, Seoul 02841, Korea

Editor: Karsten Borgwardt

Abstract

Many applications, such as human action recognition and object detection, can be formulated as a multiclass classification problem. One-vs-rest (OVR) is one of the most widely used approaches for multiclass classification due to its simplicity and excellent performance. However, many confusing classes in such applications will degrade its results. For example, *hand clap* and *boxing* are two confusing actions. *Hand clap* is easily misclassified as *boxing*, and vice versa. Therefore, precisely classifying confusing classes remains a challenging task. To obtain better performance for multiclass classifications that have confusing classes, we first develop a classifier chain model for multiclass classification (CCMC) to transfer class information between classifiers. Then, based on an analysis of our proposed model, we propose an easy-to-hard learning paradigm for multiclass classification to automatically identify easy and hard classes and then use the predictions from simpler classes to help solve harder classes. Similar to CCMC, the classifier chain (CC) model is also proposed by Read et al. (2009) to capture the label dependency for multi-label classification. However, CC does not consider the order of difficulty of the labels and achieves degenerated performance when there are many confusing labels. Therefore, it is non-trivial to learn the appropriate label order for CC. Motivated by our analysis for CCMC, we also propose the easy-to-hard learning paradigm for multi-label classification to automatically identify easy and hard labels, and then use the predictions from simpler labels to help solve harder labels. We also demonstrate that our proposed strategy can be successfully applied to a wide range of applications, such as ordinal classification and relationship prediction. Extensive empirical studies validate our analysis and the effectiveness of our proposed easy-to-hard learning strategies.

Keywords: Multiclass Classification, Multi-label Classification, Classifier Chain, Easy-to-hard Learning Paradigm

1. Introduction

Many applications can be formulated as a multiclass classification problem. For example, human action recognition aims to classify videos into different categories of human action. In the KTH data set (Schüldt et al., 2004), for example, there are six types of human action: *walk*, *jog*, *run*, *hand wave*, *hand clap* and *boxing*. These actions can be classified into two main categories: leg movements (*walk*, *jog* and *run*) and hand movements (*hand wave*, *hand clap* and *boxing*). It is easy to differentiate between leg and hand movements, such as *walk* and *hand wave*, but actions within leg or hand movements, such as hand clap and boxing, are easily confused. *Hand clap* is easily misclassified as *boxing*, and vice versa. Confusing classes are ubiquitous in real world applications, especially for data sets with many classes. For example, there are many confusing classes in the ALOI data set from the LIBSVM website[1], which contains 1,000 classes. Therefore, precisely classifying multiclass data sets with confusing classes is a challenging task.

From the confusion matrix of one-vs-rest (OVR) on the KTH data set shown in Figure 1, we observe that *walk* is the easiest action to classify and *hand clap* is the hardest action to classify, as the *walk* action can be correctly identified by OVR whereas the percentage of *hand clap* images that are misclassified as *boxing* is 22.5%. To achieve accurate prediction performance, according to Figure 1, we should classify *walk* first, followed by *run*, *jog* and *hand wave*. *Boxing* and *hand clap* are the last two actions to classify. The motivation behind this paper is to solve classification tasks from easy to hard, and to use the predictions from simpler tasks to help solve the harder tasks.

To achieve our goal, a classifier chain model for multiclass classification (CCMC) is proposed to transfer class information between classifiers. Furthermore, we generalize CCMC over a random class order and provide a theoretical analysis of the generalization error for the proposed generalized model. Our results show that the upper bound of the generalization error depends on the sum of the reciprocal of the square of the margin over the classes. Therefore, we conclude that class order does affect the performance of CCMC, and a globally optimal class order exists only when the minimization of the upper bound is achieved over this CCMC. Lastly, based on our results, we propose the easy-to-hard learning paradigm for multiclass classification to automatically identify easy and hard classes and then use the predictions from simpler classes to help solve harder classes.

Multi-label classification, where each instance can belong to multiple labels simultaneously, has garnered significant attention from researchers as a result of its various applica-

1. https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets

©2017 Weiwei Liu, Ivor W. Tsang and Klaus-Robert Müller.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-212.html.

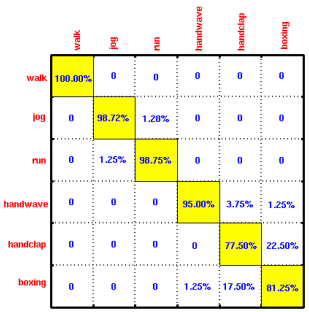


Figure 1: Confusion Matrix of OVR on the KTH Data Set. In the confusion matrix, the entry in the i-th row and j-th column is the percentage of images from action i that are misclassified as action j. Average classification accuracy rates for individual actions are listed along the diagonal, which is colored yellow.

tions, which range from document classification and gene function prediction, to automatic image annotation. For example, a document can be associated with a range of topics, such as *Sports*, *Finance* and *Education* (Schapire and Singer, 2000); a gene belongs to the functions of *protein synthesis*, *metabolism* and *transcription* (Barutcuoglu et al., 2006); an image may have both *beach* and *tree* tags (Boutell et al., 2004).

Similar to CCMC, a classifier chain (CC) model is also proposed by Read et al. (2009) to capture the label dependency for multi-label classification. It also tries to use information from previous labels to help train the classifier for the next label. However, CC's performance degenerates when there are many confusing labels, because the main drawback of CC is that it does not consider the order of difficulty of the labels. Therefore, it is non-trivial to learn the appropriate label order for CC.

Motivated by our analysis for CCMC, we first generalize CC over a random label order and provide the generalization error bound for the proposed generalized model. Then we propose the easy-to-hard learning paradigm for multi-label classification to automatically identify easy and hard labels. Lastly, we use the predictions from simpler labels to help solve harder labels. To learn the objective of our proposed easy-to-hard learning paradigms, it is very expensive to search over $q!$ different class or label orders[2], where q denotes the number of classes or labels, which is computationally infeasible for a large q. We thus propose a set of easy-to-hard learning algorithms to simplify the search process of the optimal learning sequence.

Experiments on a wide spectrum of data sets show that our proposed methods excel in all data sets for multi-label and multiclass classification problems. The results validate our analysis and the effectiveness of our proposed easy-to-hard learning algorithms. Lastly, we demonstrate that our proposed easy-to-hard learning strategies can be successfully applied

2. ! represents the factorial notation.

to a wide range of applications, such as ordinal classification (Chu and Ghahramani, 2005) and relationship prediction (Massa and Avesani, 2006).

We organize this paper as follows. Section 2 summarizes existing related works and problems. The easy-to-hard learning paradigm for multi-label and multiclass classification are proposed in Sections 3 and 4. Learning algorithms and time complexity analysis are described in Section 5. We present two applications in Section 6. Section 7 shows the comprehensive experimental results. The last section provides concluding remarks.

Notations: Assume $\mathbf{x}_t \in \mathbb{R}^d$ is a real vector representing an input or instance (feature) for $t \in \{1, \cdots, n\}$. n denotes the number of training instances. $\mathcal{Y}_t \subseteq \{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ is the corresponding output (class or label). $\mathbf{y}_t \in \{0,1\}^q$ is used to represent the set $\mathcal{Y}_t$, where $\mathbf{y}_t(j) = 1$ if and only if $\lambda_j \in \mathcal{Y}_t$. Note that, there is only one element in $\mathcal{Y}_t$ for the multiclass problem.

2. Related Work and Problems

2.1 Multiclass Classification

2.1.1 OVR and OVO

One-vs-rest (OVR) and one-vs-one (OVO) are two famous strategies for decomposing multiclass classification problems into multiple binary classification problems. Hsu and Lin (2002), Rifkin and Klautau (2004), Demirkesen and Cherifi (2008) and Lapin et al. (2015, 2016) have already shown that OVR and OVO are successful schemes that are as accurate as more complicated approaches, such as error-correcting output coding (ECOC) (Dietterich and Bakiri, 1995), the tree-based method (Beygelzimer et al., 2009a,b; Bengio et al., 2010; Yang and Tsang, 2011; Liu and Tsang, 2016) and multi-class SVM (Weston and Watkins, 1999; Lapin et al., 2015, 2016).

OVR works as follows: a binary classifier is trained for each class λ_j, with all of the instances in the j-th class having positive labels, and all other instances having negative labels. q binary classifiers are then trained on $\{\mathbf{x}_t, \mathbf{y}_t(1)\}_{t=1}^n, \cdots, \{\mathbf{x}_t, \mathbf{y}_t(q)\}_{t=1}^n$. The final output of OVR for each testing instance is the class that corresponds to the classifier with the highest output value. OVR ignores correlations between classes and each classifier is trained independently. OVO trains all possible $q(q-1)/2$ binary classifiers from a training set of q classes, where each classifier is trained on only two out of q classes.

The main differences between OVR and OVO relate to computational issues and applications: 1) Computational issues: OVO requires $\mathcal{O}(q^2)$ classifiers, while OVR trains $\mathcal{O}(q)$ classifiers. If q is very large, then the cost of OVO will be prohibitive. 2) Applications: many real world data sets are partially labelled, as a result of the heavy burden of labelling data. We call this kind of data "background data" (See also (Niu et al., 2016)). It arises in many applications, for example, multiclass image segmentation (Guillaumin et al., 2010; Pham et al., 2015), object detection (Torralba et al., 2004; Huo et al., 2016) and multi-class video segmentation (Budvytis et al., 2010). OVR can be used both in supervised and semi-supervised settings, while OVO can not be used in semi-supervised settings. Thus, OVR can be used to tackle this kind of background data, but OVO cannot. Milgram et al. (2006) have already shown that OVR appears to be significantly more accurate than OVO for handwriting recognition.

2.1.2 Rifkin and Klautau's conjecture

OVR is a lower cost approach with many more applications than OVO. However, Rifkin and Klautau (2004), who conduct a thorough study on OVR, point out that the condition of OVR working as well as any other clever schemes is that the classes are independent - we do not necessarily expect instances from class "A" to be closer to those in class "B" than those in class "C". They also speculate that an algorithm that exploits the relationship between classes could offer superior performance, and that this would remains an open problem. To the best of our knowledge, this problem has still not been well-addressed, which is why the present paper studies this problem to provide an answer.

2.2 Multi-label Classification

One popular strategy for multi-label classification is to reduce the original problem into many binary classification problems. Many works have followed this strategy. For example, binary relevance (BR) (Tsoumakas et al., 2010) is a simple approach for multi-label learning which independently trains a binary classifier for each label. Recently, Chen and Lin (2012); Liu and Tsang (2015a,b); Zhang and Zhou (2014); Gong et al. (2017); Liu and Tsang (2017) have shown that multi-label learning methods that explicitly capture label dependency will usually achieve better prediction performance. Therefore, modeling label dependency is one of the major challenges in multi-label classification problems.

To capture label dependency, Hsu et al. (2009) first use the compressed sensing technique to handle multi-label classification problems. They project the original label space into a low dimensional label space. A regression model is then trained on each transformed label. Lastly, multi-labels are recovered from the regression output, which usually involves solving a quadratic programming problem (Hsu et al., 2009). Many works have been developed in this way (Zhang and Schneider, 2011, 2012; Tai and Lin, 2012). Such methods mainly aim to use different projection methods to transform the original label space into another effective label space. However, an expensive encoding and decoding procedure prevents these methods from being practical.

Another important approach attempts to exploit the different orders (first-order, second-order and high-order) of label correlations (Zhang and Zhang, 2010; Zhang and Zhou, 2014). Following this way, some works try to provide a probabilistic interpretation for label correlations. For example, Guo and Gu (2011) model the label correlations using a conditional dependency network; PCC (Dembczynski et al., 2010) exploits a high-order Markov Chain model to capture the correlations between the labels and provide an accurate probabilistic interpretation of classifier chain (CC) (Read et al., 2009). Some other works (Kang et al., 2006; Read et al., 2009; Huang and Zhou, 2012) focus on modeling the label correlations in a deterministic way. Among them, the CC model is one of the most popular methods due to its simplicity and promising experimental results (Read et al., 2009).

CC works as follows: one classifier is trained for each label. For the $(i+1)$th label, each instance is augmented with the 1st, 2nd, $\cdots$, ith label as the input to train the $(i+1)$th classifier. Given a new instance to be classified, CC firstly predicts the value of the first label, then takes this instance together with the predicted value as the input to predict the value of the next label. CC proceeds in this way until the last label is predicted. However, here the question is: *Does the label order affect the performance of CC?* Apparently yes,

because different classifier chains involve different classifiers trained on different training sets. Thus, to reduce the influence of the label order, Read et al. (2009) propose the ensemble of classifier chains (ECC) to average the multi-label predictions of CC over a set of random ordering chains. Since the performance of CC is sensitive to the choice of label, there is another important question: *Is there any globally optimal classifier chain which can achieve the optimal prediction performance for CC?* If yes, *how can the globally optimal classifier chain be found?* This paper studies this problem and provides an answer.

2.3 Curriculum Learning

Curriculum learning (Bengio et al., 2009) can be seen as a sequence of training criteria. Each training criterion in the sequence is associated with a different set of weights in the training examples, or more generally, in a re-weighting of the training distribution. Initially, the weights favor easier examples that can be learned most easily. The next training criterion involves a slight change in the weighting of examples that increases the probability of sampling slightly more difficult examples. Overall, curriculum learning aims to find easier examples. However, up to now, curriculum learning has not defined what easy examples mean, or equivalently, how to sort the examples into a sequence that illustrates the simpler concepts first.

Inspired by curriculum learning, this paper clearly defines easy and hard tasks and provides the strategy to learn easy and hard tasks. Our empirical studies verify that our method is able to automatically identify easy and hard tasks, and use the predictions of classifiers from easier tasks to train the classifier for harder tasks.

3. The Easy-to-hard Learning Paradigm for Multiclass Classification

3.1 Classifier Chain for Multiclass Classification

In multi-label classification, each instance can belong to multiple labels simultaneously, while multiclass classification classifies instances into one of more than two classes. Therefore, multiclass classification (Hsu and Lin, 2002) is a quite different learning task compared to multi-label classification (Zhang and Zhou, 2014). Many applications, such as human action recognition and object detection, can be formulated as a multiclass classification problem. However, many confusing classes in such applications will degrade the existing solver's performance. For example, *hand clap* and *boxing* are two confusing actions. *Hand clap* is easily misclassified as boxing, and vice versa. Therefore, it is non-trivial to precisely classify confusing classes for multiclass classification.

To solve these issues, motivated by CC for multi-label classification, we propose a classifier chain model for multiclass classification (CCMC) which aims to transfer class information between classifiers. CCMC trains q binary classifiers h_j ($j \in \{1, \cdots, q\}$), with all of the instances in the j-th class having positive labels, and all other instances having negative labels. Similar to CC, classifiers of CCMC are linked along a chain. CCMC works as follows: binary classifier h_1 is first trained for class λ_1, then the augmented vector $\{\mathbf{x}_t, h_1(\mathbf{x}_t)\}_{t=1}^n$ is used as the input to train classifier h_2 for class λ_2. Similarly, $\mathbf{x}_t$ augments all previous prediction values of $h_1, \cdots, h_j$ as the input to train classifier h_{j+1} for class λ_{j+1}. CCMC proceeds in this way until the last classifier h_q for class λ_q has been trained.

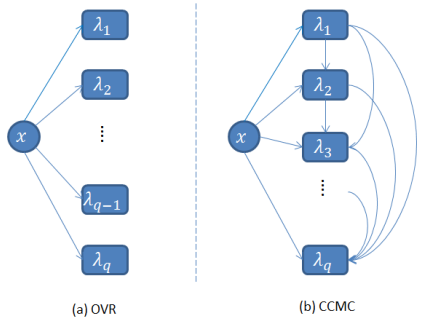


Figure 2: Schematic illustration of OVR and CCMC. A circle represents the input $\mathbf{x}$. A rectangle represents class $\lambda_i (t \in \{1, \cdots, q\})$. The starting point of the arrow denotes the input to train the classifier for the class to which the arrow points.

Given a new testing instance $\mathbf{x}$, classifier h_1 in the chain is responsible for predicting the value for λ_1 using input $\mathbf{x}$. h_2 predicts the value for λ_2 taking $\mathbf{x}$ plus the predicted value of $h_1(\mathbf{x})$ as an input. Similarly, h_{j+1} predicts the value for λ_{j+1} using $\mathbf{x}$ plus all previous prediction results from $h_1, \cdots, h_j$ as the input. CCMC proceeds in this way until the value of the last class λ_q has been predicted. Similar to OVR, the final output of CCMC for $\mathbf{x}$ is the class that corresponds to the classifier with the highest output value. CCMC exploits the class dependence by passing class information between classifiers. Figure 2 illustrates the working scheme between OVR and CCMC.

3.2 Generalized Classifier Chain for Multiclass Classification

We generalize the CCMC model over a random class order, called *generalized classifier chain for multiclass classification* (GCCMC). Assume that classes $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ are randomly reordered as $\{\zeta_1, \zeta_2, \cdots, \zeta_q\}$, where $\zeta_j = \lambda_k$ means class λ_k moves to position j from k. In the GCCMC model, classifiers are also linked along a chain where each classifier h_j deals with the binary classification problem for class ζ_j (λ_k). GCCMC follows the same training and testing procedures as CCMC, the only difference being the class order.

3.3 Analysis

In this subsection, we analyze the generalization error bound of the multiclass classification problem using GCCMC based on some techniques, such as fat shattering dimension (Kearns and Schapire, 1990) and the upper bound theorem of covering numbers (Shawe-Taylor et al., 1998).

Let $\mathbf{X}$ represent the input space. Both $\mathbf{s}$ and $\bar{\mathbf{s}}$ are m multiclass examples drawn independently according to an unknown distribution D. We denote logarithms to base 2 by log. If $\mathcal{S}$ is a set, $|\mathcal{S}|$ denotes its cardinality. $\|\cdot\|$ means the l_2 norm.

We begin with the definition of the fat shattering dimension.

Definition 1 (Kearns and Schapire (1990)) *Let $\mathcal{H}$ be a set of real valued functions. We say that a set of points P is γ-shattered by $\mathcal{H}$ relative to $r = (r_p)_{p \in P}$ if there are real numbers r_p indexed by $p \in P$ such that for all binary vectors b indexed by P, there is a function $f_b \in \mathcal{H}$ satisfying*

$$f_b(p) = \begin{cases} \geq r_p + \gamma & \text{if } b_p = 1 \\ \leq r_p - \gamma & \text{otherwise} \end{cases}$$

The fat shattering dimension $fat(\gamma)$ of the set $\mathcal{H}$ is a function from the positive real numbers to the integers which maps a value γ to the size of the largest γ-shattered set, if this is finite, or infinity otherwise.

Assume that $\mathcal{H}$ is the real valued function class and $h \in \mathcal{H}$. $l(y, h(x))$ denotes the loss function. The expected error of h is defined as $er_D[h] = E_{(x,y)\sim D}[l(y, h(x))]$, where (x, y) drawn from the unknown distribution D. Here we select 0-1 loss function. So, $er_D[h] = P_{(x,y)\sim D}(h(x) \neq y)$. The empirical risk $er_{\mathbf{s}}[h]$ is defined as $er_{\mathbf{s}}[h] = \frac{1}{n}\sum_{t=1}^{n} \mathbf{I}\big(y_t \neq h(x_t)\big)$.[3]

Suppose that $\mathcal{N}(\epsilon, \mathcal{H}, \mathbf{s})$ is the ϵ-covering number of $\mathcal{H}$ with respect to the l_∞ pseudo-metric measuring the maximum discrepancy on the example $\mathbf{s}$. The notion of the covering number can be referred to Appendix A. We introduce the following general corollary regarding the bound of the covering number:

Corollary 2 (Shawe-Taylor et al. (1998)) *Let $\mathcal{H}$ be a class of functions $X \to [a, b]$ and D a distribution over X. Choose $0 < \epsilon < 1$ and let $d = fat(\epsilon/4) \leq em$. Then*

$$E(\mathcal{N}(\epsilon, \mathcal{H}, \boldsymbol{s})) \leq 2\Big(\frac{4m(b-a)^2}{\epsilon^2}\Big)^{d\log(2em(b-a)/(d\epsilon))} \tag{1}$$

where the expectation E is over examples $\boldsymbol{s} \in X^m$ drawn according to D^m.

We study the generalization error bound of the specified GCCMC with the specified number of classes and margins. Let G be the set of classifiers of GCCMC, $G = \{h_1, h_2, \cdots, h_q\}$. $er_{\mathbf{s}}[G]$ denotes the fraction of the number of errors that GCCMC makes on $\mathbf{s}$. Define $\hat{\mathbf{x}} \in \mathbf{X} \times \{0, 1\}$, $\hat{h}_j(\hat{\mathbf{x}}) = h_j(\mathbf{x})(1 - \mathbf{y}(j)) - h_j(\mathbf{x})\mathbf{y}(j)$.

We introduce the following proposition:

Proposition 3 *If an instance $\boldsymbol{x} \in \boldsymbol{X}$ is misclassified by a GCCMC model, then $\exists h_j \in G, \hat{h}_j(\hat{\mathbf{x}}) \geq 0$.*

Proof For multiclass problem, assume that an instance $\mathbf{x}$ belongs to class ζ_i: $\mathbf{y}(i) = 1$, $\mathbf{y}(g) = 0 (\forall g \in \{1, 2, \cdots, q\}, g \neq i)$, and that it is misclassified as ζ_j. Suppose that classifier $h_j \in G$ reports the highest confidence score for this instance: $h_j(\mathbf{x})$.

3. The expression $\mathbf{I}\big(y_t \neq h(x_t)\big)$ evaluates to 1 if $y_t \neq h(x_t)$ is true and to 0 otherwise.

Case 1: $h_j(\mathbf{x}) \geq 0$. In this case, $\hat{h}_j(\hat{\mathbf{x}}) = h_j(\mathbf{x})(1-\mathbf{y}(j)) - h_j(\mathbf{x})\mathbf{y}(j) = h_j(\mathbf{x}) \geq 0$.

Case 2: $h_j(\mathbf{x}_t) < 0$. In this case, all classifiers will output negative real numbers as the result of h_j reporting the highest confidence score. Thus, $\hat{h}_i(\hat{\mathbf{x}}) = h_i(\mathbf{x})(1-\mathbf{y}(i)) - h_i(\mathbf{x})\mathbf{y}(i) = -h_i(\mathbf{x}) \geq 0$. ■

Lemma 4 *Given a specified GCCMC model with q classes and with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$ for each class satisfying $k_i = fat(\gamma^i/8)$, where fat is continuous from the right. If GCCMC has correctly classified m multiclass examples $\boldsymbol{s}$ generated independently according to the unknown (but fixed) distribution D and $\bar{\boldsymbol{s}}$ is a set of another m multiclass examples, then we can bound the following probability to be less than δ: $P^{2m}\{\boldsymbol{s}\bar{\boldsymbol{s}} : \exists$ a GCCMC model, it correctly classifies $\boldsymbol{s}$, fraction of $\bar{\boldsymbol{s}}$ misclassified $> \epsilon(m,q,\delta)\} < \delta$, where $\epsilon(m,q,\delta) = \frac{1}{m}(Q\log(32m) + \log\frac{2^q}{\delta})$ and $Q = \sum_{i=1}^{q} k_i \log(\frac{8em}{k_i})$.*

Proof (of Lemma 4). Suppose that G is a GCCMC model with q classes and with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$, the probability event in Lemma 4 can be described as

$$A = \{\mathbf{s}\bar{\mathbf{s}} : \exists G, k_i = fat(\gamma^i/8), er_{\mathbf{s}}[G] = 0, er_{\bar{\mathbf{s}}}[G] > \epsilon\}.$$

Let $\hat{\mathbf{s}}$ and $\hat{\bar{\mathbf{s}}}$ denote two different set of m multiclass examples, which are drawn i.i.d. from the distribution $D \times \{0,1\}$. Applying the definition of $\hat{\mathbf{x}}$, $\hat{h}$ and Proposition 3, the event can also be written as $A = \{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}} : \exists G, \hat{\gamma}^i = \gamma^i/2, k_i = fat(\hat{\gamma}^i/4), er_{\mathbf{s}}[G] = 0, r_i = max_t \hat{h}_i(\hat{\mathbf{x}}_t), 2\hat{\gamma}^i = -r_i, |\{\hat{\mathbf{y}} \in \hat{\bar{\mathbf{s}}} : \exists h_i \in G, \hat{h}_i(\hat{\mathbf{y}}) \geq 2\hat{\gamma}^i + r_i\}| > m\epsilon\}$. Here, $-max_t \hat{h}_i(\hat{\mathbf{x}}_t)$ means the minimal value of $|h_i(\mathbf{x})|$ which represents the margin for class ζ_i, so $2\hat{\gamma}^i = -r_i$. Let $\gamma_{k_i} = min\{\gamma' : fat(\gamma'/4) \leq k_i\}$, so $\gamma_{k_i} \leq \hat{\gamma}^i$, we define the following function:

$$\pi(\hat{h}) = \begin{cases} 0 & \text{if } \hat{h} \geq 0 \\ -2\gamma_{k_i} & \text{if } \hat{h} \leq -2\gamma_{k_i} \\ \hat{h} & \text{otherwise} \end{cases}$$

so $\pi(\hat{h}) \in [-2\gamma_{k_i}, 0]$. Let $\pi(\hat{G}) = \{\pi(\hat{h}) : h \in G\}$.

Let $B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}$ represent the minimal γ_{k_i}-cover set of $\pi(\hat{G})$ in the pseudo-metric $d_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}$. We have that for any $h_i \in G$, there exists $\tilde{f} \in B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}$, $|\pi(\hat{h}_i(\hat{\mathbf{z}})) - \pi(\tilde{f}(\hat{\mathbf{z}}))| < \gamma_{k_i}$, for all $\hat{\mathbf{z}} \in \hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}$. For all $\hat{\mathbf{x}} \in \hat{\mathbf{s}}$, by the definition of r_i, $\hat{h}_i(\hat{\mathbf{x}}) \leq r_i = -2\hat{\gamma}^i$, and $\gamma_{k_i} \leq \hat{\gamma}^i$, $\hat{h}_i(\hat{\mathbf{x}}) \leq -2\gamma_{k_i}$, $\pi(\hat{h}_i(\hat{\mathbf{x}})) = -2\gamma_{k_i}$, so $\pi(\tilde{f}(\hat{\mathbf{x}})) < -2\gamma_{k_i} + \gamma_{k_i} = -\gamma_{k_i}$. However, there are at least $m\epsilon$ points $\hat{\mathbf{y}} \in \hat{\bar{\mathbf{s}}}$ such that $\hat{h}_i(\hat{\mathbf{y}}) \geq 0$, so $\pi(\tilde{f}(\hat{\mathbf{y}})) > -\gamma_{k_i} > max_t \pi(\tilde{f}(\hat{\mathbf{x}}_t))$. Since π only reduces the separation between output values, we conclude that the inequality $\tilde{f}(\hat{\mathbf{y}}) > max_t \tilde{f}(\hat{\mathbf{x}}_t)$ holds. Moreover, the $m\epsilon$ points in $\hat{\bar{\mathbf{s}}}$ with the largest $\tilde{f}$ values must remain for the inequality to hold. By the permutation argument, at most $2^{-m\epsilon}$ of the sequences obtained by swapping corresponding points satisfy the conditions for fixed $\tilde{f}$.

As for any $h_i \in G$, there exists $\tilde{f} \in B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}$, so there are $|B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|$ possibilities of $\tilde{f}$ that satisfy the inequality for k_i. Note that $|B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|$ is a positive integer which is usually bigger than 1, and by the union bound, we obtain the following inequality:

$$P(A) \leq (E(|B^{k_1}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) + \cdots + E(|B^{k_q}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|))2^{-m\epsilon} \leq (E(|B^{k_1}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) \times \cdots \times E(|B^{k_q}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|))2^{-m\epsilon}$$

Since every set of points γ-shattered by $\pi(\hat{G})$ can be γ-shattered by $\hat{G}$, so $fat_{\pi(\hat{G})}(\gamma) \leq fat_{\hat{G}}(\gamma)$, where $\hat{G} = \{\hat{h} : h \in G\}$. Hence, by Corollary 2 (setting $[a,b]$ to $[-2\gamma_{k_i}, 0]$, ϵ to γ_{k_i} and m to $2m$),

$$E(|B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) = E(\mathcal{N}(\gamma_{k_i}, \pi(\hat{G}), \hat{\mathbf{s}}\hat{\bar{\mathbf{s}}})) \leq 2(32m)^{d\log(\frac{8em}{d})}$$

where $d = fat_{\pi(\hat{G})}(\gamma_{k_i}/4) \leq fat_{\hat{G}}(\gamma_{k_i}/4) \leq k_i$. Thus $E(|B^{k_i}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) \leq 2(32m)^{k_i \log(\frac{8em}{k_i})}$, and we obtain

$$P(A) \leq (E(|B^{k_1}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) \times \cdots \times E(|B^{k_q}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|))2^{-m\epsilon} \leq \prod_{i=1}^{q} 2(32m)^{k_i \log(\frac{8em}{k_i})} = 2^q(32m)^Q$$

where $Q = \sum_{i=1}^{q} k_i \log(\frac{8em}{k_i})$. And so $(E(|B^{k_1}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|) \times \cdots \times E(|B^{k_q}_{\hat{\mathbf{s}}\hat{\bar{\mathbf{s}}}}|))2^{-m\epsilon} < \delta$ provided

$$\epsilon(m,q,\delta) \geq \frac{1}{m}\Big(Q\log(32m) + \log\frac{2^q}{\delta}\Big)$$

as required. ■

Lemma 4 applies to a particular GCCMC model with a specified number of classes and a specified margin for each class. In practice, we will observe the margins after running the GCCMC model. Thus, we must bound the probabilities uniformly over all of the possible margins to obtain a practical bound. The generalization error bound of the multiclass classification problem using GCCMC is shown as follows:

Theorem 5 *Suppose that random m multiclass examples can be correctly classified using a GCCMC model, and suppose this GCCMC model contains q classifiers with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$ for each class. Then we can bound the generalization error with probability greater than $1-\delta$ to be less than*

$$\frac{130R^2}{m}\Big(Q'\log(8em)\log(32m) + \log\frac{2(2m)^q}{\delta}\Big)$$

where $Q' = \sum_{i=1}^{q} \frac{1}{(\gamma^i)^2}$ and R is the radius of a ball containing the support of the distribution.

Before proving Theorem 5, we state one key symmetrization lemma and Theorem 7.

Lemma 6 (Symmetrization) *Let $\mathcal{H}$ be the real valued function class. $\boldsymbol{s}$ and $\bar{\boldsymbol{s}}$ are m examples both drawn independently according to the unknown distribution D. If $m\epsilon^2 \geq 2$, then*

$$P_{\boldsymbol{s}}(\sup_{h\in\mathcal{H}} |er_D[h] - er_{\boldsymbol{s}}[h]| \geq \epsilon) \;\leq\; 2P_{\boldsymbol{s}\bar{\boldsymbol{s}}}(\sup_{h\in\mathcal{H}} |er_{\bar{\boldsymbol{s}}}[h] - er_{\boldsymbol{s}}[h]| \geq \epsilon/2) \tag{2}$$

The proof of this lemma can be found in Appendix B.

Theorem 7 (Bartlett and Shawe-Taylor (1998)) *Let $\mathcal{H}$ be restricted to points in a ball of $\boldsymbol{M}$ dimensions of radius R about the origin, then*

$$fat_{\mathcal{H}}(\gamma) \leq \min\Big\{\frac{R^2}{\gamma^2}, \boldsymbol{M}+1\Big\} \tag{3}$$

Proof (of Theorem 5). We must bound the probabilities over different margins. We first use Lemma 6 to bound the probability of error in terms of the probability of the discrepancy between the performance on two halves of a double example. Then we combine this result with Lemma 4. We must consider all possible patterns of k_i's for ζ_i. The largest value of k_i is m. Thus, for fixed q, we can bound the number of possibilities by m^q. Hence, there are m^q of applications of Lemma 4.

Let $c_i = \{\gamma^1, \gamma^2, \cdots, \gamma^q\}$ denote the i-th combination of margins varied in $\{1, \cdots, m\}^q$. $\mathcal{G}$ denotes a set of GCCMC models. The generalization error of G can be represented as $er_D[G]$ and $er_{\mathbf{s}}[G]$ is 0, where $G \in \mathcal{G}$. The uniform convergence bound of the generalization error is

$$P_{\mathbf{s}}(\sup_{G\in\mathcal{G}} |er_D[G] - er_{\mathbf{s}}[G]| \geq \epsilon)$$

Applying Lemma 6,

$$P_{\mathbf{s}}(\sup_{G\in\mathcal{G}} |er_D[G] - er_{\mathbf{s}}[G]| \geq \epsilon) \leq 2P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{G\in\mathcal{G}} |er_{\bar{\mathbf{s}}}[G] - er_{\mathbf{s}}[G]| \geq \epsilon/2)$$

Let $J_{c_i} = \{\mathbf{s}\bar{\mathbf{s}} : \exists$ a GCCMC model G with q classes and with margins $c_i : k_i = fat(\gamma^i/8), er_{\mathbf{s}}[G] = 0, er_{\bar{\mathbf{s}}}[G] \geq \epsilon/2\}$. Clearly,

$$P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{G\in\mathcal{G}} |er_{\bar{\mathbf{s}}}[G] - er_{\mathbf{s}}[G]| \geq \epsilon/2) \leq P^{m^q}\Big(\bigcup_{i=1}^{m^q} J_{c_i}\Big)$$

As k_i still satisfies $k_i = fat(\gamma^i/8)$, Lemma 4 can still be applied to each case of $P^{m^q}(J_{c_i})$. Let $\delta_k = \delta/m^q$. Applying Lemma 4 (replacing δ by $\delta_k/2$), we get:

$$P^{m^q}(J_{c_i}) < \delta_k/2$$

where $\epsilon(m, k, \delta_k/2) \geq 2/m(Q\log(32m) + \log\frac{2\times 2^q}{\delta_k})$ and $Q = \sum_{i=1}^q k_i \log(\frac{4em}{k_i})$. It suffices to show by the union bound that $P^{m^q}(\bigcup_{i=1}^{m^q} J_{c_i}) \leq \sum_{i=1}^{m^q} P^{m^q}(J_{c_i}) < \delta_k/2 \times m^q = \delta/2$. Applying Lemma 6,

$$\begin{aligned} P_{\mathbf{s}}(\sup_{G\in\mathcal{G}} |er_D[G] - er_{\mathbf{s}}[G]| \geq \epsilon) &\leq 2P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{G\in\mathcal{G}} |er_{\bar{\mathbf{s}}}[G] - er_{\mathbf{s}}[G]| \geq \epsilon/2) \\ &\leq 2P^{m^q}\Big(\bigcup_{i=1}^{m^q} J_{c_i}\Big) < \delta \end{aligned}$$

Thus, $P_{\mathbf{s}}(\sup_{G\in\mathcal{G}} |er_D[G] - er_{\mathbf{s}}[G]| \leq \epsilon) \geq 1 - \delta$. Let R be the radius of a ball containing the support of the distribution. Applying Theorem 7, we get $k_i = fat(\gamma^i/8) \leq 65R^2/(\gamma^i)^2$. Note that we have replaced the constant $8^2 = 64$ by 65 in order to ensure the continuity from the right required for the application of Lemma 4. We have upperbounded $\log(8em/k_i)$ by $\log(8em)$. Thus,

$$\begin{aligned} er_D[G] &\leq 2/m\Big(Q\log(32m) + \log\frac{2(2m)^q}{\delta}\Big) \\ &\leq \frac{130R^2}{m}\Big(Q'\log(8em)\log(32m) + \log\frac{2(2m)^q}{\delta}\Big) \end{aligned}$$

where $Q' = \sum_{i=1}^q \frac{1}{(\gamma^i)^2}$. ■

Given the training data size and the number of classes, Theorem 5 reveals an important factor in reducing the generalization error bound for the GCCMC model: the minimization of the sum of the reciprocal of the square of the margin over the classes. Thus, we obtain the following Corollary:

Corollary 8 (Globally Optimal Classifier Chain for Multiclass Classification) *Suppose that random m multiclass examples with q classes can be correctly classified using a GCCMC model, this GCCMC model is the globally optimal classifier chain if and only if the minimization of Q' in Theorem 5 is achieved over this classifier chain.*

Based on Corollary 8, we propose the following easy-to-hard learning paradigm for multiclass classification problems.

Definition 9 (Easy-to-hard Learning Paradigm for Multiclass Classification) *The easy-to-hard learning paradigm for multiclass classification problem is to minimize Q' in Theorem 5. By minimizing Q', we can automatically identify easy and hard classes.*

Remark. Classes with a larger margin are easier to identify than those with a smaller margin. Thus, the intuitive idea of the easy-to-hard learning paradigm is to identify the class with a larger margin first, followed by ones with a smaller margin.

Discussion of Rifkin and Klautau's Conjecture. Rifkin and Klautau (2004) speculate that an algorithm which exploits the relationship between classes can offer superior performance, and this remains an open problem. Theoretically, Corollary 10 provides an affirmative answer to Rifkin and Klautau (2004)'s conjecture based on Theorem 5:

Corollary 10 *By exploiting the relationship between classes, our proposed GCCMC model is able to achieve a lower generalization error bound. Furthermore, our proposed easy-to-hard learning paradigm can optimize the performance of GCCMC.*

Motivated by the above analysis, we derive the following easy-to-hard learning paradigm for multi-label classification.

4. The Easy-to-hard Learning Paradigm for Multi-label Classification

4.1 Classifier Chain for Multi-label Classification

Similar to CCMC, the *classifier chain* (CC) model (Read et al., 2009) is proposed to train q binary classifiers h_j ($j \in \{1, \cdots, q\}$) for multi-label problems. Classifiers are linked along a chain where each classifier h_j deals with the binary classification problem for label λ_j. The augmented vector $\{\mathbf{x}_t, \mathbf{y}_t(1), \cdots, \mathbf{y}_t(j)\}_{t=1}^n$ is used as the input for training classifier h_{j+1}. Given a new testing instance $\mathbf{x}$, classifier h_1 in the chain is responsible for predicting the value of $\mathbf{y}(1)$ using input $\mathbf{x}$. Then, h_2 predicts the value of $\mathbf{y}(2)$ taking $\mathbf{x}$ plus the predicted value of $\mathbf{y}(1)$ as an input. Following in this way, h_{j+1} predicts $\mathbf{y}(j+1)$ using the predicted value of $\mathbf{y}(1), \cdots, \mathbf{y}(j)$ as additional input information. CC passes label information between classifiers, allowing CC to exploit the label dependence and thus overcome the label independence problem of BR.

Different classifier chains involve different classifiers learned on different training sets and thus the order of the chain itself clearly affects the prediction performance. To solve the issue of selecting a chain order for CC, Read et al. (2009) propose the extension of CC, called *ensembled classifier chain* (ECC), to average the multi-label predictions of CC over a set of random chain ordering. ECC first randomly reorders the labels $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ many times. Then, CC is applied to the reordered labels for each time and the performance of CC is averaged over those times to obtain the final prediction performance.

4.2 Generalized Classifier Chain for Multi-label Classification

We generalize the CC model over a random label order, called *generalized classifier chain* (GCC) model. Assume the labels $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ are randomly reordered as $\{\zeta_1, \zeta_2, \cdots, \zeta_q\}$, where $\zeta_j = \lambda_k$ means label λ_k moves to position j from k. In the GCC model, classifiers are also linked along a chain where each classifier h_j deals with the binary classification problem for label ζ_j (λ_k). GCC follows the same training and testing procedures as CC, while the only difference is the label order. In the GCC model, for input $\mathbf{x}_t$, $\mathbf{y}_t(j) = 1$ if and only if $\zeta_j \in \mathcal{Y}_t$.

4.3 Analysis

Motivated by our analysis in Section 3, we analyze the generalization error bound of the multi-label classification problem using GCC.

Let $\mathbf{X}$ represent the input space. Both $\mathbf{s}$ and $\bar{\mathbf{s}}$ are m multi-label examples drawn independently according to an unknown distribution D.

We first study the generalization error bound of the specified GCC with the specified number of labels and margins. Let G be the set of classifiers of GCC, $G = \{h_1, h_2, \cdots, h_q\}$. $er_{\mathbf{s}}[G]$ denotes the fraction of the number of errors that GCC makes on $\mathbf{s}$. Define $\hat{\mathbf{x}} \in \mathbf{X} \times \{0, 1\}$, $\hat{h}_j(\hat{\mathbf{x}}) = h_j(\mathbf{x})(1 - \mathbf{y}(j)) - h_j(\mathbf{x})\mathbf{y}(j)$.

We introduce the following proposition:

Proposition 11 *If an instance $\boldsymbol{x} \in \boldsymbol{X}$ is misclassified by a GCC model, then $\exists h_j \in G, \hat{h}_j(\hat{\mathbf{x}}) \geq 0$.*

Proof For multi-label problem, it is easy to verify that if an instance $\mathbf{x} \in \mathbf{X}$ is correctly classified by h_j, then $\hat{h}_j(\hat{\mathbf{x}}) < 0$, otherwise, $\hat{h}_j(\hat{\mathbf{x}}) \geq 0$. ■

Lemma 12 *Given a specified GCC model with q labels and with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$ for each label satisfying $k_i = fat(\gamma^i/8)$, where fat is continuous from the right. If GCC has correctly classified m multi-labeled examples $\boldsymbol{s}$ generated independently according to the unknown (but fixed) distribution D and $\bar{\boldsymbol{s}}$ is a set of another m multi-labeled examples, then we can bound the following probability to be less than δ: $P^{2m}\{\boldsymbol{s}\bar{\boldsymbol{s}} : \exists$ a GCC model, it correctly classifies $\boldsymbol{s}$, fraction of $\bar{\boldsymbol{s}}$ misclassified $> \epsilon(m, q, \delta)\} < \delta$, where $\epsilon(m, q, \delta) = \frac{1}{m}(Q \log(32m) + \log \frac{2^q}{\delta})$ and $Q = \sum_{i=1}^{q} k_i \log(\frac{8em}{k_i})$.*

The proof can be adapted from the proof for Lemma 4.

Based on Lemma 12, we can bound the probabilities uniformly over all of the possible margins to obtain a practical bound. The generalization error bound of the multi-label classification problem using GCC is shown as follows:

Theorem 13 *Suppose that random m multi-labeled examples can be correctly classified using a GCC model, and suppose this GCC model contains q classifiers with margins $\gamma^1, \gamma^2, \cdots, \gamma^q$ for each label. Then we can bound the generalization error with probability greater than $1 - \delta$ to be less than*

$$\frac{130R^2}{m}\Big(Q' \log(8em) \log(32m) + \log \frac{2(2m)^q}{\delta}\Big)$$

where $Q' = \sum_{i=1}^{q} \frac{1}{(\gamma^i)^2}$ and R is the radius of a ball containing the support of the distribution.

The proof can be adapted from the proof for Theorem 5.

Theorem 13 reveals an important factor in reducing the generalization error bound for the GCC model: the minimization of the sum of the reciprocal of the square of the margin over the labels, given the training data size and the number of labels. Thus, we obtain the following Corollary:

Corollary 14 (Globally Optimal Classifier Chain for Multi-label Classification) *Suppose that random m multi-labeled examples with q labels can be correctly classified using a GCC model, this GCC model is the globally optimal classifier chain if and only if the minimization of Q' in Theorem 13 is achieved over this classifier chain.*

Based on Corollary 14, we propose the following easy-to-hard learning paradigm for multi-label classification.

Definition 15 (Easy-to-hard Learning Paradigm for Multi-label Classification) *The easy-to-hard learning paradigm for multi-label classification problem is to minimize Q' in Theorem 13. By minimizing Q', we can automatically identify easy and hard labels.*

Discussion of Label's Relationship. Recently, many works, such as Read et al. (2009) and Guo and Schuurmans (2011), have conducted extensive experiments to show that multi-label learning methods which explicitly capture the label's relationship will usually achieve better prediction performance. However, to the best of our knowledge, very few works study the reasons behind these promising empirical results. Based on Theorem 13, Corollary 16 provides theoretical support for this problem:

Corollary 16 *By exploiting the relationship between labels, our proposed GCC model is able to achieve a lower generalization error bound. Furthermore, our proposed easy-to-hard learning paradigm can optimize the performance of GCC.*

Given the number of classes or labels q, there are $q!$ different class or label orders. It is very expensive to find the globally optimal CCMC or CC, which can minimize Q', by searching over all of the class or label orders. Next, we discuss some simple easy-to-hard learning algorithms.

5. Easy-to-hard Learning Algorithm

In this section, we propose some simple easy-to-hard learning algorithms. To clearly state the algorithms, we redefine the margins with class or label order information. Given class or label set $\mathcal{M} = \{\lambda_1, \lambda_2, \cdots, \lambda_q\}$. Let $o_i (1 \leq o_i \leq q)$ denote the order of λ_i in the GCCMC or GCC model, $\gamma_i^{o_i}$ represents the margin for λ_i, with previous $o_i - 1$ classes or labels as the augmented input. If $o_i = 1$, then γ_i^1 represents the margin for λ_i, without augmented input. Then Q' is redefined as $Q' = \sum_{i=1}^{q} \frac{1}{(\gamma_i^{o_i})^2}$.

5.1 Dynamic Programming Algorithm

To simplify the search algorithm mentioned before, we propose the CCMC-DP and CC-DP algorithm to find the globally optimal CCMC and CC, respectively. Note that $Q' = \sum_{i=1}^{q} \frac{1}{(\gamma_i^{o_i})^2} = \frac{1}{(\gamma_q^{o_q})^2} + \cdots + \left[\frac{1}{(\gamma_{k+1}^{o_{k+1}})^2} + \sum_{j=1}^{k} \frac{1}{(\gamma_j^{o_j})^2}\right]$, we explore the idea of dynamic programming (DP) to iteratively optimize Q' over a subset of $\mathcal{M}$ with the length of $1, 2, \cdots, q$. Lastly, we obtain the optimal Q' over $\mathcal{M}$. Assume $i \in \{1, \cdots, q\}$. Let $V(i, \eta)$ be the optimal Q' over a subset of $\mathcal{M}$ with the length of $\eta (1 \leq \eta \leq q)$, where the class or label order ends by λ_i. M_i^η represents the corresponding class or label set for $V(i, \eta)$. When $\eta = q$, $V(i, q)$ is the optimal Q' over $\mathcal{M}$, where the class or label order ends by λ_i. The DP equation is written as:

$$V(i, \eta + 1) = \min_{j \neq i, \lambda_i \notin M_j^\eta} \left\{ \frac{1}{(\gamma_i^{\eta+1})^2} + V(j, \eta) \right\} \tag{4}$$

where $\gamma_i^{\eta+1}$ is the margin for λ_i, with M_j^η as the augmented input. The initial condition of DP is: $V(i, 1) = \frac{1}{(\gamma_i^1)^2}$ and $M_i^1 = \{\lambda_i\}$. The optimal Q' over $\mathcal{M}$ can be obtained by solving $\min_{i \in \{1, \cdots, q\}} V(i, q)$. Assume that the training time of linear SVM takes $\mathcal{O}(nd)$. The CCMC-DP or CC-DP algorithm is shown as the following bottom-up procedure: from the bottom, we first compute $V(i, 1) = \frac{1}{(\gamma_i^1)^2}$, which takes $\mathcal{O}(nd)$. Then we compute $V(i, 2) = \min_{j \neq i, \lambda_i \notin M_j^1} \{\frac{1}{(\gamma_i^2)^2} + V(j, 1)\}$, which requires at most $\mathcal{O}(qnd)$, and set $M_i^2 = M_j^1 \cup \{\lambda_i\}$. Similarly, it takes at most $\mathcal{O}(q^2 nd)$ time complexity to calculate $V(i, q)$. Lastly, we iteratively solve this DP equation, and use $\min_{i \in \{1, \cdots, q\}} V(i, q)$ to obtain the optimal solution, which requires at most $\mathcal{O}(q^3 nd)$ time complexity.

Theorem 17 (Correctness of DP) *Q' can be minimized by CCMC-DP or CC-DP, which means this Algorithm can find the globally optimal CCMC or CC.*

The proof can be found in Appendix C.

5.2 Greedy Algorithm

We propose the CCMC-Greedy and CC-Greedy algorithm to find a locally optimal CCMC and CC, respectively. To save time, we construct only one classifier chain with the locally optimal class or label order. If the maximum margin can be achieved over this class or label, without augmented input, we select this class or label from $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ as the first class or label. The first class or label is denoted by ζ_1. We then select the class or label from the remaining classes or labels as the second class or label, if the maximum margin can be achieved over this class or label with ζ_1 as the augmented input. We continue in this way until the last class or label has been selected. CCMC-Greedy and CC-Greedy take $\mathcal{O}(q^2 nd)$ time, respectively. We show the details of the CC-Greedy algorithm in Appendix D.

5.3 Fast Greedy Algorithm

In multiclass problems, CCMC-DP and CCMC-Greedy are intractable for data sets with a large number of classes. To further speed up the CCMC-Greedy algorithm, we propose fast greedy algorithm (CCMC-FG), which scales linearly with q, to greedily optimize the order of the top ω classes. Similar to CCMC-Greedy, if the maximum margin can be achieved over this class without augmented input, we select this class from $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ as the first class. The first class is denoted by ζ_1. Then we select the class from the remaining classes as the second class, if the maximum margin can be achieved over this class with prediction values of the classifier trained for class ζ_1 as the augmented input. We continue in this way until the ω class has been selected. Lastly, we use the remaining $q - \omega$ classes to form the classifier chain. CCMC-FG takes $\mathcal{O}(q\omega nd)$ time.

Remark. CCMC-Greedy converges to the locally optimal CCMC, while CCMC-FG finds the top ω locally optimal class order.

5.4 Tree-Based Algorithm

For multi-label problem, CC-DP and CC-Greedy are very time-consuming for data sets with many labels. We propose Tree-DP and Tree-Greedy algorithms to further speed up CC-DP and CC-Greedy, respectively, which scale linearly with q. We create a tree recursively in a top-down manner.

Assume that $\{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^n$ is the input data for the root node. Suppose that the label set $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ in the root node is randomly split into two subsets with about the same size for left and right child nodes: $leftset = \{\lambda_1, \cdots, \lambda_{q/2}\}$ and $rightset = \{\lambda_{q/2+1}, \cdots, \lambda_q\}$. A training example can be considered annotated with $leftset$ and $rightset$ if it is annotated with at least one of the labels in $leftset$ and $rightset$, respectively. In this way, $leftset$ and $rightset$ can be seen as two labels. Then, in the root node, we train the CC with $leftset$ and $rightset$ as two labels using CC-DP or CC-Greedy. After that, left and right child nodes only keep the examples that are annotated with $leftset$ and $rightset$, respectively. This approach recurses into each child node that contains more than a single label.

Starting from the root node, we use the trained CC classifier on this node for prediction and we follow the recursive process. Finally, this process may lead to the prediction of some labels corresponding to some leaves. We provide the following corollary pertaining to the Tree-DP.

Corollary 18 *After building a tree using the Tree-DP algorithm, we can find the globally optimal CC in each decision node of the tree.*

Proof Given the structure of the tree, according to Theorem 17, we can find the globally optimal label order in each decision node using the CC-DP algorithm. ■

For each internal node, we only deal with two labels, thus the training time of Tree-DP and Tree-Greedy only take $\mathcal{O}(8nd)$ and $\mathcal{O}(4nd)$, respectively. The number of internal nodes in such a tree is equal to $q-1$. In total, Tree-DP and Tree-Greedy take $\mathcal{O}(8(q-1)nd)$ and $\mathcal{O}(4(q-1)nd)$ training time, respectively. Assume that the testing instance goes along ι paths in our tree during the testing procedure and the depth of the tree is $\log(q)$. In each decision node, we take $\mathcal{O}(d)$ time for testing. Totally, the testing time for Tree-DP and Tree-Greedy is $\mathcal{O}(\iota \log(q)d)$.

6. Applications

This section shows that our framework can be used for various applications, such as ordinal classification and relationship prediction.

6.1 Ordinal Classification

Many practical applications involve situations exhibiting an order among the different categories. For example, a user rates movies by giving them grades based on quality. These grades represent the ranking information. For example, grade classes are ordered as $D < C < B < A$. This is a learning task for predicting ordinal classes, referred to as ordinal classification (Seah et al., 2012).

Several algorithms and methods have been developed to deal with ordinal classification, such as SVM techniques (Shashua and Levin, 2002), binary decomposition (Destercke and Yang, 2014), Gaussian processes (Chu and Ghahramani, 2005) and monotone functions (Tehrani et al., 2012). However, all these methods do not capture and use correlated information between ordinal classes. To achieve this goal, we transform ordinal classification into multiclass classification and then apply CCMC for ordinal classification.

Consider an ordinal classification problem with q ordered categories. We denote these categories as $\mathcal{Y}_t \subseteq \{\lambda_1, \lambda_2, \cdots, \lambda_q\}$ to keep the known ordering information. $\mathbf{y}_t \in \{0,1\}^q$ is used to represent the set $\mathcal{Y}_t$, where $\mathbf{y}_t(j) = 1$ if and only if $\lambda_j \in \mathcal{Y}_t$, and there is only one element in $\mathcal{Y}_t$ for the ordinal classification problem. So, we transform ordinal classification into a multiclass classification problems. Then, we apply CCMC for ordinal classification as follows: the binary classifier h_1 is first trained for the ordinal class λ_1, then the augmented vector $\{\mathbf{x}_t, h_1(\mathbf{x}_t)\}_{t=1}^n$ is used as the input to train classifier h_2 for ordinal class λ_2. Similarly, $\mathbf{x}_t$ augments all previous prediction values of $h_1, \cdots, h_j$ as the input to train classifier h_{j+1} for ordinal class λ_{j+1}. CCMC proceeds in this way until the last classifier h_q for ordinal class λ_q has been trained.

6.2 Relationship Prediction

Relationship prediction problems in the online review website Epinions (Massa and Avesani, 2006) attempt to predict whether people trust or distrust others based on their reviews. Such social networks can be modeled as a signed network where trust/distrust are modeled as positive/negative edges between entities (Leskovec et al., 2010). The problem then becomes predicting unknown relationship between any two users given the network.

Many approaches, such as Hsieh et al. (2012) and Chiang et al. (2014), perform matrix completion on an adjacency matrix and then use the sign of the completed matrix

for relationship prediction. Recently, Chiang et al. (2015) achieve state-of-the-art performance by incorporating the feature information of users. Based on feature information, we provide new insight into the design of relationship prediction algorithms. Specifically, we first transform relationship prediction into a multi-label classification problem by considering trust/distrust as positive/negative labels. Then, we apply our proposed easy-to-hard learning strategy to solve relationship prediction tasks.

Relationship prediction is represented as a graph with the adjacency matrix $R \in \{0,1\}^{n \times n}$, which denotes relationships between users as follows:

$$R_{ij} = \begin{cases} 1, & \text{if user } i \text{ and user } j \text{ have positive relationship;} \\ 0, & \text{if user } i \text{ and user } j \text{ have negative relationship} \end{cases}$$

We assume that user i and user i have a positive relationship. The attribute information of user i can be extracted as the input or instance (feature) $\mathbf{x}_i$, and the i-th row vector of R can be used as label $\mathbf{y}_i$. So, we transform relationship prediction into a multi-label classification problem. Then, CC model can be used to deal with the transformed problems, and we can apply our proposed easy-to-hard learning strategy to solve relationship prediction tasks.

7. Experiment

In this section, we perform experimental studies on a number of real world data sets to evaluate the performance of our proposed algorithms for multiclass and multi-label classification problems. To perform a fair comparison, we use the same linear classification/regression package LIBLINEAR (Fan et al., 2008) with L2-regularized square hinge loss (primal) to train the classifiers for all methods and use the default parameter settings in LIBLINEAR. All experiments are conducted on a workstation with a 3.4GHZ Intel CPU and 32GB main memory running on a Linux platform.

7.1 Experiment on Multiclass Classification

In this subsection, we first demonstrate our motivation on the recognition of human action (Schüldt et al., 2004). We then consider a variety of benchmark multiclass data sets without background from the LIBSVM website[4] to evaluate the performance of the proposed algorithms for multiclass classification. Lastly, we conduct experiments on two multiclass data sets with background collected from Silberman et al. (2012) and He et al. (2004). The training/testing partition is either predefined or the data is randomly split into 80% training and 20% testing. The statistics of each data set are reported in Table 1. We compare our algorithms with some baseline methods: OVO, OVR, ECOC (Dietterich and Bakiri, 1995) and Top-k SVM (Lapin et al., 2015, 2016). The library for ECOC is from Pedregosa et al. (2011) and the size of code for ECOC is selected using 5-fold cross validation over the range $\{2, 10, 30, 50\}$. Top-k SVM (Lapin et al., 2015, 2016) is one of the state-of-the-art generalized multiclass SVM for top-k error optimization. The code is provided by their authors. Following the similar parameter settings in Lapin et al. (2015, 2016) for Top-k SVM, k is selected using 5-fold cross validation over the range $\{1, 3, 5, 10\}$.

4. https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets

Table 1: Multiclass data sets used in the experiments.

Data set	# training	# testing	# Classes	Background
KTH	1,910	477	6	×
IRIS	120	30	3	×
SEGMENT	1,848	4,62	7	×
SEISMIC	78,823	19,705	3	×
RCV1	15,564	518,571	51	×
ALOI	86,400	21,600	1,000	×
ILSVRC2012	58,700	31,300	1,000	×
NYU	60,232	11,768	5	√
SOWERBY	528,356	110,620	6	√

Table 2: Testing error rate (in %) on the KTH data set.

OVO	OVR	ECOC	Top-k SVM	CCMC	CCMC-Greedy	CCMC-DP
8.81	8.18	7.73	8.13	7.34	6.29	5.66

7.1.1 Human action recognition

We first validate our methods by recognizing complex human actions on the KTH data set Schüldt et al. (2004). KTH contains six types of human action: *walk*, *jog*, *run*, *hand wave*, *hand clap* and *boxing*. The confusion matrix of OVR on the KTH data set is shown in Figure 1. From Figure 1, we can see that it is easy to identify *walk* from other actions, whereas it is difficult to distinguish between *boxing* and *hand clap*. This observation is consistent with commonsense. Our model aims to classify classes from simple to hard; the classifier of the hard actions will benefit significantly from the prediction of the simple actions.

The testing error rates of the different methods are shown in Table 2. The order of actions identified by CCMC-Greedy is: *run*, *hand wave*, *walk*, *jog*, *boxing* and *hand clap*. CCMC-DP finds the action order: *walk*, *run*, *jog*, *hand wave*, *boxing* and *hand clap*. From these results, we observe that:

- For the KTH data set, our methods achieve a better prediction performance than OVO, OVR, ECOC and Top-k SVM, which verifies that our model effectively uses the predictions of previous classifiers to improve the performance of OVR.

- CCMC-Greedy and CCMC-DP improve CCMC, which also verifies our motivation: we should classify the classes from easy to hard. The order of actions (from easy to hard) found by CCMC-Greedy and CCMC-DP are consistent with the confusion matrix, which demonstrates that our model can automatically identify easy and hard classes, and use the predictions of classifiers from easier classes to train the classifier for harder classes.

Figure 3 shows the confusion matrix of CCMC-DP on the KTH data set. By comparing Figure 1 with Figure 3, we observe that our method significantly improves OVR in terms of the classification accuracy rate for each action. For example, we achieve 100% accuracy for *run* and improve the accuracy for *hand clap* from 77.50% to 81.25%.

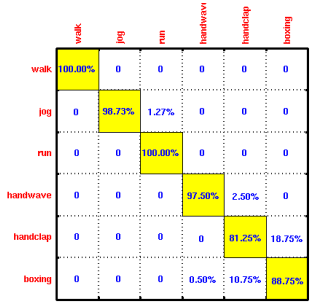


Figure 3: Confusion Matrix of CCMC-DP on the KTH Data Set.

7.1.2 Results without background data

We compare OVO, OVR and ECOC with our algorithms on the benchmark data sets without background. The classification results for our methods and baseline approaches on the IRIS, SEGMENT, SEISMIC and RCV1 data sets are reported in Table 3. Based on these results, we make the following observations.

- Our results show that OVR and OVO perform as accurate as ECOC and Top-k SVM, which is consistent with the empirical results in Rifkin and Klautau (2004), Demirkesen and Cherifi (2008) and Lapin et al. (2015, 2016).

- CCMC consistently improves the prediction performance of OVR and other baselines on all data sets. The results verify Rifkin and Klautau (2004)'s conjecture: an algorithm which exploits the relationship between classes can offer superior performance.

- When OVO outperforms OVR on certain data sets such as SEISMIC and SEGMENT, our methods are able to achieve superior prediction performance to OVO on these data sets.

- CCMC-Greedy is better than CCMC, and CCMC-DP outperforms CCMC-Greedy and CCMC. The results validate our theoretical analysis: i) Class order affects the performance of CCMC. ii) CCMC-DP is able to find the globally optimal CCMC which achieves the best prediction performance compared to CCMC-Greedy and CCMC. iii) The CCMC-Greedy algorithm achieves comparable prediction performance with CCMC-DP.

We also evaluate the performance of our fast greedy algorithm on the ALOI data set, which contains 1,000 classes. Here, ω is set to 10, 30, 50 and 100. The classification results and training time are reported in Table 4, from which we can see that CCMC-FG outperforms CCMC and improves the prediction performance of OVR, ECOC and Top-k SVM by about 9%, 6% and 5%, respectively. The prediction performance of CCMC-FG improves with the increased value of ω.

Table 3: Testing error rate (in %) on data sets without background.

Data set	OVO	OVR	ECOC	Top-k SVM	CCMC	CCMC-Greedy	CCMC-DP
IRIS	10.00	10.00	8.67	10.33	3.33	3.33	3.33
SEGMENT	5.63	7.14	7.08	7.06	6.28	5.63	5.19
SEISMIC	27.92	29.87	27.62	28.19	26.85	26.50	26.48
RCV1	11.09	11.90	11.83	11.98	11.78	11.71	11.68

Table 4: Testing error rate (in %) and training time (in second) on the ALOI data set.

Method	Testing Error	Training Time
OVO	6.88	14,748s
OVR	13.69	1,559s
ECOC	10.31	85,316s
Top-k SVM	9.4	65s
CCMC	5.46	12,715s
CCMC-FG($\omega = 10$)	5.27	16,327s
CCMC-FG($\omega = 30$)	5.08	31,886s
CCMC-FG($\omega = 50$)	4.94	67,582s
CCMC-FG($\omega = 100$)	4.78	163,891s

7.1.3 Results with background data

The OVO, ECOC and Top-k SVM approaches cannot be directly applied to background data. Table 5 shows the classification results for our methods and OVR on the NYU and SOWERBY data sets. From the results of Table 5, we can see that:

- CCMC consistently outperforms OVR.
- CCMC-Greedy achieves better prediction performance than CCMC and is comparable to CCMC-DP.
- CCMC-DP improves the prediction performance of OVR on the data sets with background by 3%.

7.1.4 Training time

This section studies the training time of the proposed methods and baselines on all data sets. The results are shown in Tables 4 and 6. From these results, we can see that:

- Top-k SVM is much faster than other methods on the ALOI data set with 1000 classes.
- Compared to OVR, CCMC maintains the training time over an acceptable threshold, while CCMC consistently improves OVR.

Table 5: Testing error rate (in %) on data sets with background.

Data set	OVR	CCMC	CCMC-Greedy	CCMC-DP
NYU	21.07	20.61	19.39	18.82
SOWERBY	18.03	16.27	15.54	15.24

Table 6: Training time (in second).

Data set	OVO	OVR	ECOC	Top-k SVM	CCMC	CCMC-Greedy	CCMC-DP
KTH	0.09s	0.20s	0.16s	0.13s	0.24s	0.60s	1.37s
IRIS	0.009s	0.008s	0.008s	0.02s	0.012s	0.019s	0.019s
SEGMENT	0.04s	0.05s	0.30s	0.06s	0.32s	1.10s	5.85s
SEISMIC	2.94s	4.60s	167.28s	2.23s	22.08s	32.88s	98.28s
RCV1	56.06s	18.16s	31.15s	346.79s	30.27s	982.89s	16,450.55s
NYU	N/A	4.74s	N/A	N/A	4.58s	14.46s	83.75s
SOWERBY	N/A	23.63s	N/A	N/A	39.73s	105.42s	558.02s

- CCMC-Greedy is much faster than CCMC-DP.
- With the increasing value of ω, the training time of CCMC-FG rises, but the prediction performance of CCMC-FG becomes better. ω can be set according to the time and accuracy requirements of applications.

The testing time of the proposed methods is similar to OVR. Although the training time of the proposed approaches is slower than OVR, the time required to test is of more importance than the time required to train for many applications.

7.1.5 Comparisons with deep learning methods

ADIOS (Cissé et al., 2016) is a state-of-the-art deep learning architecture for solving multiple class and label tasks. Unlike traditional deep learning methods that use a flat output layer, ADIOS aims to capture the complex dependency between labels/classes to improve deep learning methods. Their approach is to split the label/class set into two subsets, G_1 and G_2, such that given G_1, the labels/classes in G_2 are independent. Our strategy to leverage label/class dependency is very different from that of ADIOS, as shown in Figure 2, we use a classifier chain model for multi-class classification and find the optimal class ordering. After that, we use the predictions of classifiers from easier classes to train the classifiers for harder classes. As such, the assumptions and constraints used in ADOIS are not applicable in our model.

This subsection conducts the experiments on the ILSVRC2012 data set[5]. It contains 1,000 object categories (Liu et al., 2017). Due to the limit of computational resources, we randomly sample 58,700 training instances and 31,300 testing instances from the ILSVRC2012 data set. We use the source code provided by the authors of ADIOS with default parameters. According to Cissé et al. (2016), we use one hidden layer with 1024 rectified linear units (ReLUs) (Glorot et al., 2011) between inputs and G_1, and another 512-dimensional ReLUs between the hidden layer before G_1 and G_2 as well as direct connections between G_1 and G_2. We also compare with VGG (Simonyan and Zisserman, 2014) and residual nets (ResNet) (He et al., 2016). Both VGG and ResNet use a flat output layer, in which do not model the dependency between the classes(Cissé et al., 2016), so the rich structure information among classes is missing in VGG and ResNet. We use the source code provided by the respective authors with default parameters. Following He et al. (2016), we use the 34-layer residual nets due to the limit of computational resources, and also extract

5. http://www.image-net.org/challenges/LSVRC/2012/

Table 7: Testing error rate (in %) of VGG, ResNet-34, ADIOS and CCMC-FG on the ILSVRC2012 data set.

Method	VGG	ResNet-34	ADIOS	CCMC-FG+VGG features	CCMC-FG+ResNet features
Testing Error	23.95	21.51	21.28	23.98 ($\omega = 10$)	21.85 ($\omega = 10$)
				22.67 ($\omega = 30$)	20.11 ($\omega = 30$)
				20.73 ($\omega = 50$)	19.84 ($\omega = 50$)

2048-dimensional features by the ResNet-34. According to Simonyan and Zisserman (2014), we extract 4096-dimensional features from the 16-layer of the VGG. Here, ω is set to 10, 30 and 50 for our method.

The classification results are reported in Table 7. From this table, we can observe that

- ADIOS outperforms VGG and ResNet-34, which verifies ADIOS's claim: existing deep learning approaches do not take into account the often unknown but nevertheless rich relationships between classes, this knowledge about the rich class structure (and other deep structure in data) is sometime referred to as dark knowledge (e.g. by Hinton et al. (2015) and Ba and Caruana (2014)).
- Without the restriction of the assumptions and constraints used in ADOIS, our method achieves better performance than ADIOS with the increasing value of ω.
- CCMC-FG with ResNet features obtains better performance than CCMC-FG with VGG features, which demonstrates that our proposed method can be further improved based on better features.
- Based on the deep learning features, CCMC-FG consistently improves VGG and ResNet-34 with the increasing value of ω. The results validate our analysis and the better ordering of classes obtains the better performance. Note that the above mentioned results were obtained using 58,700 training data points. We conclude that with limited data, the usage of structure information is helpful. We conjecture that this advantage may ultimately vanish as more and more data becomes available for training.

7.2 Experiment on Multi-label Classification

We conduct experiments on ten real-world multi-label data sets with various domains from three websites.[6][7][8] The EURLEX_SM and EURLEX_ED data sets are preprocessed according to the experimental settings in Dembczynski et al. (2010) and Zhang and Schneider (2012). The statistics of data sets are presented in Table 8. We compare our algorithms with some baseline methods: BR, CC, ECC, CCA (Zhang and Schneider, 2011) and MMOC (Zhang and Schneider, 2012). ECC is averaged over several CC predictions with random

6. http://mulan.sourceforge.net
7. http://meka.sourceforge.net/#datasets
8. http://cse.seu.edu.cn/people/zhangml/Resources.htm#data

Table 8: Multi-label data sets used in the experiments.

Data	# inst.	# attr.	# labels	Domain
YEAST	2,417	103	14	biology
IMAGE	2,000	294	5	image
SLASHDOT	3,782	1,079	22	text
ENRON	1,702	1,001	53	text
LLOG	799	1,004	10	linguistics
ART	6,849	23,146	10	art
EURLEX_SM_10	11,454	5,000	10	text
EURLEX_ED_10	6,540	5,000	10	text
EURLEX_SM	19,348	5,000	201	text
EURLEX_ED	19,348	5,000	3,993	text

order and the ensemble size in ECC is set to 10 according to Dembczynski et al. (2010); Read et al. (2009). In our experiment, the running time of PCC and EPCC (Dembczynski et al., 2010) on most data sets, like SLASHDOT and ART, takes more than one week. From the results in Dembczynski et al. (2010), ECC is comparable with EPCC and outperforms PCC, so we do not consider PCC and EPCC here. CCA and MMOC are two state-of-the-art encoding-decoding (Hsu et al., 2009) methods. We cannot get the results of CCA and MMOC on ART , EURLEX_SM_10 and EURLEX_ED_10 data sets in one week.

We consider the following evaluation measurements Mao et al. (2013) to measure the prediction performance of all methods fairly:

- Example-F1: computes the F-1 score for all the labels of each testing example and then takes the average of the F-1 score.
- Macro-F1: calculates the F-1 score for each label and then takes the average of the F-1 score.
- Micro-F1: computes true positives, true negatives, false positives and false negatives over all labels, and then calculates an overall F-1 score.

The larger the value of those measurements, the better the performance. We perform 5-fold cross-validation on each data set and report the mean and standard error of each evaluation measurement.

7.2.1 Small-scale results

Three measurement results for CC-Greedy, CC-DP and baseline approaches in respect to the different small-scale data sets are reported in Tables 9, 10 and 11. We conduct the pairwise t-test at a 5% significance level to show that our methods perform significantly better than the compared methods. From the results, we can see that:

- BR generally underperforms in terms of Macro-F1 and Micro-F1 and it is much inferior to other methods in terms of Example-F1. Our experiment provides empirical evidence that the label correlations exist in many real word data sets and because BR ignores the information about the correlations between the labels, BR achieves poor performance on most data sets.

Table 9: Results of Example-F1 on the various small-scale data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" denotes the training time is more than one week.

Data set	BR	CC	ECC	CCA	MMOC	CC-Greedy	CC-DP
YEAST	0.6076±0.02[6]	0.5850±0.03[7]	0.6096±0.02[5]	0.6109±0.02[4]	0.6132±0.02[3]	**0.6144**±0.02[1]	0.6135±0.02[2]
IMAGE	0.5247±0.03[7]	**0.5991**±0.02[1]	0.5947±0.02[4]	0.5947±0.01[4]	0.5960±0.01[3]	0.5939±0.02[6]	0.5976±0.02[2]
SLASHDOT	0.4898±0.02[6]	0.5246±0.03[4]	0.5123±0.03[5]	0.5260±0.02[3]	0.4895±0.02[7]	0.5266±0.02[2]	**0.5268**±0.02[1]
ENRON	0.4792±0.02[7]	0.4799±0.01[6]	0.4848±0.01[4]	0.4812±0.02[5]	**0.4940**±0.02[1]	0.4894±0.02[2]	0.4880±0.02[3]
LLOG	0.3138±0.02[6]	0.3219±0.03[4]	0.3223±0.03[3]	0.2978±0.03[7]	0.3153±0.03[5]	0.3269±0.02[2]	**0.3298**±0.03[1]
ART	0.4840±0.02[5]	0.5013±0.02[4]	0.5070±0.02[3]	-	-	0.5131±0.02[2]	**0.5135**±0.02[1]
EURLEX_SM_10	0.8594±0.00[5]	**0.8609**±0.00[1]	0.8606±0.00[3]	-	-	0.8600±0.00[4]	**0.8609**±0.00[1]
EURLEX_ED_10	0.7170±0.01[5]	0.7176±0.01[4]	0.7183±0.01[2]	-	-	0.7183±0.01[2]	**0.7190**±0.01[1]
Average Rank	5.88	3.88	3.63	4.60	3.80	2.63	1.50

Table 10: Results of Macro-F1 on the various small-scale data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" denotes the training time is more than one week.

Data set	BR	CC	ECC	CCA	MMOC	CC-Greedy	CC-DP
YEAST	0.3543±0.01[4]	**0.3993**±0.03[1]	0.3763±0.02[2]	0.3496±0.02[5]	0.3431±0.02[7]	0.3441±0.02[6]	0.3596±0.02[3]
IMAGE	0.5852±0.01[7]	**0.6013**±0.02[1]	0.5988±0.01[4]	0.6010±0.01[2]	0.5975±0.01[6]	0.5987±0.02[5]	0.6010±0.01[2]
SLASHDOT	0.3416±0.01[4]	0.3485±0.02[2]	0.3331±0.01[7]	**0.3512**±0.02[1]	0.3334±0.01[6]	0.3431±0.01[3]	0.3408±0.01[5]
ENRON	0.2089±0.02[2]	0.2066±0.02[5]	0.2088±0.02[3]	0.1594±0.03[6]	0.1539±0.02[7]	**0.2090**±0.02[1]	0.2082±0.02[4]
LLOG	0.3452±0.03[2]	0.3428±0.03[4]	0.3425±0.04[5]	0.3189±0.04[7]	0.3303±0.04[6]	0.3448±0.03[3]	**0.3471**±0.04[1]
ART	0.4836±0.01[4]	0.4816±0.01[5]	0.4851±0.02[3]	-	-	0.4876±0.01[2]	**0.4884**±0.02[1]
EURLEX_SM_10	0.8546±0.00[5]	0.8558±0.00[2]	0.8554±0.00[3]	-	-	0.8550±0.00[4]	**0.8559**±0.00[1]
EURLEX_ED_10	0.7201±0.01[5]	0.7202±0.01[4]	0.7205±0.01[3]	-	-	0.7208±0.01[2]	**0.7217**±0.01[1]
Average Rank	4.13	3.00	3.75	4.20	6.40	3.25	2.25

- CC improves on the performance of BR, however, it underperforms compared to ECC. This result verifies the answer to our first question stated in Section 2.2: the label order does affect the performance of CC; ECC, which averages over several CC predictions with random order, improves the performance of CC.

Table 11: Results of Micro-F1 on the various small-scale data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" denotes the training time is more than one week.

Data set	BR	CC	ECC	CCA	MMOC	CC-Greedy	CC-DP
YEAST	0.6320±0.02[4]	0.6185±0.03[7]	0.6306±0.02[5]	**0.6362**±0.03[1]	0.6361±0.02[2]	0.6303±0.02[6]	0.6328±0.02[3]
IMAGE	0.5840±0.02[7]	0.5994±0.02[2]	0.5955±0.01[5]	**0.6003**±0.01[1]	0.5958±0.01[4]	0.5946±0.02[6]	0.5980±0.01[3]
SLASHDOT	0.5233±0.02[6]	0.5278±0.03[3]	0.5175±0.03[7]	**0.5844**±0.02[1]	0.5720±0.02[2]	0.5266±0.02[5]	0.5272±0.02[4]
ENRON	0.5052±0.01[6]	0.5013±0.01[7]	0.5056±0.01[5]	0.5335±0.02[2]	**0.5401**±0.01[1]	0.5104±0.01[3]	0.5096±0.01[4]
LLOG	**0.3768**±0.03[1]	0.3712±0.03[6]	0.3730±0.04[5]	0.3623±0.03[7]	0.3760±0.03[3]	0.3744±0.03[4]	0.3762±0.03[2]
ART	0.5122±0.02[5]	0.5130±0.02[4]	0.5156±0.02[3]	-	-	**0.5184**±0.01[1]	**0.5184**±0.02[1]
EURLEX_SM_10	0.8718±0.00[5]	0.8727±0.00[2]	0.8725±0.00[3]	-	-	0.8722±0.00[4]	**0.8733**±0.00[1]
EURLEX_ED_10	0.7419±0.01[5]	0.7421±0.01[4]	0.7424±0.01[3]	-	-	0.7425±0.01[2]	**0.7432**±0.01[1]
Average Rank	4.88	4.38	4.50	2.40	2.40	3.88	2.38

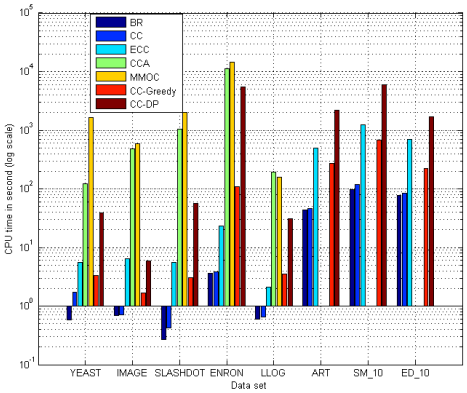


Figure 4: Training time (in second) of all methods on the small-scale data sets. EURLEX_SM_10 and EURLEX_ED_10 are abbreviated to SM_10 and ED_10.

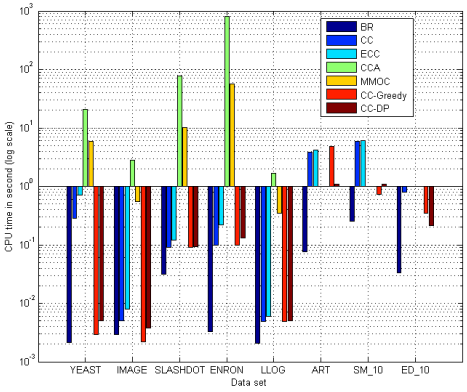


Figure 5: Testing time (in second) of all methods on the small-scale data sets. EURLEX_SM_10 and EURLEX_ED_10 are abbreviated to SM_10 and ED_10.

- Our algorithms outperform CCA and MMOC. This study verifies that optimal CC achieves competitive results compared with state-of-the-art encoding-decoding approaches.

- Our proposed CC-DP and CC-Greedy algorithms are successful on most data sets. This empirical result verifies the effectiveness of our easy-to-hard learning strategies, and we provide an answer to the last two questions stated in Section 2.2: a globally optimal CC exists and CC-DP is able to find the globally optimal CC that achieves the best prediction performance. The CC-Greedy algorithm achieves comparable prediction performance with CC-DP.

Figures 4 and 5 show the training and testing time of CC-Greedy, CC-DP and the baseline methods on the small-scale data sets, respectively. According to these two figures, we can see that:

- Our proposed algorithms are much faster than CCA and MMOC in terms of both training and testing time.
- CC-Greedy and CC-DP achieve comparable testing time with BR, CC and ECC. Though the training time of our algorithms are slower than BR, CC and ECC, our extensive empirical studies show that our algorithms achieve superior prediction performance than those baselines.
- The CC-Greedy algorithm is much faster than CC-DP in terms of training time, and it achieves comparable prediction performance with CC-DP.

7.2.2 Large-scale results

This subsection studies the performance of Tree-Greedy, Tree-DP and other baselines on the EURLEX_SM and EURLEX_ED data sets with many labels. We cannot get the results of CCA and MMOC on EURLEX_SM and EURLEX_ED data sets in one week. And we also cannot get the results of ECC on EURLEX_ED data set in one week. The prediction performance of Tree-Greedy, Tree-DP and the baselines are reported in Tables 12, 13 and 14. We conduct the pairwise t-test at a 5% significance level to show that our methods perform significantly better than the compared methods. From the results, we can see that: our proposed Tree-Greedy and Tree-DP algorithms consistently outperform BR, CC and ECC on the data sets with many labels.

The training and testing time of Tree-Greedy, Tree-DP and the baselines on the EURLEX_SM and EURLEX_ED data sets are shown in Figure 6. According to this figure, we can observe that compared to BR, CC and ECC, our algorithms maintain the testing time over an acceptable threshold, while our methods are much faster than the baselines in terms of training time.

7.3 Experiment on Ordinal Classification

This subsection conducts experiments on four ordinal data sets with various domains from website[9]. The statistics of these data sets are presented in Table 15. We compare our algorithm with some baseline methods: SVM (Shashua and Levin, 2002), MAP (Chu and Ghahramani, 2005), EP (Chu and Ghahramani, 2005) and BD (Destercke and Yang, 2014).

9. http://www.gatsby.ucl.ac.uk/ chuwei/ordinalregression.html

Table 12: Results of Example-F1 on EURLEX_SM and EURLEX_ED data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" denotes the training time is more than one week.

Data set	BR	CC	ECC	Tree-Greedy	Tree-DP
EURLEX_SM	0.6970±0.02[5]	0.7233±0.01[4]	0.7263±0.01[3]	0.7292±0.01[2]	**0.7301**±0.01[1]
EURLEX_ED	0.4345±0.03[4]	0.4528±0.02[3]	-	0.4550±0.01[2]	**0.4563**±0.01[1]
Average Rank	4.5	3.5	3	2	1

Table 13: Results of Macro-F1 on EURLEX_SM and EURLEX_ED data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" denotes the training time is more than one week.

Data set	BR	CC	ECC	Tree-Greedy	Tree-DP
EURLEX_SM	0.4777±0.02[5]	0.4785±0.02[4]	0.4800±0.02[3]	0.4817±0.01[2]	**0.4834**±0.01[1]
EURLEX_ED	0.1660±0.01[4]	0.1812±0.00[3]	-	0.1844±0.00[2]	**0.1848**±0.00[1]
Average Rank	4.5	3.5	3	2	1

The results are shown in Table 16. From this table, we can see that our proposed CCMC-DP outperforms the other baselines on all data sets, which verifies that our method is able to capture and use the correlated information between ordinal classes, and boost the performance of ordinal classification problems.

7.4 Experiment on Relationship Prediction

This subsection conducts experiments on the Epinions data set (Massa and Avesani, 2006). According to Chiang et al. (2015), we collect 10,000 users and 41 features in this data set. We compare our proposed Tree-DP algorithm with some popular methods: IMC (Jain and Dhillon, 2013), MF-ALS (Hsieh et al., 2012), HOC-3 (Chiang et al., 2014), HOC-5 (Chiang et al., 2014) and DirtyIMC (Chiang et al., 2015). We perform 5-fold cross-validation on this data set and report the mean and standard error of Example-F1. The results are shown in Table 17. From this table, we can see that 1) DirtyIMC outperforms the other baselines, which verifies the effectiveness of using feature information and is consistent with the empirical results in Chiang et al. (2015). 2) Our proposed Tree-DP algorithm is able to achieve the best accuracy among all baselines, which demonstrates the superior performance of the easy-to-hard learning strategy.

8. Conclusion

To precisely classify multiclass data sets with confusing classes, this paper aims to solve classification tasks from easy to hard, and then use the predictions from simpler tasks to help solve the harder tasks. To achieve our goal, we first build the classifier chain model for

Table 14: Results of Micro-F1 on EURLEX_SM and EURLEX_ED data sets (mean ± standard deviation). The best results are in bold. Numbers in square brackets indicate the rank. "-" denotes the training time is more than one week.

Data set	BR	CC	ECC	Tree-Greedy	Tree-DP
EURLEX_SM	0.7321±0.01[5]	0.7422±0.02[3]	0.7363±0.01[4]	0.7440±0.01[2]	**0.7454**±0.01[1]
EURLEX_ED	0.4200±0.01[4]	0.4477±0.01[3]	-	0.4527±0.00[2]	**0.4549**±0.00[1]
Average Rank	4.5	3	4	2	1

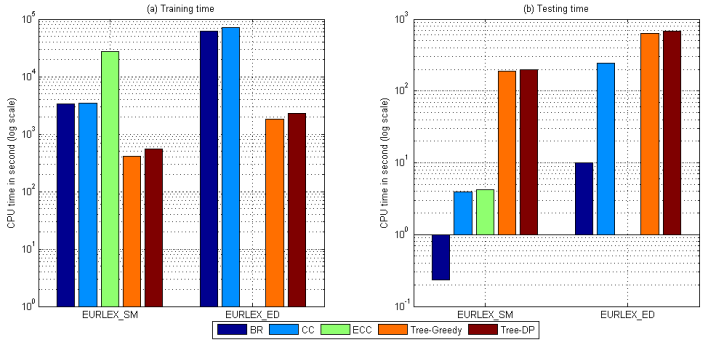


Figure 6: Training and testing time (in second) of BR, CC, ECC, Tree-Greedy and Tree-DP on EURLEX_SM and EURLEX_ED data sets.

multiclass classification (CCMC) to transfer class information between classifiers. Then, we generalize the CCMC model over a random class order and provide a theoretical analysis of the generalization error for the proposed generalized model. Our results show that the upper bound of the generalization error depends on the sum of the reciprocal of the square of the margin over the classes. Based on our results, we propose the easy-to-hard learning paradigm for multiclass classification to automatically identify easy and hard classes and then use the predictions from simpler classes to help solve harder classes.

Similar to CCMC, a CC model is also proposed by Read et al. (2009) to capture the label dependency for multi-label classification. However, confusing labels decrease the generalization performance of CC, especially when there are many confusing labels, because it ignores the label's order of difficulty. Thus, it is imperative to learn the appropriate label order for CC. Motivated by our analysis of CCMC, we first generalize the CC over a random label order and provide the generalization error bound for the proposed generalized model, and then we also propose the easy-to-hard learning paradigm for multi-label classification

Table 15: Ordinal data sets used in the experiments.

Data set	# Training	# Testing	# Rank
Stocks Domain	600	350	5
Machine CPU	150	59	5
Abalone	1,000	3,177	6
Boston Housing	300	206	5

Table 16: Testing error rate (in %) on ordinal data sets.

Data set	SVM	MAP	EP	BD	CCMC-DP
Stocks Domain	10.84	11.99	12.00	10.05	9.87
Machine CPU	19.15	18.47	18.56	18.71	17.34
Abalone	22.93	23.22	23.37	23.43	21.62
Boston Housing	26.72	26.04	25.85	25.86	23.50

to automatically identify easy and hard labels. Lastly, we use the predictions from simpler labels to help solve harder labels.

It is very expensive to search over $q!$ different class or label orders for learning the objective of our proposed easy-to-hard learning paradigms, which is computationally infeasible for a large q. We thus propose the CCMC-DP and CC-DP algorithms to find the globally optimal solution, respectively, which requires $\mathcal{O}(q^3nd)$ complexity. To speed up the CCMC-DP and CC-DP algorithm, we propose the CCMC-Greedy and CC-Greedy algorithms to find a locally optimal CCMC and CC, respectively, which takes $\mathcal{O}(q^2nd)$ time. Fast greedy and tree-based algorithms are further developed to handle large data sets with many classes and labels, respectively, which scale linearly with q.

Comprehensive experiments on extensive multiclass data sets, without and with background, demonstrate that our proposed methods consistently improve the prediction performance of OVR and outperforms ECOC and Top-k multiclass SVM. Our human action recognition experiment results also validate our analysis and the success of our proposed easy-to-hard learning strategies: we can automatically identify easy and hard classes, and use the predictions of classifiers from easier classes to train the classifiers for harder classes. Furthermore, this paper also provide an affirmative answer to Rifkin and Klautau's conjecture.

Empirical results on ten real-world multi-label data sets from different domains verify the effectiveness of our easy-to-hard learning strategies, and we provide an answer to the last two questions stated in Section 2.2: a globally optimal CC exists, and CC-DP is able to find the globally optimal CC which achieves the best prediction performance. Moreover, we provide theoretical support for the argument: multi-label learning methods which explicitly capture the label's relationship will usually achieve better prediction performance.

Table 17: Results of Example-F1 on the Epinions data sets (mean ± standard deviation).

Data set	IMC	MF-ALS	HOC-3	HOC-5	DirtyIMC	Tree-DP
Epinions	0.910 ± 0.002	0.936 ± 0.001	0.926 ± 0.001	0.927 ± 0.002	0.939 ± 0.001	0.946 ± 0.001

Lastly, we demonstrate our proposed easy-to-hard learning strategies can be successfully applied to a wide range of applications, such as ordinal classification and relationship prediction. From a more philosophical point of view, our work has shown that the proper usage of structure information in multiclass and multi-label problems yields better modeling, in other words structuring output class information may be an attractive path to incorporate more dark knowledge into learning models.

Acknowledgments

We would like to thank the action editor and reviewers for their valuable comments and constructive suggestions on our paper. This project was partially supported by the ARC Future Fellowship FT130100746, ARC grant LP150100671, DP170101628, DP150102728, DP150103071, NSFC 61232006 and NSFC 61672235. Klaus-Robert Müller acknowledges partial funding from DFG & BMBF and was also supported by Institute for Information & Communications Technology Promotion (IITP) grant funded by the Korea government (No. 2017-0-00451).

Appendix A. Covering Numbers

Definition 19 (Covering Numbers) *Let (X, d) be a (pseudo-) metric space, A be a subset of X and $\epsilon > 0$. A set $B \subseteq X$ is an ϵ-cover for A, if for every $a \in A$, there exists $b \in B$ such that $d(a, b) < \epsilon$. The ϵ-covering number of A, $\mathcal{N}_d(\epsilon, A)$, is the minimal cardinality of an ϵ-cover for A (if there is no such finite cover then it is defined to be ∞).*

In the main paper, let $\mathcal{N}(\epsilon, \mathcal{H}, \mathbf{s})$ be the ϵ-covering number of $\mathcal{H}$ with respect to the l_∞ pseudo-metric measuring the maximum discrepancy on the sample $\mathbf{s}$, that is, with respect to the distance $d(f, g) = \max_{1 \le t \le m} |f(x_t) - g(x_t)|$, for $f, g \in \mathcal{H}$.

Appendix B. Proof of Lemma 6

Proof (of Lemma 6). For each $\mathbf{s}$, let $\bar{h}_{\mathbf{s}}$ be a function for which $|er_D[\bar{h}_{\mathbf{s}}] - er_{\mathbf{s}}[\bar{h}_{\mathbf{s}}]| \ge \epsilon$ if such a function exists, and any fixed function in $\mathcal{H}$ otherwise. Then

$$\begin{aligned} P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{h\in\mathcal{H}} |er_{\bar{\mathbf{s}}}[h] - er_{\mathbf{s}}[h]| \ge \epsilon/2) \ge & P_{\mathbf{s}\bar{\mathbf{s}}}(|er_{\bar{\mathbf{s}}}[\bar{h}_{\mathbf{s}}] - er_{\mathbf{s}}[\bar{h}_{\mathbf{s}}]| \ge \epsilon/2) \\ \ge & P_{\mathbf{s}\bar{\mathbf{s}}}(\{|er_D[\bar{h}_{\mathbf{s}}] - er_{\mathbf{s}}[\bar{h}_{\mathbf{s}}]| \ge \epsilon\} \bigcap \{|er_{\bar{\mathbf{s}}}[\bar{h}_{\mathbf{s}}] - er_D[\bar{h}_{\mathbf{s}}]| \le \epsilon/2\}) \\ = & E_{\mathbf{s}}[\mathbf{I}(|er_D[\bar{h}_{\mathbf{s}}] - er_{\mathbf{s}}[\bar{h}_{\mathbf{s}}]| \ge \epsilon) P_{\bar{\mathbf{s}}}(|er_{\bar{\mathbf{s}}}[\bar{h}_{\mathbf{s}}] - er_D[\bar{h}_{\mathbf{s}}]| \le \epsilon/2)] \end{aligned} \tag{5}$$

Now the conditional probability inside can be bounded using Chebyshev's inequality:

$$P_{\bar{\mathbf{s}}}(|er_{\bar{\mathbf{s}}}[\bar{h}_{\mathbf{s}}] - er_D[\bar{h}_{\mathbf{s}}]| \le \epsilon/2) \ge 1 - \frac{\mathbf{Var}_{\bar{\mathbf{s}}}[er_{\bar{\mathbf{s}}}[\bar{h}_{\mathbf{s}}]]}{\epsilon^2/4} \tag{6}$$

Since $\bar{\mathbf{s}} \sim D^m$ and $er_{\bar{\mathbf{s}}}[\bar{h}_{\mathbf{s}}]$ is $1/m$ times a Binomial random variable with parameters $(m, er_D[\bar{h}_{\mathbf{s}}])$, we have $\mathbf{Var}_{\bar{\mathbf{s}}}[er_{\bar{\mathbf{s}}}[\bar{h}_{\mathbf{s}}]] = \frac{er_D[\bar{h}_{\mathbf{s}}](1-er_D[\bar{h}_{\mathbf{s}}])}{m} \le \frac{1}{4m}$. This gives

$$P_{\bar{\mathbf{s}}}(|er_{\bar{\mathbf{s}}}[\bar{h}_{\mathbf{s}}] - er_D[\bar{h}_{\mathbf{s}}]| \le \epsilon/2) \ge 1 - \frac{1}{m\epsilon^2} \ge \frac{1}{2} \tag{7}$$

whenever $m\epsilon^2 \ge 2$. Thus we get

$$\begin{aligned} P_{\mathbf{s}\bar{\mathbf{s}}}(\sup_{h\in\mathcal{H}} |er_{\bar{\mathbf{s}}}[h] - er_{\mathbf{s}}[h]| \ge \epsilon/2) \ge & \frac{1}{2} P_{\mathbf{s}}(|er_D[\bar{h}_{\mathbf{s}}] - er_{\mathbf{s}}[\bar{h}_{\mathbf{s}}]| \ge \epsilon) \\ = & \frac{1}{2} P_{\mathbf{s}}(\sup_{h\in\mathcal{H}} |er_D[h] - er_{\mathbf{s}}[h]| \ge \epsilon) \end{aligned} \tag{8}$$

where the last step of Eq. (8) is by definition of $\bar{h}_{\mathbf{s}}$. ■

Appendix C. Proof of Theorem 15

Proof (of Theorem 15). We proof the theorem using the mathematical induction. For $i \in \{1, \cdots, q\}$,

Case 1: $V(i,1) = \frac{1}{(\gamma_i^1)^2}$, where γ_i^1 is the margin for λ_i, without augmented input and $M_i^1 = \{\lambda_i\}$.

Case 2: $V(i,2) = \min_{j \neq i, \lambda_i \notin M_j^1}\{(\frac{1}{(\gamma_i^2)^2} + V(j,1)\}$, where γ_i^2 is the margin for λ_i, with M_j^1 as the augmented input. As in case 1, we already calculated $V(i,1)$, so we can easily find the solution of $V(i,2)$. Assume $V(j,1)$ is the optimal value for computing $V(i,2)$, then we can get $M_i^2 = M_j^1 \cup \{\lambda_i\}$.

Case 3: Assume $V(i,k-1), k \leq q$ is the optimal Q' over a subset of $\mathcal{M}$ with the length of $k-1$, where the class or label order ends by λ_i and M_i^{k-1} denote the corresponding class or label set for $V(i,k-1)$.

Case 4: $V(i,k) = \min_{j \neq i, \lambda_i \notin M_j^{k-1}}\{\frac{1}{(\gamma_i^k)^2} + V(j,k-1)\}$, where γ_i^k is the margin for λ_i, with M_j^{k-1} as the augmented input. Based on the assumption in case 3, we can obtain $V(i,k), i \in \{1, \cdots, q\}$. Thus, we can find the optimal Q' over $\mathcal{M}$ by using CCMC-DP or CC-DP algorithm. ■

Appendix D. Greedy Algorithm

This section presents the details of the CC-Greedy algorithm. Let $\{\mathbf{x}_t\}_{t=1}^n$ be the feature and $\{\mathbf{y}_t(\zeta_j)\}_{t=1}^n$ be the label, the output parameter of SVM is defined as $[\mathbf{w}_j, b] = SVM(\{\mathbf{x}_t, \mathbf{y}_t(\zeta_1), \cdots, \mathbf{y}_t(\zeta_{j-1})\}_{t=1}^n, \{\mathbf{y}_t(\zeta_j)\}_{t=1}^n)$.

Algorithm 1 CC-Greedy

Input: training data $\{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^n$ with size n and label set $\{\lambda_1, \lambda_2, \cdots, \lambda_q\}$.
Set $\mathcal{M} = \{\lambda_1, \lambda_2, \cdots, \lambda_q\}$.
for $\lambda_j \in \mathcal{M}$ **do**
  Calculate $[\mathbf{w}_j, b] = SVM(\{\mathbf{x}_t\}_{t=1}^n, \{\mathbf{y}_t(\lambda_j)\}_{t=1}^n)$.
  Calculate $\gamma_j^1 = \frac{1}{||\mathbf{w}_j||^2}$.
end for
Calculate $\nu = \arg_{\lambda_j \in \mathcal{M}} \min \frac{1}{(\gamma_j^1)^2}$.
Set $\mathcal{M} = \mathcal{M} - \{\lambda_\nu\}$
Set $C[1] = \lambda_\nu$.
for $s = 2$ **to** q **do**
  for $\lambda_k \in \mathcal{M}$ **do**
    Calculate $[\mathbf{w}_k, b] = SVM(\{\mathbf{x}_t, \mathbf{y}_t(C[1]), \cdots, \mathbf{y}_t(C[s-1])\}_{t=1}^n, \{\mathbf{y}_t(\lambda_k)\}_{t=1}^n)$.
    Calculate $\gamma_k^s = \frac{1}{||\mathbf{w}_k||^2}$.
  end for
  Calculate $\nu = \arg_{\lambda_k \in \mathcal{M}} \min \frac{1}{(\gamma_k^s)^2}$.
  Set $\mathcal{M} = \mathcal{M} - \{\lambda_\nu\}$.
  Set $C[s] = \lambda_\nu$.
end for
Output this locally optimal CC.

References

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NIPS*, pages 2654–2662, 2014.

Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In *Advances in Kernel Methods - Support Vector Learning*, pages 43–54. MIT Press, Cambridge, MA, USA, 1998.

Zafer Barutcuoglu, Robert E. Schapire, and Olga G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In *Advances in Neural Information Processing Systems 23*, pages 163–171, 2010.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, pages 41–48, 2009.

Alina Beygelzimer, John Langford, Yury Lifshits, Gregory B. Sorkin, and Alexander L. Strehl. Conditional probability tree estimation analysis and algorithms. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 51–58, 2009a.

Alina Beygelzimer, John Langford, and Pradeep Ravikumar. Error-correcting tournaments. In *Proceedings of the 20th Conference on Algorithmic Learning Theory*, pages 247–262, 2009b.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and C.M.Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla. Label propagation in complex video sequences using semi-supervised learning. In *British Machine Vision Conference*, pages 1–12. British Machine Vision Association, 2010.

Yao-Nan Chen and Hsuan-Tien Lin. Feature-aware label space dimension reduction for multi-label classification. In *Advances in Neural Information Processing Systems 25*, pages 1538–1546, 2012.

Kai-Yang Chiang, Cho-Jui Hsieh, Nagarajan Natarajan, Inderjit S. Dhillon, and Ambuj Tewari. Prediction and clustering in signed networks: a local to global perspective. *Journal of Machine Learning Research*, 15(1):1177–1213, 2014.

Kai-Yang Chiang, Cho-Jui Hsieh, and Inderjit S. Dhillon. Matrix completion with noisy side information. In *NIPS*, pages 3447–3455, 2015.

Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.

Moustapha Cissé, Maruan Al-Shedivat, and Samy Bengio. ADIOS: architectures deep in output space. In *ICML*, pages 2770–2779, 2016.

Krzysztof Dembczynski, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning*, pages 279–286, Haifa, Israel, 2010. Omnipress.

Can Demirkesen and Hocine Cherifi. An evaluation of divide-and-combine strategies for image categorization by multi-class support vector machines. In *23rd International Symposium on Computer and Information Sciences*, pages 1–6, 2008.

Sébastien Destercke and Gen Yang. Cautious ordinal classification by binary decomposition. In *ECML/PKDD*, pages 323–337, 2014.

Thomas G. Dietterich and Ghulum Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *AISTATS*, pages 315–323, 2011.

Chen Gong, Dacheng Tao, Wei Liu, Liu Liu, and Jie Yang. Label propagation via teaching-to-learn and learning-to-teach. *IEEE Trans. Neural Netw. Learning Syst.*, 28(6):1452–1465, 2017.

Matthieu Guillaumin, Jakob J. Verbeek, and Cordelia Schmid. Multimodal semi-supervised learning for image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 902–909. IEEE Computer Society, 2010.

Yuhong Guo and Suicheng Gu. Multi-label classification using conditional dependency networks. In Toby Walsh, editor, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 1300–1305, Barcelona, Catalonia, Spain, 2011. AAAI Press.

Yuhong Guo and Dale Schuurmans. Adaptive large margin training for multilabel classification. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

Xuming He, Richard S. Zemel, and Miguel Á. Carreira-Perpiñán. Multiscale conditional random fields for image labeling. In *CVPR*, pages 695–702, 2004.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

Cho-Jui Hsieh, Kai-Yang Chiang, and Inderjit S. Dhillon. Low rank modeling of signed networks. In *KDD*, pages 507–515, 2012.

Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks and Learning Systems*, 13(2):415–425, 2002.

Daniel Hsu, Sham Kakade, John Langford, and Tong Zhang. Multi-label prediction via compressed sensing. In *Advances in Neural Information Processing Systems*, pages 772–780, 2009.

Sheng-Jun Huang and Zhi-Hua Zhou. Multi-label learning by exploiting label correlations locally. In Jörg Hoffmann and Bart Selman, editors, *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, Toronto, Ontario, Canada, 2012. AAAI Press.

Lina Huo, Licheng Jiao, Shuang Wang, and Shuyuan Yang. Object-level saliency detection with color attributes. *Pattern Recognition*, 49:162–173, 2016.

Prateek Jain and Inderjit S. Dhillon. Provable inductive matrix completion. *CoRR*, abs/1306.0626, 2013.

Feng Kang, Rong Jin, and Rahul Sukthankar. Correlated label propagation with application to multi-label learning. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1719–1726, NY, USA, 2006. IEEE Computer Society.

Michael J. Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts. In *Proceedings of the 31st Symposium on the Foundations of Computer Science*, pages 382–391, Los Alamitos, CA, 1990. IEEE Computer Society Press.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Top-k multiclass SVM. In *Advances in Neural Information Processing Systems 28*, pages 325–333, 2015.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Loss functions for top-k error: Analysis and insights. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Jure Leskovec, Daniel P. Huttenlocher, and Jon M. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*, pages 641–650, 2010.

Weiwei Liu and Ivor W. Tsang. Large margin metric learning for multi-label prediction. In *AAAI*, pages 2800–2806, 2015a.

Weiwei Liu and Ivor W. Tsang. On the optimality of classifier chain for multi-label classification. In *NIPS*, pages 712–720, 2015b.

Weiwei Liu and Ivor W. Tsang. Sparse perceptron decision tree for millions of dimensions. In *AAAI*, pages 1881–1887, 2016.

Weiwei Liu and Ivor W. Tsang. Making decision trees feasible in ultrahigh feature and label dimensions. *Journal of Machine Learning Research*, 18:1–36, 2017.

Weiwei Liu, Xiaobo Shen, and Ivor W. Tsang. Sparse embedded k-means clustering. In *NIPS*, 2017.

Qi Mao, Ivor Wai-Hung Tsang, and Shenghua Gao. Objective-guided image annotation. *IEEE Transactions on Image Processing*, 22(4):1585–1597, 2013.

Paolo Massa and Paolo Avesani. Trust-aware bootstrapping of recommender systems. In *Proceedings of ECAI 2006 Workshop on Recommender Systems*, pages 29–33, 2006.

Jonathan Milgram, Mohamed Cheriet, and Robert Sabourin. "one against one" or "one against all": Which one is better for handwriting recognition with SVMs? In *Tenth International Workshop on Frontiers in Handwriting Recognition*, 2006.

Gang Niu, Marthinus Christoffel du Plessis, Tomoya Sakai, Yao Ma, and Masashi Sugiyama. Theoretical comparisons of positive-unlabeled learning against positive-negative learning. In *NIPS*, pages 1199–1207, 2016.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Trung T. Pham, Ian Reid, Yasir Latif, and Stephen Gould. Hierarchical higher-order regression forest fields: An application to 3D indoor scene labelling. In *ICCV*, 2015.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. In Wray L. Buntine, Marko Grobelnik, Dunja Mladenic, and John Shawe-Taylor, editors, *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II*, pages 254–269, Berlin, Heidelberg, 2009. Springer-Verlag.

Ryan M. Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2-3):135–168, 2000.

Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *17th International Conference on Pattern Recognition*, pages 32–36. IEEE Computer Society, 2004.

Chun-Wei Seah, Ivor W. Tsang, and Yew-Soon Ong. Transductive ordinal regression. *TNNLS*, 23(7):1074–1086, 2012.

Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In *NIPS*, pages 937–944, 2002.

John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *12th European Conference on Computer Vision*, pages 746–760. Springer, 2012.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

Farbound Tai and Hsuan-Tien Lin. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.

Ali Fallah Tehrani, Weiwei Cheng, and Eyke Hüllermeier. Preference learning using the choquet integral: The case of multipartite ranking. *IEEE Trans. Fuzzy Systems*, 20(6): 1102–1113, 2012.

Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems*, pages 1401–1408, 2004.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, pages 667–685. Springer US, 2010.

Jason Weston and Chris Watkins. Support vector machines for multi-class pattern recognition. In *7th European Symposium on Artificial Neural Networks*, pages 219–224, 1999.

Jian-Bo Yang and Ivor W. Tsang. Hierarchical maximum margin learning for multi-class classification. In *UAI*, pages 753–760, 2011.

Min-Ling Zhang and Kun Zhang. Multi-label learning by exploiting label dependency. In Bharat Rao, Balaji Krishnapuram, Andrew Tomkins, and Qiang Yang, editors, *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 999–1008, Washington, DC, USA, 2010. ACM.

Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.

Yi Zhang and Jeff Schneider. Maximum margin output coding. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning*, pages 1575–1582, New York, NY, USA, 2012. Omnipress.

Yi Zhang and Jeff G. Schneider. Multi-label output codes using canonical correlation analysis. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 873–882, Fort Lauderdale, USA, 2011. JMLR.org.

Journal of Machine Learning Research 18 (2017) 1-32 Submitted 1/17; Revised 4/17; Published 9/17

Fisher Consistency for Prior Probability Shift

Dirk Tasche* dirk.tasche@gmx.net
Swiss Financial Market Supervisory Authority FINMA
Laupenstrasse 27
3003 Bern
Switzerland

Editor: Charles Elkan

Abstract

We introduce Fisher consistency in the sense of unbiasedness as a desirable property for estimators of class prior probabilities. Lack of Fisher consistency could be used as a criterion to dismiss estimators that are unlikely to deliver precise estimates in test data sets under prior probability and more general data set shift. The usefulness of this unbiasedness concept is demonstrated with three examples of classifiers used for quantification: Adjusted Count, EM-algorithm and CDE-Iterate. We find that Adjusted Count and EM-algorithm are Fisher consistent. A counter-example shows that CDE-Iterate is not Fisher consistent and, therefore, cannot be trusted to deliver reliable estimates of class probabilities.

Keywords: Classification, quantification, class distribution estimation, Fisher consistency, data set shift

1. Introduction

The application of a classifier to a test data set often is based on the assumption that the data for training the classifier is representative of the test data. While this assumption might be true sometimes or even most of the time, there may be circumstances when the distributions of the classes or the features or both are genuinely different in the test set and the training set. Spam emails represent a familiar example of this situation: The typical contents of spam emails and their proportion of the total daily number of emails received may significantly vary over time. Spam email filters developed on the set of emails received last week may become less effective this week due to changes in the composition of the email traffic. In the machine learning community, this phenomenon is called *data set shift* or population drift. The area of research of how to learn a model appropriate for a test set on a differently composed training set is called domain adaptation.

The simplest type of data set shift occurs when the training set and the test set differ only in the distribution of the classes of the instances. In this case, only the prior (or unconditional) class probabilities (or *class prevalences*) change between the training set and the test set; this type of data set shift is called *prior probability shift*. In the context of supervised or semi-supervised learning, the class prevalences in the labelled portion of the training set are always known. In contrast, the class prevalences in the test set may be known or unknown at the time of the application of the classifier to the test set, depending on the cause of the data set shift. For example, in a binary classification exercise the class prevalence of the majority class in the training set might deliberately be reduced by removing instances of that class at random, in order to facilitate the training of the classifier. If the original training set were a random sample of the test set then the test set class prevalences would be equal to the original training set class prevalences and therefore known. The earlier mentioned spam email filter problem represents an example of the situation where the test set class prevalences are unknown due to possible data set shift.

In this paper, we primarily study the question of how to estimate the unknown class prevalences in a test data set when the training set and the test set are related by prior probability shift. This problem was coined *quantification* by Forman (2008). Solutions to the problem are known at least since the 1960s (Gart and Buck, 1966) but research for more and better solutions has been ongoing ever since. It seems, however, as if the criteria of what should be deemed a 'good' solution are not fully clear.

In the words of Cox and Hinkley (1974, p. 287) *Fisher consistency* is described as "Roughly this requires that if the whole 'population' of random variables is observed, then the method of estimation should give exactly the right answer". Obtaining the right answer when looking for the value of a parameter is an intuitive concept that exhibits similarity to the concept of unbiasedness. In particular, if a wrong decision based on the estimated value of a parameter can entail severe financial loss or cause damage to a person's health, getting the value of the parameter right is very important. Hofer and Krempl (2013) and Tasche (2014) present examples related to the estimation of credit losses. Nonetheless, requiring that the right answer be obtained also on finite samples from the population would be too harsh, given that by the nature of randomness the empirical distribution associated with a finite sample can be quite different from the population distribution.

For these reasons, we argue that as a minimum, it should be required that class prevalence estimators (or *quantifiers*) be Fisher consistent. Cox and Hinkley (1974, p. 287) comment that "Consistency, however, gives us no idea of the magnitude of the errors of estimation likely for any given n [the sample size]". Therefore, being Fisher consistent cannot be a sufficient criterion for a class prevalence estimator to be useful. Rather, by the logic of a necessary criterion, lack of Fisher consistency should be considered a reason to dismiss a candidate estimator. For even for very large sample sizes, with such an estimator there would not be any guarantee of obtaining an approximation of the true parameter value.

Accordingly, focussing on *binary* classification and quantification, with this paper we make three contributions to the literature on quantification of class prevalences:

- We formally introduce the concept of unbiasedness expressed as Fisher consistency, in a prior probability shift context and more generally for quantification of class prevalences in the presence of any data set shift.

- We illustrate the usefulness of the notion of Fisher consistency, by demonstrating with three popular quantification approaches that Fisher consistency may be used as a criterion to cull inapt quantifiers. In other words, Fisher consistency can serve as a filter to eliminate quantifiers that are unlikely to provide precise estimates.

*. The author currently works at the Swiss Financial Market Supervisory Authority (FINMA). The opinions expressed in this note are those of the author and do not necessarily reflect views of FINMA.
The author thanks the reviewers of this article for suggesting numerous improvements.

©2017 Dirk Tasche.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-048.html.

- We show that Fisher consistency of an estimator is not a global concept that can be expected to hold for all types of data set shift. To demonstrate this fact, we suggest a new type of data set shift, called 'invariant density ratio'-type data set shift, which generalises prior probability shift. We also propose a method for generating non-trivial examples of this type of data set shift.

'Invariant density ratio'-type data set shift is interesting of its own for the following two reasons:

- It can be described as a covariate shift without the 'contamination' effect that can entail the test set class prevalences to be very similar to the training set class prevalences (Tasche, 2013, p. 83).
- Prior probability shift is a special case of 'invariant density ratio'-type data set shift (see Section 2.4 below). Therefore, estimators which are Fisher consistent with respect to 'Invariant density ratio'-type data set shift potentially have a wider scope of application than estimators that are Fisher consistent only with respect to prior probability shift.

In Section 2.4.3, this list is supplemented by a number of other interesting properties of 'invariant density ratio'-type data set shift. The plan for the paper is as follows:

- In Section 2, we recall the concepts from binary classification and quantification that are relevant for the subject of the paper. These concepts include Bayes classifiers and some types of data set shift, including prior probability shift. In addition, we motivate and define the notion of Fisher consistency.
- In Section 3, we describe three approaches to class prevalence estimation which serve to illustrate the role of Fisher consistency for this task. The three approaches are Adjusted Count (Forman, 2008), the EM-algorithm (Saerens et al., 2001) and CDE-Iterate (Xue and Weiss, 2009).
- In Section 4, we explore by numerical examples under which circumstances the three approaches discussed in Section 3 cease to be Fisher consistent for the estimation of class prevalences. The most important finding here is that CDE-Iterate is not Fisher consistent under prior probability shift. Hence there is no guarantee that CDE-Iterate will find the true class prevalences even in the presence of that quite benign type of data set shift.
- We conclude in Section 5 with some comments on the findings of the paper.
- Appendix A presents some tables with computation results for additional information while Appendix B provides a mathematically rigorous derivation of the equation that characterises the limit of CDE-Iterate.

Notation. Concepts and notation used in this paper are formally introduced where needed. For additional quick reference, here is a short list of the most important symbols:

- $\mathcal{X}$: Feature space.

- $\{0,1\}$: Two classes, 0 positive and 1 negative.
- (X,Y): X is the vector of the features of an instance, Y denotes the class of the instance.
- P: Probability distribution of the training set.
- Q: Probability distribution of the test set.
- $\mathrm{P}[Y=i\,|\,X]$: Feature-conditional probability of class i in the training set (analogous for Q).
- $\mathrm{P}[X\in S\,|\,Y=i]$: $\mathrm{P}[X\,|\,Y]$ denotes the class-conditional distribution of the features in the training set. $\mathrm{P}[X\in S\,|\,Y=i]$ stands for the probability that the realisation of X is an element of the set S, conditional on the class of the instance being i (analogous for Q).
- $g:\mathcal{X}\to\{0,1\}$: Classifier that assigns class $g(x)$ to a realisation x of the features of an instance.
- f_0, f_1: Class-conditional feature densities on the training set.
- h_0, h_1: Class-conditional feature densities on the test set.

2. Estimating Prior Probabilities

In this section, we recall some basic concepts from the theory of binary classification and quantification to build the basis for the discussion of prevalence estimation approaches in Section 3 and the numerical examples in Section 4. The concepts discussed include Bayes classifiers and different types of data set shift including prior probability shift and its extension called here 'invariant density ratio'-type data set shift. In addition, we introduce the notion of unbiasedness of prevalence estimators in the shape of Fisher consistency which is most appropriate in the quantification context central for this paper.

2.1 Binary Classification and Bayes Error at Population Level

The basic model we consider is a random vector (X,Y) with values in a product $\mathcal{X}\times\{0,1\}$ of sets. For example, in applications for credit scoring, we might have $\mathcal{X}=\mathbb{R}^d$ for some positive integer d. A realisation of $X\in\mathcal{X}$ is interpreted as the vector of features of an observed instance. $Y\in\{0,1\}$ is the class of the observed instance. For the purpose of this paper, 0 is the interesting (positive) class, as in Hernández-Orallo et al. (2012).

Classification problem. On the basis of the observed features X of an instance, make a guess (prediction) $g(X)\in\{0,1\}$ of the instance's class Y such that the probability of an error is minimal. In mathematical terms: Find $g^*:\mathcal{X}\to\{0,1\}$ such that

$$\mathrm{P}[g^*(X)\neq Y] \;=\; \min_g \mathrm{P}[g(X)\neq Y]. \tag{1a}$$

The functions g in (1a) used for predicting Y are called (crisp) classifiers. The value of the minimum in (1a) is called *Bayes error*. (1a) accounts for two possibilities to make

classification errors:

$$\text{'Predict 0 if the true class is 1'} = \{g(X) = 0, Y = 1\}, \qquad \text{and}$$
$$\text{'Predict 1 if the true class is 0'} = \{g(X) = 1, Y = 0\}.$$

Cost-sensitive errors. In practice, the consequences of these two erroneous predictions might have different severities. Say the cost related to 'predict 0 if the true class is 1' is $c_1 \geq 0$, and the cost related to 'predict 1 if the true class is 0' is $c_0 \geq 0$. To deal with the issue of different severities, a cost-sensitive version of (1a) can be studied:

$$c_1 \, \mathrm{P}[g^*(X) = 0, \, Y = 1] + c_0 \, \mathrm{P}[g^*(X) = 1, \, Y = 0] = \min_g c_1 \, \mathrm{P}[g(X) = 0, \, Y = 1] + c_0 \, \mathrm{P}[g(X) = 1, \, Y = 0]. \quad (1b)$$

To make this problem non-trivial, of course one has to assume that $c_0 + c_1 > 0$.

Bayes classifier. A solution g^* to (1b) and therefore also to (1a) (case of $c_0 = c_1 = 1$) exists and is well-known (see Section 2.2 of van Trees, 1968, or Section 1.3 of Elkan, 2001):

$$g^*(X) \;=\; \begin{cases} 0, & \text{if } \mathrm{P}[Y = 0 \,|\, X] > \frac{c_1}{c_0 + c_1}, \\ 1, & \text{if } \mathrm{P}[Y = 0 \,|\, X] \leq \frac{c_1}{c_0 + c_1}. \end{cases} \quad (2)$$

In this equation, $\mathrm{P}[Y = 0 \,|\, X]$ denotes the (non-elementary) conditional probability of the event $Y = 0$ given X, as defined in standard textbooks on statistical learning and probability theory (see Appendix A.7 of Devroye et al., 1996, and Section 4.1 of Durrett, 1996, respectively). Being a function of X, $\mathrm{P}[Y = 0 \,|\, X]$ is also a non-constant random variable whenever X and Y are not stochastically independent. In the following, we also call $\mathrm{P}[Y = 0 \,|\, X]$ *feature-conditional class probability*. The function $g^*(X)$ as defined in (2) is called a Bayes classifier.

A proof of (2) is also provided in Appendix B below (see Lemma 5). That proof shows, in particular, that the solution g^* to (1b) is unique in the sense of $\mathrm{P}[g^*(X) = \tilde{g}(X)] = 1$ for any other minimiser $\tilde{g}$ of (1b), as long as the distribution of the ratio of the class-conditional feature densities is continuous (see Section 2.4 for the definition of the density ratio).

2.2 Binary Classification and Bayes Error at Sample Level

In theory, the binary classification problem with cost-sensitive errors is completely solved in Section 2.1. In practice, however, there are issues that can make the Bayes classifier (2) unfeasible:

- Typically, the joint probability distribution of (X, Y) is not exactly known but has to be inferred from a finite sample (called 'training set') $(x_{1,tr}, y_{1,tr}), \ldots, (x_{m,tr}, y_{m,tr}) \in \mathcal{X} \times \{0, 1\}$. If $\mathcal{X}$ is high-dimensional this may be difficult and require a large sample size to reduce errors due to random variation.

- As the Bayes classifier is explicitly given by (2), one can try to avoid estimating the entire distribution of (X, Y) and, instead, only estimate the feature-conditional probability $\mathrm{P}[Y = 0 \,|\, X]$ (also called posterior probability). However, this task is not significantly simpler than estimating the distribution of (X, Y), as illustrated by the fact that methods for the estimation of non-elementary conditional probabilities constitute a major branch of applied statistics.

By structural assumptions on the nature of the classification problem (1b), the approach based on direct estimation of the feature-conditional probability can be rendered more accessible. Logistic regression provides the possibly most important example for this approach (see, e.g., Cramer, 2003). But the price of the underlying assumptions may be high and include significant deterioration of goodness of fit. That is why alternative approaches based on direct implementation of the optimisation of the right-hand side of (1b) are popular. Lessmann et al. (2015) give a survey of the variety of methods available, just for application to credit scoring.

As mentioned in the introduction, one of the topics of this paper is an investigation into the question of whether certain quantifiers are Fisher consistent. In Section 4 below, the demonstration of lack of Fisher consistency is based on counter-examples which are non-trivial but simple enough to allow for the exact computation of the feature-conditional class probabilities and hence also the Bayes classifiers.

2.3 Data Set Shift

Even if one has succeeded in estimating a Bayes classifier or at least a reasonably good approximate Bayes classifier, issues may arise that spoil its effective deployment. Quite often, it is assumed that any instance with known features vector x but unknown class y that is presented for classification has been drawn at random from the same population as the training set. However, for a number of reasons this assumption can be wrong (see, e.g., Quiñonero-Candela et al., 2009; Kull and Flach, 2014; Moreno-Torres et al., 2012; Dal Pozzolo et al., 2015). A lot of research has been undertaken and is ongoing on methods to deal with this problem of so-called data set shift. In this paper, we consider the following variant of the problem:

- There is a training data set $(x_{1,tr}, y_{1,tr}), \ldots, (x_{m,tr}, y_{m,tr}) \in \mathcal{X} \times \{0, 1\}$ which is assumed to be an independent and identically distributed sample from the population distribution $\mathrm{P}(X, Y)$ of the random vector (X, Y) as described in Section 2.1.

- There is another data set, the test data set, $(x_{1,te}, y_{1,te}), \ldots, (x_{n,te}, y_{n,te}) \in \mathcal{X} \times \{0, 1\}$ which is assumed to be an independent and identically distributed sample from a possibly different population distribution $\mathrm{Q}(X, Y)$. Moreover, training and test data have been independently generated.

- For the instances in the training data set, their class labels are visible and can be made use of for learning classifiers (i.e. solving optimisation problem (1b)).

- For the instances in the test data set, their class labels are invisible to us or become visible only with large delay.

This assumption is intended to describe a setting where the test instances arrive batch-wise, not as a stream. We also assume that the sizes m of the training set and n of the test set are reasonably large such that trying to infer properties of the respective population distributions makes sense.

As far as the theory for this paper is concerned, we will ignore the issues caused by the fact that we know the training data set distribution $\mathrm{P}(X, Y)$ and the test data set

distribution $\mathrm{Q}(X, Y)$ only by inference from finite samples. Instead we will assume that we can directly deal with the population distributions $\mathrm{P}(X, Y)$ and $\mathrm{Q}(X, Y)$. Throughout the whole paper, we make the assumption that there are both positive and negative instances in both of the populations, i.e. it holds that

$$0 < \mathrm{P}[Y = 0] < 1 \qquad \text{and} \qquad 0 < \mathrm{Q}[Y = 0] < 1. \tag{3}$$

The problem that $\mathrm{P}(X, Y)$ and $\mathrm{Q}(X, Y)$ may not be the same is treated under different names in the literature (Moreno-Torres et al., 2012): data set shift, domain adaptation, population drift and others. There are several facets of the problem:

- Classifiers may have to be adapted to the test set or re-trained.
- The feature-conditional probabilities may have to be adapted to the test set or re-estimated.
- The unconditional class probabilities (also called prior probabilities or prevalences) may have to be re-adjusted for the test set or re-estimated.

Quantification. In this paper, we focus on the estimation of the prevalences $\mathrm{Q}[Y = 0]$ and $\mathrm{Q}[Y = 1]$ in the test set, as parameters of distribution $\mathrm{Q}(X, Y)$. This problem is called quantification (Forman, 2008) and of its own interest beyond its auxiliary function for classification and estimation of the feature-conditional probabilities (González et al., 2016).

2.4 Quantification in the Presence of Prior Probability Shift

In technical terms, the quantification problem as presented in Section 2.3 can be described as follows:

- We know the joint distribution of features and class labels (X, Y) under the training set probability distribution P.
- We know the distribution of the features X under the test set probability distribution Q.
- How can we infer the prevalences of the classes under Q, i.e. the probabilities $\mathrm{Q}[Y = 0]$ and $\mathrm{Q}[Y = 1] = 1 - \mathrm{Q}[Y = 0]$, by making best possible use of our knowledge of $\mathrm{P}(X, Y)$ and $\mathrm{Q}(X)$?

This question cannot be answered without assuming that the test set probability distribution Q shares some properties with the training set probability distribution P. This means to make more specific assumptions about the structure of the data set shift between training set and test set. In the literature, a variety of different types of data set shift have been discussed. See Moreno-Torres et al. (2012), Kull and Flach (2014) or Hofer (2015) for a number of examples, including 'covariate shift' and 'prior probability shift' which possibly are the two most studied types of data set shift. This paper focusses on prior probability shift.

2.4.1 Prior Probability Shift

This type of data set shift also has been called 'global shift' (Hofer and Krempl, 2013). The assumption of prior probability shift is, in particular, appropriate for circumstances where the features of an instance are caused by the instance's class membership (Fawcett and Flach, 2005). Technically, prior probability shift can be described as 'the class-conditional feature distributions of the training and test sets are the same', i.e.

$$\mathrm{Q}[X \in S \,|\, Y = 0] = \mathrm{P}[X \in S \,|\, Y = 0] \quad \text{and} \quad \mathrm{Q}[X \in S \,|\, Y = 1] = \mathrm{P}[X \in S \,|\, Y = 1], \tag{4}$$

for all measurable sets $S \subset \mathcal{X}$. Note that (4) does *not* imply $\mathrm{Q}[X \in S] = \mathrm{P}[X \in S]$ for all measurable $S \subset \mathcal{X}$ because the training set class distribution $\mathrm{P}(Y)$ and the test set class distribution $\mathrm{Q}(Y)$ still can be different.

In this paper, we will revisit three approaches to the estimation of the prevalences in population Q in the presence of prior probability shift as defined by (4). Specifically, we will check both in theory and by simple examples if these three approaches satisfy the basic estimation quality criterion of Fisher consistency. For this purpose, a slight generalisation of prior probability shift called 'invariant density ratio'-type data set shift will prove useful. Before we introduce it, let us briefly recall some facts on conditional probabilities and probability densities.

2.4.2 Feature-Conditional Class Probabilities and Class-Conditional Feature Densities

Typically, the class-conditional feature distributions $\mathrm{P}(X \,|\, Y = i)$, $i = 0, 1$, of a data set have got densities f_0 and f_1 respectively with respect to some reference measure like the d-dimensional Lebesgue measure. Then also the unconditional feature distribution $\mathrm{P}(X)$ has a density f which can be represented as

$$f(x) \;=\; \mathrm{P}[Y = 0]\, f_0(x) + (1 - \mathrm{P}[Y = 0])\, f_1(x) \quad \text{for all } x \in \mathcal{X}. \tag{5a}$$

Moreover, the feature-conditional class probability $\mathrm{P}[Y = 0 \,|\, X]$ can be expressed in terms of the densities f_0 and f_1:

$$\mathrm{P}[Y = 0 \,|\, X](x) \;=\; \frac{\mathrm{P}[Y = 0]\, f_0(x)}{\mathrm{P}[Y = 0]\, f_0(x) + (1 - \mathrm{P}[Y = 0])\, f_1(x)}, \quad x \in \mathcal{X}. \tag{5b}$$

Conversely, assume that there is a density f of the unconditional feature distribution $\mathrm{P}(X)$ and the feature-conditional class probability $\mathrm{P}[Y = 0 \,|\, X]$ is known. Then the class-conditional feature densities f_0 and f_1 are determined as follows:

$$\begin{aligned} f_0(x) &= \frac{\mathrm{P}[Y = 0 \,|\, X](x)}{\mathrm{P}[Y = 0]}\, f(x), \quad x \in \mathcal{X}, \\ f_1(x) &= \frac{1 - \mathrm{P}[Y = 0 \,|\, X](x)}{1 - \mathrm{P}[Y = 0]}\, f(x), \quad x \in \mathcal{X}. \end{aligned} \tag{6}$$

2.4.3 Invariant Density Ratio

Assume that there are densities f_0 and f_1 for the class-conditional feature distributions $\mathrm{P}(X \,|\, Y = i)$, $i = 0, 1$, of the training set population. We then say that the data set shift

from population P to Q is of 'invariant density ratio'-type if there are also densities h_0 and h_1 respectively for the class-conditional distributions $\mathrm{Q}(X \mid Y = i)$, $i = 0, 1$, of the test set, and it holds that

$$\frac{f_0(x)}{f_1(x)} \;=\; \frac{h_0(x)}{h_1(x)} \quad \text{for all } x \in \mathcal{X}. \tag{7}$$

Note that by (6), the density ratio $\frac{f_0}{f_1}$ can be rewritten as

$$\frac{f_0(x)}{f_1(x)} \;=\; \frac{\mathrm{P}[Y = 0 \mid X](x)}{1 - \mathrm{P}[Y = 0 \mid X](x)} \, \frac{1 - \mathrm{P}[Y = 0]}{\mathrm{P}[Y = 0]}, \quad x \in \mathcal{X}, \tag{8}$$

as long as $\mathrm{P}[Y = 0 \mid X](x) < 1$. Hence the density ratio can be calculated without knowledge of the values of the densities if the feature-conditional class probabilities or reasonable approximations are known.

Obviously, 'invariant density ratio' is implied by prior probability shift if all involved class-conditional distributions have got densities. 'Invariant density ratio'-type data set shift was discussed in some detail by Tasche (2014). It is an interesting type of data set shift for several reasons:

1) 'Invariant density ratio' extends the concept of prior probability shift.

2) Conceptually, 'invariant density ratio' is similiar to covariate shift. To see this recall first that covariate shift is defined by the property that the feature-conditional class probabilities of the training and test sets are the same (Moreno-Torres et al., 2012, Section 4.1):

$$\mathrm{P}[Y = 0 \mid X] \;=\; \mathrm{Q}[Y = 0 \mid X]. \tag{9a}$$

   Then, if there are densities of the class-conditional feature distributions under P and Q like for (7), (9a) can be rewritten as

$$\frac{\mathrm{P}[Y = 0]}{1 - \mathrm{P}[Y = 0]} \, \frac{f_0(x)}{f_1(x)} \;=\; \frac{\mathrm{Q}[Y = 0]}{1 - \mathrm{Q}[Y = 0]} \, \frac{h_0(x)}{h_1(x)} \quad \text{for all } x \in \mathcal{X}. \tag{9b}$$

   Hence, covariate shift also can be described in terms of the ratio of the class-conditional densities, like 'invariant density ratio'. The two types of data set shift coincide if $\mathrm{P}[Y = 0] = \mathrm{Q}[Y = 0]$, i.e. if the class prevalences in the training and test sets are the same.

3) The maximum likelihood estimates of class prevalences under prior probability shift actually are maximum likelihood estimates under 'invariant density ratio', too (Tasche, 2014, p. 152 and Remark 1). Hence they can be calculated with the EM (expectation-maximisation) algorithm as described by Saerens et al. (2001).

4) In terms of the data set shift taxonomy of Moreno-Torres et al. (2012), 'invariant density ratio' is a non-trivial but manageable 'other' shift, i.e. it is neither a prior-probability shift, nor a covariate shift, nor a concept shift.

By properties 1) and 2), 'invariant density ratio' has got conceptual similarities with both prior probability shift and covariate shift. In Section 4 below, we will make use of its manageability property 4) to construct an example of data set shift that reveals the limitations of some common approaches to the estimation of class prevalences.

2.5 Fisher Consistency

In Section 1, Fisher consistency of an estimator has been described as the property that the true value of a parameter related to a probability distribution is recovered when the estimator is applied to the distribution on the whole population. In this section, we apply the notion of Fisher consistency to the quantification of binary class prevalences.

In practice, estimators are often Fisher consistent and asymptotically consistent (weakly or strongly, see Section 10.4 of van der Vaart, 1998, for the definitions) at the same time. Gerow (1989) discusses the questions of when this is the case and what concept Fisher (1922) originally defined. Our focus on Fisher consistency is not meant to imply that asymptotic consistency is a less important property. In the context of statistical learning, asymptotic consistency seems to have enjoyed quite a lot of attention, as shown for instance by the existence of books like Devroye et al. (1996). In addition, the convergence aspect of asymptotic consistency often can be checked empirically by observing the behaviour of large samples. However, in some cases it might be unclear if the limit of a seemingly asymptotically consistent large sample is actually the right one. This is why, thanks to its similarity to the concept of unbiasedness which also refers to getting the right value of a parameter, Fisher consistency becomes an important property.

Definition 1 (Fisher consistency) *In the data set shift setting of this paper as described in Section 2.3, we say that an estimator $T(\mathrm{Q})$, applied to the elements Q of a family $\mathcal{Q}$ of possible population distributions of the test set, is Fisher consistent in $\mathcal{Q}$ for the prevalence of class 0 if it holds that*

$$T(\mathrm{Q}) \;=\; \mathrm{Q}[Y = 0] \quad \textit{for all } \mathrm{Q} \in \mathcal{Q}. \tag{10}$$

This definition of Fisher consistency is more restrictive than the definition by Cox and Hinkley (1974) quoted in Section 1. For it requires the specification of a family of distributions to which the parameter 'recovery' property applies. The family $\mathcal{Q}$ of most interest for the purpose of this paper is the set of distributions Q that are related to one fixed training set distribution P by prior probability shift, i.e. by (4).

3. Three Approaches to Estimating Class Prevalences under Prior Probability Shift

In this section, we study three approaches to the estimation of binary class prevalences:

- Adjusted Count (AC) (Forman, 2008), called 'confusion matrix approach' by Saerens et al. (2001), but in use since long before (Gart and Buck, 1966),
- the EM (expectation maximisation) algorithm by Saerens et al. (2001), described before as maximum likelihood approach by Peters and Coberly (1976), and
- CDE-Iterate (CDE for class distribution estimation) by Xue and Weiss (2009).

A variety of other approaches have been and are being studied in the literature (see the discussion in Hofer, 2015, for a recent overview). The selection of approaches to be discussed in this paper was driven by findings of Xue and Weiss (2009) and more recently Karpov

et al. (2016). According to that research, for the estimation of binary class prevalences the CDE-Iterate approach seems to perform equally well or even stronger than AC which by itself was found to outperform the popular EM-algorithm. In this section, we recall the technical details of the approaches which are needed to implement the numerical examples of Section 4 below.

In addition, we check the three estimators on a theoretical basis for Fisher consistency. In particular with regard to CDE-Iterate, the theory is inconclusive with regard to its possible Fisher consistency. However, the example of Section 4.2 below shows that CDE-Iterate is not Fisher consistent for class 0 prevalence under prior probability shift, in contrast to both AC and the EM-algorithm.

3.1 Adjusted Count (AC)

Let $g : \mathcal{X} \to \{0, 1\}$ be any classifier. Under prior probability shift as described by (4), we then obtain

$$\begin{aligned} \mathrm{Q}[g(X) = 0] &= \mathrm{Q}[Y = 0]\,\mathrm{Q}[g(X) = 0 \,|\, Y = 0] + (1 - \mathrm{Q}[Y = 0])\,\mathrm{Q}[g(X) = 0 \,|\, Y = 1] \\ &= \mathrm{Q}[Y = 0]\,\mathrm{P}[g(X) = 0 \,|\, Y = 0] + (1 - \mathrm{Q}[Y = 0])\,\mathrm{P}[g(X) = 0 \,|\, Y = 1]. \quad (11) \end{aligned}$$

If $\mathrm{P}[g(X) = 0 \,|\, Y = 0] \neq \mathrm{P}[g(X) = 0 \,|\, Y = 1]$, i.e. if $g(X)$ and Y are not stochastically independent, (11) is equivalent to

$$\mathrm{Q}[Y = 0] \;=\; \frac{\mathrm{Q}[g(X) = 0] - \mathrm{P}[g(X) = 0 \,|\, Y = 1]}{\mathrm{P}[g(X) = 0 \,|\, Y = 0] - \mathrm{P}[g(X) = 0 \,|\, Y = 1]}. \quad (12)$$

Equation (12) is called the AC approach to the quantification of binary class prevalences. Let us recall some useful facts about AC:

- Equation (12) has been around for a long time, at least since the 1960s. In this paper, the quantification approach related to (12) is called 'Adjusted Count' as in Forman (2008) because this term nicely describes what is done.
- $\mathrm{Q}[g(X) = 0]$ is the proportion of instances in the test set (measured by counting) that are classified (predicted) positive by classifier $g(X)$.
- $\mathrm{P}[g(X) = 0 \,|\, Y = 1]$ is the 'false positive rate', as measured for classifier $g(X)$ on the training set.
- $\mathrm{P}[g(X) = 0 \,|\, Y = 0]$ is the 'true positive rate', as measured for classifier $g(X)$ on the training set.
- Forman (2008) discusses AC in detail and provides a number of variations of the theme in order to account for its deficiencies.
- Possibly, the main issue with AC is that in practice the result of the right-hand side of (12) can turn out to be negative or greater than 1. This can happen for one or more of the following reasons:
  1. The data set shift in question actually is no prior probability shift, i.e. (4) does not hold.
  2. The estimates of the true positive and true negative rates are inaccurate.
  3. The estimation of the scoring function underlying the classifier from limited training data may be inaccurate (both biased and subject to high variance).
- In theory, if the data set shift is indeed a prior probability shift, the result of the right-hand side of (12) should be the same, regardless of which admissible (i.e. such that the denominator is not zero) classifier is deployed for determining the proportion of instances in the test set classified positive. Hence, whenever in practice different classifiers give significantly different results, that could suggest that the assumption of prior probability shift is wrong.
- As long as $g(X)$ and Y are at least somewhat dependent, possible lack of power of the classifier g should not be an issue for the applicability of (12) because the denominator on the right-hand side of (12) is then different from zero.

For a training set distribution P denote by $\mathcal{Q}_{\mathrm{prior}} = \mathcal{Q}_{\mathrm{prior}}(\mathrm{P})$ the family of distributions Q that are related to P by prior probability shift in the sense of Definition 1, i.e.

$$\mathcal{Q}_{\mathrm{prior}} \;=\; \{\mathrm{Q} : \mathrm{Q} \text{ is probability measure satisfying (4)}\}. \quad (13)$$

Then, for fixed training set distribution P and fixed classifier $g(X)$ such that $g(X)$ and Y are not independent under P, the AC approach is Fisher consistent in $\mathcal{Q}_{\mathrm{prior}}$ for the prevalence of class 0 by construction: Define the operator $T = T_{g,\mathrm{P}}$ by

$$T(\mathrm{Q}) \;=\; \frac{\mathrm{Q}[g(X) = 0] - \mathrm{P}[g(X) = 0 \,|\, Y = 1]}{\mathrm{P}[g(X) = 0 \,|\, Y = 0] - \mathrm{P}[g(X) = 0 \,|\, Y = 1]}.$$

Then (12) implies (10) for $\mathrm{Q} \in \mathcal{Q}_{\mathrm{prior}}$. However, denote—again for some fixed training set distribution P—by $\mathcal{Q}_{\mathrm{invariant}} = \mathcal{Q}_{\mathrm{invariant}}(\mathrm{P})$ the family of distributions Q that are related to P by 'invariant density ratio'-type data set shift in the sense of (7), i.e.

$$\mathcal{Q}_{\mathrm{invariant}} \;=\; \{\mathrm{Q} : \mathrm{Q} \text{ is probability measure satisfying (7)}\}. \quad (14)$$

Then in general the AC approach is not Fisher consistent in $\mathcal{Q}_{\mathrm{invariant}}$. For it is shown in Section 4.3 below that there are a classifier g^*, a distribution P^* and a related distribution $\mathrm{Q}^* \in \mathcal{Q}_{\mathrm{invariant}}(\mathrm{P}^*)$ such that

$$T_{g^*,\mathrm{P}^*}(\mathrm{Q}^*) \neq \mathrm{Q}^*[Y = 0].$$

3.2 EM-Algorithm

Saerens et al. (2001) made the EM-algorithm popular for the estimation of class prevalences, as a necessary step for the re-adjustment of thresholds of soft classifiers. A closer inspection of the article by Peters and Coberly (1976) shows that they actually had studied the same algorithm and provided conditions for its convergence. In particular, this observation again draws attention to the fact that the EM-algorithm, deployed on data samples, should result in unique maximum likelihood estimates of the class prevalences.

As noticed by Du Plessis and Sugiyama (2014), the population level equivalent of sample level maximum likelihood estimation of the class prevalences under an assumption of prior

probability shift is minimisation of the Kullback-Leibler distance between the estimated test set feature distribution and the observed test set feature distribution. Moreover, Tasche (2014) observed that the EM-algorithm finds the true values of the class prevalences not only under prior probability shift but also under 'invariant density ratio'-type data set shift. In other words, the EM-algorithm is Fisher consistent both in $\mathcal{Q}_{\text{prior}}$ and $\mathcal{Q}_{\text{invariant}}$, as defined in (13) and (14) respectively, for the prevalence of class 0. See Proposition 2 below for a formal proof.

In the case of two classes, the maximum-likelihood version of the EM-algorithm in the sense of solving the likelihood equation is more efficient than the EM-algorithm itself. This statement applies even more to the population level calculations. In this paper, therefore, we describe the result of the EM-algorithm as the unique solution of a specific equation, as presented in Tasche (2014).

3.3 Calculating the Result of the EM-Algorithm.

In the population setting of Section 2.3 with training set distribution $\mathrm{P}(X, Y)$ and test set distribution $\mathrm{Q}(X, Y)$, assume that the class-conditional feature distributions $\mathrm{P}(X \,|\, Y = i)$, $i = 0, 1$, of the training set have got densities f_0 and f_1. Define the density ratio R by

$$R(x) \;=\; \frac{f_0(x)}{f_1(x)}, \quad \text{for } x \in \mathcal{X}. \tag{15}$$

We then define the estimation operator $T_R(\mathrm{Q})$ for the prevalence of class 0 as the unique solution $q \in (0, 1)$ of the equation (Tasche, 2014)

$$0 \;=\; \mathrm{E}_{\mathrm{Q}}\left[\frac{R(X) - 1}{1 + q\,(R(X) - 1)}\right], \tag{16a}$$

where E_{Q} denotes the expectation operator with respect to Q. Unfortunately, not always does a solution of (16a) exist in $(0, 1)$. There exists a solution in $(0, 1)$ if and only if

$$\mathrm{E}_{\mathrm{Q}}[R(X)] > 1 \quad \text{and} \quad \mathrm{E}_{\mathrm{Q}}\left[\frac{1}{R(X)}\right] > 1, \tag{16b}$$

and if there is a solution in $(0, 1)$ it is unique (Tasche, 2014, Remark 2(a)).

Proposition 2 *The operator $T_R(\mathrm{Q})$ (EM-algorithm), as defined by* (16a)*, is Fisher consistent in $\mathcal{Q}_{prior}$ and $\mathcal{Q}_{invariant}$ for the prevalence of class 0.*

Proof We only have to prove the claim for $\mathcal{Q}_{\text{invariant}}$ because $\mathcal{Q}_{\text{prior}}$ is a subset of $\mathcal{Q}_{\text{invariant}}$. Let any $\mathrm{Q} \in \mathcal{Q}_{\text{invariant}}$ be given and denote by h_i, $i = 0, 1$, its class-conditional feature densities. By (7) and (5b), it then follows that

$$\begin{aligned}
\mathrm{E}_{\mathrm{Q}}\left[\frac{R(X) - 1}{1 + \mathrm{Q}[Y = 0]\,(R(X) - 1)}\right] &= \mathrm{E}_{\mathrm{Q}}\left[\frac{h_0 - h_1}{h_1 + \mathrm{Q}[Y = 0]\,(h_0 - h_1)}\right] \\
&= \frac{\mathrm{E}_{\mathrm{Q}}\big[\mathrm{Q}[Y = 0 \,|\, X]\big]}{\mathrm{Q}[Y = 0]} - \frac{\mathrm{E}_{\mathrm{Q}}\big[\mathrm{Q}[Y = 1 \,|\, X]\big]}{\mathrm{Q}[Y = 1]} \\
&= 1 - 1 = 0.
\end{aligned}$$

Hence, the prevalence $\mathrm{Q}[Y = 0]$ of class 0 is a solution of (16a) and, therefore, the only solution. As a consequence, $T_R(\mathrm{Q})$ is well-defined and satisfies $T_R(\mathrm{Q}) = \mathrm{Q}[Y = 0]$. ■

In practice, for fixed q the right-hand side of (16a) could be estimated on a test set sample $(x_{1,te}, y_{1,te}), \ldots, (x_{n,te}, y_{n,te})$ as in Section 2.3 by the sample average

$$\frac{1}{n}\sum_{i=1}^{n} \frac{R(x_{i,te}) - 1}{1 + q\,(R(x_{i,te}) - 1)},$$

where R could be plugged in as a training set estimate of the density ratio by means of (8).

3.4 CDE-Iterate

In order to successfully apply the AC quantification approach as described in Section 3.1, we must get hold of reliable estimates of the training set true and false positive rates of the classifier deployed. If the positive class is the minority class, the estimation of the true positive rate can be subject to large uncertainties and hence may be hard to achieve with satisfactory accuracy. Similarly, if the negative class is the minority class, the estimation of the false positive rate can be rather difficult.

Application of the EM-algorithm as introduced by Saerens et al. (2001) or described in Section 3.2 requires reliable estimation of the feature-conditional class probabilities or the density ratio. Again, such estimates in general are hard to achieve. That is why alternative methods for quantification are always welcome. In particular, methods that are based exclusively on learning one or more crisp classifiers are promising. For learning classifiers is a well-investigated problem for which efficient solution approaches are available (see, for instance, Lessmann et al., 2015, for a survey related to credit scoring).

Xue and Weiss (2009) proposed 'CDE-Iterate' (CDE for class distribution estimation) which is appropriately summarised by Karpov et al. (2016) as follows: "The main idea of this method is to retrain a classifier at each iteration, where the iterations progressively improve the quantification accuracy of performing the 'classify and count' method via the generated cost-sensitive classifiers." Xue and Weiss motivated the CDE-Iterate algorithm as kind of an equivalent of the EM-algorithm, with the training set feature-conditional class probabilities replaced by Bayes classifiers (or approximations of the Bayes classifiers) learnt on the training set.

In this paper, we do not retrain a classifier but make use of the fact that we have got a closed-form representation of the optimal classifier resulting from the retraining, by (2). Taking this into account and using notation adapted for this paper, we obtain the following description of the CDE-Iterate procedure:

CDE-Iterate algorithm

1) Set initial parameters: $k = 0$, $c_0^{(0)} = 1$, $c_1^{(0)} = 1$.

2) Find Bayes classifier under training distribution $\mathrm{P}(X, Y)$:

$$g_k(X) \;=\; \begin{cases} 0, & \text{if } \mathrm{P}[Y = 0 \,|\, X] > \frac{c_1^{(k)}}{c_0^{(k)} + c_1^{(k)}}, \\ 1, & \text{if } \mathrm{P}[Y = 0 \,|\, X] \le \frac{c_1^{(k)}}{c_0^{(k)} + c_1^{(k)}}. \end{cases}$$

3) Under test feature distribution $\mathrm{Q}(X)$ compute $q_k = \mathrm{Q}[g_k(X) = 0]$.

4) Increment k by 1.

5) Reset cost parameters: $c_1^{(k)} = \frac{1-q_{k-1}}{1-\mathrm{P}[Y=0]}$, $c_0^{(k)} = \frac{q_{k-1}}{\mathrm{P}[Y=0]}$.

6) If convergence is reached or $k = k_{\max}$ then stop, and accept q_{k-1} as the CDE-Iterate estimate of $\mathrm{Q}[Y = 0]$. Else continue with step 2.

Xue and Weiss (2009) did not provide a proof of convergence or unbiasedness for CDE-Iterate. In Section 6 of their paper, they state that "one improvement would be to adapt the CDE-Iterative method to automatically terminate once the class distribution estimate converges. This might improve overall performance over any specific CDE-Iterate-n method and would eliminate the problem of identifying the appropriate number of iterations. It is possible that such a CDE-converge method would outperform CDE-AC." In this paper, we prove convergence of CDE-Iterate and also show that it does not outperform 'CDE-AC' (Adjusted Count in the notation of this paper) for class distribution estimation under prior probability shift, thus answering the quoted research questions of Xue and Weiss.

The proof of the convergence of CDE-Iterate as described above is provided in Proposition 6 in Appendix B below. There it is also shown that the limit $q^* = \lim_{k\to\infty} q_k$ solves the following equation[1]:

$$q^* \;=\; \begin{cases} \mathrm{Q}\left[R(X) \geq \frac{1-q^*}{q^*}\right], & \text{if } q_0 \geq q_1 \text{ and } q_k > q^* \text{ for all } k, \\ \mathrm{Q}\left[R(X) > \frac{1-q^*}{q^*}\right], & \text{otherwise,} \end{cases} \tag{17}$$

where $R(x)$ is defined by (15).

The limit result (17) is quite general in so far as it is not based on any assumption with regard to the type of data set shift between the training set distribution $\mathrm{P}(X, Y)$ and the test set distribution $\mathrm{Q}(X, Y)$. If we restrict the type of data set shift to prior probability shift or 'invariant density ratio'-type data set shift as defined in Section 2.4, does (17) then imply Fisher consistency for the prevalence of class 0 in either of these two families of distributions? As we show by example in the following section, the answer to this question is 'no'.

4. Numerical Examples

In Section 3, we have found that AC as an estimator of class prevalences is Fisher consistent for prior probability shift while the EM-algorithm is even Fisher consistent for the more general 'invariant density ratio'-type data set shift. We have not yet answered the question if CDE-Iterate is Fisher consistent for either of these two data set shift types.

The property of an estimator to be Fisher consistent is something that has to be proved. In contrast, lack of Fisher consistency of an estimator is conveniently shown by providing a counter-example. This is the purpose of the following subsections: We show by examples that

1. Subject to the technical condition that $\mathrm{P}(X)$ has a density f such that $\mathrm{Q}[f(X) > 0] = 1$.

- CDE-Iterate is not Fisher consistent for prior probability shift (and hence for 'invariant density ratio'-type data set shift neither),
- AC is not Fisher consistent for 'invariant density ratio'-type data set shift, and
- the EM-algorithm is no longer Fisher consistent if the 'invariant density ratio'-type data set shift is slightly modified.

We present the counter-examples as a simulation and estimation experiment that is executed for each of the three following example models:

- Section 4.2: Binormal model with equal variances for both training and test set (prior probability shift).
- Section 4.3: Binormal model with equal variances for training set and model with non-normal class-conditional densities but identical density ratio for test set ('invariant density ratio'-type data set shift).
- Section 4.4: Binormal model with equal variances for training set and model with non-normal class-conditional densities and a different density ratio for test set (neither prior probability shift nor 'invariant density ratio'-type data set shift).

The experimental design is described in Section 4.1 below. The classical binormal model with equal variances has been selected as the training set model for the following reasons:

- Logistic regression finds the correct feature-conditional class probabilities.
- Needed algorithms are available in common software packages like R.
- The ratio of the class-conditional feature densities has a particularly simple shape, see (21b) below.
- The binormal model with equal variances has been found useful before for a similar experiment (Tasche, 2016).

Thresholds for Bayes classifier under data set shift. We deploy the logistic regression as coded by R Core Team (2014). Therefore, it is convenient to always use the feature-conditional class-probability $\mathrm{P}[Y = 0 \,|\, X]$ as the Bayes classifier, both for the training set and for the test set after a possible data set shift (then with a modified threshold). In order to be able to do so, we observe that, under prior probability shift or even 'invariant density ratio'-type data set shift, the test set Bayes classifier $g_{\mathrm{test}}(X)$ for the cost-sensitive error criterion (1b) can be represented both as

$$g_{\mathrm{test}}(X) \overset{(2)}{=} \begin{cases} 0, & \text{if } \mathrm{Q}[Y = 0 \,|\, X] > \frac{c_1}{c_0+c_1}, \\ 1, & \text{if } \mathrm{Q}[Y = 0 \,|\, X] \leq \frac{c_1}{c_0+c_1}, \end{cases}$$

and as (see Saerens et al., 2001, Section 2.2)

$$g_{\text{test}}(X) \;=\; \begin{cases} 0, & \text{if } \mathrm{P}[Y=0\,|\,X] > \dfrac{c_1 \frac{1-\mathrm{Q}[Y=0]}{1-\mathrm{P}[Y=0]}}{c_1 \frac{1-\mathrm{Q}[Y=0]}{1-\mathrm{P}[Y=0]} + c_0 \frac{\mathrm{Q}[Y=0]}{\mathrm{P}[Y=0]}}, \\ 1, & \text{if } \mathrm{P}[Y=0\,|\,X] \leq \dfrac{c_1 \frac{1-\mathrm{Q}[Y=0]}{1-\mathrm{P}[Y=0]}}{c_1 \frac{1-\mathrm{Q}[Y=0]}{1-\mathrm{P}[Y=0]} + c_0 \frac{\mathrm{Q}[Y=0]}{\mathrm{P}[Y=0]}}. \end{cases} \tag{18}$$

4.1 Design of the Experiment

In the subsequent sections 4.2, 4.3 and 4.4, we conduct the following experiment and report its results[2]:

- For a given training set (sample and population) represented by distribution $\mathrm{P}(X,Y)$, we determine the Bayes classifier that is optimal for minimising the Bayes error (1a). We represent the Bayes classifier by a decision threshold applied to the feature-conditional probability of class 0 as in (2), with $c_0 = 1 = c_1$.
- We create test sets (by simulation or as population distribution), represented by distributions $\mathrm{Q}(X,Y)$, which are related to the training set by certain types of data set shift, including prior probability shift and 'invariant density ratio'-type data set shift.
- On the test sets, we deploy three different quantification methods for the estimation of the prevalence of class 0 (see also Definition 3 below): CDE-Iterate (defined in Section 3.4), Adjusted Count (defined in Section 3.1), and EM-algorithm (defined in Section 3.2).
- Based on the estimated class 0 prevalences, we adapt the threshold of the Bayes classifier according to (18) such that it would be optimal for minimising the Bayes error (equivalently for maximising the classification accuracy) if the estimated prevalence were equal to the true test set class 0 prevalence and the test set were related to the training set by prior probability shift or by 'invariant density ratio'-type data set shift.
- We report the following results for samples and populations:
  - Classification accuracy and F-measure (see (23a) and (23b) for the definitions) of the adapted Bayes classifier when applied to the test set, because these measures were used by Xue and Weiss (2009).
  - Estimated prevalences of class 0, for direct comparison of estimation results and true values.
  - Relative error: If q is the true probability and $\tilde{q}$ the estimated probability, then we tabulate

$$\max\left(\frac{|\tilde{q}-q|}{q}, \frac{|1-\tilde{q}-(1-q)|}{1-q}\right) \;=\; \frac{|\tilde{q}-q|}{\min(q,1-q)}. \tag{19}$$

2. The R-scripts used for creating the tables and figures of this paper can be received upon request from the author.

Relative error behaves similar to Kullback-Leibler distance used by Karpov et al. (2016), but is defined also for $\tilde{q} = 0$ and $\tilde{q} = 1$ and, moreover, has a more intuitive interpretation.

Modelled class prevalences. The setting is broadly the same as for the artificial data set in Karpov et al. (2016). For each model, we consider a training set with class probabilities 50%, combined with test sets with class 0 probabilities 1%, 5%, 10%, 30%, 50%, 70%, 90%, 95% and 99%. For the samples as well as for the population distributions, we deploy the estimation approaches whose acronyms are given in the following definition to estimate the test set class 0 prevalences.

Definition 3 (Acronyms for estimation approaches)
The following estimation approaches are used in this section:

- *CDE-Iterate in three variants:*
  - *CDE1: First iteration of the algorithm described in Section 3.4. Identical with Classify & Count of Forman (2008).*
  - *CDE2: Second iteration of the algorithm described in Section 3.4.*
  - *CDE∞: CDE-Iterate converged, as described in Section 3.4.*
- *AC: Adjusted Count as described in Section 3.1.*
- *EM: EM-algorithm as described in Section 3.2.*

4.2 Training Set: Binormal; Test Set: Binormal

We consider the classical binormal model with equal variances as an example that fits well into the prior probability shift setting of Section 2.4. We specify the binormal model by defining the class-conditional feature distributions.

- *Training set:* Both class-conditional feature distributions are normal, with equal variances, i.e.

$$\mathrm{P}(X\,|\,Y=0) = \mathcal{N}(\mu,\sigma^2), \qquad \mathrm{P}(X\,|\,Y=1) = \mathcal{N}(\nu,\sigma^2), \tag{20a}$$

  with $\mu < \nu$ and $\sigma > 0$.
- *Test set:* Same as training set.
- For this section's numerical experiment, the following parameter values have been chosen:

$$\mu = 0, \qquad \nu = 2, \qquad \sigma = 1. \tag{20b}$$

For the sake of brevity, in the following we sometimes refer to the setting of this section as 'double' binormal. The feature-conditional class probability $\mathrm{P}[Y=0\,|\,X]$ in the training set is given by

$$\mathrm{P}[Y=0\,|\,X](x) \;=\; \frac{1}{1+\exp(a\,x+b)}, \quad x \in \mathbb{R}, \tag{21a}$$

with $a = \frac{\nu-\mu}{\sigma^2} > 0$ and $b = \frac{\mu^2-\nu^2}{2\,\sigma^2} + \log\left(\frac{1-\mathrm{P}[Y=0]}{\mathrm{P}[Y=0]}\right)$. For the density ratio R according to (15), we obtain

$$R(x) \;=\; \exp\left(x\,\tfrac{\mu-\nu}{\sigma^2} + \tfrac{\nu^2-\mu^2}{2\,\sigma^2}\right), \quad x \in \mathbb{R}. \tag{21b}$$

For the sample version of the example in this section, we create by Monte-Carlo simulation a training sample $\big((x_{1,tr}, y_{1,tr}), \ldots, (x_{m,tr}, y_{m,tr})\big) \in (\mathbb{R} \times \{0,1\})^m$ and test samples $\big((x_{1,te}, y_{1,te}),\, \ldots,\, (x_{n,te}, y_{n,te})\big) \in (\mathbb{R} \times \{0,1\})^n$ with class-conditional feature distributions given by (20a) that approximate the training set population distribution P and the test set population distributions Q as described in general terms in Section 2.4 and more specifically here by (20a), (20b) and the respective class 0 prevalences.

- In principle, $(x_{1,tr}, y_{1,tr}), \ldots, (x_{m,tr}, y_{m,tr})$ is an independent and identically distributed (iid) sample from P as specified by (20a), (20b) and 'training' class 0 prevalence $\mathrm{P}[Y=0] = 0.5$.

- In principle, $(x_{1,te}, y_{1,te}), \ldots,\, (x_{n,te}, y_{n,te})$ is an iid sample from Q as specified by (20a), (20b) and 'test' class 0 prevalences $\mathrm{Q}[Y=0] \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 0.99\}$.

- However, following the precedent of Xue and Weiss (2009), for both data sets we have used stratified sampling such that the proportion of $(x_{i,tr}, y_{i,tr})$ with $y_{i,tr} = 0$ in the training set is exactly $\mathrm{P}[Y=0]$, and the proportion of $(x_{i,te}, y_{i,te})$ with $y_{i,te} = 0$ in the test set is exactly $\mathrm{Q}[Y=0]$.

The sample sizes for both the training and the test set samples have been chosen to be 10,000, i.e.

$$m \;=\; n \;=\; 10,000. \tag{22}$$

In our experimental design, the sampling is conducted mainly for illustration purposes because at the same time we also calculate the results at population (i.e. sample size ∞) level such that we know the theoretical outcomes. Therefore, for each parametrisation of each model, there is no repeated sampling, i.e. only one sample is created.

Table 1 shows the class 0 prevalence estimates made in the double binormal setting of this section. Note that in the lower panel of the table, the population estimates by AC and EM are exact—as they should be since in Sections 3.1 and 3.2 we have proved that both estimators are Fisher consistent for the prevalence of class 0 in the family of prior probability shifted distributions. The numbers from the lower panel also show that, in general, neither of the three CDE-Iterate[3] variants CDE1, CDE2 and CDE∞ are Fisher consistent for class 0 prevalence under prior probability shift, except for the case of identical training and test set distributions.

As mentioned in Section 2.1, in the setting of the binormal model with equal variances, the Bayes classifier is unique, irrespective of its specific representation. Hence, in this example, there is no chance to work-around the lack of Fisher consistency for CDE-Iterate by trying to find alternative Bayes classifiers.

3. Note that in any case the tabulated estimates by CDE1, CDE2 and CDE∞ confirm the monotonicity statement in Proposition 6 of Appendix B for the convergence of CDE-Iterate.

Q[Y=0]	0.01	0.05	0.10	0.30	0.50	0.70	0.90	0.95	0.99
Prevalence estimates on samples									
CDE1	0.1637	0.1910	0.2288	0.3639	0.4989	0.6351	0.7720	0.8014	0.8356
CDE2	0.0402	0.0682	0.1127	0.2991	0.4986	0.7014	0.8851	0.9230	0.9619
CDE∞	0.0000	0.0000	0.0040	0.2416	0.4985	0.7653	0.9929	1.0000	1.0000
AC	-0.0010	0.0391	0.0947	0.2935	0.4921	0.6924	0.8938	0.9370	0.9873
EM	0.0070	0.0475	0.0968	0.2988	0.4967	0.6981	0.9007	0.9467	0.9890
Prevalence estimates on populations									
CDE1	0.1655	0.1928	0.2269	0.3635	0.5000	0.6365	0.7731	0.8072	0.8345
CDE2	0.0406	0.0715	0.1131	0.2994	0.5000	0.7006	0.8869	0.9285	0.9594
CDE∞	0.0000	0.0000	0.0121	0.2389	0.5000	0.7611	0.9879	1.0000	1.0000
AC	0.0100	0.0500	0.1000	0.3000	0.5000	0.7000	0.9000	0.9500	0.9900
EM	0.0100	0.0500	0.1000	0.3000	0.5000	0.7000	0.9000	0.9500	0.9900

Table 1: Class 0 prevalence estimates on the test sets. Training set: Binormal with equal variances. Test sets: Binormal with equal variances. 'Q[Y=0]' is the true test set prevalence of class 0. See Definition 3 for the other acronyms.

However, from the sample estimation numbers in the upper panel of Table 1, we can conclude that in practice CDE2 may provide estimates of the class 0 prevalence that are better or at least not much worse than the AC and EM estimates. Table 4 in Appendix A with the relative errors of the estimates confirms this observation. Such incidences could explain the favourable performance of CDE-Iterate observed by Xue and Weiss (2009) and Karpov et al. (2016).

For the sake of completeness, in Tables 5 and 6 in Appendix A, we report the classification accuracies and F-measures respectively, for the training set Bayes classifier with adapted thresholds according to (18), computed on the different test sets. The metrics classification accuracy and F-measure here are of interest because Xue and Weiss (2009) used them for measuring the performance of the classifiers discussed in their study. For any classifier $g : \mathcal{X} \to \{0,1\}$, we make use of the following population level formulae for classification accuracy and F-measure:

$$\text{Classification accuracy} \;=\; 1 - \text{Classification Error} \;=\; \mathrm{Q}[g(X) = Y], \tag{23a}$$

$$\begin{aligned}\text{F-measure} \;&=\; \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \\ &=\; \frac{2\,\mathrm{Q}[g(X)=0\,|\,Y=0]\,\mathrm{Q}[Y=0\,|\,g(X)=0]}{\mathrm{Q}[g(X)=0\,|\,Y=0] + \mathrm{Q}[Y=0\,|\,g(X)=0]}.\end{aligned} \tag{23b}$$

In these formulae, Q is used to indicate the test set probability distribution in accordance with the general assumption of this paper.

Basically, Tables 5 and 6 in Appendix A demonstrate that it is hard to draw conclusions on quantification performance by looking at classification accuracy or F-measure as

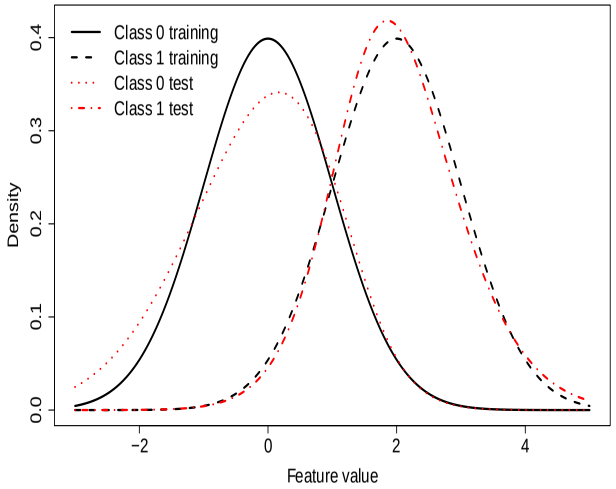


Figure 1: Training set and test set class-conditional densities for Section 4.3. The ratio of the training set densities and the ratio of the test set densities are equal.

performance metrics. Both at sample and at population level, the tabulated classification accuracy values of CDE∞, AC and EM are almost indistinguishable. While the slightly better accuracy values for EM-algorithm and AC compared to CDE∞ at population level are indeed sure evidence of better performance, the better values at sample level might just be random effects. A similar statement applies to the F-measure values of CDE2, AC and EM. The NaNs in Table 6 are caused by zero and negative estimates of the class 0 prevalence.

4.3 Training Set: Binormal; Test Set: Non-normal Densities, Binormal Density Ratio

The combination of a binormal model with equal variances for the training set and a model with the same density ratio but non-normal class-conditional feature distributions provides an example that fits conveniently into the 'invariant density ratio'-type data set shift setting of Section 2.4. We again specify the two models by their class-conditional feature densities.

Training set. Both class-conditional feature distributions are normal, with equal variances, as specified in (20a) and (20b).

Test set. We specify the test set distribution by class-conditional feature densities h_0 and h_1 chosen in such a way that their ratio $\frac{h_0(x)}{h_1(x)}$, $x \in \mathbb{R}$, is given by (21b). Then it equals the feature density ratio in the double binormal model from Section 4.2. In addition, we

require that the resulting model is easy to handle numerically but still not too close to the double binormal model. To achieve this, we apply the following steps:

- We start with a normal density h^* characterised by the parameters

$$\text{Mean} \;=\; \vartheta, \qquad \text{variance} \;=\; \tau^2 \;>\; 0. \tag{24a}$$

  For the purpose of this paper, we have chosen

$$\vartheta \;=\; 0.5, \qquad \tau \;=\; 1.4. \tag{24b}$$

- Then we apply Theorem 3 of Tasche (2014) to decompose h^* into a mixture $q^*\, h_0 + (1-q^*)\, h_1$ of h_0 and h_1, subject to the condition $\frac{h_0(x)}{h_1(x)} = \exp\left(x\,\frac{\mu-\nu}{\sigma^2} + \frac{\nu^2-\mu^2}{2\,\sigma^2}\right)$, $x \in \mathbb{R}$, with μ, ν and σ given by (20b). The important step for the decomposition is to determine q^*. This can be done as suggested in Section 3.2, by solving a version of Equation (16a) for the variable q:

$$0 \;=\; \int_{-\infty}^{\infty} \frac{R(x)-1}{1+q\,(R(x)-1)}\, h^*(x)\, dx, \tag{25a}$$

  where $R(x)$ is given by (21b). There is a unique solution $0 < q = q^* < 1$ if and only if

$$\int R(x)\, h^*(x)\, dx > 1 \quad \text{and} \quad \int R(x)^{-1}\, h^*(x)\, dx > 1.$$

  With parameters set as in (20b) and (24b), the solution for q is

$$q^* \;=\; 0.7239184.$$

- If q^* denotes the solution of (25a) the resulting class conditional feature densities are determined as

$$h_0(x) \;=\; \frac{R(x)\, h^*(x)}{1+q^*\,(R(x)-1)} \quad \text{and} \quad h_1(x) \;=\; \frac{h^*(x)}{1+q^*\,(R(x)-1)}, \qquad x \in \mathbb{R}. \tag{25b}$$

Once the densities h_0 and h_1 have been made available by (25b), the test set class-conditional feature distributions $\mathrm{Q}(X \,|\, Y = 0)$ and $\mathrm{Q}(X \,|\, Y = 1)$ can be defined as the distributions determined by these densities. Figure 1 shows the test set class-conditional densities from (25b) and the normal densities related to the training set.

For the population-related calculations and the sample simulations, we use the sample sizes as specified in (22), training set class 0 prevalence $\mathrm{P}[Y = 0] = 0.5$ and test set class 0 prevalences $\mathrm{Q}[Y = 0] \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 0.99\}$ as in the double binormal case of Section 4.2. Again, the samples are stratified with separate sampling from classes 0 and 1. This is straightforward for the binormal model of the training set, but less straightforward for the test set distributions given by the class-conditional densities (25b). For this sampling we have applied the simple accept-reject algorithm described in Robert and Casella (2004, Corollary 2.17).

Q[Y=0]	0.01	0.05	0.10	0.30	0.50	0.70	0.90	0.95	0.99
Prevalence estimates on samples									
CDE1	0.1548	0.1865	0.2207	0.3526	0.4931	0.6136	0.7508	0.7850	0.8161
CDE2	0.0313	0.0644	0.1014	0.2773	0.4901	0.6736	0.8728	0.9228	0.9542
CDE∞	0.0001	0.0010	0.0236	0.2084	0.4848	0.7462	0.9994	1.0000	1.0000
AC	-0.0141	0.0325	0.0828	0.2768	0.4835	0.6608	0.8626	0.9129	0.9587
EM	0.0074	0.0467	0.0957	0.2932	0.4994	0.6853	0.8990	0.9503	0.9913
Prevalence estimates on populations									
CDE1	0.1641	0.1907	0.2240	0.3572	0.4904	0.6236	0.7568	0.7901	0.8167
CDE2	0.0359	0.0653	0.1049	0.2855	0.4853	0.6890	0.8794	0.9217	0.9531
CDE∞	0.0000	0.0032	0.0264	0.2168	0.4794	0.7731	1.0000	1.0000	1.0000
AC	0.0080	0.0470	0.0958	0.2908	0.4859	0.6810	0.8761	0.9249	0.9639
EM	0.0100	0.0500	0.1000	0.3000	0.5000	0.7000	0.9000	0.9500	0.9900

Table 2: Class 0 prevalence estimates on the test sets. Training set: Binormal with equal variances. Test sets: Non-normal densities, binormal density ratio. 'Q[Y=0]' is the true test set prevalence of class 0. See Definition 3 for the other acronyms.

Table 2 shows the class 0 prevalence estimates made in the 'binormal – non-binormal with binormal density ratio' setting of this section. In the lower panel of the table, the population estimates by EM are exact—as they should be since in Section 3.2 we have proved that the EM-algorithm is Fisher consistent for the prevalence of class 0 in the family of test set distributions subject to 'invariant density ratio'-type data set shift. The numbers from the lower panel also show that, in general, neither AC nor any of the three CDE-Iterate variants CDE1, CDE2 and CDE∞ are Fisher consistent for class 0 prevalence under 'invariant density ratio'-type data set shift, not even in the case of training and test sets with equal class 0 prevalences. However, while the performance of CDE∞ and CDE1 is really poor, CDE2 at least is not worse than AC in this setting.

This observation is confirmed by looking at the relative error table 7 in Appendix A and the upper 'sample' panel of Table 2. Hence, outside of the prior probability setting, CDE2 may well outperform AC. The 'sample' error figures also demonstrate that while in theory the EM-algorithm should deliver unbiased estimates of the class 0 test set prevalence under 'invariant density ratio'-type data set shift, in practice for extreme prevalences like 1% or 99% the estimation error may be significant also for the EM-algorithm.

Since we have already seen in Section 4.2 that measurements of accuracy and F-measure are not very helpful for assessing quantification accuracy, we do not present them for the model of this section.

4.4 Training Set: Binormal; Test Set: Non-normal Densities, Non-binormal Density Ratio

The setting of this section for the training and test set distributions is exactly the same as in Section 4.3, with the exception that the ratio $\frac{h_0}{h_1}$ of the class-conditional feature densities

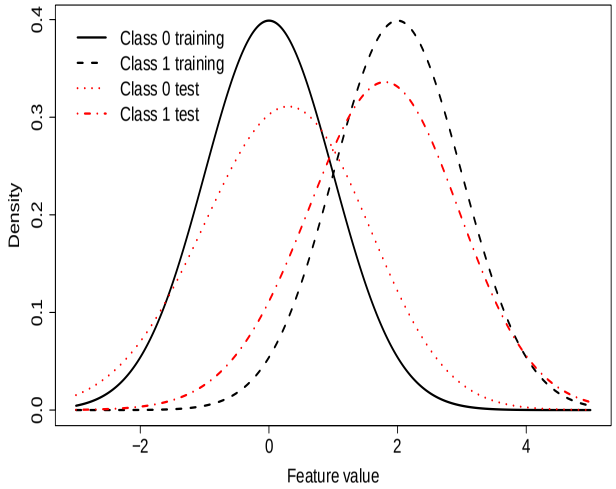


Figure 2: Training set and test set class-conditional densities for Section 4.4. The ratio of the test set densities is equal to the square root of the ratio of the training set densities.

is not equal to the test set density as given by (21b) but to its square root:

$$\frac{h_0(x)}{h_1(x)} \;=\; \sqrt{R(X)} \;=\; \exp\left(\frac{2\,x\,(\mu-\nu)+\nu^2-\mu^2}{4\,\sigma^2}\right). \tag{26}$$

Otherwise, the class-conditional feature densities h_0 and h_1 are again determined by the solution of (25a) and (25b) (with $R(x)$ replaced by $\sqrt{R(X)}$). This time, the solution for q is

$$q^* \;=\; 0.8152434.$$

Figure 2 shows the test set class-conditional densities from (25b) and the normal densities related to the training set in this case. Also the sample simulations are conducted in the same way as in Section 4.3, again with an accept-reject algorithm deployed for the test set simulations.

Table 3 and Table 8 in Appendix A show that the EM-algorithm is no longer Fisher consistent for the prevalence of class 0 when the test set distribution under consideration has not been generated from the training set distribution by 'invariant density ratio'-type data set shift. Unsurprisingly, given that AC is not Fisher consistent on test sets which are not generated by prior probability shift, it is not Fisher consistent in the family of test set distributions generated by this modified 'invariant density ratio'-type data set shift

Q[Y=0]	0.01	0.05	0.10	0.30	0.50	0.70	0.90	0.95	0.99
Prevalence estimates on samples									
Q[Y=0]	0.01	0.05	0.10	0.30	0.50	0.70	0.90	0.95	0.99
CDE1	0.2593	0.2766	0.3015	0.3987	0.4926	0.5774	0.6697	0.7007	0.7124
CDE2	0.1458	0.1666	0.2033	0.3428	0.4883	0.6196	0.7596	0.8083	0.8221
CDE∞	0.0001	0.0321	0.0920	0.2812	0.4830	0.6612	0.8993	0.9786	0.9998
AC	0.1396	0.1650	0.2017	0.3447	0.4828	0.6075	0.7433	0.7889	0.8061
EM	0.1274	0.1505	0.1952	0.3448	0.4946	0.6251	0.7785	0.8295	0.8482
Prevalence estimates on populations									
CDE1	0.2664	0.2850	0.3081	0.4008	0.4934	0.5861	0.6788	0.7019	0.7205
CDE2	0.1512	0.1775	0.2109	0.3486	0.4899	0.6320	0.7715	0.8055	0.8324
CDE∞	0.0000	0.0398	0.0914	0.2886	0.4859	0.6875	0.9020	0.9681	1.0000
AC	0.1579	0.1850	0.2189	0.3547	0.4904	0.6261	0.7619	0.7958	0.8230
EM	0.1307	0.1627	0.2015	0.3500	0.4947	0.6394	0.7875	0.8259	0.8576

Table 3: Class 0 prevalence estimates on the test sets. Training set: Binormal with equal variances. Test sets: Non-normal densities, non-binormal density ratio. 'Q[Y=0]' is the true test set prevalence of class 0. See Definition 3 for the other acronyms.

either. While both CDE1 and CDE2 perform really poorly in this section's model setting, CDE∞ performs better, both at sample and at population level. At sample level, even the CDE∞-estimates of the small class 0 prevalences look sensible.

5. Conclusions

In this paper, we have discussed the notion of Fisher consistency as a basic unbiasedness requirement for class prevalence quantifiers in the presence of data set shift. The usefulness of Fisher consistency has been demonstrated with three examples of classifiers serving as quantifiers: Adjusted Count, EM-algorithm, and CDE-Iterate. We have shown by example that CDE-Iterate is not Fisher consistent even for simple prior probability shift. Adjusted Count and EM-algorithm are Fisher consistent for prior probability shift but lose this property under data set shifts deviating not much from prior probability shift. Hence before relying on prevalence estimates by Adjusted Count or EM-algorithm, users should carefully check what kind of data set shift they are confronted with. As a further contribution to quantification-related research, we have suggested a method, based on the concept of 'invariant density ratio'-type data set shift, for conveniently generating non-trivial data set shift beyond prior probability shift and covariate shift but conceptually close to both of these types of data set shift.

Appendix A. Additional Tables

Q[Y=0]	0.01	0.05	0.10	0.30	0.50	0.70	0.90	0.95	0.99
Relative error of prevalence estimates on samples									
CDE1	15.3700	2.8200	1.2880	0.2130	0.0022	0.2163	1.2800	2.9720	15.4400
CDE2	3.0200	0.3640	0.1270	0.0030	0.0028	0.0047	0.1490	0.5400	2.8100
CDE∞	1.0000	1.0000	0.9600	0.1947	0.0030	0.2177	0.9290	1.0000	1.0000
AC	1.1030	0.2174	0.0527	0.0218	0.0159	0.0253	0.0621	0.2592	0.2651
EM	0.2987	0.0496	0.0321	0.0040	0.0065	0.0064	0.0073	0.0657	0.1042
Relative error of prevalence estimates on populations									
CDE1	15.5482	2.8558	1.2692	0.2115	0.0000	0.2115	1.2692	2.8558	15.5482
CDE2	3.0631	0.4304	0.1311	0.0019	0.0000	0.0019	0.1311	0.4304	3.0631
CDE∞	1.0000	1.0000	0.8789	0.2038	0.0000	0.2038	0.8789	1.0000	1.0000
AC	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
EM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 4: Relative error of class 0 prevalence estimates on the test sets. Training set: Binormal with equal variances. Test sets: Binormal with equal variances. 'Q[Y=0]' is the true test set prevalence of class 0. See Definition 3 for the other acronyms.

Q[Y=0]	0.01	0.05	0.10	0.30	0.50	0.70	0.90	0.95	0.99
Classification accuracy on samples									
CDE1	0.9598	0.9414	0.9139	0.8581	0.8430	0.8606	0.9141	0.9334	0.9615
CDE2	0.9868	0.9591	0.9255	0.8610	0.8431	0.8614	0.9223	0.9581	0.9882
CDE∞	0.9900	0.9500	0.9040	0.8594	0.8431	0.8573	0.9069	0.9500	0.9900
AC	0.9900	0.9591	0.9266	0.8612	0.8427	0.8617	0.9230	0.9593	0.9903
EM	0.9905	0.9592	0.9265	0.8609	0.8434	0.8613	0.9240	0.9599	0.9903
Classification accuracy on populations									
CDE1	0.9609	0.9397	0.9170	0.8591	0.8413	0.8591	0.9170	0.9397	0.9609
CDE2	0.9879	0.9588	0.9297	0.8613	0.8413	0.8613	0.9297	0.9588	0.9879
CDE∞	0.9900	0.9500	0.9109	0.8589	0.8413	0.8589	0.9109	0.9500	0.9900
AC	0.9905	0.9595	0.9299	0.8613	0.8413	0.8613	0.9299	0.9595	0.9905
EM	0.9905	0.9595	0.9299	0.8613	0.8413	0.8613	0.9299	0.9595	0.9905

Table 5: Classification accuracy on the test sets. Training set: Binormal with equal variances. Test sets: Binormal with equal variances. 'Q[Y=0]' is the true test set prevalence of class 0. See Definition 3 for the other acronyms.

Q[Y=0]	0.01	0.05	0.10	0.30	0.50	0.70	0.90	0.95	0.99
F-measure on samples									
CDE1	0.1992	0.5042	0.5952	0.7631	0.8428	0.9005	0.9519	0.9644	0.9803
CDE2	0.2143	0.4723	0.5509	0.7561	0.8429	0.9033	0.9576	0.9781	0.9940
CDE∞	NaN	NaN	0.0769	0.7404	0.8429	0.9026	0.9508	0.9744	0.9950
AC	NaN	0.4029	0.5360	0.7552	0.8420	0.9032	0.9581	0.9788	0.9951
EM	0.0952	0.4270	0.5380	0.7559	0.8431	0.9031	0.9587	0.9792	0.9951
F-measure on populations									
CDE1	0.2274	0.5035	0.6106	0.7649	0.8413	0.8994	0.9536	0.9679	0.9799
CDE2	0.3174	0.4855	0.5815	0.7562	0.8413	0.9030	0.9617	0.9785	0.9939
CDE∞	NaN	NaN	0.2050	0.7382	0.8413	0.9034	0.9528	0.9744	0.9950
AC	0.1697	0.4404	0.5681	0.7563	0.8413	0.9030	0.9619	0.9790	0.9952
EM	0.1697	0.4404	0.5681	0.7563	0.8413	0.9030	0.9619	0.9790	0.9952

Table 6: Classifier F-measure on the test sets. Training set: Binormal with equal variances. Test sets: Binormal with equal variances. 'Q[Y=0]' is the true test set prevalence of class 0. See Definition 3 for the other acronyms.

Q[Y=0]	0.01	0.05	0.10	0.30	0.50	0.70	0.90	0.95	0.99
Relative error of prevalence estimates on samples									
CDE1	14.4800	2.7300	1.2070	0.1753	0.0138	0.2880	1.4920	3.3000	17.3900
CDE2	2.1300	0.2880	0.0140	0.0757	0.0198	0.0880	0.2720	0.5440	3.5800
CDE∞	0.9900	0.9800	0.7640	0.3053	0.0304	0.1540	0.9940	1.0000	1.0000
AC	2.4122	0.3498	0.1718	0.0772	0.0330	0.1307	0.3739	0.7417	3.1336
EM	0.2638	0.0667	0.0433	0.0227	0.0012	0.0489	0.0096	0.0056	0.1325
Relative error of prevalence estimates on populations									
CDE1	15.4091	2.8146	1.2402	0.1907	0.0192	0.2547	1.4324	3.1988	17.3302
CDE2	2.5914	0.3051	0.0490	0.0483	0.0295	0.0367	0.2059	0.5655	3.6945
CDE∞	1.0000	0.9355	0.7365	0.2774	0.0411	0.2437	1.0000	1.0000	1.0000
AC	0.2038	0.0604	0.0425	0.0305	0.0281	0.0633	0.2389	0.5024	2.6102
EM	0.0029	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0001	0.0010

Table 7: Relative error of class 0 prevalence estimates on the test sets. Training set: Binormal with equal variances. Test sets: Non-normal densities, binormal density ratio. 'Q[Y=0]' is the true test set prevalence of class 0. See Definition 3 for the other acronyms.

Q[Y=0]	0.01	0.05	0.10	0.30	0.50	0.70	0.90	0.95	0.99
Relative error of prevalence estimates on samples									
CDE1	24.9300	4.5320	2.0150	0.3290	0.0148	0.4087	2.3030	4.9860	27.7600
CDE2	13.5800	2.3320	1.0330	0.1427	0.0234	0.2680	1.4040	2.8340	16.7900
CDE∞	0.9900	0.3580	0.0800	0.0627	0.0340	0.1293	0.0070	0.5720	0.9800
AC	12.9600	2.3010	1.0168	0.1489	0.0344	0.3082	1.5669	3.2218	18.3881
EM	11.7408	2.0095	0.9518	0.1495	0.0108	0.2497	1.2146	2.4093	14.1813
Relative error of prevalence estimates on populations									
CDE1	25.6419	4.6990	2.0812	0.3359	0.0131	0.3796	2.2122	4.9611	26.9524
CDE2	14.1179	2.5494	1.1085	0.1618	0.0201	0.2268	1.2851	2.8896	15.7629
CDE∞	1.0000	0.2049	0.0863	0.0381	0.0281	0.0418	0.0204	0.3618	1.0000
AC	14.7852	2.7000	1.1893	0.1822	0.0192	0.2462	1.3813	3.0839	16.7047
EM	12.0686	2.2549	1.0149	0.1667	0.0105	0.2019	1.1254	2.4814	13.2365

Table 8: Relative error of class 0 prevalence estimates on the test sets. Training set: Binormal with equal variances. Test sets: Non-normal densities, non-binormal density ratio. 'Q[Y=0]' is the true test set prevalence of class 0. See Definition 3 for the other acronyms.

Appendix B. Proof of the Convergence of CDE-Iterate

In order to provide a fully rigorous proof of Equation (17) that characterises the limit of CDE-Iterate, we adopt measure-theoretic notation in this section. See standard textbooks on probability theory like Billingsley (1995) or Durrett (1996) for reference.

We discuss the problem in a 'mixture model' probabilitistic context specified by the following assumption.

Assumption 4 *P_0 and P_1 are probability measures on some measurable space $(\Omega, \mathcal{H})$. Both P_0 and P_1 are absolutely continuous with respect to some measure μ on $(\Omega, \mathcal{H})$. The density of P_i with respect to μ is f_i, $i = 0, 1$.*

Note that in the setting of Sections 2 and 3.4, Assumption 4 is satisfied when the training set class-conditional feature distributions have got densities. Choose in that case $\Omega = \mathcal{X}$, $\mathrm{P}_0[H] = \mathrm{P}[X \in H \,|\, Y = 0]$ and $\mathrm{P}_1[H] = \mathrm{P}[X \in H \,|\, Y = 1]$. The σ-field $\mathcal{H}$ is any appropriate σ-field on $\mathcal{X}$, for instance the Borel-σ-field in case $\mathcal{X} = \mathbb{R}^d$.

For events $H \in \mathcal{H}$ we denote the complement of H in Ω by H^c, i.e. we have $H^c = \Omega \backslash H$. Then, in the setting of this section, crisp classifiers $g(X)$ with values 0 or 1 are described as events by the relations $\{g(X) = 0\} = H$ and $\{g(X) = 1\} = H^c$.

The following lemma translates the optimisation problem (1b) and its solution (2) into this section's notation and enhances them with a statement on the uniqueness of the solution.

Lemma 5 *Let $a_0, a_1 \geq 0$. Under Assumption 4, then for all $H \in \mathcal{H}$, it holds that*

$$a_0\,\mathrm{P}_0[H^c] + a_1\,\mathrm{P}_1[H] \;\geq\; a_0\,\mathrm{P}_0[a_1\,f_1 \geq a_0\,f_0] + a_1\,\mathrm{P}_1[a_1\,f_1 < a_0\,f_0]. \tag{27}$$

Equality in (27) *holds if and only if* $0 = \mu\big(H \cap \{a_1\, f_1 > a_0\, f_0\}\big)$ *and* $0 = \mu\big(H^c \cap \{a_1\, f_1 < a_0\, f_0\}\big)$.

Proof We inspect the following chain of equations and inequalities:

$$\begin{aligned}
a_0\,\mathrm{P}_0[H^c] + a_1\,\mathrm{P}_1[H] &= a_0 + a_1\,\mathrm{P}_1[H] - a_0\,\mathrm{P}_0[H] \\
&= a_0 + \int_H a_1\, f_1 - a_0\, f_0\, d\mu \\
&= a_0 + \int_{H\cap\{a_1\, f_1 < a_0\, f_0\}} a_1\, f_1 - a_0\, f_0\, d\mu + \int_{H\cap\{a_1\, f_1 > a_0\, f_0\}} a_1\, f_1 - a_0\, f_0\, d\mu \\
&\overset{\text{a)}}{\geq} a_0 + \int_{H\cap\{a_1\, f_1 < a_0\, f_0\}} a_1\, f_1 - a_0\, f_0\, d\mu \\
&\overset{\text{b)}}{\geq} a_0 + \int_{\{a_1\, f_1 < a_0\, f_0\}} a_1\, f_1 - a_0\, f_0\, d\mu \\
&= a_0\,\mathrm{P}_0[a_1\, f_1 \geq a_0\, f_0] + a_1\,\mathrm{P}_1[a_1\, f_1 < a_0\, f_0].
\end{aligned}$$

This proves (27). By inequality a), equality in (27) implies $0 = \int_{H\cap\{a_1\, f_1 > a_0\, f_0\}} a_1\, f_1 - a_0\, f_0\, d\mu$ and, therefore, $0 = \mu\big(H \cap \{a_1\, f_1 > a_0\, f_0\}\big)$. Similarly, equality in (27) implies $0 = \mu\big(H^c \cap \{a_1\, f_1 < a_0\, f_0\}\big)$ because of inequality b). ■

Lemma 5 characterises the solutions H^* of the optimisation problem

$$a_0\,\mathrm{P}_0[(H^*)^c] + a_1\,\mathrm{P}_1[H^*] \;=\; \min_{H\in\mathcal{H}} a_0\,\mathrm{P}_0[H^c] + a_1\,\mathrm{P}_1[H]. \tag{28}$$

One solution is the event $\{a_1\, f_1 < a_0\, f_0\} \in \mathcal{H}$. However, the solution is not unique. For instance, $\{a_1\, f_1 \leq a_0\, f_0\} \in \mathcal{H}$ is another solution as it easily can be checked that the two conditions for equality in (27) are satisfied. If we have $\mu(a_1\, f_1 = a_0\, f_0) = 0$, then the minimising event $\{a_1\, f_1 < a_0\, f_0\}$ from Lemma 5 is unique in the following sense: If $H^* \in \mathcal{H}$ is another minimising event then it follows that[4]

$$\begin{aligned}
0 \;&=\; \mu\big(\Delta(H^*, \{a_1\, f_1 < a_0\, f_0\})\big) \\
&=\; \mu\big((H^*)^c \cap \{a_1\, f_1 < a_0\, f_0\}\big) + \mu\big(H^* \cap \{a_1\, f_1 < a_0\, f_0\}^c\big).
\end{aligned}$$

Hence H^* and $\{a_1\, f_1 < a_0\, f_0\}$ are almost everywhere equal.

Assume, similarly to Section 2.3, that there is another probability measure Q on $(\Omega, \mathcal{H})$. Q is interpreted as the unconditional distribution of the features on a test set whose class distribution is (not yet) known. In the notation of this section, then the CDE-Iterate algorithm of Xue and Weiss (2009) can be described as follows:

CDE-Iterate algorithm

1) Set initial parameters: $k = 0$, $a_0^{(0)} > 0$, $a_1^{(0)} > 0$.

2) Find optimal classifier under Assumption 4: $H_k = \{a_1^{(k)}\, f_1 < a_0^{(k)}\, f_0\}$.

4. For two sets A and B, the term $\Delta(A, B)$ denotes the set difference $\Delta(A, B) = ((A \cap B^c) \cup (A^c \cap B)$.

3) Under probability Q compute $q_k = \mathrm{Q}[H_k]$.

4) Increment k by 1.

5) Reset cost parameters: $a_0^{(k)} = q_{k-1}$, $a_1^{(k)} = 1 - q_{k-1}$.

6) If convergence is reached or $k = k_{\max}$ then stop, else continue with step 2).

Convergence of the CDE-algorithm as given above or in the paper by Xue and Weiss (2009) is not obvious. However, we can state the following result.

Proposition 6 *Under Assumption 4, the sequence* $(q_k)_{k\geq 0}$ *determined by the CDE-algorithm as described in this section converges for any probability measure* Q *on* $(\Omega, \mathcal{H})$ *and any choice of the intial parameters* $a_0^{(0)} > 0$ *and* $a_1^{(0)} > 0$. *The limit* $q^* = \lim_{k\to\infty} q_k$ *satisfies the equation*

$$q^* \;=\; \begin{cases} \mathrm{Q}\big[(1-q^*)\, f_1 \leq q^*\, f_0,\; f_0 + f_1 > 0\big], & \text{if } q_0 \geq q_1 \text{ and } q_n > q^* \text{ for all } k. \\ \mathrm{Q}\big[(1-q^*)\, f_1 < q^*\, f_0\big], & \text{otherwise.} \end{cases} \tag{29}$$

Proof Suppose that $q_k \leq q_{k+1}$ for some k. Then it follows that

$$\begin{aligned}
q_{k+2} &= \mathrm{Q}\big[(1-q_{k+1})\, f_1 < q_{k+1}\, f_0\big] \\
&= \mathrm{Q}\big[f_1 < q_{k+1}\,(f_0 + f_1)\big] \\
&\geq \mathrm{Q}\big[(1-q_k)\, f_1 < q_k\, f_0\big] \\
&= q_{k+1}.
\end{aligned}$$

Hence $(q_k)_{k\geq 0}$ is non-decreasing if $q_0 \leq q_1$. Similarly, it can be shown that $(q_k)_{k\geq 0}$ is non-increasing if $q_0 \geq q_1$. It follows that $q^* = \lim_{k\to\infty} q_k$ exists.

Note that $\mathrm{Q}\big[f_1 < x\,(f_0 + f_1)\big] = \mathrm{Q}[f_0 + f_1 > 0]\,\mathrm{Q}\big[\frac{f_1}{f_0+f_1} < x \,|\, f_0 + f_1 > 0\big]$ is the left-continuous version of a distribution function. Therefore, it follows that

$$\begin{aligned}
\lim_{y\uparrow x} \mathrm{Q}\big[f_1 < y\,(f_0 + f_1)\big] &= \mathrm{Q}\big[f_1 < x\,(f_0 + f_1)\big], \\
\lim_{y\downarrow x} \mathrm{Q}\big[f_1 < y\,(f_0 + f_1)\big] &= \mathrm{Q}\big[f_1 \leq x\,(f_0 + f_1),\; f_0 + f_1 > 0\big]
\end{aligned}$$

By definition of q_k, this implies (29). ■

References

P. Billingsley. *Probability and measure*. John Wiley & Sons, third edition, 1995.

D.R. Cox and D.V. Hinkley. *Theoretical Statistics*. Chapman and Hall, 1974.

J.S. Cramer. *Logit Models From Economics and Other Fields*. Cambridge University Press, 2003.

A. Dal Pozzolo, O. Caelen, R.A. Johnson, and G. Bontempi. Calibrating Probability with Undersampling for Unbalanced Classification. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 159–166. IEEE, 2015.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

M.C. Du Plessis and M. Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. *Neural Networks*, 50:110–119, 2014.

R. Durrett. *Probability: Theory and Examples*. Duxbury Press, second edition, 1996.

C. Elkan. The foundations of cost-sensitive learning. In B. Nebel, editor, *Seventeenth International Joint Conference on Artificial Intelligence, IJCAI 2001*, pages 973–978. Morgan Kaufmann, 2001.

T. Fawcett and P.A. Flach. A response to Webb and Ting's On the Application of ROC Analysis to Predict Classification Performance under Varying Class Distributions. *Machine Learning*, 58(1):33–38, 2005.

R.A. Fisher. On the Mathematical Foundations of Theoretical Statistics. *Philosophical Transactions of the Royal Society of London, Series A*, 222:309–368, 1922.

G. Forman. Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206, 2008.

J.J. Gart and A.A. Buck. Comparison of a screening test and a reference test in epidemiologic studies. II. A probabilistic model for the comparison of diagnostic tests. *American Journal of Epidemiology*, 83(3):593–602, 1966.

K. Gerow. Fisher Consistency – the Evolution of a Concept: It's Hard to Get It Right the First Time. Technical Report BU-1022-M, Cornell University, Biometrics Unit, 1989.

P. González, J. Díez, N. Chawla, and J.J. del Coz. Why is quantification an interesting learning problem? *Progress in Artificial Intelligence*, pages 1–6, 2016.

J. Hernández-Orallo, P. Flach, and C. Ferri. A unified view of performance metrics: Translating threshold choice into expected classification loss. *Journal of Machine Learning Research*, 13:2813–2869, 2012.

V. Hofer. Adapting a classification rule to local and global shift when only unlabelled data are available. *European Journal of Operational Research*, 243(1):177–189, 2015.

V. Hofer and G. Krempl. Drift mining in data: A framework for addressing drift in classification. *Computational Statistics & Data Analysis*, 57(1):377–391, 2013.

N. Karpov, A. Porshnev, and K. Rudakov. NRU-HSE at SemEval-2016 Task 4: Comparative Analysis of Two Iterative Methods Using Quantification Library. In *10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 171–177, San Diego, California, June 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/S16-1025`.

M. Kull and P. Flach. Patterns of dataset shift. In *First International Workshop on Learning over Multiple Contexts (LMCE) at ECML-PKDD*, 2014.

S. Lessmann, B. Baesens, H.-V. Seow, and L.C. Thomas. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*, 247(1):124–136, 2015.

J.G. Moreno-Torres, T. Raeder, R. Alaiz-Rodriguez, N.V. Chawla, and F. Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

C. Peters and W.A. Coberly. The numerical evaluation of the maximum-likelihood estimate of mixture proportions. *Communications in Statistics – Theory and Methods*, 5(12):1127–1135, 1976.

J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N.D. Lawrence, editors. *Dataset shift in machine learning*. MIT Press, 2009.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org/`.

C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, second edition, 2004.

M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure. *Neural Computation*, 14(1):21–41, 2001.

D. Tasche. The art of probability-of-default curve calibration. *Journal of Credit Risk*, 9(4): 63–103, 2013.

D. Tasche. Exact fit of simple finite mixture models. *Journal of Risk and Financial Management*, 7(4):150–164, 2014.

D. Tasche. Does quantification without adjustments work? arXiv preprint arXiv:1602.08780, 2016.

A.W. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

H.L. van Trees. *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, 1968.

J.C. Xue and G.M. Weiss. Quantification and Semi-supervised Classification Methods for Handling Changes in Class Distribution. In *15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 897–906, New York, 2009.

Journal of Machine Learning Research 18 (2017) 1-5 Submitted 3/17; Revised 7/17; Published 10/17

openXBOW – Introducing the Passau Open-Source Crossmodal Bag-of-Words Toolkit

Maximilian Schmitt maximilian.schmitt@uni-passau.de
Björn Schuller* bjoern.schuller@uni-passau.de
Chair of Complex and Intelligent Systems, University of Passau, 94032 Passau, Germany

Editor: Geoff Holmes

Abstract

We introduce openXBOW, an open-source toolkit for the generation of bag-of-words (BoW) representations from multimodal input. In the BoW principle, word histograms were first used as features in document classification, but the idea was and can easily be adapted to, e. g., acoustic or visual descriptors, introducing a prior step of vector quantisation. The openXBOW toolkit supports arbitrary numeric input features and text input and concatenates computed sub-bags to a final bag. It provides a variety of extensions and options. To our knowledge, openXBOW is the first publicly available toolkit for the generation of crossmodal bags-of-words. The capabilities of the tool have been exemplified in different scenarios: sentiment analysis in tweets, classification of snore sounds, and time-dependent emotion recognition based on acoustic, linguistic, and visual information, where improved results over other feature representations were observed.

Keywords: bag-of-words, multimodal signal processing, histogram feature representations, feature learning

1. Introduction

The bag-of-words (BoW) principle is a common practice in *natural language processing* (NLP) (Weninger et al., 2013). In this method, *word histograms* are generated, i. e., within a text document, the frequencies of each word from a dictionary are counted. The resulting *term frequency* (TF) vector is then input to a classifier, such as a *support vector machine* (SVM), i. e., a machine learning scheme which is known to cope well with possibly irrelevant features and large, yet sparse feature vectors.

BoW has been adopted by the visual computing community, where it is known under the name bag-of-visual-words (BoVW) (Wu et al., 2011). Instead of lexical words, local image features are extracted from an image and their general distribution is modelled by a histogram. In recent years, the principle has also been employed successfully in the field of audio classification, where it is known under the term bag-of-audio-words (BoAW). Acoustic *low-level descriptors* (LLDs), such as *mel-frequency cepstral coefficients*, are extracted from the audio signal; then, the LLD vectors from single frames are quantised according to a codebook (Pancoast and Akbacak, 2014). This codebook can be the result of, e. g., a k-means clustering (Pokorny et al., 2015) or a random sampling of LLDs (Rawat et al., 2013). A histogram finally describes the distribution of the codebook vectors over the whole audio segment. Major applications of BoAW are acoustic and multimedia event detection (Rawat et al., 2013; Pancoast and Akbacak, 2013; Lim et al., 2015), but they have also been successfully employed for music information retrieval (Riley et al., 2008), emotion recognition (Schmitt et al., 2016b), and medical diagnosis (Schmitt et al., 2016a).

In this contribution, we introduce the first open-source toolkit for the generation of BoW representations across modalities, thus named openXBOW ("open crossbow" – 'X' stands for CROSSmodal). The motivation behind openXBOW is to ease the generation of a fused BoW-based representation from different modalities. These modalities can be the acoustic or the visual domain, providing numeric LLDs, and written documents or transcriptions of speech, providing text or *symbolic* input. However, arbitrary modalities, such as, e. g., stemming from physiological measurement or feature streams as used in brain computing, can be processed. In case of multimodal or 'crossmodal' usage, the output of the toolkit is a concatenated feature vector consisting of histogram representations per modality or combinations of these. Crossmodal BoW have already been employed, e. g., for depression monitoring (Joshi et al., 2013), exploiting both the audio and the video domain.

openXBOW provides a multitude of options, e. g., different modes of vector quantisation, codebook generation, TF weighting, and methods known from natural language processing to process the textual features. To the knowledge of the authors, such a toolkit has not been published, so far, whereas there are already some libraries implementing BoVW, such as *DBoW2* (Gálvez-López and Tardos, 2012).

In the next section, we give an overview of the openXBOW tool, its structure and its options. In Section 3, we give results from an exemplary application of the tool. We conclude and give an outlook on future developments in openXBOW in Section 4.

2. Overview

openXBOW is implemented in Java and can thus be used on any common platform. It has been published on GitHub as a public repository[1], including both the source code and a compiled jar file for users who do not have a Java Development Kit installed. The software and the source code are published under GPLv3. openXBOW supports three formats for input and output: *CSV* (comma separated values), *ARFF* (attribute-relation file format), used in the broadly applied machine learning software *Weka* (Hall et al., 2009), and *LIBSVM* file format (only for output), used in LIBSVM (Chang and Lin, 2011) and LIBLINEAR (Fan et al., 2008). All options, which are configured through the command line call, are displayed when calling openXBOW ('java -jar openXBOW.jar').

The input is processed in the way shown in Figure 1. First, the input features can be pre-processed, particularly, (min-max)-normalisation or standardisation can be applied to the numeric descriptors, which is meaningful in case feature types with (significantly) different ranges of values are combined. The corresponding parameters are stored in the *codebook file*, in order to be applied also to a given test file (online approach).

For codebook generation, there are four different methods; *random sampling*, which means that the codebook vectors are picked randomly from the input, whereas *random sampling++* favours far-off vectors as proposed by Arthur and Vassilvitskii (2007). This implements basically the initialisation step of *k-means++ clustering*, which is also available besides the classical *k-means clustering*. The codebook generation can also be done in a *supervised* manner, learning a codebook from the input descriptors of one class separately,

∗. B. Schuller is also with GLAM – Group on Language, Audio & Music, Imperial College London, U. K.

1. https://github.com/openXBOW/openXBOW

©2017 Maximilian Schmitt and Björn Schuller.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-113.html.

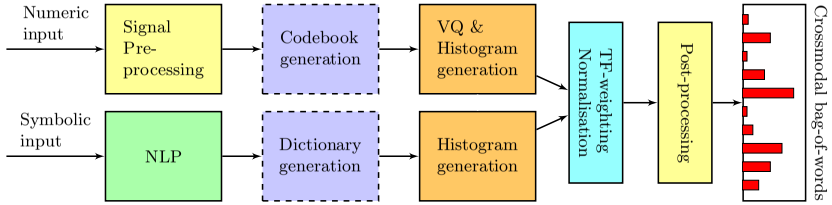


Figure 1: Overview of the basic workflow of openXBOW.

first, and then concatenating these codebooks to form a *super-codebook* (Grzeszick et al., 2015). Besides, it is also possible to split the input into well-defined sub-vectors in order to generate different sub-codebooks for different feature types. The vector quantisation (VQ) of the input is based on the minimum *Euclidean distance* between an input vector and all words in the codebook, sometimes referred to as *templates*. The bag-of-words are then generated from the term frequencies, i. e., the number of occurrences of each template, within the whole document. In the input file, all feature vectors belonging to the same instance need to have the same unique name as an attribute. The input can also be subject to *multiple assignments*, where not only the word with the closest distance is considered but a specified number of close templates. A way of doing *soft* vector quantisation is applying *Gaussian encoding* (Pancoast and Akbacak, 2014), where in each word assignment step, the increment is weighted with the distance to the templates.

For the text domain, standard processing techniques known from NLP are available, such as *stopping*, *n-grams*, and *n-character-grams*. The numeric codebook and the symbolic dictionary are stored in a file to be used for independent test data.

The BoW representations of different domains are fused into a single feature vector (early fusion). Logarithmic TF-weighting and inverse document-frequency (IDF) weighting can be applied (Riley et al., 2008). The bags can then be normalised, a step which might be meaningful if the input instances have different lengths. Finally, post-processing in terms of (min-max-)normalisation and standardisation is available for the resulting BoW.

3. Experiments

The usage of openXBOW is now shortly exemplified in one scenario: *crossmodal emotion recognition*. The above mentioned GitHub repository includes a *tutorial* which allows the user to reproduce results for different scenarios and several features types, including *sentiment analysis* of *tweets* in a data set of more than 1 million instances.

Emotion recognition in speech has been conducted on the SEWA[2] (Automatic Sentiment Analysis in the Wild) corpus, more specifically, on video chat recordings of 64 German subjects. The overall length of this audio-visual data is approximately 89 minutes, the data was split into subject-independent training, development (devel), and test partitions (34/14/16 subjects). Emotion has been annotated continuously in terms of the two emotional dimensions *arousal* and *valence* (Schmitt et al., 2016b). Results are presented in terms of the *concordance correlation coefficient* (CCC) between the prediction and the annotation. For

2. `http://sewaproject.eu/`

Modality	Arousal (devel)	Arousal (test)	Valence (devel)	Valence (test)
Acoustic	.535	.470	.402	.426
Visual	.434	.314	.385	.344
Linguistic	.364	.293	.328	.320
Crossmodal (early fusion)	.560	.432	.536	.509
Crossmodal (late fusion)	.588	.499	.495	.523

Table 1: Performance (in terms of CCC) of emotion recognition on the SEWA German video chats using crossmodal BoW. The codebook size for numeric features is 1 000, the number of assignments is 10. The dictionary consists of 346 words.

the acoustic domain, the 65 LLDs from the ComParE feature set have been extracted using the toolkit openSMILE (Eyben et al., 2013). For the visual domain, we used 49 *facial landmarks* extracted with the Chehra Face Tracker (Asthana et al., 2014). Manual transcriptions of the speech were used as linguistic input. As the target of the prediction is time-dependent, the input was segmented into overlapping blocks of 8 *s* width and 0.1 *s* hop size, as described by Schmitt et al. (2016b). For decoding of the BoW, LIBLINEAR was used; the complexity parameter was optimised on the development set. The results reported in Table 1 show that, both early and late fusion—by training another SVM based on the predictions from single modalities—are suitable for the prediction of emotional speaker states. Further results with BoAW on well-established databases with comparison to other approaches have been reported by Schmitt et al. (2016a,b).

The performance of openXBOW is quite decent. On a system with an *Intel Core i7-4770 (3.4 GHz)* CPU, *16 GB RAM*, *Windows 10* operating system, and *Java Version 8, Update 121*, the computation of the aforementioned crossmodal BoW (with early fusion) took 263 *s* for training and 67 *s* for prediction. Calling openXBOW for a short video segment of 8 *s* duration takes 0.57 *s*. Thus, the toolkit is suitable for real-time applications ($< 10\,\%$ real-time factor for crossmodal input). For the sentiment analysis task mentioned in the *tutorial*, the generation of the BoW representation for 1 million tweets took 118 *s*.

4. Conclusions and Outlook

We introduced our novel openXBOW toolkit—a first of its kind—for the generation of BoW representations from crossmodal symbolic (including text), but also numeric (such as audio or video feature streams) information representations. We showed the potential of the toolkit and the underlying BoW principle in a crossmodal emotion recognition task on a state-of-the-art database.

Future work on openXBOW will include further codebook generation techniques such as *EM clustering* or *non-negative matrix factorisation* and alternative methods of soft vector quantisation.

Acknowledgments

The research leading to these results has received funding from the European Union's Horizon 2020 and 7th Framework Programmes under GA No. 645094 (H2020 IA SEWA) and No. 338164 (ERC StG iHEARu).

References

D. Arthur and S. Vassilvitskii. K-means++: the advantages of careful seeding. In *ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035, 2007.

A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic. Incremental face alignment in the wild. In *CVPR*, pages 1859–1866, 2014.

C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. *ACM TIST*, 2:27:1–27:27, 2011.

F. Eyben, F. Weninger, F. Groß, and B. Schuller. Recent developments in openSMILE, the munich open-source multimedia feature extractor. In *ACM MM*, pages 835–838, 2013.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: a library for large linear classification. *JMLR*, 9:1871–1874, 2008.

D. Gálvez-López and J. D. Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE T-RO*, 28(5):1188–1197, 2012.

R. Grzeszick, A. Plinge, and G. A. Fink. Temporal acoustic words for online acoustic event detection. In *GCPR*, pages 142–153, 2015.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explor. Newsletter*, 11(1):10–18, 2009.

J. Joshi, R. Goecke, S. Alghowinem, A. Dhall, M. Wagner, J. Epps, G. Parker, and M. Breakspear. Multimodal assistive technologies for depression diagnosis and monitoring. *Journal on MultiModal User Interfaces*, 7(3):217–228, 2013.

H. Lim, M. J. Kim, and H. Kim. Robust sound event classification using LBP-HOG based bag-of-audio-words feature representation. In *INTERSPEECH*, pages 3325–3329, 2015.

S. Pancoast and M. Akbacak. N-gram extension for bag-of-audio-words. In *ICASSP*, pages 778–782, 2013.

S. Pancoast and M. Akbacak. Softening quantization in bag-of-audio-words. In *ICASSP*, pages 1370–1374, 2014.

F. Pokorny, F. Graf, F. Pernkopf, and B. Schuller. Detection of negative emotions in speech signals using bags-of-audio-words. In *WASA/ACII*, pages 879–884, 2015.

S. Rawat, P. F. Schulam, S. Burger, D. Ding, Y. Wang, and F. Metze. Robust audio-codebooks for large-scale event detection in consumer videos. In *INTERSPEECH*, pages 2929–2933, 2013.

M. Riley, E. Heinen, and J. Ghosh. A text retrieval approach to content-based audio hashing. In *ISMIR*, pages 295–300, 2008.

M. Schmitt, C. Janott, V. Pandit, K. Qian, C. Heiser, W. Hemmert, and B. Schuller. A bag-of-audio-words approach for snore sounds' excitation localisation. In *ITG Speech Communication*, pages 230–234, 2016a.

M. Schmitt, F. Ringeval, and B. Schuller. At the border of acoustics and linguistics: bag-of-audio-words for the recognition of emotions in speech. In *INTERSPEECH*, pages 495–499, 2016b.

F. Weninger, P. Staudt, and B. Schuller. Words that fascinate the listener: predicting affective ratings of on-line lectures. *International Journal of Distance Education Technologies, Special Issue on Emotional Intelligence for Online Learning*, 11(2):110–123, 2013.

J. Wu, W.-C. Tan, and J. M. Rehg. Efficient and effective visual codebook generation using additive kernels. *JMLR*, 12:3097–3118, 2011.

Journal of Machine Learning Research 18 (2017) 1-47 Submitted 3/17; Revised 8/17; Published 10/17

Optimal Rates for Multi-pass Stochastic Gradient Methods

Junhong Lin* JUNHONG.LIN@IIT.IT
Laboratory for Computational and Statistical Learning
Istituto Italiano di Tecnologia and Massachusetts Institute of Technology
Bldg. 46-5155, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

Lorenzo Rosasco LROSASCO@MIT.EDU
DIBRIS, Università di Genova
Via Dodecaneso, 35 — 16146 Genova, Italy
Laboratory for Computational and Statistical Learning
Istituto Italiano di Tecnologia and Massachusetts Institute of Technology
Bldg. 46-5155, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

Editor: Leon Bottou

Abstract

We analyze the learning properties of the stochastic gradient method when multiple passes over the data and mini-batches are allowed. We study how regularization properties are controlled by the step-size, the number of passes and the mini-batch size. In particular, we consider the square loss and show that for a universal step-size choice, the number of passes acts as a regularization parameter, and optimal finite sample bounds can be achieved by early-stopping. Moreover, we show that larger step-sizes are allowed when considering mini-batches. Our analysis is based on a unifying approach, encompassing both batch and stochastic gradient methods as special cases. As a byproduct, we derive optimal convergence results for batch gradient methods (even in the non-attainable cases).

1. Introduction

Modern machine learning applications require computational approaches that are at the same time statistically accurate and numerically efficient (Bousquet and Bottou, 2008). This has motivated a recent interest in stochastic gradient methods (SGM), since on the one hand they enjoy good practical performances, especially in large scale scenarios, and on the other hand they are amenable to theoretical studies. In particular, unlike other learning approaches, such as empirical risk minimization or Tikhonov regularization, theoretical results on SGM naturally integrate statistical and computational aspects.

Most generalization studies on SGM consider the case where only one pass over the data is allowed and the step-size is appropriately chosen, see (Cesa-Bianchi et al., 2004; Nemirovski et al., 2009; Ying and Pontil, 2008; Tarres and Yao, 2014; Dieuleveut and Bach, 2016; Orabona, 2014) and references therein, possibly considering averaging (Poljak, 1987). In particular, recent works show how the step-size can be seen to play the role of a regularization parameter whose choice controls the bias and variance properties of the obtained solution (Ying and Pontil, 2008; Tarres and Yao, 2014; Dieuleveut and Bach, 2016; Lin et al., 2016a). These latter works show that balancing these contributions, it is possible to derive a step-size choice leading to optimal learning bounds. Such a choice typically depends on some unknown properties of the data generating distributions and it can be chosen by cross-validation in practice.

While processing each data point only once is natural in streaming/online scenarios, in practice SGM is often used to process large data-sets and multiple passes over the data are typically considered. In this case, the number of passes over the data, as well as the step-size, need then to be determined. While the role of multiple passes is well understood if the goal is empirical risk minimization (see e.g., Boyd and Mutapcic, 2007), its effect with respect to generalization is less clear. A few recent works have recently started to tackle this question. In particular, results in this direction have been derived in (Hardt et al., 2016) and (Lin et al., 2016a). The former work considers a general stochastic optimization setting and studies stability properties of SGM allowing to derive convergence results as well as finite sample bounds. The latter work, restricted to supervised learning, further develops these results to compare the respective roles of step-size and number of passes, and show how different parameter settings can lead to optimal error bounds. In particular, it shows that there are two extreme cases: while one between the step-size or the number of passes is fixed a priori, while the other one acts as a regularization parameter and needs to be chosen adaptively. The main shortcoming of these latter results is that they are for the worst case, in the sense that they do not consider the possible effect of benign assumptions on the problem (Zhang, 2005; Caponnetto and De Vito, 2007) that can lead to faster rates for other learning approaches such as Tikhonov regularization. Further, these results do not consider the possible effect on generalization of mini-batches, rather than a single point in each gradient step (Shalev-Shwartz et al., 2011; Dekel et al., 2012; Sra et al., 2012; Ng, 2016). This latter strategy is often considered especially for parallel implementation of SGM.

The study in this paper fills in these gaps in the case where the loss function is the least squares loss. We consider a variant of SGM for least squares, where gradients are sampled uniformly at random and mini-batches are allowed. The number of passes, the step-size and the mini-batch size are then parameters to be determined. Our main results highlight the respective roles of these parameters and show how can they be chosen so that the corresponding solutions achieve optimal learning errors in a variety of settings. In particular, we show for the first time that multi-pass SGM with early stopping and a universal step-size choice can achieve optimal convergence rates, matching those of ridge regression (Smale and Zhou, 2007; Caponnetto and De Vito, 2007). Further, our analysis shows how the mini-batch size and the step-size choice are tightly related. Indeed, larger mini-batch sizes allow considering larger step-sizes while keeping the optimal learning bounds. This result gives insights on how to exploit mini-batches for parallel computations while preserving optimal statistical accuracy. Finally, we note that a recent work (Rosasco and Villa, 2015) is related to the analysis in the paper. The generalization properties of a multi-pass incremental gradient are analyzed in (Rosasco and Villa, 2015), for a cyclic, rather than a stochastic, choice of the gradients and with no mini-batches. The analysis in this latter case appears to be harder and results in (Rosasco and Villa, 2015) give good learning bounds only in restricted setting and considering iterates rather than the excess risk. Compared to (Rosasco and Villa, 2015) our results show how stochasticity can be exploited to get fast rates and analyze the role of mini-batches. The basic idea of our proof is to approximate the SGM learning sequence in terms of the batch gradient descent sequence, see Subsection 3.7 for further details. This allows to study batch and stochastic gradient methods simultaneously, and may be also useful for analyzing other learning algorithms.

This paper is an extended version of a prior conference paper (Lin and Rosasco, 2016). In (Lin and Rosasco, 2016), we give convergence results with optimal rates for the attainable case

*. J.L. is now with the Swiss Federal Institute of Technology in Lausanne (jhlin5@hotmail.com)

©2017 Junhong Lin and Lorenzo Rosasco.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-176.html.

JMLR 18(97):1-47, 2017

(i.e., assuming the existence of at least one minimizer of the expected risk over the hypothesis space) in a fixed step-size setting. In this new version, we give convergence results with optimal rates, for both the attainable and non-attainable cases, and consider more general step-size choices. The extension from the attainable case to the non-attainable case is non-trivial. As will be seen from the proof, in contrast to the attainable case, a different and refined estimation is needed for the non-attainable case. Interestingly, as a byproduct of this paper, we also derived optimal rates for the batch gradient descent methods in the non-attainable case. To the best of our knowledge, such a result may be the first kind for batch gradient methods, without requiring any extra unlabeled data as that in (Caponnetto and Yao, 2010). Finally, we also add novel convergence results for the iterates showing that they converge to the minimal norm solution of the expected risk with optimal rates.

The rest of this paper is organized as follows. Section 2 introduces the learning setting and the SGM algorithm. Main results with discussions and proof sketches are presented in Section 3. Preliminary lemmas necessary for the proofs will be given in Section 4 while detailed proofs will be conducted in Sections 5 to 8. Finally, simple numerical simulations are given in Section 9 to complement our theoretical results.

Notation For any $a, b \in \mathbb{R}$, $a \vee b$ denotes the maximum of a and b. $\mathbb{N}$ is the set of all positive integers. For any $T \in \mathbb{N}$, $[T]$ denotes the set $\{1, \cdots, T\}$. For any two positive sequences $\{a_t\}_{t\in[T]}$ and $\{b_t\}_{t\in[T]}$, the notation $a_t \lesssim b_t$ for all $t \in [T]$ means that there exists a positive constant $C \geq 0$ such that C is independent of t and that $a_t \leq Cb_t$ for all $t \in [T]$.

2. Learning with SGM

We begin by introducing the learning setting we consider, and then describe the SGM learning algorithm. Following (Rosasco and Villa, 2015), the formulation we consider is close to the setting of functional regression, and covers the reproducing kernel Hilbert space (RKHS) setting as a special case, see Appendix A. In particular, it reduces to standard linear regression for finite dimensions.

2.1 Learning Problems

Let H be a separable Hilbert space, with inner product and induced norm denoted by $\langle\cdot,\cdot\rangle_H$ and $\|\cdot\|_H$, respectively. Let the input space $X \subseteq H$ and the output space $Y \subseteq \mathbb{R}$. Let ρ be an unknown probability measure on $Z = X \times Y$, $\rho_X(\cdot)$ the induced marginal measure on X, and $\rho(\cdot|x)$ the conditional probability measure on Y with respect to $x \in X$ and ρ.

Considering the square loss function, the problem under study is the minimization of the *risk*,

$$\inf_{\omega\in H} \mathcal{E}(\omega), \quad \mathcal{E}(\omega) = \int_{X\times Y} (\langle\omega, x\rangle_H - y)^2 d\rho(x, y), \tag{1}$$

when the measure ρ is known only through a sample $\mathbf{z} = \{z_i = (x_i, y_i)\}_{i=1}^m$ of size $m \in \mathbb{N}$, independently and identically distributed (i.i.d.) according to ρ. In the following, we measure the quality of an approximate solution $\hat{\omega} \in H$ (an estimator) considering *the excess risk*, i.e.,

$$\mathcal{E}(\hat{\omega}) - \inf_{\omega\in H} \mathcal{E}(\omega). \tag{2}$$

Throughout this paper, we assume that there exists a constant $\kappa \in [1, \infty[$, such that

$$\langle x, x'\rangle_H \leq \kappa^2, \quad \forall x, x' \in X. \tag{3}$$

2.2 Stochastic Gradient Method

We study the following variant of SGM, possibly with mini-batches. Unlike some of the variants studied in the literature, the algorithm we consider in this paper does not involve any explicit penalty term or any projection step, in which case one does not need to tune the penalty/projection parameter.

Algorithm 1 *Let $b \in [m]$. Given any sample $\mathbf{z}$, the b-minibatch stochastic gradient method is defined by $\omega_1 = 0$ and*

$$\omega_{t+1} = \omega_t - \eta_t \frac{1}{b} \sum_{i=b(t-1)+1}^{bt} (\langle\omega_t, x_{j_i}\rangle_H - y_{j_i}) x_{j_i}, \qquad t = 1, \ldots, T, \tag{4}$$

where $\{\eta_t > 0\}$ is a step-size sequence. Here, $j_1, j_2, \cdots, j_{bT}$ are i.i.d. random variables from the uniform distribution on $[m]$ [1].

We add some comments on the above algorithm. First, different choices for the mini-batch size b can lead to different algorithms. In particular, for $b = 1$, the above algorithm corresponds to a simple SGM, while for $b = m$, it is a stochastic version of the batch gradient descent. In this paper, we are particularly interested in the cases of $b = 1$ and $b = \sqrt{m}$. Second, other choices on the initial value, rather than $\omega_1 = 0$, is possible. In fact, following from our proofs in this paper, the interested readers can see that the convergence results stated in the next subsections still hold for other choices of initial values. Finally, the number of total iterations T can be bigger than the number of sample points m. This indicates that we can use the sample more than once, or in another words, we can run the algorithm with multiple passes over the data. Here and in what follows, the number of 'passes' over the data is referred to $\lceil \frac{bt}{m} \rceil$ at t iterations of the algorithm.

The aim of this paper is to derive excess risk bounds for Algorithm 1. Throughout this paper, we assume that $\{\eta_t\}_t$ is non-increasing, and $T \in \mathbb{N}$ with $T \geq 3$. We denote by $\mathbf{J}_t$ the set $\{j_l : l = b(t-1)+1, \cdots, bt\}$ and by $\mathbf{J}$ the set $\{j_l : l = 1, \cdots, bT\}$.

3. Main Results with Discussions

In this section, we first state some basic assumptions. Then, we present and discuss our main results.

3.1 Assumptions

The following assumption is related to a moment assumption on $|y|^2$. It is weaker than the often considered bounded output assumption, such as the binary classification problems where $Y = \{-1, 1\}$.

Assumption 1 *There exists constants $M \in]0, \infty[$ and $v \in]1, \infty[$ such that*

$$\int_Y y^{2l} d\rho(y|x) \leq l! M^l v, \quad \forall l \in \mathbb{N}, \tag{5}$$

ρ_X-almost surely.

1. Note that, the random variables $j_1, \cdots, j_{bT}$ are conditionally independent given the sample $\mathbf{z}$.

To present our next assumption, we introduce the operator $\mathcal{L}_\rho : L^2(H,\rho_X) \to L^2(H,\rho_X)$, defined by $\mathcal{L}_\rho(f) = \int_X \langle x,\cdot\rangle_H f(x)\rho_X(x)$. Here, $L^2(H,\rho_X)$ is the Hilbert space of square integral functions from H to $\mathbb{R}$ with respect to ρ_X, with norm,

$$\|f\|_\rho = \left(\int_X |f(x)|^2 d\rho_X(x)\right)^{1/2}.$$

Under Assumption (3), $\mathcal{L}_\rho$ can be proved to be positive trace class operators (Cucker and Zhou, 2007), and hence $\mathcal{L}_\rho^\zeta$ with $\zeta \in \mathbb{R}$ can be defined by using the spectral theory.

It is well known (see e.g., Cucker and Zhou, 2007) that the function minimizing $\int_Z (f(x)-y)^2 d\rho(z)$ over all measurable functions $f : H \to \mathbb{R}$ is the regression function, given by

$$f_\rho(x) = \int_Y y d\rho(y|x), \qquad x \in X. \tag{6}$$

Define another Hilbert space $H_\rho = \{f : X \to \mathbb{R} | \exists \omega \in H \text{ with } f(x) = \langle \omega, x\rangle_H, \rho_X\text{-almost surely}\}$. Under Assumption (3), it is easy to see that H_ρ is a subspace of $L^2(H,\rho_X)$. Let $f_\mathcal{H}$ be the projection of the regression function f_ρ onto the closure of H_ρ in $L^2(H,\rho_X)$. It is easy to see that the search for a solution of Problem (1) is equivalent to the search of a linear function in H_ρ to approximate $f_\mathcal{H}$. From this point of view, bounds on the excess risk of a learning algorithm on H_ρ or H, naturally depend on the following assumption, which quantifies how well, the target function $f_\mathcal{H}$ can be approximated by H_ρ.

Assumption 2 *There exist $\zeta > 0$ and $R > 0$, such that $\|\mathcal{L}_\rho^{-\zeta} f_\mathcal{H}\|_\rho \le R$.*

The above assumption is fairly standard in non-parametric regression (Cucker and Zhou, 2007; Rosasco and Villa, 2015). The bigger ζ is, the more stringent the assumption is, since

$$\mathcal{L}_\rho^{\zeta_1}(L^2(H,\rho_X)) \subseteq \mathcal{L}_\rho^{\zeta_2}(L^2(H,\rho_X)) \quad \text{when } \zeta_1 \ge \zeta_2.$$

In particular, for $\zeta = 0$, we are making no assumption, while for $\zeta = 1/2$, we are requiring $f_\mathcal{H} \in H_\rho$, since (Rosasco and Villa, 2015)

$$H_\rho = \mathcal{L}_\rho^{1/2}(L^2(H,\rho_X)). \tag{7}$$

In the case of $\zeta \ge 1/2$, $f_\mathcal{H} \in H_\rho$, which implies Problem (1) has at least one solution in the space H. In this case, we denote $\omega^\dagger$ as the solution with the minimal H-norm.

Finally, the last assumption relates to the capacity of the hypothesis space.

Assumption 3 *For some $\gamma \in]0,1]$ and $c_\gamma > 0$, $\mathcal{L}_\rho$ satisfies*

$$\mathrm{tr}(\mathcal{L}_\rho(\mathcal{L}_\rho + \lambda I)^{-1}) \le c_\gamma \lambda^{-\gamma}, \quad \text{for all } \lambda > 0. \tag{8}$$

The left hand-side of of (8) is called as the effective dimension (Caponnetto and De Vito, 2007), or the degrees of freedom (Zhang, 2005). It can be related to covering/entropy number conditions, see (Steinwart and Christmann, 2008) for further details. Assumption 3 is always true for $\gamma = 1$ and $c_\gamma = \kappa^2$, since $\mathcal{L}_\rho$ is a trace class operator which implies the eigenvalues of $\mathcal{L}_\rho$, denoted as σ_i, satisfy $\mathrm{tr}(\mathcal{L}_\rho) = \sum_i \sigma_i \le \kappa^2$. This is referred to as the capacity independent setting. Assumption 3 with $\gamma \in]0,1]$ allows to derive better error rates. It is satisfied, e.g., if the eigenvalues of $\mathcal{L}_\rho$ satisfy a polynomial decaying condition $\sigma_i \sim i^{-1/\gamma}$, or with $\gamma = 0$ if $\mathcal{L}_\rho$ is finite rank.

3.2 Optimal Rates for SGM and Batch GM: Simplified Versions

We start with the following corollaries, which are the simplified versions of our main results stated in the next subsections.

Corollary 1 (Optimal Rate for SGM) *Under Assumptions 2 and 3, let $|y| \le M$ almost surely for some $M > 0$. Let $p_* = \lceil m^{\frac{1}{2\zeta+\gamma}} \rceil$ if $2\zeta+\gamma > 1$, or $p_* = \lceil m^{1-\epsilon} \rceil$ with $\epsilon \in]0,1[$ otherwise. Consider the SGM with*
1) $b = 1$, $\eta_t \simeq \frac{1}{m}$ for all $t \in [(p_* m)]$, and $\tilde\omega_{p_*} = \omega_{p_* m+1}$.
If $\delta \in]0,1]$ and $m \ge m_\delta$, then with probability[2] at least $1-\delta$, it holds

$$\mathbb{E}_\mathbf{J}[\mathcal{E}(\tilde\omega_{p_*})] - \inf_H \mathcal{E} \le C \begin{cases} m^{-\frac{2\zeta}{2\zeta+\gamma}} & \text{when } 2\zeta+\gamma > 1; \\ m^{-2\zeta(1-\epsilon)} & \text{otherwise.} \end{cases} \tag{9}$$

Furthermore, the above also holds for the SGM with[3]
2) $b = \sqrt{m}$, $\eta_t \simeq \frac{1}{\sqrt{m}}$ for all $t \in [(p_*\sqrt{m})]$, and $\tilde\omega_{p_*} = \omega_{p_*\sqrt{m}+1}$.
In the above, m_δ and C are positive constants depending on $\kappa^2, \|\mathcal{T}_\rho\|, M, \zeta, R, c_\gamma, \gamma$, a polynomial of $\log m$ and $\log(1/\delta)$, and m_δ also on δ (and also on $\|f_\mathcal{H}\|_\infty$ in the case that $\zeta < 1/2$).

We add some comments on the above result. First, the above result asserts that, at p_* passes over the data, the SGM with two different fixed step-size and fixed mini-batch size choices, achieves optimal learning error bounds, matching (or improving) those of ridge regression (Smale and Zhou, 2007; Caponnetto and De Vito, 2007). Second, according to the above result, using mini-batch allows to use a larger step-size while achieving the same optimal error bounds. Finally, the above result can be further simplified in some special cases. For example, if we consider the capacity independent case, i.e., $\gamma = 1$, and assuming that $f_\mathcal{H} \in H_\rho$, which is equivalent to making Assumption 2 with $\zeta = 1/2$ as mentioned before, the error bound is $O(m^{-1/2})$, while the number of passes $p_* = \lceil \sqrt{m} \rceil$.

Remark 1 (Finite Dimensional Case) *With a simple modification of our proofs, we can derive similar results for the finite dimensional case, i.e., $H = \mathbb{R}^d$, where in this case, $\gamma = 0$. In particular, letting $\zeta = 1/2$, under the same assumptions of Corollary 1, if one considers the SGM with $b = 1$ and $\eta_t \simeq \frac{1}{m}$ for all $t \in [m^2]$, then with high probability, $\mathbb{E}_\mathbf{J}[\mathcal{E}(\omega_{m^2+1})] - \inf_H \mathcal{E} \lesssim d/m$, provided that $m \gtrsim d \log d$.*

Remark 2 *From the proofs, one can easily see that if $f_\mathcal{H}$ and $\mathcal{E}(\tilde\omega_{p_*}) - \inf_H \mathcal{E}$ are replaced respectively by $f_* \in L^2(H,\rho_X)$ and $\|\langle\cdot,\tilde\omega_{p_*}\rangle_H - f_*\|_\rho^2$, in both the assumptions and the error bounds, then all theorems and their corollaries of this paper are still true, as long as f_* satisfies $\int_X (f_* - f_\rho)(x) K_x d\rho_X = 0$. As a result, if we assume that f_ρ satisfies Assumption 2 (with $f_\mathcal{H}$ replaced by f_ρ), as typically done in (Smale and Zhou, 2007; Caponnetto and De Vito, 2007; Steinwart et al., 2009; Caponnetto and Yao, 2010) for the RKHS setting, we have that with high probability,*

$$\mathbb{E}_\mathbf{J}\|\langle\cdot,\tilde\omega_{p_*}\rangle_H - f_\rho\|_\rho^2 \le C \begin{cases} m^{-\frac{2\zeta}{2\zeta+\gamma}} & \text{when } 2\zeta+\gamma > 1; \\ m^{-2\zeta(1-\epsilon)} & \text{otherwise.} \end{cases}$$

In this case, the factor $\|f_\mathcal{H}\|_\infty$ from the upper bounds for the case $\zeta < 1/2$ is exactly $\|f_\rho\|_\infty$ and can be controlled by the condition $|y| \le M$ (and more generally, by Assumption 1). Since

2. Here, 'high probability' refers to the sample $\mathbf{z}$.
3. Here, we assume that $\sqrt{m}$ is an integer.

many common RKHSs are universally consistent (Steinwart and Christmann, 2008), making Assumption 2 on f_ρ is natural and moreover, deriving error bounds with respect to f_ρ seems to be more interesting in this case.

As a byproduct of our proofs in this paper, we derive the following optimal results for batch gradient methods (GM), defined by $\nu_1 = 0$ and

$$\nu_{t+1} = \nu_t - \eta_t \frac{1}{m}\sum_{i=1}^{m}(\langle \nu_t, x_i\rangle_H - y_i)x_i, \qquad t = 1, \dots, T. \tag{10}$$

Corollary 2 (Optimal Rate for Batch GM) *Under the assumptions and notations of Corollary 1, consider batch GM (10) with $\eta_t \simeq 1$. If m is large enough, then with high probability, (9) holds for $\tilde{\omega}_{p_*} = \nu_{p_*+1}$.*

In the above corollary, the convergence rates are optimal for $2\zeta + \gamma > 1$. To the best of our knowledge, these results are the first ones with minimax rates (Caponnetto and De Vito, 2007; Blanchard and Mücke, 2016) for the batch GM in the non-attainable case. Particularly, they improve the results in the previous literature, see Subsection 3.6 for more discussions.

Corollaries 1 and 2 cover the main contributions of this paper. In the following subsections, we will present the main theorems of this paper, following with several corollaries and simple discussions, from which one can derive the simplified versions stated in this subsection. In the next subsection, we present results for SGM in the attainable case while results in the non-attainable case will be given in Subsection 3.4, as the bounds for these two cases are different and particularly their proofs require different estimations. At last, results with more specific convergence rates for batch GM will be presented in Subsection 3.5.

3.3 Main Results for SGM: Attainable Case

In this subsection, we present convergence results in the attainable case, i.e., $\zeta \geq 1/2$, following with simple discussions. One of our main theorems in the attainable case is stated next, and provides error bounds for the studied algorithm. For the sake of readability, we only present results in a fixed step-size setting in this section. Results in a general setting ($\eta_t = \eta_1 t^{-\theta}$ with $0 \leq \theta < 1$ can be found in Section 7.

Theorem 1 *Under Assumptions 1, 2 and 3, let $\zeta \geq 1/2$, $\delta \in]0,1[$, $\eta_t = \eta\kappa^{-2}$ for all $t \in [T]$, with $\eta \leq \frac{1}{8(\log T+1)}$. If $m \geq m_\delta$, then the following holds with probability at least $1-\delta$: for all $t \in [T]$,*

$$\begin{aligned}\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})] - \inf_H \mathcal{E} \leq q_1(\eta t)^{-2\zeta} + q_2 m^{-\frac{2\zeta}{2\zeta+\gamma}}(1 + m^{-\frac{1}{2\zeta+\gamma}}\eta t)^2 \log^2 T \log^2\frac{1}{\delta} \\ + q_3 \eta b^{-1}(1 \vee m^{-\frac{1}{2\zeta+\gamma}}\eta t)\log T.\end{aligned} \tag{11}$$

Here, m_δ, q_1, q_2 and q_3 are positive constants depending on $\kappa^2, \|\mathcal{T}_\rho\|, M, v, \zeta, R, c_\gamma, \gamma$, and m_δ also on δ (which will be given explicitly in the proof).

There are three terms in the upper bounds of (11). The first term depends on the regularity of the target function and it arises from bounding the bias, while the last two terms result from estimating the sample variance and the computational variance (due to the random choices of the points), respectively. To derive optimal rates, it is necessary to balance these three terms.

Solving this trade-off problem leads to different choices on η, T, and b, corresponding to different regularization strategies, as shown in subsequent corollaries.

The first corollary gives generalization error bounds for simple SGM, with a universal step-size depending on the number of sample points.

Corollary 3 *Under Assumptions 1, 2 and 3, let $\zeta \geq 1/2$, $\delta \in]0,1[$, $b = 1$ and $\eta_t \simeq \frac{1}{m}$ for all $t \in [m^2]$. If $m \geq m_\delta$, then with probability at least $1-\delta$, there holds*

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})] - \inf_H \mathcal{E} \lesssim \left\{\left(\frac{m}{t}\right)^{2\zeta} + m^{-\frac{2\zeta+2}{2\zeta+\gamma}}\left(\frac{t}{m}\right)^2\right\} \cdot \log^2 m \log^2\frac{1}{\delta}, \quad \forall t \in [m^2], \tag{12}$$

and in particular,

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{T^*+1})] - \inf_H \mathcal{E} \lesssim m^{-\frac{2\zeta}{2\zeta+\gamma}} \log^2 m \log^2\frac{1}{\delta}, \tag{13}$$

where $T^* = \lceil m^{\frac{2\zeta+\gamma+1}{2\zeta+\gamma}} \rceil$. Here, m_δ is exactly the same as in Theorem 1.

Remark 3 *Ignoring the logarithmic term and letting $t = pm$, Eq. (12) becomes*

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{pm+1})] - \inf_H \mathcal{E} \lesssim p^{-2\zeta} + m^{-\frac{2\zeta+2}{2\zeta+\gamma}} p^2.$$

A smaller p may lead to a larger bias, while a larger p may lead to a larger sample error. From this point of view, p has a regularization effect.

The second corollary provides error bounds for SGM with a fixed mini-batch size and a fixed step-size (which depend on the number of sample points).

Corollary 4 *Under Assumptions 1, 2 and 3, let $\zeta \geq 1/2$, $\delta \in]0,1[$, $b = \lceil \sqrt{m} \rceil$ and $\eta_t \simeq \frac{1}{\sqrt{m}}$ for all $t \in [m^2]$. If $m \geq m_\delta$, then with probability at least $1-\delta$, there holds*

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})] - \inf_H \mathcal{E} \lesssim \left\{\left(\frac{\sqrt{m}}{t}\right)^{2\zeta} + m^{-\frac{2\zeta+2}{2\zeta+\gamma}}\left(\frac{t}{\sqrt{m}}\right)^2\right\} \log^2 m \log^2\frac{1}{\delta}, \quad \forall t \in [m^2], \tag{14}$$

and particularly,

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{T^*+1})] - \inf_H \mathcal{E} \lesssim m^{-\frac{2\zeta}{2\zeta+\gamma}} \log^2 m \log^2\frac{1}{\delta}, \tag{15}$$

where $T^* = \lceil m^{\frac{1}{2\zeta+\gamma}+\frac{1}{2}} \rceil$.

The above two corollaries follow from Theorem 1 with the simple observation that the dominating terms in (11) are the terms related to the bias and the sample variance, when a small step-size is chosen. The only free parameter in (12) and (14) is the number of iterations/passes. The ideal stopping rule is achieved by balancing the two terms related to the bias and the sample variance, showing the regularization effect of the number of passes. Since the ideal stopping rule depends on the unknown parameters ζ and γ, a hold-out cross-validation procedure is often used to tune the stopping rule in practice. Using an argument similar to that in Chapter 6 from (Steinwart and Christmann, 2008), it is possible to show that this procedure can achieve the same convergence rate.

We give some further remarks. First, the upper bound in (13) is optimal up to a logarithmic factor, in the sense that it matches the minimax lower rate in (Caponnetto and De Vito, 2007; Blanchard and Mücke, 2016). Second, according to Corollaries 3 and 4, $\frac{bT^*}{m} \simeq m^{\frac{1}{2\zeta+\gamma}}$ passes

over the data are needed to obtain optimal rates in both cases. Finally, in comparing the simple SGM and the mini-batch SGM, Corollaries 3 and 4 show that a larger step-size is allowed to use for the latter.

In the next result, both the step-size and the stopping rule are tuned to obtain optimal rates for simple SGM with multiple passes. In this case, the step-size and the number of iterations are the regularization parameters.

Corollary 5 *Under Assumptions 1, 2 and 3, let $\zeta \geq 1/2$, $\delta \in]0,1[$, $b = 1$ and $\eta_t \simeq m^{-\frac{2\zeta}{2\zeta+\gamma}}$ for all $t \in [m^2]$. If $m \geq m_\delta$, and $T^* = \lceil m^{\frac{2\zeta+1}{2\zeta+\gamma}} \rceil$, then (13) holds with probability at least $1-\delta$.*

The next corollary shows that for some suitable mini-batch sizes, optimal rates can be achieved with a constant step-size (which is nearly independent of the number of sample points) by early stopping.

Corollary 6 *Under Assumptions 1, 2 and 3, let $\zeta \geq 1/2$, $\delta \in]0,1[$, $b = \lceil m^{\frac{2\zeta}{2\zeta+\gamma}} \rceil$ and $\eta_t \simeq \frac{1}{\log m}$ for all $t \in [m]$. If $m \geq m_\delta$, and $T^* = \lceil m^{\frac{1}{2\zeta+\gamma}} \rceil$, then (13) holds with probability at least $1-\delta$.*

According to Corollaries 5 and 6, around $m^{\frac{1-\gamma}{2\zeta+\gamma}}$ passes over the data are needed to achieve the best performance in the above two strategies. In comparisons with Corollaries 3 and 4 where around $m^{\frac{\zeta+1}{2\zeta+\gamma}}$ passes are required, the latter seems to require fewer passes over the data. However, in this case, one might have to run the algorithms multiple times to tune the step-size, or the mini-batch size.

Remark 4 *1) If we make no assumption on the capacity, i.e., $\gamma = 1$, Corollary 5 recovers the result in (Ying and Pontil, 2008) for one pass SGM.*
2) If we make no assumption on the capacity and assume that $f_{\mathcal{H}} \in H_\rho$, from Corollaries 5 and 6, we see that the optimal convergence rate $O(m^{-1/2})$ can be achieved after one pass over the data in both of these two strategies. In this special case, Corollaries 5 and 6 recover the results for one pass SGM in, e.g., (Shamir and Zhang, 2013; Dekel et al., 2012).

The next result gives generalization error bounds for 'batch' SGM with a constant step-size (nearly independent of the number of sample points).

Corollary 7 *Under Assumptions 1, 2 and 3, let $\zeta \geq 1/2$, $\delta \in]0,1[$, $b = m$ and $\eta_t \simeq \frac{1}{\log m}$ for all $t \in [m]$. If $m \geq m_\delta$, and $T^* = \lceil m^{\frac{1}{2\zeta+\gamma}} \rceil$, then (13) holds with probability at least $1-\delta$.*

Theorem 1 and its corollaries give convergence results with respect to the target function values. In the next theorem and corollary, we will present convergence results in H-norm.

Theorem 2 *Under the assumptions of Theorem 1, the following holds with probability at least $1-\delta$: for all $t \in [T]$*

$$\mathbb{E}_{\mathbf{J}}[\|\omega_t - \omega^\dagger\|_H^2] \leq q_1(\eta t)^{1-2\zeta} + q_2 m^{-\frac{2\zeta-1}{2\zeta+\gamma}}(1 + m^{-\frac{1}{2\zeta+\gamma}}\eta t)^2 \log^2 T \log^2 \frac{1}{\delta} + q_3 \eta^2 t b^{-1}. \tag{16}$$

Here, q_1, q_2 and q_3 are positive constants depending on $\kappa^2, \|\mathcal{T}_\rho\|, M, v, \zeta, R, c_\gamma$, and γ (which can be given explicitly in the proof).

The proof of the above theorem is similar as that for Theorem 1, and will be given in Subsection 8. Again, the upper bound in (16) is composed of three terms related to bias, sample variance, and computational variance. Balancing these three terms leads to different choices on η, T, and b, as shown in the following corollary.

Corollary 8 *With the same assumptions and notations from any one of Corollaries 3 to 7, the following holds with probability at least $1-\delta$:*

$$\mathbb{E}_{\mathbf{J}}[\|\omega_{T^*+1} - \omega^\dagger\|_H^2] \lesssim m^{-\frac{2\zeta-1}{2\zeta+\gamma}} \log^2 m \log^2 \frac{1}{\delta}.$$

The convergence rate in the above corollary is optimal up to a logarithmic factor, as it matches the minimax rate shown in (Blanchard and Mücke, 2016).

In the next subsection, we will present convergence results in the non-attainable case, i.e., $\zeta < 1/2$.

3.4 Main Results for SGM: Non-attainable Case

Our main theorem in the non-attainable case is stated next, and provides error bounds for the studied algorithm. Here, we present results with a fixed step-size, whereas general results with a decaying step-size will be given in Section 7.

Theorem 3 *Under Assumptions 1, 2 and 3, let $\zeta \leq 1/2$, $\delta \in]0,1[$, $\eta_t = \eta\kappa^{-2}$ for all $t \in [T]$, with $0 < \eta \leq \frac{1}{8(\log T+1)}$. Then the following holds for all $t \in [T]$ with probability at least $1-\delta$:*
1) if $2\zeta + \gamma > 1$ and $m \geq m_\delta$, then

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})] - \inf_H \mathcal{E} \leq \left(q_1(\eta t)^{-2\zeta} + q_2 m^{-\frac{2\zeta}{2\zeta+\gamma}}\right)(1 \vee m^{-\frac{1}{2\zeta+\gamma}}\eta t)^3 \log^4 T \log^2 \frac{1}{\delta} + q_3 \eta b^{-1}(1 \vee m^{-\frac{1}{2\zeta+\gamma}}\eta t) \log T; \tag{17}$$

2) if $2\zeta + \gamma \leq 1$ and for some $\epsilon \in]0,1[$, $m \geq m_{\delta,\epsilon}$, then

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})] - \inf_H \mathcal{E} \leq \left(q_1(\eta t)^{-2\zeta} + q_2 m^{\gamma(1-\epsilon)-1}\right)(1 \vee \eta m^{\epsilon-1} t)^3 \log^4 T \log^2 \frac{1}{\delta} + q_3 \eta b^{-1}(1 \vee m^{\epsilon-1}\eta t) \log T.$$

Here, m_δ (or $m_{\delta,\epsilon}$), q_1, q_2 and q_3 are positive constants depending only on $\kappa^2, \|\mathcal{T}_\rho\|, M, v, \zeta, R, c_\gamma, \gamma$, $\|f_{\mathcal{H}}\|_\infty$, and m_δ (or $m_{\delta,\epsilon}$) also on δ (and ϵ).

The upper bounds in (11) (for the attainable case) and (17) (for the non-attainable case) are similar, whereas the latter has an extra logarithmic factor. Consequently, in the subsequent corollaries, we derive $O(m^{-\frac{2\zeta}{2\zeta+\gamma}} \log^4 m)$ for the non-attainable case. In comparison with that for the attainable case, the convergence rate for the non-attainable case has an extra $\log^2 m$ factor.

Similar to Corollaries 3 and 4, and as direct consequences of the above theorem, we have the following generalization error bounds for the studied algorithm with different choices of parameters in the non-attainable case.

Corollary 9 *Under Assumptions 1, 2 and 3, let $\zeta \leq 1/2$, $\delta \in]0,1[$, $b = 1$ and $\eta_t \simeq \frac{1}{m}$ for all $t \in [m^2]$. With probability at least $1-\delta$, the following holds:*
1) if $2\zeta + \gamma > 1$, $m \geq m_\delta$ and $T^* = \lceil m^{\frac{1+2\zeta+\gamma}{2\zeta+\gamma}} \rceil$, then

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{T^*+1})] - \inf_H \mathcal{E} \lesssim m^{-\frac{2\zeta}{2\zeta+\gamma}} \log^4 m \log^2 \frac{1}{\delta}; \tag{18}$$

2) if $2\zeta+\gamma \leq 1$, *and for some* $\epsilon \in]0,1[$, $m \geq m_{\delta,\epsilon}$, *and* $T^* = \lceil m^{2-\epsilon} \rceil$, *then*

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{T^*+1})] - \inf_{H} \mathcal{E} \lesssim m^{-2\zeta(1-\epsilon)} \log^4 m \log^2 \frac{1}{\delta}. \tag{19}$$

Here, m_δ *and* $m_{\delta,\epsilon}$ *are given by Theorem 3.*

Corollary 10 *Under Assumptions 1, 2 and 3, let* $\zeta \leq 1/2$, $\delta \in]0,1[$, $b \simeq \sqrt{m}$ *and* $\eta_t \simeq \frac{1}{\sqrt{m}}$ *for all* $t \in [m^2]$. *With probability at least* $1-\delta$, *there holds*
1) if $2\zeta+\gamma > 1$, $m \geq m_\delta$ *and* $T^* = \lceil m^{\frac{1}{2\zeta+\gamma}+\frac{1}{2}} \rceil$, *then (18) holds;*
2) if $2\zeta+\gamma \leq 1$, *for some* $\epsilon \in]0,1[$, $m \geq m_{\delta,\epsilon}$, *and* $T^* = \lceil m^{\frac{3}{2}-\epsilon} \rceil$, *then (19) holds.*

The convergence rates in the above corollaries, i.e., $m^{-\frac{2\zeta}{2\zeta+\gamma}}$ if $2\zeta+\gamma > 1$ or $m^{-2\zeta(1-\epsilon)}$ otherwise, match those in (Dieuleveut and Bach, 2016) for one pass SGM with averaging, up to a logarithmic factor. Also, in the capacity independent case, i.e., $\gamma = 1$, the convergence rates in the above corollary read as $m^{-\frac{2\zeta}{2\zeta+1}}$ (since $2\zeta+\gamma$ is always bigger than 1), which are exactly the same as those in (Ying and Pontil, 2008) for one pass SGM.

Similar results to Corollaries 5–7 can be also derived for the non-attainable case by applying Theorem 3. Refer to Appendix B for more details.

3.5 Main Results for Batch GM

In this subsection, we present convergence results for batch GM. As a byproduct of our proofs in this paper, we have the following convergence rates for batch GM.

Theorem 4 *Under Assumptions 1, 2 and 3, set* $\eta_t \simeq 1$, *for all* $t \in [m]$. *Let* $T^* = \lceil m^{\frac{1}{2\zeta+\gamma}} \rceil$ *if* $2\zeta+\gamma > 1$, *or* $T^* = \lceil m^{1-\epsilon} \rceil$ *with* $\epsilon \in]0,1[$ *otherwise. Then with probability at least* $1-\delta$ *(*$0 < \delta < 1$*), the following holds for the learning sequence generated by (10):*
1) if $\zeta > 1/2$ *and* $m \geq m_\delta$, *then*

$$\mathcal{E}(\nu_{T_*+1}) - \inf_{H} \mathcal{E} \lesssim m^{-\frac{2\zeta}{2\zeta+\gamma}} \log^2 m \log^2 \frac{1}{\delta};$$

2) if $\zeta \leq 1/2$, $2\zeta+\gamma > 1$ *and* $m \geq m_\delta$, *then*

$$\mathcal{E}(\nu_{T_*+1}) - \inf_{H} \mathcal{E} \lesssim m^{-\frac{2\zeta}{2\zeta+\gamma}} \log^4 m \log^2 \frac{1}{\delta};$$

3) if $2\zeta+\gamma \leq 1$ *and* $m \geq m_{\delta,\epsilon}$, *then*

$$\mathcal{E}(\nu_{T_*+1}) - \inf_{H} \mathcal{E} \lesssim m^{-2\zeta(1-\epsilon)} \log^4 m \log^2 \frac{1}{\delta}.$$

Here, m_δ *(or* $m_{\delta,\epsilon}$*), and all the constants in the upper bounds are positive and depend only on* $\kappa^2, \|\mathcal{T}_\rho\|, M, v, \zeta, R, c_\gamma, \gamma, \|f_{\mathcal{H}}\|_\infty$, *and* m_δ *(or* $m_{\delta,\epsilon}$*) also on* δ *(and* ϵ*).*

3.6 Discussions

We must compare our results with previous works. For non-parametric regression with the square loss, one pass SGM has been studied in, e.g., (Ying and Pontil, 2008; Shamir and Zhang, 2013; Tarres and Yao, 2014; Dieuleveut and Bach, 2016). In particular, Ying and Pontil (2008) proved

capacity independent rate of order $O(m^{-\frac{2\zeta}{2\zeta+1}} \log m)$ with a fixed step-size $\eta \simeq m^{-\frac{2\zeta}{2\zeta+1}}$, and Dieuleveut and Bach (2016) derived capacity dependent error bounds of order $O(m^{-\frac{2\min(\zeta,1)}{2\min(\zeta,1)+\gamma}})$ (when $2\zeta+\gamma > 1$) for the average. Note also that a regularized version of SGM has been studied in (Tarres and Yao, 2014), where the derived convergence rate is of order $O(m^{-\frac{2\zeta}{2\zeta+1}})$ assuming that $\zeta \in [\frac{1}{2}, 1]$. In comparison with these existing convergence rates, our rates from (13) are comparable, either involving the capacity condition, or allowing a broader regularity parameter ζ (which thus improves the rates). For finite dimensional cases, it has been shown in (Bach and Moulines, 2013) that one pass SGM with averaging with a constant step-size achieves the optimal convergence rate of $O(d/m)$. In comparisons, our results for multi-pass SGM with a smaller step-size seems to be suboptimal in the computational complexity, as we need m passes over the data to achieve the same rate. The reason for this may arise from "the computational error" that will be introduced later, or the fact that we do not consider an averaging step as done in (Bach and Moulines, 2013). We hope that in the future by considering a larger step-size and averaging, one can reduce the computational complexity of multi-pass SGM while achieving the same rate.

More recently, Rosasco and Villa (2015) studied multiple passes SGM with a fixed ordering at each pass, also called incremental gradient method. Making no assumption on the capacity, rates of order $O(m^{-\frac{\zeta}{\zeta+1}})$ (in $L^2(H, \rho_X)$-norm) with a universal step-size $\eta \simeq 1/m$ are derived. In comparisons, Corollary 3 achieves better rates, while considering the capacity assumption. Note also that Rosasco and Villa (2015) proved sharp rate in H-norm for $\zeta \geq 1/2$ in the capacity independent case. In comparisons, we derive optimal capacity-dependent rate, considering mini-batches.

The idea of using mini-batches (and parallel implements) to speed up SGM in a general stochastic optimization setting can be found, e.g., in (Shalev-Shwartz et al., 2011; Dekel et al., 2012; Sra et al., 2012; Ng, 2016). Our theoretical findings, especially the interplay between the mini-batch size and the step-size, can give further insights on parallelization learning. Besides, it has been shown in (Cotter et al., 2011; Dekel et al., 2012) that for one pass mini-batch SGM with a fixed step-size $\eta \simeq b/\sqrt{m}$ and a smooth loss function, assuming the existence of at least one solution in the hypothesis space for the expected risk minimization, the convergence rate is of order $O(\sqrt{1/m} + b/m)$ by considering an averaging scheme. When adapting to the learning setting we consider, this reads as that if $f_{\mathcal{H}} \in H_\rho$, i.e., $\zeta = 1/2$, the convergence rate for the average is $O(\sqrt{1/m} + b/m)$. Note that, $f_{\mathcal{H}}$ does not necessarily belong to H_ρ in general. Also, our derived convergence rate from Corollary 4 is better, when the regularity parameter ζ is greater than $1/2$, or γ is smaller than 1.

For batch GM in the attainable case, convergent results with optimal rates have been derived in, e.g, (Bauer et al., 2007; Caponnetto and Yao, 2010; Blanchard and Mücke, 2016; Dicker et al., 2017). In particular, Bauer et al. (2007) proved convergence rates $O(m^{-\frac{2\zeta}{2\zeta+1}})$ without considering Assumption 3, and Caponnetto and Yao (2010) derived convergence rates $O(m^{-\frac{2\zeta}{2\zeta+\gamma}})$. For the non-attainable case, convergent results with suboptimal rates $O(m^{\frac{-2\zeta}{2\zeta+2}})$ can be found in (Yao et al., 2007), and to the best of our knowledge, the only result with optimal rate $O(m^{\frac{-2\zeta}{2\zeta+\gamma}})$ is the one derived by Caponnetto and Yao (2010), but the result requires extra unlabeled data. In contrast, Theorem 4 of this paper does not require any extra unlabeled data, while achieving the same optimal rates (up to a logarithmic factor). To the best of our knowledge, Theorem 4 may be the first optimal result in the non-attainable case for batch GM.

We end this discussion with some further comments on batch GM and simple SGM. First, according to Corollaries 1 and 2, it seems that both simple SGM (with step-size $\eta_t \simeq m^{-1}$) and batch GM (with step-size $\eta_t \simeq 1$) have the same computational complexities (which are related to the number of passes) and the same orders of upper bounds. However, there is a subtle difference between these two algorithms. As we see from (22) in the coming subsection, every m iterations of simple SGM (with step-size $\eta_t \simeq m^{-1}$) corresponds to one iteration of batch GM (with step-size $\eta_t \simeq 1$). In this sense, SGM discretizes and refines the regularization path of batch GM, which thus may lead to smaller generalization errors. This phenomenon can be further understood by comparing our derived bounds, (11) and (73), for these two algorithms. Indeed, if one can ignore the computational error, one can easily show that the minimization (over t) of right hand-side of (11) with $\eta \simeq m^{-1}$ is always smaller than that of (73) with $\eta \simeq 1$. At last, by Corollary 6, using a larger step-size for SGM allows one to stop earlier (while sharing the same optimal rates), which thus reduces the computational complexity. This suggests that SGM may have some computational advantage over batch GM.

3.7 Proof Sketch (Error Decomposition)

The key to our proof is a novel error decomposition, which may be also used in analysing other learning algorithms. One may also use the approach in (Bousquet and Bottou, 2008; Lin et al., 2016b,a) which is based on the following error decomposition,

$$\mathbb{E}\mathcal{E}(\omega_t) - \inf_H \mathcal{E} = [\mathbb{E}(\mathcal{E}(\omega_t) - \mathcal{E}_{\mathbf{z}}(\omega_t)) + \mathbb{E}\mathcal{E}_{\mathbf{z}}(\tilde{\omega}) - \mathcal{E}(\tilde{\omega})] + \mathbb{E}(\mathcal{E}_{\mathbf{z}}(\omega_t) - \mathcal{E}_{\mathbf{z}}(\tilde{\omega})) + \mathcal{E}(\tilde{\omega}) - \inf_H \mathcal{E},$$

where $\tilde{\omega} \in H$ is some suitably intermediate element and $\mathcal{E}_{\mathbf{z}}$ denotes the empirical risk over $\mathbf{z}$, i.e.,

$$\mathcal{E}_{\mathbf{z}}(\cdot) = \frac{1}{m}\sum_{i=1}^{m} (\langle \cdot, x_i\rangle - y_i)^2 . \tag{20}$$

However, one can only derive a sub-optimal convergence rate, since the proof procedure involves upper bounding the learning sequence to estimate the sample error (the first term of right-hand side). Also, in this case, the 'regularity' of the regression function can not be fully utilized for estimating the bias (the last term). Thanks to the property of squares loss, we can exploit a different error decomposition leading to better results.

To describe the decomposition, we need to introduce two sequences. The *population iteration* is defined by $\mu_1 = 0$ and

$$\mu_{t+1} = \mu_t - \eta_t \int_X (\langle \mu_t, x\rangle_H - f_\rho(x))x d\rho_X(x), \qquad t = 1, \ldots, T. \tag{21}$$

The above iterated procedure is ideal and can not be implemented in practice, since the distribution ρ_X is unknown in general. Replacing ρ_X by the empirical measure and $f_\rho(x_i)$ by y_i, we derive the *sample iteration* (associated with the sample $\mathbf{z}$), i.e., (10). Clearly, μ_t is deterministic and ν_t is a H-valued random variable depending on $\mathbf{z}$. Given the sample $\mathbf{z}$, the sequence $\{\nu_t\}_t$ has a natural relationship with the learning sequence $\{\omega_t\}_t$, since

$$\mathbb{E}_{\mathbf{J}}[\omega_t] = \nu_t. \tag{22}$$

Indeed, taking the expectation with respect to $\mathbf{J}_t$ on both sides of (4), and noting that ω_t depends only on $\mathbf{J}_1, \cdots, \mathbf{J}_{t-1}$ (given any $\mathbf{z}$), one has

$$\mathbb{E}_{\mathbf{J}_t}[\omega_{t+1}] = \omega_t - \eta_t \frac{1}{m}\sum_{i=1}^{m} (\langle \omega_t, x_i\rangle_H - y_i)x_i,$$

and thus,

$$\mathbb{E}_{\mathbf{J}}[\omega_{t+1}] = \mathbb{E}_{\mathbf{J}}[\omega_t] - \eta_t \frac{1}{m}\sum_{i=1}^{m} (\langle \mathbb{E}_{\mathbf{J}}[\omega_t], x_i\rangle_H - y_i)x_i, \quad t = 1, \ldots, T,$$

which satisfies the iterative relationship given in (10). By an induction argument, (22) can then be proved.

Let $\mathcal{S}_\rho : H \to L^2(H, \rho_X)$ be the linear map defined by $(\mathcal{S}_\rho \omega)(x) = \langle \omega, x\rangle_H, \forall \omega, x \in H$. We have the following error decomposition.

Proposition 1 *We have*

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_t)] - \inf_H \mathcal{E} \le 2\|\mathcal{S}_\rho \mu_t - f_{\mathcal{H}}\|_\rho^2 + 2\|\mathcal{S}_\rho \nu_t - \mathcal{S}_\rho \mu_t\|_\rho^2 + \mathbb{E}_{\mathbf{J}}[\|\mathcal{S}_\rho \omega_t - \mathcal{S}_\rho \nu_t\|_\rho^2]. \tag{23}$$

Proof For any $\omega \in H$, we have (Rosasco and Villa, 2015)

$$\mathcal{E}(\omega) - \inf_{\omega \in H} \mathcal{E}(\omega) = \|\mathcal{S}_\rho \omega - f_{\mathcal{H}}\|_\rho^2.$$

Thus, $\mathcal{E}(\omega_t) - \inf_H \mathcal{E} = \|\mathcal{S}_\rho \omega_t - f_{\mathcal{H}}\|_\rho^2$, and

$$\begin{aligned}\mathbb{E}_{\mathbf{J}}[\|\mathcal{S}_\rho \omega_t - f_{\mathcal{H}}\|_\rho^2] &= \mathbb{E}_{\mathbf{J}}[\|\mathcal{S}_\rho \omega_t - \mathcal{S}_\rho \nu_t + \mathcal{S}_\rho \nu_t - f_{\mathcal{H}}\|_\rho^2] \\ &= \mathbb{E}_{\mathbf{J}}[\|\mathcal{S}_\rho \omega_t - \mathcal{S}_\rho \nu_t\|_\rho^2 + \|\mathcal{S}_\rho \nu_t - f_{\mathcal{H}}\|_\rho^2] + 2\mathbb{E}_{\mathbf{J}}\langle \mathcal{S}_\rho \omega_t - \mathcal{S}_\rho \nu_t, \mathcal{S}_\rho \nu_t - f_{\mathcal{H}}\rangle_\rho.\end{aligned}$$

Using (22) in the above equality, we get,

$$\mathbb{E}_{\mathbf{J}}[\|\mathcal{S}_\rho \omega_t - f_{\mathcal{H}}\|_\rho^2] = \mathbb{E}_{\mathbf{J}}[\|\mathcal{S}_\rho \omega_t - \mathcal{S}_\rho \nu_t\|_\rho^2 + \|\mathcal{S}_\rho \nu_t - f_{\mathcal{H}}\|_\rho^2].$$

The proof is finished by considering,

$$\|\mathcal{S}_\rho \nu_t - f_{\mathcal{H}}\|_\rho^2 = \|\mathcal{S}_\rho \nu_t - \mathcal{S}_\rho \mu_t + \mathcal{S}_\rho \mu_t - f_{\mathcal{H}}\|_\rho^2 \le 2\|\mathcal{S}_\rho \nu_t - \mathcal{S}_\rho \mu_t\|_\rho^2 + 2\|\mathcal{S}_\rho \mu_t - \mathcal{S}_\rho f_{\mathcal{H}}\|_\rho^2.$$

■

There are three terms in the upper bound of the error decomposition (23). We refer to the deterministic term $\|\mathcal{S}_\rho \mu_t - f_{\mathcal{H}}\|_\rho^2$ as the *bias*, the term $\|\mathcal{S}_\rho \nu_t - \mathcal{S}_\rho \mu_t\|_\rho^2$ depending on $\mathbf{z}$ as the *sample variance*, and $\mathbb{E}_{\mathbf{J}}[\|\mathcal{S}_\rho \omega_t - \mathcal{S}_\rho \nu_t\|_\rho^2]$ as the *computational variance*. The bias term, which is deterministic, has been well studied in the literature, see e.g., (Yao et al., 2007) and also (Rosasco and Villa, 2015). The main novelties of this paper are the estimate of the sample and computational variances and the difficult part is the estimate of the computational variances. The proof of these results is quite lengthy and makes use of some ideas from (Yao et al., 2007; Smale and Zhou, 2007; Bauer et al., 2007; Ying and Pontil, 2008; Tarres and Yao, 2014; Rudi et al., 2015). These three error terms will be estimated in Sections 5 and 6. The bounds in Theorems 1 and 3 thus follow plugging these estimations in the error decomposition, see Section 7 for more details. The proof for Theorem 2 is similar, see Section 8 for the details.

4. Preliminary Analysis

In this section, we introduce some notation and preliminary lemmas that are necessary to our proofs.

4.1 Notation

We first introduce some notations. For $t \in \mathbb{N}$, $\Pi_{t+1}^{T}(L) = \prod_{k=t+1}^{T}(I - \eta_k L)$ for $t \in [T-1]$ and $\Pi_{T+1}^{T}(L) = I$, for any operator $L : \mathcal{H} \to \mathcal{H}$, where $\mathcal{H}$ is a Hilbert space and I denotes the identity operator on $\mathcal{H}$. $\mathbb{E}[\xi]$ denotes the expectation of a random variable ξ. For a given bounded operator $L : L^2(H, \rho_X) \to H$, $\|L\|$ denotes the operator norm of L, i.e., $\|L\| = \sup_{f \in L^2(H,\rho_X), \|f\|_\rho = 1} \|Lf\|_H$. We will use the conventional notations on summation and production: $\prod_{i=t+1}^{t} = 1$ and $\sum_{i=t+1}^{t} = 0$.

We next introduce some auxiliary operators. Let $\mathcal{S}_\rho : H \to L^2(H, \rho_X)$ be the linear map $\omega \to \langle \omega, \cdot \rangle_H$, which is bounded by κ under Assumption (3). Furthermore, we consider the adjoint operator $\mathcal{S}_\rho^* : L^2(H, \rho_X) \to H$, the covariance operator $\mathcal{T}_\rho : H \to H$ given by $\mathcal{T}_\rho = \mathcal{S}_\rho^* \mathcal{S}_\rho$, and the operator $\mathcal{L}_\rho : L^2(H, \rho_X) \to L^2(H, \rho_X)$ given by $\mathcal{S}_\rho \mathcal{S}_\rho^*$. It can be easily proved that $\mathcal{S}_\rho^* g = \int_X x g(x) d\rho_X(x)$ and $\mathcal{T}_\rho = \int_X \langle \cdot, x \rangle_H x d\rho_X(x)$. The operators $\mathcal{T}_\rho$ and $\mathcal{L}_\rho$ can be proved to be positive trace class operators (and hence compact). For any $\omega \in H$, it is easy to prove the following isometry property (Steinwart and Christmann, 2008)

$$\|\mathcal{S}_\rho \omega\|_\rho = \|\sqrt{\mathcal{T}_\rho}\omega\|_H. \tag{24}$$

We define the sampling operator $\mathcal{S}_{\mathbf{x}} : H \to \mathbb{R}^m$ by $(\mathcal{S}_{\mathbf{x}}\omega)_i = \langle \omega, x_i \rangle_H$, $i \in [m]$, where the norm $\|\cdot\|_{\mathbb{R}^m}$ in $\mathbb{R}^m$ is the Euclidean norm times $1/\sqrt{m}$. Its adjoint operator $\mathcal{S}_{\mathbf{x}}^* : \mathbb{R}^m \to H$, defined by $\langle \mathcal{S}_{\mathbf{x}}^* \mathbf{y}, \omega \rangle_H = \langle \mathbf{y}, \mathcal{S}_{\mathbf{x}} \omega \rangle_{\mathbb{R}^m}$ for $\mathbf{y} \in \mathbb{R}^m$ is thus given by $\mathcal{S}_{\mathbf{x}}^* \mathbf{y} = \frac{1}{m} \sum_{i=1}^m y_i x_i$. Moreover, we can define the empirical covariance operator $\mathcal{T}_{\mathbf{x}} : H \to H$ such that $\mathcal{T}_{\mathbf{x}} = \mathcal{S}_{\mathbf{x}}^* \mathcal{S}_{\mathbf{x}}$. Obviously,

$$\mathcal{T}_{\mathbf{x}} = \frac{1}{m} \sum_{i=1}^{m} \langle \cdot, x_i \rangle_H x_i.$$

With these notations, (21) and (10) can be rewritten as

$$\mu_{t+1} = \mu_t - \eta_t (\mathcal{T}_\rho \mu_t - \mathcal{S}_\rho^* f_\rho), \qquad t = 1, \ldots, T, \tag{25}$$

and

$$\nu_{t+1} = \nu_t - \eta_t (\mathcal{T}_{\mathbf{x}} \nu_t - \mathcal{S}_{\mathbf{x}}^* \mathbf{y}), \qquad t = 1, \ldots, T, \tag{26}$$

respectively.

Using the projection theorem, one can prove that

$$\mathcal{S}_\rho^* f_\rho = \mathcal{S}_\rho^* f_{\mathcal{H}}. \tag{27}$$

Indeed, since $f_{\mathcal{H}}$ is the projection of the regression function f_ρ onto the closure of H_ρ in $L^2(H, \rho_X)$, according to the projection theorem, one has

$$\langle f_{\mathcal{H}} - f_\rho, \mathcal{S}_\rho \omega \rangle_\rho = 0, \qquad \forall \omega \in H,$$

which can be written as

$$\langle \mathcal{S}_\rho^* f_{\mathcal{H}} - \mathcal{S}_\rho^* f_\rho, \omega \rangle_H = 0, \qquad \forall \omega \in H,$$

and thus leading to (27).

4.2 Concentration Inequality

We need the following concentration result for Hilbert space valued random variable used in (Caponnetto and De Vito, 2007) and based on the results in (Pinelis and Sakhanenko, 1986).

Lemma 11 *Let $w_1, \cdots, w_m$ be i.i.d random variables in a Hilbert space with norm $\|\cdot\|$. Suppose that there are two positive constants B and σ^2 such that*

$$\mathbb{E}[\|w_1 - \mathbb{E}[w_1]\|^l] \leq \frac{1}{2} l! B^{l-2} \sigma^2, \quad \forall l \geq 2. \tag{28}$$

Then for any $0 < \delta < 1$, the following holds with probability at least $1 - \delta$,

$$\left\| \frac{1}{m} \sum_{k=1}^{m} w_m - \mathbb{E}[w_1] \right\| \leq 2 \left(\frac{B}{m} + \frac{\sigma}{\sqrt{m}} \right) \log \frac{2}{\delta}.$$

In particular, (28) holds if

$$\|w_1\| \leq B/2 \quad a.s., \quad \text{and} \quad \mathbb{E}[\|w_1\|^2] \leq \sigma^2. \tag{29}$$

4.3 Basic Estimates

Finally, we introduce the following three basic estimates, whose proofs can be found in Appendix C.

Lemma 12 *Let $\theta \in [0, 1[$, and $t \in \mathbb{N}$. Then*

$$\frac{t^{1-\theta}}{2} \leq \sum_{k=1}^{t} k^{-\theta} \leq \frac{t^{1-\theta}}{1-\theta}.$$

Lemma 13 *Let $\theta \in \mathbb{R}$ and $t \in \mathbb{N}$. Then*

$$\sum_{k=1}^{t} k^{-\theta} \leq t^{\max(1-\theta, 0)} (1 + \log t).$$

Lemma 14 *Let $q \in \mathbb{R}$ and $t \in \mathbb{N}$ with $t \geq 3$. Then*

$$\sum_{k=1}^{t-1} \frac{1}{t-k} k^{-q} \leq 2 t^{-\min(q,1)} (1 + \log t).$$

In the next sections, we begin prooving the main results. The proofs are quite lengthy and they are divided into several steps. For the ease of readability, we list some of the notations and definitions in Appendix D. We also remark that we are particularly interested in developing error bounds in terms of the stepsize η_t ($= \eta_1 t^{-\theta}$), the number of iterations t or T, the 'regularization' parameter $\lambda > 0$, the sample size m, the minibatch size b, and the failing profitability δ. Other parameters such as $\kappa^2, \|\mathcal{T}_\rho\|, M, v, R, c_\gamma$ and $\|f_{\mathcal{H}}\|_\infty$ can be always viewed as some constants, which are less important in our error bounds.

5. Estimating Bias and Sample Variance

In this section, we estimate the bias and the sample variance.

5.1 Bias

In this subsection, we develop upper bounds for the bias, i.e., $\|\mathcal{S}_\rho \mu_t - f_{\mathcal{H}}\|_\rho^2$. Towards this end, we introduce the following lemma, whose proof borrows idea from (Ying and Pontil, 2008; Tarres and Yao, 2014).

Lemma 15 *Let L be a compact, positive operator on a separable Hilbert space H. Assume that $\eta_1\|L\| \leq 1$. Then for $t \in \mathbb{N}$ and any non-negative integer $k \leq t-1$,*

$$\|\Pi_{k+1}^t(L)L^\zeta\| \leq \left(\frac{\zeta}{\mathrm{e}\sum_{j=k+1}^t \eta_j}\right)^\zeta. \tag{30}$$

Proof Let $\{\sigma_i\}$ be the sequence of eigenvalues of L. We have

$$\|\Pi_{k+1}^t(L)L^\zeta\| = \sup_i \prod_{l=k+1}^t (1-\eta_l\sigma_i)\sigma_i^\zeta.$$

Using the basic inequality

$$1 + x \leq \mathrm{e}^x \qquad \text{for all } x \geq -1, \tag{31}$$

with $\eta_l\|L\| \leq 1$, we get

$$\begin{aligned}\|\Pi_{k+1}^t(L)L^\zeta\| &\leq \sup_i \exp\left\{-\sigma_i \sum_{l=k+1}^t \eta_l\right\}\sigma_i^\zeta \\ &\leq \sup_{x\geq 0}\exp\left\{-x\sum_{l=k+1}^t \eta_l\right\}x^\zeta.\end{aligned}$$

The maximum of the function $g(x) = \mathrm{e}^{-cx}x^\zeta$(with $c>0$) over $\mathbb{R}_+$ is achieved at $x_{\max} = \zeta/c$, and thus

$$\sup_{x\geq 0}\mathrm{e}^{-cx}x^\zeta = \left(\frac{\zeta}{\mathrm{e}c}\right)^\zeta. \tag{32}$$

Using this inequality, one can get the desired result (30). ∎

With the above lemma and Lemma 12, we can derive the following result for the bias.

Proposition 2 *Under Assumption 2, let $\eta_1\kappa^2 \leq 1$. Then, for any $t \in \mathbb{N}$,*

$$\|\mathcal{S}_\rho\mu_{t+1} - f_{\mathcal{H}}\|_\rho \leq R\left(\frac{\zeta}{2\sum_{j=1}^t \eta_j}\right)^\zeta. \tag{33}$$

In particular, if $\eta_t = \eta t^{-\theta}$ for all $t \in \mathbb{N}$, with $\eta \in]0, \kappa^{-2}]$ and $\theta \in [0,1[$, then

$$\|\mathcal{S}_\rho\mu_{t+1} - f_{\mathcal{H}}\|_\rho \leq R\zeta^\zeta\eta^{-\zeta}t^{(\theta-1)\zeta}. \tag{34}$$

The above result is essentially proved in (Yao et al., 2007), see also (Rosasco and Villa, 2015) when step-size is fixed. For the sake of completeness, we provide a proof in Appendix C. The following lemma gives upper bounds for the sequence $\{\mu_t\}_{t\in\mathbb{N}}$ in H-norm. It will be used for the estimation on the sample variance in the next section.

Lemma 16 *Under Assumption 2, let $\eta_1\kappa^2 \leq 1$. The following holds for all $t \in \mathbb{N}$:*
1) If $\zeta \geq 1/2$,

$$\|\mu_{t+1}\|_H \leq R\kappa^{2\zeta-1}. \tag{35}$$

2) If $\zeta \in]0, 1/2]$,

$$\|\mu_{t+1}\|_H \leq R\left\{\kappa^{2\zeta-1} \vee \left(\sum_{k=1}^t \eta_k\right)^{\frac{1}{2}-\zeta}\right\}. \tag{36}$$

Proof The proof can be found in Appendix C. The proof for a fixed step-size (i.e., $\eta_t = \eta$ for all t) can be also found in (Rosasco and Villa, 2015). For a general step-size, the proof is similar. Note also that our proof for the non-attainable case is simpler than that in (Rosasco and Villa, 2015). ∎

5.2 Sample Variance

In this subsection, we estimate the sample variance, i.e., $\mathbb{E}[\|\mathcal{S}_\rho\mu_t - \mathcal{S}_\rho\nu_t\|_\rho^2]$. Towards this end, we need some preliminary analysis. We first introduce the following key inequality, which also provides the basic idea on estimating $\mathbb{E}[\|\mathcal{S}_\rho\mu_t - \mathcal{S}_\rho\nu_t\|_\rho^2]$.

Lemma 17 *For all $t \in [T]$, we have*

$$\|\mathcal{S}_\rho\nu_{t+1} - \mathcal{S}_\rho\mu_{t+1}\|_\rho \leq \sum_{k=1}^t \eta_k \left\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\right\|_H, \tag{37}$$

where

$$N_k = (\mathcal{T}_\rho\mu_k - \mathcal{S}_\rho^* f_\rho) - (\mathcal{T}_{\mathbf{x}}\mu_k - \mathcal{S}_{\mathbf{x}}^*\mathbf{y}), \qquad \forall k \in [T]. \tag{38}$$

Proof Since ν_{t+1} and μ_{t+1} are given by (26) and (25), respectively,

$$\begin{aligned}\nu_{t+1} - \mu_{t+1} &= \nu_t - \mu_t + \eta_t\left\{(\mathcal{T}_\rho\mu_t - \mathcal{S}_\rho^* f_\rho) - (\mathcal{T}_{\mathbf{x}}\nu_t - \mathcal{S}_{\mathbf{x}}^*\mathbf{y})\right\} \\ &= (I - \eta_t\mathcal{T}_{\mathbf{x}})(\nu_t - \mu_t) + \eta_t\left\{(\mathcal{T}_\rho\mu_t - \mathcal{S}_\rho^* f_\rho) - (\mathcal{T}_{\mathbf{x}}\mu_t - \mathcal{S}_{\mathbf{x}}^*\mathbf{y})\right\},\end{aligned}$$

which is exactly

$$\nu_{t+1} - \mu_{t+1} = (I - \eta_t\mathcal{T}_{\mathbf{x}})(\nu_t - \mu_t) + \eta_t N_t.$$

Applying this relationship iteratively, with $\nu_1 = \mu_1 = 0$,

$$\nu_{t+1} - \mu_{t+1} = \Pi_1^t(\mathcal{T}_{\mathbf{x}})(\nu_1 - \mu_1) + \sum_{k=1}^t \eta_k\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k = \sum_{k=1}^t \eta_k\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k. \tag{39}$$

By (24), we have

$$\|\mathcal{S}_\rho\nu_{t+1} - \mathcal{S}_\rho\mu_{t+1}\|_\rho = \left\|\sum_{k=1}^t \eta_k\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\right\|_H,$$

which leads to the desired result (37). ∎

The above lemma shows that in order to upper bound $\mathbb{E}[\|\mathcal{S}_\rho\mu_t - \mathcal{S}_\rho\nu_t\|_\rho^2]$, one may only need to

bound $\left\| \mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}}) N_k \right\|_H$. A detailed look at this latter term indicates that one may analyze the terms $\mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})$ and N_k separately, since $\mathbb{E}_{\mathbf{z}}[N_k] = 0$ and the properties of the deterministic sequence $\{\mu_k\}_k$ have been derived in Section 5.1. Moreover, to exploit the capacity condition from Assumption 3, we estimate $\|(\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}} N_k\|_H$ (with $\lambda > 0$ properly chosen later), rather than $\|N_k\|_H$, as follows.

Lemma 18 *Under Assumptions 1, 2 and 3, let $\{N_t\}_t$ be as in (38). Then for any fixed $\lambda > 0$, and $T \geq 2$,*
1) if $\zeta \geq 1/2$, with probability at least $1 - \delta_1$, the following holds for all $k \in \mathbb{N}$:

$$\|(\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}} N_k\|_H \leq 4(R\kappa^{2\zeta} + \sqrt{M}) \left(\frac{\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{2\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}} \right) \log \frac{4}{\delta_1}. \tag{40}$$

2) if $\zeta \in]0, 1/2[$, with probability at least $1 - \delta_1$, the following holds for all $k \in [T]$:

$$\|(\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}} N_k\|_H \leq 2 \left(3\|f_{\mathcal{H}}\|_\infty + 2\sqrt{M} + \kappa R \right) \left(\frac{\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{2\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}} \right) \log \frac{3T}{\delta_1} + \frac{2\kappa^2 R \left(\sum_{i=1}^k \eta_i \right)^{\frac{1}{2}-\zeta}}{m\sqrt{\lambda}} \log \frac{3T}{\delta_1} + \frac{2\kappa R}{\sqrt{m\lambda}} \left(\frac{1}{\sum_{i=1}^k \eta_i} \right)^\zeta \log \frac{3T}{\delta_1}. \tag{41}$$

Proof We will apply Bernstein inequality from Lemma 11 to prove the result.
Attainable Case: $\zeta \geq 1/2$. See Appendix C for the proof.
Non-attainable case: $0 < \zeta < 1/2$.
Let $w_i = (f_{\mathcal{H}}(x_i) - y_i)(\mathcal{T}_\rho + \lambda)^{-1/2} x_i$, for all $i \in [m]$. Noting that by (27), and taking the expectation with respect to the random variable (x, y) (from the distribution ρ),

$$\mathbb{E}[\omega] = \mathbb{E}[(f_{\mathcal{H}}(x) - f_\rho(x))(\mathcal{T}_\rho + \lambda)^{-1/2} x] = 0.$$

Applying Hölder's inequality, for any $l \geq 2$,

$$\mathbb{E}[\|w - \mathbb{E}[w]\|_H^l] = \mathbb{E}[\|w\|_H^l] \leq 2^{l-1} \mathbb{E}[(|f_{\mathcal{H}}(x)|^l + |y|^l)\|(\mathcal{T}_\rho + \lambda)^{-1/2} x\|_H^l]$$
$$\leq 2^{l-1} \int_X (\|f_{\mathcal{H}}\|_\infty^l + \int_Y |y|^l d\rho(y|x)) \|(\mathcal{T}_\rho + \lambda)^{-1/2} x\|_H^l d\rho_X(x).$$

Using Cauchy-Schwarz's inequality and Assumption 1 which implies,

$$\int_Y y^l d\rho(y|x) \leq \left(\int_Y |y|^{2l} d\rho(y|x) \right)^{\frac{1}{2}} \leq \sqrt{l! M^l v} \leq l! (\sqrt{M})^l \sqrt{v}, \tag{42}$$

we get

$$\mathbb{E}[\|w - \mathbb{E}[w]\|_H^l] \leq 2^{l-1} (\|f_{\mathcal{H}}\|_\infty^l + l! (\sqrt{M})^l \sqrt{v}) \int_X \|(\mathcal{T}_\rho + \lambda)^{-1/2} x\|_H^l d\rho_X(x). \tag{43}$$

By Assumption (3),

$$\|(\mathcal{T}_\rho + \lambda I)^{-\frac{1}{2}} x\|_H \leq \frac{\|x\|_H}{\sqrt{\lambda}} \leq \frac{\kappa}{\sqrt{\lambda}}. \tag{44}$$

Besides, using the fact that $\mathbb{E}[\|\xi\|_H^2] = \mathbb{E}[\mathrm{tr}(\xi \otimes \xi)] = \mathrm{tr}(\mathbb{E}[\xi \otimes \xi])$ and $\mathbb{E}[x \otimes x] = \mathcal{T}_\rho$, we know that

$$\int_X \|(\mathcal{T}_\rho + \lambda I)^{-\frac{1}{2}} x\|_H^2 d\rho_X(x) = \mathrm{tr}((\mathcal{T}_\rho + \lambda I)^{-\frac{1}{2}} \mathcal{T}_\rho (\mathcal{T}_\rho + \lambda I)^{-\frac{1}{2}}) = \mathrm{tr}((\mathcal{T}_\rho + \lambda I)^{-1} \mathcal{T}_\rho),$$

and as a result of the above and Assumption 3,

$$\int_X \|(\mathcal{T}_\rho + \lambda I)^{-\frac{1}{2}} x\|_H^2 d\rho_X(x) \leq c_\gamma \lambda^{-\gamma}.$$

It thus follows that

$$\int_X \|(\mathcal{T}_\rho + \lambda)^{-1/2} x\|_H^l d\rho_X(x) \leq \left(\frac{\kappa}{\sqrt{\lambda}} \right)^{l-2} \int_X \|(\mathcal{T}_\rho + \lambda)^{-1/2} x\|_H^2 d\rho_X(x) \leq \left(\frac{\kappa}{\sqrt{\lambda}} \right)^{l-2} c_\gamma \lambda^{-\gamma}. \tag{45}$$

Plugging the above inequality into (43),

$$\begin{aligned} \mathbb{E}[\|w - \mathbb{E}[w]\|_H^l] &\leq 2^{l-1} (\|f_{\mathcal{H}}\|_\infty^l + l! (\sqrt{M})^l \sqrt{v}) \left(\frac{\kappa}{\sqrt{\lambda}} \right)^{l-2} c_\gamma \lambda^{-\gamma} \\ &\leq \frac{1}{2} l! \left(\frac{2\kappa(\|f_{\mathcal{H}}\|_\infty + \sqrt{M})}{\sqrt{\lambda}} \right)^{l-2} 4 c_\gamma \sqrt{v} (\|f_{\mathcal{H}}\|_\infty + \sqrt{M})^2 \lambda^{-\gamma}. \end{aligned}$$

Therefore, using Lemma 11, we get that with probability at least $1 - \delta$,

$$\left\| (\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}} \frac{1}{m} \sum_{i=1}^m (f_{\mathcal{H}}(x_i) - y_i) x_i \right\|_H = \left\| \frac{1}{m} \sum_{i=1}^m (\mathbb{E}[w_i] - w_i) \right\|_H$$
$$\leq 4(\sqrt{M} + \|f_{\mathcal{H}}\|_\infty) \left(\frac{\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}} \right) \log \frac{2}{\delta}. \tag{46}$$

We next let $\xi_i = (\mathcal{T}_\rho + \lambda)^{-1/2} (\langle \mu_k, x_i \rangle - f_{\mathcal{H}}(x_i)) x_i$, for all $i \in [m]$. We assume that $k \geq 2$. (The proof for the case $k = 1$ is simpler as $\mu_1 = 0$.) It is easy to see that the expectation of each ξ_i with respect to the random variable (x_i, y_i) is

$$\mathbb{E}[\xi] = (\mathcal{T}_\rho + \lambda)^{-1/2} (\mathcal{T}_\rho \mu_k - \mathcal{S}_\rho^* f_{\mathcal{H}}) = (\mathcal{T}_\rho + \lambda)^{-1/2} (\mathcal{T}_\rho \mu_k - \mathcal{S}_\rho^* f_\rho),$$

and

$$\|\xi\|_H \leq (\|\mathcal{S}_\rho \mu_k\|_\infty + \|f_{\mathcal{H}}\|_\infty) \|(\mathcal{T}_\rho + \lambda)^{-1/2} x\|_H.$$

By Assumption (3), $\|\mathcal{S}_\rho \mu_k\|_\infty \leq \kappa \|\mu_k\|_H$. It thus follows from the above and (44) that

$$\|\xi\|_H \leq (\kappa \|\mu_k\|_H + \|f_{\mathcal{H}}\|_\infty) \frac{\kappa}{\sqrt{\lambda}}.$$

Besides,

$$\mathbb{E}\|\xi\|_H^2 \leq \frac{\kappa^2}{\lambda} \mathbb{E}(\mu_k(x) - f_{\mathcal{H}}(x))^2 = \frac{\kappa^2}{\lambda} \|\mathcal{S}_\rho \mu_k - f_{\mathcal{H}}\|_\rho^2 \leq \frac{\kappa^2 R^2}{\lambda} \left(\frac{\zeta}{2 \sum_{i=1}^{k-1} \eta_i} \right)^{2\zeta} \leq \frac{\kappa^2 R^2}{\lambda} \left(\frac{1}{\sum_{i=1}^k \eta_i} \right)^{2\zeta},$$

where for the last inequality, we used (33). Applying Lemma 11 and (36), we get that with probability at least $1-\delta$,

$$\begin{aligned}
&\left\|(\mathcal{T}_\rho+\lambda)^{-1/2}[\frac{1}{m}\sum_{i=1}^m(\mu_k(x_i)-f_{\mathcal{H}}(x_i))x_i-(\mathcal{T}_\rho\mu_k-\mathcal{S}_\rho^*f_\rho)]\right\|_H\\
\leq\ & 2\kappa\left(\frac{\kappa\|\mu_k\|_H+\|f_{\mathcal{H}}\|_\infty}{m\sqrt{\lambda}}+\frac{R}{\sqrt{m\lambda}}\left(\frac{1}{\sum_{i=1}^k\eta_i}\right)^\zeta\right)\log\frac{2}{\delta}\\
\leq\ & 2\kappa\left(\frac{\kappa R+\|f_{\mathcal{H}}\|_\infty}{m\sqrt{\lambda}}+\frac{\kappa R\left(\sum_{i=1}^k\eta_i\right)^{\frac{1}{2}-\zeta}}{m\sqrt{\lambda}}+\frac{R}{\sqrt{m\lambda}}\left(\frac{1}{\sum_{i=1}^k\eta_i}\right)^\zeta\right)\log\frac{2}{\delta}.
\end{aligned}$$

Introducing the above estimate and (46) into the following inequality

$$\begin{aligned}
\|(\mathcal{T}_\rho+\lambda)^{-1/2}N_k\|_H\leq&\left\|(\mathcal{T}_\rho+\lambda)^{-\frac{1}{2}}\frac{1}{m}\sum_{i=1}^m\left(f_{\mathcal{H}}(x_i)-y_i\right)x_i\right\|_H\\
&+\left\|(\mathcal{T}_\rho+\lambda)^{-1/2}[\frac{1}{m}\sum_{i=1}^m(\mu_k(x_i)-f_{\mathcal{H}}(x_i))x_i-(\mathcal{T}_\rho\mu_k-\mathcal{S}_\rho^*f_\rho)]\right\|_H,
\end{aligned}$$

and then substituting with (36), by a simple calculation, one can prove the desired result by scaling δ. ∎

The next lemma is from Rudi et al. (2015), and is derived applying a recent Bernstein inequality from (Tropp, 2012; Minsker, 2011) for a sum of random operators.

Lemma 19 *Let $\delta_2\in(0,1)$ and $\frac{9\kappa^2}{m}\log\frac{m}{\delta_2}\leq\lambda\leq\|\mathcal{T}_\rho\|$. Then the following holds with probability at least $1-\delta_2$,*

$$\|(\mathcal{T}_{\mathbf{x}}+\lambda I)^{-\frac{1}{2}}\mathcal{T}_\rho^{\frac{1}{2}}\|\leq\|(\mathcal{T}_{\mathbf{x}}+\lambda I)^{-\frac{1}{2}}(\mathcal{T}_\rho+\lambda I)^{\frac{1}{2}}\|\leq 2. \tag{47}$$

Now we are in a position to estimate the sample variance.

Proposition 3 *Under Assumptions 1, 2 and 3, let $\eta_1\kappa^2\leq 1$ and $0<\lambda\leq\|\mathcal{T}_\rho\|$. Assume that (47) holds. Then for all $t\in[T]$:*
1) if $\zeta\geq 1/2$, and (40) hold, then for $t\in\mathbb{N}$,

$$\begin{aligned}
&\|\mathcal{S}_\rho\nu_{t+1}-\mathcal{S}_\rho\mu_{t+1}\|_\rho\\
\leq&4(R\kappa^{2\zeta}+\sqrt{M})\left(\frac{\kappa}{m\sqrt{\lambda}}+\frac{\sqrt{2\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\left(\sum_{k=1}^{t-1}\frac{2\eta_k}{\sum_{i=k+1}^t\eta_i}+4\lambda\sum_{k=1}^{t-1}\eta_k+\sqrt{2}\kappa^2\eta_t\right)\log\frac{4}{\delta_1}.
\end{aligned}\tag{48}$$

2) if $\zeta<1/2$, and (41) hold for any $t\in[T]$, then for $t\in[T]$:

$$\begin{aligned}
\|\mathcal{S}_\rho\nu_{t+1}-\mathcal{S}_\rho\mu_{t+1}\|_\rho\leq&\left(\sum_{k=1}^{t-1}\frac{2\eta_k}{\sum_{i=k+1}^t\eta_i}+4\lambda\sum_{k=1}^{t-1}\eta_k+\sqrt{2}\kappa^2\eta_t\right)\\
&\times\left(2\left(3\|f_{\mathcal{H}}\|_\infty+3\sqrt{M}+\kappa R\right)\left(\frac{\kappa}{m\sqrt{\lambda}}+\frac{\sqrt{2\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)+\frac{2\kappa^2R\left(\sum_{i=1}^t\eta_i\right)^{\frac{1}{2}-\zeta}}{m\sqrt{\lambda}}\right)\log\frac{3T}{\delta_1}\\
&+\frac{2\kappa R}{\sqrt{m\lambda}}\log\frac{3T}{\delta_1}\left(\sum_{k=1}^{t-1}\frac{2\eta_k}{\left(\sum_{i=1}^k\eta_i\right)^\zeta\sum_{i=k+1}^t\eta_i}+4\lambda\sum_{k=1}^{t-1}\frac{\eta_k}{\left(\sum_{i=1}^k\eta_i\right)^\zeta}+\frac{\sqrt{2}\kappa^2\eta_t}{\left(\sum_{i=1}^t\eta_i\right)^\zeta}\right).
\end{aligned}\tag{49}$$

Proof For notational simplicity, we let $\mathcal{T}_{\rho,\lambda}=\mathcal{T}_\rho+\lambda I$ and $\mathcal{T}_{\mathbf{x},\lambda}=\mathcal{T}_{\mathbf{x}}+\lambda I$. Note that by Lemma 17, we have (37). When $k\in[t-1]$, by rewriting $\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k$ as

$$\mathcal{T}_\rho^{\frac{1}{2}}\mathcal{T}_{\mathbf{x},\lambda}^{-\frac{1}{2}}\mathcal{T}_{\mathbf{x},\lambda}^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\mathcal{T}_{\mathbf{x},\lambda}^{\frac{1}{2}}\mathcal{T}_{\mathbf{x},\lambda}^{-\frac{1}{2}}\mathcal{T}_{\rho,\lambda}^{\frac{1}{2}}\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_k,$$

we can upper bound $\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\|_H$ as

$$\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\|_H\leq\|\mathcal{T}_\rho^{\frac{1}{2}}\mathcal{T}_{\mathbf{x},\lambda}^{-\frac{1}{2}}\|\|\mathcal{T}_{\mathbf{x},\lambda}^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\mathcal{T}_{\mathbf{x},\lambda}^{\frac{1}{2}}\|\|\mathcal{T}_{\mathbf{x},\lambda}^{-\frac{1}{2}}\mathcal{T}_{\rho,\lambda}^{\frac{1}{2}}\|\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_k\|_H.$$

Applying (47), the above can be relaxed as

$$\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\|_H\leq 4\|\mathcal{T}_{\mathbf{x},\lambda}^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\mathcal{T}_{\mathbf{x},\lambda}^{\frac{1}{2}}\|\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_k\|_H,$$

which is equivalent to

$$\|\mathcal{T}_{\rho,\lambda}^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\|_H\leq 4\|\mathcal{T}_{\mathbf{x},\lambda}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\|\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_k\|_H.$$

Thus, following from $\eta_k\kappa^2\leq 1$ which implies $\eta_k\|\mathcal{T}_{\mathbf{x}}\|\leq 1$,

$$\begin{aligned}
\|\mathcal{T}_{\mathbf{x},\lambda}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\|&\leq\|\mathcal{T}_{\mathbf{x}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\|+\|\lambda\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\|\\
&\leq\|\mathcal{T}_{\mathbf{x}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\|+\lambda.
\end{aligned}$$

Applying Lemma 15 with $\zeta=1$ to bound $\|\mathcal{T}_{\mathbf{x}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\|$, we get

$$\|\mathcal{T}_{\mathbf{x},\lambda}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\|\leq\frac{1}{\mathrm{e}\sum_{j=k+1}^t\eta_j}+\lambda.$$

When $k=t$,

$$\begin{aligned}
\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\|_H=\|\mathcal{T}_\rho^{\frac{1}{2}}N_t\|_H&\leq\|\mathcal{T}_\rho^{\frac{1}{2}}\|\|\mathcal{T}_{\rho,\lambda}^{\frac{1}{2}}\|\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_t\|_H\\
&\leq\|\mathcal{T}_\rho\|^{\frac{1}{2}}(\|\mathcal{T}_\rho\|+\lambda)^{\frac{1}{2}}\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_t\|_H.
\end{aligned}$$

Since $\lambda\leq\|\mathcal{T}_\rho\|\leq\mathrm{tr}(\mathcal{T}_\rho)\leq\kappa^2$, we derive

$$\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_t\|_H\leq\sqrt{2}\kappa^2\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_t\|_H.$$

From the above analysis, we see that $\sum_{k=1}^{t}\eta_k\left\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^{t}(\mathcal{T}_{\mathbf{x}})N_k\right\|_H$ can be upper bounded by

$$\leq\left(\sum_{k=1}^{t-1}\frac{\eta_k/2\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_k\|_H}{\sum_{i=k+1}^{t}\eta_i}+\lambda\sum_{k=1}^{t-1}\eta_k\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_k\|_H+\sqrt{2}\kappa^2\eta_t\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}N_t\|_H\right).$$

Plugging (40) (or (41)) into the above, and then combining with (37), we get the desired bound (48) (or (49)). The proof is complete. ■

Setting $\eta_t=\eta_1 t^{-\theta}$ in the above proposition, with the basic estimates from Section 4, we get the following explicit bounds for the sample variance.

Proposition 4 *Under Assumptions 1, 2 and 3, let $\eta_t=\eta_1 t^{-\theta}$ with $\eta_1\in]0,\kappa^{-2}]$ and $\theta\in[0,1[$. Assume that (47) holds. Then the following holds for all $t\in[T]$ and any $0<\lambda\leq\|\mathcal{T}_\rho\|$:*
1) If $\zeta\geq 1/2$, and (40) holds for all $t\in[T]$,

$$\begin{aligned}&\|\mathcal{S}_\rho\nu_{t+1}-\mathcal{S}_\rho\mu_{t+1}\|_\rho\\ \leq&4(R\kappa^{2\zeta}+\sqrt{M})\left(\frac{8\lambda\eta_1 t^{1-\theta}}{1-\theta}+4\log t+4+\sqrt{2}\eta_1\kappa^2\right)\left(\frac{\kappa}{m\sqrt{\lambda}}+\frac{\sqrt{2\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{4}{\delta_1}.\end{aligned}\tag{50}$$

2) If $\zeta<1/2$, and (41) holds for all $t\in[T]$,

$$\begin{aligned}&\|\mathcal{S}_\rho\nu_{t+1}-\mathcal{S}_\rho\mu_{t+1}\|_\rho\leq\left(\frac{8\lambda\eta_1 t^{1-\theta}}{1-\theta}+4\log t+4+\sqrt{2}\eta_1\kappa^2\right)\log\frac{3T}{\delta_1}\\ \times&\left(2\left(3\|f_{\mathcal{H}}\|_\infty+3\sqrt{M}+\kappa R\right)\left(\frac{\kappa}{m\sqrt{\lambda}}+\frac{\sqrt{2\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)+\left(\frac{\kappa}{1-\theta}\sqrt{\frac{\eta_1 t^{1-\theta}}{m}}+1\right)\frac{4\kappa R}{\sqrt{m\lambda}}\frac{1}{(\eta_1 t^{1-\theta})^\zeta}\right).\end{aligned}\tag{51}$$

Proof By Proposition 3, we have (48) or (49). Note that

$$\sum_{k=1}^{t-1}\frac{\eta_k}{\sum_{i=k+1}^{t}\eta_i}=\sum_{k=1}^{t-1}\frac{k^{-\theta}}{\sum_{i=k+1}^{t}i^{-\theta}}\leq\sum_{k=1}^{t-1}\frac{k^{-\theta}}{(t-k)t^{-\theta}}.$$

Applying Lemma 14, we get

$$\sum_{k=1}^{t-1}\frac{\eta_k}{\sum_{i=k+1}^{t}\eta_i}\leq 2+2\log t,$$

and by Lemma 12,

$$\sum_{k=1}^{t-1}\eta_k=\eta_1\sum_{k=1}^{t-1}k^{-\theta}\leq\frac{2\eta_1 t^{1-\theta}}{1-\theta}.$$

Introducing the last two estimates into (48) and (49), one can get (50) and that

$$\begin{aligned}&\|\mathcal{S}_\rho\nu_{t+1}-\mathcal{S}_\rho\mu_{t+1}\|_\rho\leq\left(\frac{8\lambda\eta_1 t^{1-\theta}}{1-\theta}+4\log t+4+\sqrt{2}\eta_1\kappa^2\right)\\ &\times\left(2\left(3\|f_{\mathcal{H}}\|_\infty+3\sqrt{M}+\kappa R\right)\left(\frac{\kappa}{m\sqrt{\lambda}}+\frac{\sqrt{2\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)+\frac{4\kappa^2 R\left(\eta_1 t^{1-\theta}\right)^{\frac{1}{2}-\zeta}}{(1-\theta)m\sqrt{\lambda}}\right)\log\frac{3T}{\delta_1}\\ &+\frac{2\kappa R}{\sqrt{m\lambda}}\log\frac{3T}{\delta_1}\left(\sum_{k=1}^{t-1}\frac{2\eta_k}{\left(\sum_{i=1}^{k}\eta_i\right)^\zeta\sum_{i=k+1}^{t}\eta_i}+4\lambda\sum_{k=1}^{t-1}\frac{\eta_k}{\left(\sum_{i=1}^{k}\eta_i\right)^\zeta}+\frac{\sqrt{2}\kappa^2\eta_t}{\left(\sum_{i=1}^{t}\eta_i\right)^\zeta}\right).\end{aligned}$$

To prove (51), it remains to estimate the last term of the above. Again, using Lemmas 12, 13 and 14, we get

$$\begin{aligned}\sum_{k=1}^{t-1}\frac{\eta_k}{\left(\sum_{i=1}^{k}\eta_i\right)^\zeta\sum_{i=k+1}^{t}\eta_i}&=\frac{1}{\eta_1^\zeta}\sum_{k=1}^{t-1}\frac{k^{-\theta}}{\left(\sum_{i=1}^{k}i^{-\theta}\right)^\zeta\sum_{i=k+1}^{t}i^{-\theta}}\\ \leq&\frac{1}{\eta_1^\zeta}\sum_{k=1}^{t-1}\frac{k^{-\theta}}{(k^{1-\theta}/2)^\zeta(t-k)t^{-\theta}}=\frac{2^\zeta}{\eta_1^\zeta}t^\theta\sum_{k=1}^{t-1}\frac{k^{-(\theta+\zeta(1-\theta))}}{t-k}\\ \leq&\frac{2^\zeta}{\eta_1^\zeta}t^\theta 2t^{-(\theta+\zeta(1-\theta))}(1+\log t)\leq\frac{4(1+\log t)}{(\eta_1 t^{1-\theta})^\zeta},\end{aligned}$$

$$\sum_{k=1}^{t-1}\frac{\eta_k}{\left(\sum_{i=1}^{k}\eta_i\right)^\zeta}=\eta_1^{1-\zeta}\sum_{k=1}^{t-1}\frac{k^{-\theta}}{\left(\sum_{i=1}^{k}i^{-\theta}\right)^\zeta}\leq 2^\zeta\eta_1^{1-\zeta}\sum_{k=1}^{t-1}k^{-(\theta+\zeta(1-\theta))}\leq\frac{2(\eta_1 t^{1-\theta})^{1-\zeta}}{(1-\theta)},\quad\text{and}$$

$$\frac{\eta_t}{\left(\sum_{i=1}^{t}\eta_i\right)^\zeta}=\frac{\eta_1 t^{-\theta}}{\left(\sum_{i=1}^{t}\eta_1 i^{-\theta}\right)^\zeta}\leq 2^\zeta\frac{\eta_1 t^{-\theta}}{(\eta_1 t^{1-\theta})^\zeta}\leq\frac{\sqrt{2}\eta_1}{(\eta_1 t^{1-\theta})^\zeta}.$$

Therefore,

$$\begin{aligned}&\|\mathcal{S}_\rho\nu_{t+1}-\mathcal{S}_\rho\mu_{t+1}\|_\rho\leq\left(\frac{8\lambda\eta_1 t^{1-\theta}}{1-\theta}+4\log t+4+\sqrt{2}\eta_1\kappa^2\right)\\ &\times\left(2\left(3\|f_{\mathcal{H}}\|_\infty+3\sqrt{M}+\kappa R\right)\left(\frac{\kappa}{m\sqrt{\lambda}}+\frac{\sqrt{2\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)+\frac{4\kappa^2 R\left(\eta_1 t^{1-\theta}\right)^{\frac{1}{2}-\zeta}}{(1-\theta)m\sqrt{\lambda}}\right)\log\frac{3T}{\delta_1}\\ &+\frac{2\kappa R}{\sqrt{m\lambda}}\log\frac{3T}{\delta_1}\left(8+8\log t+\frac{8\lambda\eta_1 t^{1-\theta}}{1-\theta}+2\kappa^2\eta_1\right)\frac{1}{(\eta_1 t^{1-\theta})^\zeta}.\end{aligned}$$

Rearranging terms, we can prove the second part. ■

In conclusion, we get the following result for the sample variance.

Theorem 5 *Under Assumptions 1, 2 and 3, let $\delta_1,\delta_2\in]0,1[$ and $\frac{9\kappa^2}{m}\log\frac{m}{\delta_2}\leq\lambda\leq\|\mathcal{T}_\rho\|$. Let $\eta_t=\eta_1 t^{-\theta}$ for all $t\in[T]$, with $\eta_1\in]0,\kappa^{-2}]$ and $\theta\in[0,1[$. Then with probability at least $1-\delta_1-\delta_2$, the following holds for all $t\in[T]$:*
1) if $\zeta\geq 1/2$, we have (50).
2) if $\zeta<1/2$, we have (51).

6. Estimating Computational Variance

In this section, we estimate the computational variance, $\mathbb{E}[\|\mathcal{S}_\rho\omega_t - \mathcal{S}_\rho\nu_t\|_\rho^2]$. For this, a series of lemmas is introduced.

6.1 Cumulative Error

We have the following lemma, which shows that the computational variance can be controlled by a sum of weighted empirical risks.

Lemma 20 *We have*

$$\mathbb{E}_{\mathbf{J}}\|\mathcal{S}_\rho\omega_{t+1} - \mathcal{S}_\rho\nu_{t+1}\|_\rho^2 \leq \frac{\kappa^2}{b}\sum_{k=1}^{t}\eta_k^2\left\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\right\|^2\mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_k)]. \tag{52}$$

Proof Since ω_{t+1} and ν_{t+1} are given by (4) and (26), respectively,

$$\begin{aligned}\omega_{t+1} - \nu_{t+1} &= (\omega_t - \nu_t) + \eta_t\left\{(\mathcal{T}_{\mathbf{x}}\nu_t - \mathcal{S}_{\mathbf{x}}^*\mathbf{y}) - \frac{1}{b}\sum_{i=b(t-1)+1}^{bt}(\langle\omega_t, x_{j_i}\rangle_H - y_{j_i})x_{j_i}\right\}\\ &= (I - \eta_t\mathcal{T}_{\mathbf{x}})(\omega_t - \nu_t) + \frac{\eta_t}{b}\sum_{i=b(t-1)+1}^{bt}\{(\mathcal{T}_{\mathbf{x}}\omega_t - \mathcal{S}_{\mathbf{x}}^*\mathbf{y}) - (\langle\omega_t, x_{j_i}\rangle_H - y_{j_i})x_{j_i}\}.\end{aligned}$$

Applying this relationship iteratively,

$$\omega_{t+1} - \nu_{t+1} = \Pi_1^t(\mathcal{T}_{\mathbf{x}})(\omega_1 - \nu_1) + \frac{1}{b}\sum_{k=1}^{t}\sum_{i=b(k-1)+1}^{bk}\eta_k\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i},$$

where we denote

$$M_{k,i} = (\mathcal{T}_{\mathbf{x}}\omega_k - \mathcal{S}_{\mathbf{x}}^*\mathbf{y}) - (\langle\omega_k, x_{j_i}\rangle_H - y_{j_i})x_{j_i}. \tag{53}$$

Since $\omega_1 = \nu_1 = 0$, then

$$\omega_{t+1} - \nu_{t+1} = \frac{1}{b}\sum_{k=1}^{t}\sum_{i=b(k-1)+1}^{bk}\eta_k\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i}.$$

Therefore,

$$\begin{aligned}\mathbb{E}_{\mathbf{J}}\|\mathcal{S}_\rho\omega_{t+1} - \mathcal{S}_\rho\nu_{t+1}\|_\rho^2 &= \frac{1}{b^2}\mathbb{E}_{\mathbf{J}}\left\|\sum_{k=1}^{t}\sum_{i=b(k-1)+1}^{bk}\eta_k\mathcal{S}_\rho\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i}\right\|_\rho^2\\ &= \frac{1}{b^2}\sum_{k=1}^{t}\sum_{i=b(k-1)+1}^{bk}\eta_k^2\mathbb{E}_{\mathbf{J}}\left\|\mathcal{S}_\rho\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i}\right\|_\rho^2,\end{aligned} \tag{54}$$

where for the last equality, we use the fact that if $k \neq k'$, or $k = k'$ but $i \neq i'$[4], then

$$\mathbb{E}_{\mathbf{J}}\langle\mathcal{S}_\rho\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i}, \mathcal{S}_\rho\Pi_{k'+1}^t(\mathcal{T}_{\mathbf{x}})M_{k',i'}\rangle_\rho = 0.$$

4. This is possible only when $b \geq 2$.

Indeed, if $k \neq k'$, without loss of generality, we consider the case $k < k'$. Recalling that $M_{k,i}$ is given by (53) and that given any $\mathbf{z}$, ω_k is depending only on $\mathbf{J}_1, \cdots, \mathbf{J}_{k-1}$, we thus have

$$\begin{aligned}&\mathbb{E}_{\mathbf{J}}\langle\mathcal{S}_\rho\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i}, \mathcal{S}_\rho\Pi_{k'+1}^t(\mathcal{T}_{\mathbf{x}})M_{k',i'}\rangle_\rho\\ &= \mathbb{E}_{\mathbf{J}_1,\cdots,\mathbf{J}_{k'-1}}\langle\mathcal{S}_\rho\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i}, \mathcal{S}_\rho\Pi_{k'+1}^t(\mathcal{T}_{\mathbf{x}})\mathbb{E}_{\mathbf{J}_{k'}}[M_{k',i'}]\rangle_\rho = 0.\end{aligned}$$

If $k = k'$ but $i \neq i'$, without loss of generality, we assume $i < i'$. By noting that ω_k is depending only on $\mathbf{J}_1, \cdots, \mathbf{J}_{k-1}$ and $M_{k,i}$ is depending only on ω_k and z_{j_i} (given any sample $\mathbf{z}$),

$$\begin{aligned}&\mathbb{E}_{\mathbf{J}}\langle\mathcal{S}_\rho\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i}, \mathcal{S}_\rho\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i'}\rangle_\rho\\ &= \mathbb{E}_{\mathbf{J}_1,\cdots,\mathbf{J}_{k-1}}\langle\mathcal{S}_\rho\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\mathbb{E}_{j_i}[M_{k,i}], \mathcal{S}_\rho\Pi_{k'+1}^t(\mathcal{T}_{\mathbf{x}})\mathbb{E}_{j_{i'}}[M_{k,i'}]\rangle_\rho = 0.\end{aligned}$$

Using the isometry property (24) to (54),

$$\mathbb{E}_{\mathbf{J}}\left\|\mathcal{S}_\rho\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i}\right\|_\rho^2 = \mathbb{E}_{\mathbf{J}}\left\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})M_{k,i}\right\|_H^2 \leq \left\|\mathcal{T}_\rho^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})\right\|^2\mathbb{E}_{\mathbf{J}}\left\|M_{k,i}\right\|_H^2,$$

and by applying the inequality $\mathbb{E}[\|\xi - \mathbb{E}[\xi]\|_H^2] \leq \mathbb{E}[\|\xi\|_H^2]$,

$$\mathbb{E}_{\mathbf{J}}\left\|M_{k,i}\right\|_H^2 \leq \mathbb{E}_{\mathbf{J}}\left\|(\langle\omega_k, x_{j_i}\rangle_H - y_{j_i})x_{j_i}\right\|_H^2 \leq \kappa^2\mathbb{E}_{\mathbf{J}}[(\langle\omega_k, x_{j_i}\rangle_H - y_{j_i})^2] = \kappa^2\mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_k)],$$

where for the last inequality we use (3). Therefore, we can get the desired result. ■

To estimate the computational variance from (52), we need to further develop upper bounds for the empirical risks and the weighted factors, which will be given in the following two subsections.

6.2 Bounding the Empirical Risk

This subsection is devoted to upper bounding $\mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_l)]$. The process relies on some tools from convex analysis and a decomposition related to the weighted averages and the last iterates from (Shamir and Zhang, 2013; Lin et al., 2016b). We begin by introducing the following lemma, a fact based on the square loss' special properties.

Lemma 21 *Given any sample* $\mathbf{z}$*, and* $l \in \mathbb{N}$*, let* $\omega \in H$ *be independent from* $\mathbf{J}_l$*, then*

$$\eta_l\left(\mathcal{E}_{\mathbf{z}}(\omega_l) - \mathcal{E}_{\mathbf{z}}(\omega)\right) \leq \|\omega_l - \omega\|_H^2 - \mathbb{E}_{\mathbf{J}_l}\|\omega_{l+1} - \omega\|_H^2 + \eta_l^2\kappa^2\mathcal{E}_{\mathbf{z}}(\omega_l). \tag{55}$$

Proof Since ω_{t+1} is given be (4), subtracting both sides of (4) by ω, taking the square H-norm, and expanding the inner product,

$$\begin{aligned}\|\omega_{l+1} - \omega\|_H^2 = \|\omega_l - \omega\|_H^2 &+ \frac{\eta_l^2}{b^2}\left\|\sum_{i=b(l-1)+1}^{bl}(\langle\omega_l, x_{j_i}\rangle_H - y_{j_i})x_{j_i}\right\|_H^2\\ &+\frac{2\eta_l}{b}\sum_{i=b(l-1)+1}^{bl}(\langle\omega_l, x_{j_i}\rangle_H - y_{j_i})\langle\omega - \omega_l, x_{j_i}\rangle_H.\end{aligned}$$

By Assumption (3), $\|x_{j_i}\|_H \leq \kappa$, and thus

$$\left\| \sum_{i=b(l-1)+1}^{bl} (\langle \omega_l, x_{j_i}\rangle_H - y_{j_i}) x_{j_i} \right\|_H^2 \leq \left(\sum_{i=b(l-1)+1}^{bl} |\langle \omega_l, x_{j_i}\rangle_H - y_{j_i}| \kappa \right)^2 \leq \kappa^2 b \sum_{i=b(l-1)+1}^{bl} (\langle \omega_l, x_{j_i}\rangle_H - y_{j_i})^2,$$

where for the last inequality, we used Cauchy-Schwarz inequality. Thus,

$$\|\omega_{l+1} - \omega\|_H^2 \leq \|\omega_l - \omega\|_H^2 + \frac{\eta_l^2 \kappa^2}{b} \sum_{i=b(l-1)+1}^{bl} (\langle \omega_l, x_{j_i}\rangle_H - y_{j_i})^2 + \frac{2\eta_l}{b} \sum_{i=b(l-1)+1}^{bl} (\langle \omega_l, x_{j_i}\rangle_H - y_{j_i})(\langle \omega, x_{j_i}\rangle_H - \langle \omega_l, x_{j_i}\rangle_H).$$

Using the basic inequality $a(b-a) \leq (b^2 - a^2)/2, \forall a, b \in \mathbb{R}$,

$$\|\omega_{l+1} - \omega\|_H^2 \leq \|\omega_l - \omega\|_H^2 + \frac{\eta_l^2 \kappa^2}{b} \sum_{i=b(l-1)+1}^{bl} (\langle \omega_l, x_{j_i}\rangle_H - y_{j_i})^2 + \frac{\eta_l}{b} \sum_{i=b(l-1)+1}^{bl} \left((\langle \omega, x_{j_i}\rangle_H - y_{j_i})^2 - (\langle \omega_l, x_{j_i}\rangle_H - y_{j_i})^2 \right).$$

Noting that ω_l and ω are independent from $\mathbf{J}_l$, and taking the expectation on both sides with respect to $\mathbf{J}_l$,

$$\mathbb{E}_{\mathbf{J}_l} \|\omega_{l+1} - \omega\|_H^2 \leq \|\omega_l - \omega\|_H^2 + \eta_l^2 \kappa^2 \mathcal{E}_{\mathbf{z}}(\omega_l) + \eta_l \left(\mathcal{E}_{\mathbf{z}}(\omega) - \mathcal{E}_{\mathbf{z}}(\omega_l) \right),$$

which leads to the desired result by rearranging terms. The proof is complete. ■

Using the above lemma and a decomposition related to the weighted averages and the last iterates from (Shamir and Zhang, 2013; Lin et al., 2016b), we can prove the following relationship.

Lemma 22 *Let $\eta_1 \kappa^2 \leq 1/2$ for all $t \in \mathbb{N}$. Then*

$$\eta_t \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_t)] \leq 4\mathcal{E}_{\mathbf{z}}(0) \frac{1}{t} \sum_{l=1}^{t} \eta_l + 2\kappa^2 \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k}^{t-1} \eta_i^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)]. \tag{56}$$

Proof For $k = 1, \cdots, t-1$,

$$\begin{aligned} & \frac{1}{k} \sum_{i=t-k+1}^{t} \eta_i \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] - \frac{1}{k+1} \sum_{i=t-k}^{t} \eta_i \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] \\ = & \frac{1}{k(k+1)} \left\{ (k+1) \sum_{i=t-k+1}^{t} \eta_i \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] - k \sum_{i=t-k}^{t} \eta_i \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] \right\} \\ = & \frac{1}{k(k+1)} \sum_{i=t-k+1}^{t} (\eta_i \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] - \eta_{t-k} \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_{t-k})]). \end{aligned}$$

Summing over $k = 1, \cdots, t-1$, and rearranging terms, we get (Lin et al., 2016b)

$$\eta_t \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_t)] = \frac{1}{t} \sum_{i=1}^{t} \eta_i \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] + \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k+1}^{t} (\eta_i \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] - \eta_{t-k} \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_{t-k})]).$$

Since $\{\eta_t\}_t$ is decreasing and $\mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_{t-k})]$ is non-negative, the above can be relaxed as

$$\eta_t \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_t)] \leq \frac{1}{t} \sum_{i=1}^{t} \eta_i \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] + \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k+1}^{t} \eta_i \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i) - \mathcal{E}_{\mathbf{z}}(\omega_{t-k})]. \tag{57}$$

In the rest of the proof, we will upper bound the last two terms of the above.

To bound the first term of the right side of (57), we apply Lemma 21 with $\omega = 0$ to get

$$\eta_l \mathbb{E}_{\mathbf{J}} \left(\mathcal{E}_{\mathbf{z}}(\omega_l) - \mathcal{E}_{\mathbf{z}}(0) \right) \leq \mathbb{E}_{\mathbf{J}}[\|\omega_l\|_H^2 - \|\omega_{l+1}\|_H^2] + \eta_l^2 \kappa^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_l)].$$

Rearranging terms,

$$\eta_l (1 - \eta_l \kappa^2) \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_l)] \leq \mathbb{E}_{\mathbf{J}}[\|\omega_l\|_H^2 - \|\omega_{l+1}\|_H^2] + \eta_l \mathcal{E}_{\mathbf{z}}(0).$$

It thus follows from the above and $\eta_l \kappa^2 \leq 1/2$ that

$$\eta_l \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_l)]/2 \leq \mathbb{E}_{\mathbf{J}}[\|\omega_l\|_H^2 - \|\omega_{l+1}\|_H^2] + \eta_l \mathcal{E}_{\mathbf{z}}(0).$$

Summing up over $l = 1, \cdots, t$,

$$\sum_{l=1}^{t} \eta_l \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_l)]/2 \leq \mathbb{E}_{\mathbf{J}}[\|w_1\|_H^2 - \|\omega_{t+1}\|_H^2] + \mathcal{E}_{\mathbf{z}}(0) \sum_{l=1}^{t} \eta_l.$$

Introducing with $\omega_1 = 0, \|\omega_{t+1}\|_H^2 \geq 0$, and then multiplying both sides by $2/t$, we get

$$\frac{1}{t} \sum_{l=1}^{t} \eta_l \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_l)] \leq 2\mathcal{E}_{\mathbf{z}}(0) \frac{1}{t} \sum_{l=1}^{t} \eta_l. \tag{58}$$

It remains to bound the last term of (57). Let $k \in [t-1]$ and $i \in \{t-k, \cdots, t\}$. Note that given the sample $\mathbf{z}$, ω_i is depending only on $\mathbf{J}_1, \cdots, \mathbf{J}_{i-1}$ when $i > 1$ and $\omega_1 = 0$. Thus, we can apply Lemma 21 with $\omega = \omega_{t-k}$ to derive

$$\eta_i \left(\mathcal{E}_{\mathbf{z}}(\omega_i) - \mathcal{E}_{\mathbf{z}}(\omega_{t-k}) \right) \leq \|\omega_i - \omega_{t-k}\|_H^2 - \mathbb{E}_{\mathbf{J}_i} \|\omega_{i+1} - \omega_{t-k}\|_H^2 + \eta_i^2 \kappa^2 \mathcal{E}_{\mathbf{z}}(\omega_i).$$

Therefore,

$$\eta_i \mathbb{E}_{\mathbf{J}} \left[\mathcal{E}_{\mathbf{z}}(\omega_i) - \mathcal{E}_{\mathbf{z}}(\omega_{t-k}) \right] \leq \mathbb{E}_{\mathbf{J}}[\|\omega_i - \omega_{t-k}\|_H^2 - \|\omega_{i+1} - \omega_{t-k}\|_H^2] + \eta_i^2 \kappa^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)].$$

Summing up over $i = t-k, \cdots, t$,

$$\sum_{i=t-k}^{t} \eta_i \mathbb{E}_{\mathbf{J}} \left[\mathcal{E}_{\mathbf{z}}(\omega_i) - \mathcal{E}_{\mathbf{z}}(\omega_{t-k}) \right] \leq \kappa^2 \sum_{i=t-k}^{t} \eta_i^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)].$$

Note that the left hand side is exactly $\sum_{i=t-k+1}^{t} \eta_i \mathbb{E}_{\mathbf{J}} \left[\mathcal{E}_{\mathbf{z}}(\omega_i) - \mathcal{E}_{\mathbf{z}}(\omega_{t-k})\right]$. We thus know that the last term of (57) can be upper bounded by

$$\begin{aligned} & \kappa^2 \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k}^{t} \eta_i^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] \\ = \quad & \kappa^2 \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k}^{t-1} \eta_i^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] + \kappa^2 \eta_t^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_t)] \sum_{k=1}^{t-1} \frac{1}{k(k+1)}. \end{aligned}$$

Using the fact that

$$\sum_{k=1}^{t-1} \frac{1}{k(k+1)} = \sum_{k=1}^{t-1} \left(\frac{1}{k} - \frac{1}{k+1} \right) = 1 - \frac{1}{t} \leq 1,$$

and $\kappa^2 \eta_t \leq 1/2$, we get that the last term of (57) can be bounded as

$$\begin{aligned} & \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k+1}^{t} \eta_i (\mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] - \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_{t-k})]) \\ \leq \quad & \kappa^2 \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k}^{t-1} \eta_i^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)] + \eta_t \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_t)]/2. \end{aligned}$$

Plugging the above and (58) into the decomposition (57), and rearranging terms

$$\eta_t \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_t)]/2 \leq 2\mathcal{E}_{\mathbf{z}}(0) \frac{1}{t} \sum_{l=1}^{t} \eta_l + \kappa^2 \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k}^{t-1} \eta_i^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)],$$

which leads to the desired result by multiplying both sides by 2. The proof is complete. ■

We also need the following lemma, whose proof can be done using an induction argument.

Lemma 23 *Let $\{u_t\}_{t=1}^T$, $\{A_t\}_{t=1}^T$ and $\{B_t\}_{t=1}^T$ be three sequences of non-negative numbers such that $u_1 \leq A_1$ and*

$$u_t \leq A_t + B_t \sup_{i \in [t-1]} u_i, \qquad \forall t \in \{2, 3, \cdots, T\}. \tag{59}$$

Let $\sup_{t \in [T]} B_t \leq B < 1$. Then for all $t \in [T]$,

$$\sup_{k \in [t]} u_k \leq \frac{1}{1-B} \sup_{k \in [t]} A_k. \tag{60}$$

Proof When $t = 1$, (60) holds trivially since $u_1 \leq A_1$ and $B < 1$. Now assume for some $t \in \mathbb{N}$ with $2 \leq t \leq T$,

$$\sup_{i \in [t-1]} u_i \leq \frac{1}{1-B} \sup_{i \in [t-1]} A_i.$$

Then, by (59), the above hypothesis, and $B_t \leq B$, we have

$$u_t \leq A_t + B_t \sup_{i \in [t-1]} u_i \leq A_t + \frac{B_t}{1-B} \sup_{i \in [t-1]} A_i \leq \sup_{i \in [t]} A_i \left(1 + \frac{B_t}{1-B} \right) \leq \sup_{i \in [t]} A_i \frac{1}{1-B}.$$

Consequently,

$$\sup_{k \in [t]} u_k \leq \frac{1}{1-B} \sup_{k \in [t]} A_k,$$

thereby showing that indeed (60) holds for t. By mathematical induction, (60) holds for every $t \in [T]$. The proof is complete. ■

Now we can bound $\mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_k)]$ as follows.

Lemma 24 *Let $\eta_1 \kappa^2 \leq 1/2$ and for all $t \in [T]$ with $t \geq 2$,*

$$\frac{1}{\eta_t} \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k}^{t-1} \eta_i^2 \leq \frac{1}{4\kappa^2}. \tag{61}$$

Then for all $t \in [T]$,

$$\sup_{k \in [t]} \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_k)] \leq 8\mathcal{E}_{\mathbf{z}}(0) \sup_{k \in [t]} \left\{ \frac{1}{\eta_k k} \sum_{l=1}^{k} \eta_l \right\}. \tag{62}$$

Proof By Lemma 22, we have (56). Dividing both sides by η_t, we can relax the inequality as

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_t)] \leq 4\mathcal{E}_{\mathbf{z}}(0) \frac{1}{\eta_t t} \sum_{l=1}^{t} \eta_l + 2\kappa^2 \frac{1}{\eta_t} \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k}^{t-1} \eta_i^2 \sup_{i \in [t-1]} \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_i)].$$

In Lemma 23, we let $u_t = \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_t)]$, $A_t = 4\mathcal{E}_{\mathbf{z}}(0) \frac{1}{\eta_t t} \sum_{l=1}^{t} \eta_l$ and

$$B_t = 2\kappa^2 \frac{1}{\eta_t} \sum_{k=1}^{t-1} \frac{1}{k(k+1)} \sum_{i=t-k}^{t-1} \eta_i^2.$$

Condition (61) guarantees that $\sup_{t \in [T]} B_t \leq 1/2$. Thus, (60) holds, and the desired result follows by plugging with $B = 1/2$. The proof is complete. ■

Finally, we need the following lemma to bound $\mathcal{E}_{\mathbf{z}}(0)$, whose proof follows from applying the Bernstein inequality from Lemma 11.

Lemma 25 *Under Assumption 1, with probability at least $1 - \delta_3$ ($\delta_3 \in]0, 1[$), there holds*

$$\mathcal{E}_{\mathbf{z}}(0) \leq Mv + 2Mv \left(\frac{1}{m} + \frac{\sqrt{2}}{\sqrt{m}} \right) \log \frac{2}{\delta_3}.$$

In particular, if $m \geq 32 \log^2 \frac{2}{\delta_3}$, then

$$\mathcal{E}_{\mathbf{z}}(0) \leq 2Mv. \tag{63}$$

Proof Following from (5),

$$\int_Z y^{2l} d\rho \leq \frac{1}{2} l! M^{l-2} \cdot (2M^2 v), \qquad \forall l \in \mathbb{N},$$

and

$$\int_Z y^2 d\rho \leq Mv.$$

Therefore,

$$\begin{aligned}\int_Z |y^2 - \mathbb{E}y^2|^l d\rho &\leq \int_Z \max(|y|^{2l}, (\mathbb{E}y^2)^l) d\rho \\ &\leq \int_Z (|y|^{2l} + (\mathbb{E}y^2)^l) d\rho \\ &\leq \frac{1}{2} l! M^{l-2} \cdot (2M^2 v) + (Mv)^l \\ &\leq \frac{1}{2} l! (Mv)^{l-2} (2Mv)^2,\end{aligned}$$

where for the last inequality we used $v \geq 1$. Applying Lemma 11, with $\omega_i = y_i^2$ for all $i \in [m]$, $B = Mv$ and $\sigma = 2Mv$, we know that with probability at least $1 - \delta_3$, there holds

$$\frac{1}{n}\sum_{i=1}^n y_i^2 - \int_Z y^2 d\rho \leq 2Mv \left(\frac{1}{m} + \frac{2}{\sqrt{m}}\right) \log \frac{2}{\delta_3}.$$

The proof is complete. ∎

6.3 Bounding $\left\| \mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x}) \right\|$

We bound the weighted factor $\left\| \mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x}) \right\|$ as follows.

Lemma 26 *Assume (47) holds for some $\lambda > 0$ and $\eta_1 \kappa^2 \leq 1$. Then*

$$\|\mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\|^2 \leq \frac{1}{\sum_{i=k+1}^t \eta_i} + 4\lambda.$$

Proof Note that we have

$$\|\mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\| \leq \|\mathcal{T}_\rho^{\frac{1}{2}} (\mathcal{T}_\mathbf{x} + \lambda I)^{-\frac{1}{2}}\| \|(\mathcal{T}_\mathbf{x} + \lambda I)^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\|.$$

Using (47), we can relax the above as

$$\|\mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\| \leq 2\|(\mathcal{T}_\mathbf{x} + \lambda I)^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\|,$$

which leads to

$$\|\mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\|^2 \leq 4\|(\mathcal{T}_\mathbf{x} + \lambda I)^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\|^2.$$

Since

$$\begin{aligned}\|(\mathcal{T}_\mathbf{x} + \lambda I)^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\|^2 &= \|(\mathcal{T}_\mathbf{x} + \lambda I) \Pi_{k+1}^t(\mathcal{T}_\mathbf{x}) \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\| \\ &\leq \|\mathcal{T}_\mathbf{x} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x}) \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\| + \lambda \\ &= \|\mathcal{T}_\mathbf{x}^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\|^2 + \lambda,\end{aligned}$$

and with $\eta_t \kappa^2 \leq 1$, $\|\mathcal{T}_\mathbf{x}\| \leq \mathrm{tr}(\mathcal{T}_\mathbf{x}) \leq \kappa^2$, by Lemma 15,

$$\|\mathcal{T}_\mathbf{x}^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x})\|^2 \leq \frac{1}{2\mathrm{e}\sum_{i=k+1}^t \eta_i} \leq \frac{1}{4\sum_{i=k+1}^t \eta_i},$$

we thus derive the desired result. The proof is complete. ∎

6.4 Deriving Error Bounds

With Lemmas 20–26, we are ready to estimate the computational variance , $\mathbb{E}_\mathbf{J}\|\mathcal{S}_\rho \omega_{t+1} - \mathcal{S}_\rho \nu_{t+1}\|_\rho^2$, as follows.

Proposition 5 *Under Assumption 1, assume (47) holds for some $\lambda > 0$, $\eta_1 \kappa^2 \leq 1/2$, (61) and (63). Then, we have for all $t \in [T]$,*

$$\mathbb{E}_\mathbf{J}\|\mathcal{S}_\rho \omega_{t+1} - \mathcal{S}_\rho \nu_{t+1}\|_\rho^2 \leq \frac{16Mv\kappa^2}{b} \sup_{k\in[t]} \left\{ \frac{1}{\eta_k k} \sum_{l=1}^k \eta_l \right\} \left(\sum_{k=1}^{t-1} \frac{\eta_k^2}{\sum_{i=k+1}^t \eta_i} + 4\lambda \sum_{k=1}^{t-1} \eta_k^2 + \eta_t^2 \kappa^2 \right). \tag{64}$$

Proof According to Lemmas 20 and 24, we have (52) and (62). It thus follows that

$$\mathbb{E}_\mathbf{J}\|\mathcal{S}_\rho \omega_{t+1} - \mathcal{S}_\rho \nu_{t+1}\|_\rho^2 \leq \frac{8\mathcal{E}_\mathbf{z}(0)\kappa^2}{b} \sup_{k\in[t]} \left\{ \frac{1}{\eta_k k} \sum_{l=1}^k \eta_l \right\} \sum_{k=1}^t \eta_k^2 \left\| \mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x}) \right\|^2.$$

Now the proof can be finished by applying Lemma 26 which tells us that

$$\begin{aligned}\sum_{k=1}^t \eta_k^2 \left\| \mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x}) \right\|^2 &= \sum_{k=1}^{t-1} \eta_k^2 \left\| \mathcal{T}_\rho^{\frac{1}{2}} \Pi_{k+1}^t(\mathcal{T}_\mathbf{x}) \right\|^2 + \eta_t^2 \left\| \mathcal{T}_\rho^{\frac{1}{2}} \right\|^2 \\ &\leq \sum_{k=1}^{t-1} \frac{\eta_k^2}{\sum_{i=k+1}^t \eta_i} + 4\lambda \sum_{k=1}^{t-1} \eta_k^2 + \eta_t^2 \kappa^2,\end{aligned}$$

and (63) to the above inequality. The proof is complete. ∎

Setting $\eta_t = \eta_1 t^{-\theta}$ for some appropriate η_1 and θ in the above proposition, we get the following explicitly upper bounds for $\mathbb{E}_\mathbf{J}\|\mathcal{S}_\rho \omega_t - \mathcal{S}_\rho \omega_t\|_\rho^2$.

Proposition 6 *Under Assumption 1, assume (47) holds for some $\lambda > 0$ and (63). Let $\eta_t = \eta_1 t^{-\theta}$ for all $t \in [T]$, with $\theta \in [0, 1[$ and*

$$0 < \eta_1 \leq \frac{t^{\min(\theta, 1-\theta)}}{8\kappa^2(\log t + 1)}, \qquad \forall t \in [T]. \tag{65}$$

Then, for all $t \in [T]$,

$$\mathbb{E}_\mathbf{J}\|\omega_{t+1} - \nu_{t+1}\|_\rho^2 \leq \frac{16Mv\kappa^2}{b(1-\theta)} \left(5\eta_1 t^{-\min(\theta, 1-\theta)} + 8\lambda\eta_1^2 t^{(1-2\theta)_+} \right) (1 \vee \log t). \tag{66}$$

Proof We will use Proposition 5 to prove the result. Thus, we need to verify the condition (61). Note that

$$\sum_{k=1}^{t-1}\frac{1}{k(k+1)}\sum_{i=t-k}^{t-1}\eta_i^2=\sum_{i=1}^{t-1}\eta_i^2\sum_{k=t-i}^{t-1}\frac{1}{k(k+1)}=\sum_{i=1}^{t-1}\eta_i^2\left(\frac{1}{t-i}-\frac{1}{t}\right)\le\sum_{i=1}^{t-1}\frac{\eta_i^2}{t-i}.$$

Substituting with $\eta_i=\eta i^{-\theta}$, and by Lemma 14,

$$\sum_{k=1}^{t-1}\frac{1}{k(k+1)}\sum_{i=t-k}^{t-1}\eta_i^2\le\eta_1^2\sum_{i=1}^{t-1}\frac{i^{-2\theta}}{t-i}\le 2\eta_1^2 t^{-\min(2\theta,1)}(\log t+1).$$

Dividing both sides by $\eta_t\ (=\eta_1 t^{-\theta})$, and then using (65),

$$\frac{1}{\eta_t}\sum_{k=1}^{t-1}\frac{1}{k(k+1)}\sum_{i=t-k}^{t-1}\eta_i^2\le 2\eta_1 t^{-\min(\theta,1-\theta)}(\log t+1)\le\frac{1}{4\kappa^2}.$$

This verifies (61). Note also that by taking $t=1$ in (65), for all $t\in[T]$,

$$\eta_t\kappa^2\le\eta_1\kappa^2\le\frac{1}{8\kappa^2}\le\frac{1}{2}.$$

We thus can apply Proposition 5 to derive (64). What remains is to control the right hand side of (64). Since

$$\sum_{k=1}^{t-1}\frac{\eta_k^2}{\sum_{i=k+1}^{t}\eta_i}=\eta_1\sum_{k=1}^{t-1}\frac{k^{-2\theta}}{\sum_{i=k+1}^{t}i^{-\theta}}\le\eta_1\sum_{k=1}^{t-1}\frac{k^{-2\theta}}{(t-k)t^{-\theta}},$$

combining with Lemma 14,

$$\sum_{k=1}^{t-1}\frac{\eta_k^2}{\sum_{i=k+1}^{t}\eta_i}\le 2\eta_1 t^{-\min(\theta,1-\theta)}(\log t+1).$$

Also, by Lemma 12,

$$\frac{1}{\eta_k k}\sum_{l=1}^{k}\eta_l=\frac{1}{k^{1-\theta}}\sum_{l=1}^{k}l^{-\theta}\le\frac{1}{1-\theta},$$

and by Lemma 13,

$$\sum_{k=1}^{t-1}\eta_k^2=\eta_1^2\sum_{k=1}^{t-1}k^{-2\theta}\le\eta_1^2 t^{\max(1-2\theta,0)}(\log t+1).$$

Introducing the last three estimates into (64) and using that $\eta_t^2\kappa^2\le\eta_1 t^{-\theta}$ by (65), we get the desired result. The proof is complete. ■

Collect some of the above analysis, we get the following result for the computational variance.

Theorem 6 *Under Assumptions 1, let* $\delta_2\in]0,1[$, $\frac{9\kappa^2}{m}\log\frac{m}{\delta_2}\le\lambda\le\|\mathcal{T}_\rho\|$, $\delta_3\in]0,1[$, $m\ge 32\log^2\frac{2}{\delta_3}$, *and* $\eta_t=\eta_1 t^{-\theta}$ *for all* $t\in[T]$, *with* $\theta\in[0,1[$ *and* η_1 *such that (65). Then, with probability at least* $1-\delta_2-\delta_3$, *(66) holds for all* $t\in[T]$.

7. Deriving Total Error Bounds

The purpose of this section is to derive total error bounds.

7.1 Attainable Case

We have the following general theorem for $\zeta\ge 1/2$, with which we prove our main results stated in Section 3.

Theorem 7 *Under Assumptions 1, 2 and 3, let* $\zeta\ge 1/2$, $T\in\mathbb{N}$ *with* $T\ge 3$, $\delta\in]0,1[$, $\eta_t=\eta\kappa^{-2}t^{-\theta}$ *for all* $t\in[T]$, *with* $\theta\in[0,1[$ *and* η *such that*

$$0<\eta\le\frac{t^{\min(\theta,1-\theta)}}{8(\log t+1)},\qquad\forall t\in[T].\tag{67}$$

If for some $\epsilon\in]0,1]$,

$$m\ge\left(\frac{18\kappa^2}{\epsilon\|\mathcal{T}_\rho\|}\log\left(\frac{27\kappa^2}{\epsilon\|\mathcal{T}_\rho\|\delta}\right)\right)^{1/\epsilon},\tag{68}$$

then the following holds with probability at least $1-\delta$: *for all* $t\in[T]$,

$$\begin{aligned}\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})]-\inf_H\mathcal{E}\le{}& q_1(\eta t^{1-\theta})^{-2\zeta}+q_2 m^{\gamma(1-\epsilon)-1}(1\vee\eta^2 m^{2\epsilon-2}t^{2-2\theta})(\log T)^2\log^2\frac{12}{\delta}\\&+q_3\eta b^{-1}(t^{-\min(\theta,1-\theta)}\vee m^{\epsilon-1}\eta t^{(1-2\theta)_+})\log T.\end{aligned}\tag{69}$$

Here, $q_1=2R^2\zeta^{2\zeta}$, $q_2=\frac{10^4(R\kappa^{2\zeta}+\sqrt{M})^2(\kappa/\sqrt{\|\mathcal{T}_\rho\|}+\sqrt{2\sqrt{v}c_\gamma/\|\mathcal{T}_\rho\|^\gamma})^2}{(1-\theta)^2}$, *and* $q_3=\frac{208Mv}{1-\theta}$.

Proof Let $\lambda=\|\mathcal{T}_\rho\|m^{\epsilon-1}$. Clearly, $\lambda\le\|\mathcal{T}_\rho\|$. For any $A\ge 0$ and $B\ge 1$, by applying (32) with $\zeta=1, x=(Bm)^\epsilon$ and $c=\frac{\epsilon}{2AB^\epsilon}$,

$$A\log(Bm)=\frac{A}{\epsilon}\log((Bm)^\epsilon)\le\frac{A}{\epsilon}\log\left(\frac{2AB^\epsilon}{\mathrm{e}\epsilon}\right)+\frac{1}{2}m^\epsilon\le\frac{A}{\epsilon}\log\left(\frac{AB}{\epsilon}\right)+\frac{1}{2}m^\epsilon.\tag{70}$$

Using the above inequality with $A=\frac{9\kappa^2}{\|\mathcal{T}_\rho\|}$ and $B=\frac{1}{\delta_2}$, one can prove that the condition (68) ensures that $\frac{9\kappa^2}{m}\log\frac{m}{\delta_2}\le\lambda$ is satisfied with $\delta_2=\frac{\delta}{3}$, Therefore, by Lemma 19, (47) holds with probability at least $1-\delta_2$. Similarly the condition (68) implies that $m\ge 32\log^2\frac{2}{\delta_3}$ is satisfied with $\delta_3=\frac{\delta}{3}$, and thus by Lemma 25, (63) holds with probability at least $1-\delta_3$. Combining with Lemma 18, by taking the union bound, we know that with probability at least $1-\delta_1-\delta_2-\delta_3$, (47), (63) and (40) hold for all $k\in[T]$. Now, we can apply Propositions 4 and 6 to get (50) and (66). Noting that by (67), $\sqrt{2}\eta\le 1$, and by a simple calculation, we derive from (50) that

$$\begin{aligned}&\|\mathcal{S}_\rho\nu_{t+1}-\mathcal{S}_\rho\mu_{t+1}\|_\rho^2\\\le&\frac{4624(R\kappa^{2\zeta}+\sqrt{M})^2(\kappa/\sqrt{\|\mathcal{T}_\rho\|}+\sqrt{2\sqrt{v}c_\gamma/\|\mathcal{T}_\rho\|^\gamma})^2}{(1-\theta)^2}m^{\gamma(1-\epsilon)-1}(1\vee\lambda^2\eta^2\kappa^{-4}t^{2-2\theta}\vee\log^2 t)\log^2\frac{4}{\delta_1}\\\le&\frac{4624(R\kappa^{2\zeta}+\sqrt{M})^2(\kappa/\sqrt{\|\mathcal{T}_\rho\|}+\sqrt{2\sqrt{v}c_\gamma/\|\mathcal{T}_\rho\|^\gamma})^2}{(1-\theta)^2}m^{\gamma(1-\epsilon)-1}(1\vee\eta^2 m^{2\epsilon-2}t^{2-2\theta})(\log T)^2\log^2\frac{4}{\delta_1},\end{aligned}$$

where for the last inequality, we used $\|\mathcal{T}_\rho\|\le\kappa^2$. Similarly, by a simple calculation, we get from (66) that

$$\begin{aligned}\mathbb{E}_{\mathbf{J}}\|\mathcal{S}_\rho\omega_{t+1}-\mathcal{S}_\rho\nu_{t+1}\|_\rho^2&\le\frac{208Mv}{b(1-\theta)}(\eta t^{-\min(\theta,1-\theta)}\vee\lambda\eta^2\kappa^{-2}t^{(1-2\theta)_+})(1\vee\log t)\\&\le\frac{208Mv}{b(1-\theta)}(\eta t^{-\min(\theta,1-\theta)}\vee m^{\epsilon-1}\eta^2 t^{(1-2\theta)_+})\log T.\end{aligned}$$

Letting $\delta_1 = \frac{\delta}{3}$, and introducing the above estimates and (34) into (23), we get (69). The proof is complete. ■

Proof [of Theorem 1] By choosing $\epsilon = 1 - \frac{1}{2\zeta+\gamma}$ and $\theta = 0$ in Theorem 7, then the condition (68) reduces to $m \geq m_\delta$, where

$$m_\delta = \left(\frac{18\kappa^2 p}{\|\mathcal{T}_\rho\|}\log\left(\frac{27\kappa^2 p}{\|\mathcal{T}_\rho\|\delta}\right)\right)^p, \quad p = \frac{2\zeta+\gamma}{2\zeta+\gamma-1}. \tag{71}$$

The desired result thus follows by applying Theorem 7. ■

7.2 Non Attainable Case

For the non-attainable case, we have the following general results on generalization errors for SGM.

Theorem 8 *Under Assumptions 1, 2 and 3, let $\zeta \leq 1/2$, $T \in \mathbb{N}$ with $T \geq 3$, $\delta \in]0,1[$, $\eta_t = \eta\kappa^{-2}t^{-\theta}$ for all $t \in [T]$, with $\theta \in [0,1[$ and η such that (67) and for some $\epsilon \in]0,1]$, (68) holds. Then the following holds with probability at least $1-\delta$: for all $t \in [T]$,*

$$\begin{aligned}\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})] - \inf_H \mathcal{E} \lesssim &\left((\eta t^{1-\theta})^{-2\zeta} + m^{\gamma(1-\epsilon)-1}\right)(1 \vee \eta m^{\epsilon-1}t^{1-\theta})^3 \log^4 T \log^2\frac{1}{\delta} \\ &+\eta b^{-1}(t^{-\min(\theta,1-\theta)} \vee m^{\epsilon-1}\eta t^{(1-2\theta)_+})\log T.\end{aligned} \tag{72}$$

Here, the constant in the upper bounds is positive and depends only on $\kappa^2, \|\mathcal{T}_\rho\|, M, v, \zeta, R, c_\gamma$, γ and $\|f_{\mathcal{H}}\|_\infty$.

Proof The proof is similar to that for Theorem 7. We include the sketch only and omit the constants appeared. Similar to the proof of Theorem 7, with $\lambda = \|\mathcal{T}_\rho\|m^{\epsilon-1}$, one can prove that with probability at least $1-\delta_1-\delta_2-\delta_3$, (47), (63) and (41) hold for all $k \in [T]$. Now, we can apply Propositions 4 and 6 to get (51) and (66). Noting that by (65), $\sqrt{2}\eta \leq 1$, and by a simple calculation, we derive from (51) that

$$\begin{aligned}&\|\mathcal{S}_\rho\nu_{t+1} - \mathcal{S}_\rho\mu_{t+1}\|_\rho^2 \\ \lesssim& m^{\gamma(1-\epsilon)-1}(1 \vee \eta^2 m^{2\epsilon-2}t^{2-2\theta})\log^4 T\log^2\frac{1}{\delta} + (\eta t^{1-\theta})^{-2\zeta}(1 \vee \eta t^{1-\theta}m^{-1})(1 \vee \eta^2 m^{2\epsilon-2}t^{2-2\theta})\log^4 T\log^2\frac{1}{\delta} \\ \lesssim& m^{\gamma(1-\epsilon)-1}(1 \vee \eta^2 m^{2\epsilon-2}t^{2-2\theta})\log^4 T\log^2\frac{1}{\delta} + (\eta t^{1-\theta})^{-2\zeta}(1 \vee \eta m^{\epsilon-1}t^{1-\theta})^3\log^4 T\log^2\frac{1}{\delta}.\end{aligned}$$

The rest of the proof parallelizes to that for Theorem 7. ■

Now, we are in a position to prove Theorem 3.

Proof [of Theorem 3] The second part of the theorem follows directly from applying Theorem 8 with $\theta = 0$. The first part can be proved by applying Theorem 8 with $\theta = 0$ and $\epsilon = 1 - \frac{1}{2\zeta+\gamma}$, combining with the same argument from the proof of Theorem 1 to verify the condition (68). We omit the details. ■

7.3 Batch GM

Following the proof of Theorems 1 and 3, we know that the following results hold for batch GM, from which one can prove Theorem 4.

Theorem 9 *Under Assumptions 1, 2 and 3, set $\eta_t = \eta\kappa^{-2}$ with $\eta \leq 1$, for all $t \in [m]$. With probability at least $1-\delta$ $(0 < \delta < 1)$, the following holds for the learning sequence generated by (10):*
1) if $\zeta > 1/2$ and $m \geq m_\delta$ with m_δ given by (71), then

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})] - \inf_H \mathcal{E} \lesssim (\eta t)^{-2\zeta} + m^{-\frac{2\zeta}{2\zeta+\gamma}}(1 + m^{-\frac{1}{2\zeta+\gamma}}\eta t)^2\log^2 T\log^2\frac{1}{\delta}; \tag{73}$$

2) if $\zeta \leq 1/2$, $2\zeta+\gamma > 1$ and $m \geq m_\delta$ with m_δ given by (71), then

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})] - \inf_H \mathcal{E} \lesssim \left((\eta t)^{-2\zeta} + m^{-\frac{2\zeta}{2\zeta+\gamma}}\right)(1 \vee m^{-\frac{1}{2\zeta+\gamma}}\eta t)^3\log^4 T\log^2\frac{1}{\delta};$$

3) if $2\zeta+\gamma \leq 1$ and for some $\epsilon \in]0,1]$, (68) hold, then

$$\mathbb{E}_{\mathbf{J}}[\mathcal{E}(\omega_{t+1})] - \inf_H \mathcal{E} \lesssim \left((\eta t)^{-2\zeta} + m^{\gamma(1-\epsilon)-1}\right)(1 \vee \eta m^{\epsilon-1}t)^3\log^4 T\log^2\frac{1}{\delta}.$$

Here, all the constants in the upper bounds are positive and depend only on $\kappa^2, \|\mathcal{T}_\rho\|, M, v, \zeta, R, c_\gamma$ and γ (and also on $\|f_{\mathcal{H}}\|_\infty$ when $\zeta < 1/2$) .

8. Convergence in H-norm

In this section, we will give convergence results in H-norm for Algorithm 1 in the attainable case. For the sake of simplicity, we will only consider a fixed step-size sequence, i.e, $\eta_t = \eta$ for all t.

Using a similar procedure as that for (23), we can prove the following error decomposition,

$$\mathbb{E}_{\mathbf{J}}[\|\omega_t - \omega^\dagger\|_H^2] \lesssim \|\mu_t - \omega^\dagger\|_H^2 + \|\mu_t - \nu_t\|_H^2 + \mathbb{E}_{\mathbf{J}}[\|\omega_t - \nu_t\|_H^2]. \tag{74}$$

To estimate the bias term, $\|\mu_t - \omega^\dagger\|_H^2$, we introduce the following lemma from (Yao et al., 2007; Rosasco and Villa, 2015). Its proof is similar as that for (34) and will be given in Appendix C for the sake of completeness.

Lemma 27 *Under Assumption 2, let $\zeta \geq 1/2$ and $\eta_t = \eta$ for all $t \in \mathbb{N}$, with $\eta \in]0, \kappa^{-2}]$, then*

$$\|\mu_{t+1} - \omega^\dagger\|_H \leq R\left(\frac{\zeta-1/2}{\eta t}\right)^{\zeta-1/2}. \tag{75}$$

To estimate the sample variance term, $\|\mu_t - \nu_t\|_H^2$, we use (39) and get that

$$\begin{aligned}\|\nu_{t+1} - \mu_{t+1}\|_H &= \left\|\sum_{k=1}^t \eta_k \Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\right\|_H \\ \leq \sum_{k=1}^t \eta_k \left\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}\right\| \left\|\mathcal{T}_{\rho,\lambda}^{-\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\right\|_H &\leq \frac{1}{\sqrt{\lambda}}\sum_{k=1}^t \eta_k \left\|\mathcal{T}_{\rho,\lambda}^{\frac{1}{2}}\Pi_{k+1}^t(\mathcal{T}_{\mathbf{x}})N_k\right\|_H.\end{aligned}$$

From the proof of Theorem 5, we know that $\sum_{k=1}^{t} \eta_k \left\| \mathcal{T}_{\rho,\lambda}^{\frac{1}{2}} \Pi_{k+1}^{t}(\mathcal{T}_{\mathbf{x}}) N_k \right\|_H$ is upper bounded by the right-hand side of (50). With $\eta_t = \eta$ and $\lambda = \|\mathcal{T}_\rho\| m^{-\frac{1}{2\zeta+\gamma}}$, we thus have

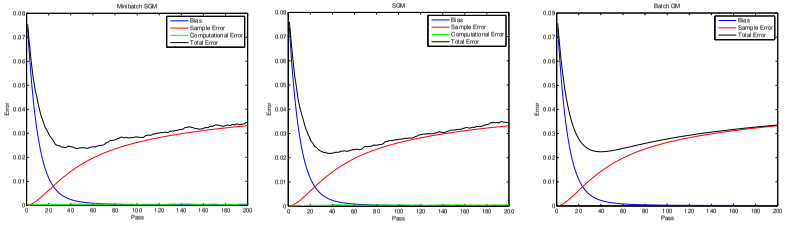
$$\|\nu_{t+1} - \mu_{t+1}\|_H \lesssim m^{-\frac{\zeta-1/2}{2\zeta+\gamma}} (1 + m^{-\frac{1}{2\zeta+\gamma}} \eta t) \log t \log \frac{1}{\delta}. \tag{76}$$

Finally, for the computational variance term, $\mathbb{E}_{\mathbf{J}}[\|\omega_t - \nu_t\|_H^2]$, we use a same procedure as that for (52) to get

$$\mathbb{E}_{\mathbf{J}} \|\omega_{t+1} - \nu_{t+1}\|_H^2 \leq \frac{\kappa^2}{b} \sum_{k=1}^{t} \eta^2 \left\| \Pi_{k+1}^{t}(\mathcal{T}_{\mathbf{x}}) \right\|^2 \mathbb{E}_{\mathbf{J}}[\mathcal{E}_{\mathbf{z}}(\omega_k)] \lesssim \frac{\eta^2 t}{b}, \tag{77}$$

where we used (62) and (63) in the last inequality. Introducing (75), (76) and (77) into the error decomposition (74), we can prove Theorem 2.

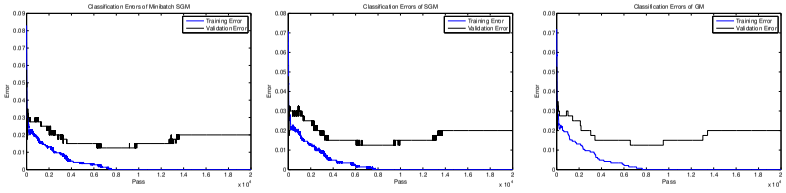
9. Numerical Simulations



(a) Minibatch SGM (b) SGM (c) Batch GM

Figure 1: Error decompositions for gradient-based learning algorithms on *synthesis data*, where $m = 100$.

In order to illustrate our theoretical results and the error decomposition, we first performed some simulations on a simple problem. We constructed $m = 100$ i.i.d. training examples of the form $y = f_\rho(x_i) + \omega_i$. Here, the regression function is $f_\rho(x) = |x - 1/2| - 1/2$, the input point x_i is



(a) Minibatch SGM (b) SGM (c) Batch GM

Figure 2: Misclassification Errors for gradient-based learning algorithms on *BreastCancer* dataset.

uniformly distributed in $[0, 1]$, and ω_i is a Gaussian noise with zero mean and standard deviation 1, for each $i \in [m]$. We perform three experiments with the same H, a RKHS associated with a Gaussian kernel $K(x, x') = \exp(-(x-x')^2/(2\sigma^2))$ where $\sigma = 0.2$. In the first experiment, we run mini-batch SGM, where the mini-batch size $b = \sqrt{m}$, and the step-size $\eta_t = 1/(8\sqrt{m})$. In the second experiment, we run simple SGM where the step-size is fixed as $\eta_t = 1/(8m)$, while in the third experiment, we run batch GM using the fixed step-size $\eta_t = 1/8$. For mini-batch SGM and SGM, the total error $\|\mathcal{S}_\rho \omega_t - f_\rho\|_{L^2_{\hat{\rho}}}^2$, the bias $\|\mathcal{S}_\rho \hat{\mu}_t - f_\rho\|_{L^2_{\hat{\rho}}}^2$, the sample variance $\|\mathcal{S}_\rho \nu_t - \mathcal{S}_\rho \hat{\mu}_t\|_{L^2_{\hat{\rho}}}^2$ and the computational variance $\|\mathcal{S}_\rho \omega_t - \mathcal{S}_\rho \nu_t\|_{L^2_{\hat{\rho}}}^2$, averaged over 50 trials, are depicted in Figures 1a and 1b, respectively. For batch GM, the total error $\|\mathcal{S}_\rho \nu_t - f_\rho\|_{L^2_{\hat{\rho}}}^2$, the bias $\|\mathcal{S}_\rho \hat{\mu}_t - f_\rho\|_{L^2_{\hat{\rho}}}^2$ and the sample variance $\|\mathcal{S}_\rho \nu_t - \hat{\mu}_t\|_{L^2_{\hat{\rho}}}^2$, averaged over 50 trials are depicted in Figure 1c. Here, we replace the unknown marginal distribution ρ_X by an empirical measure $\hat{\rho} = \frac{1}{2000} \sum_{i=1}^{2000} \delta_{\hat{x}_i}$, where each $\hat{x}_i$ is uniformly distributed in $[0, 1]$. From Figure 1a or 1b, we see that as the number of passes increases[5], the bias decreases, while the sample error increases. Furthermore, we see that in comparisons with the bias and the sample error, the computational error is negligible. In all these experiments, the minimal total error is achieved when the bias and the sample error are balanced. These empirical results show the effects of the three terms from the error decomposition, and complement the derived bound (11), as well as the regularization effect of the number of passes over the data. Finally, we tested the simple SGM, mini-batch SGM, and batch GM, using similar step-sizes as those in the first simulation, on the *BreastCancer* dataset[6]. The classification errors on the training set and the testing set of these three algorithms are depicted in Figure 2. We see that all of these algorithms perform similarly, which complement the bounds in Corollaries 3, 4 and 7.

Acknowledgments

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216. L. R. acknowledges the financial support of the Italian Ministry of Education, University and Research FIRB project RBFR12M3AC.

Appendices

A. Learning with Kernel Methods

Let the input space Ξ be a closed subset of Euclidean space $\mathbb{R}^d$, the output space $Y \subseteq \mathbb{R}$. Let μ be an unknown but fixed Borel probability measure on $\Xi \times Y$. Assume that $\{(\xi_i, y_i)\}_{i=1}^m$ are i.i.d. from the distribution μ. A reproducing kernel K is a symmetric function $K : \Xi \times \Xi \to \mathbb{R}$ such that $(K(u_i, u_j))_{i,j=1}^\ell$ is positive semidefinite for any finite set of points $\{u_i\}_{i=1}^\ell$ in Ξ. The kernel K defines a reproducing kernel Hilbert space (RKHS) $(\mathcal{H}_K, \|\cdot\|_K)$ as the completion of the linear span of the set $\{K_\xi(\cdot) := K(\xi, \cdot) : \xi \in \Xi\}$ with respect to the inner product $\langle K_\xi, K_u \rangle_K := K(\xi, u)$. For any $f \in \mathcal{H}_K$, the reproducing property holds: $f(\xi) = \langle K_\xi, f \rangle_K$.

5. Note that the terminology 'running the algorithm with p passes' means 'running the algorithm with $\lceil mp/b \rceil$ iterations', where b is the mini-batch size.
6. `https://archive.ics.uci.edu/ml/datasets/`

Example 1 (Sobolev Spaces) *Let $X = [0, 1]$ and the kernel*

$$K(x, x') = \begin{cases} (1-y)x, & x \le y; \\ (1-x)y, & x \ge y. \end{cases}$$

Then the kernel induces a Sobolev Space $H = \{f : X \to \mathbb{R} | f$ is absolutely continuous , $f(0) = f(1) = 0, f \in L^2(X)\}$.

In learning with kernel methods, one considers the following minimization problem

$$\inf_{f \in \mathcal{H}_K} \int_{\Xi \times Y} (f(\xi) - y)^2 d\mu(\xi, y).$$

Since $f(\xi) = \langle K_\xi, f \rangle$ by the reproducing property, the above can be rewritten as

$$\inf_{f \in \mathcal{H}_K} \int_{\Xi \times Y} (\langle f, K_\xi \rangle - y)^2 d\mu(\xi, y).$$

Letting $X = \{K_\xi : \xi \in \Xi\}$ and defining another probability measure $\rho(K_\xi, y) = \mu(\xi, y)$, the above reduces to the learning setting in Section 2.

B. Further Corollaries for SGM in the non-attainable case

In this section, we state the convergence results for the SGM with different parameter choices similar as those in Corollaries 5–7, in the non-attainable case. These results are direct consequences of Theorem 3.

Corollary 28 *Under Assumptions 1, 2 and 3, let $\zeta \le 1/2$, $\delta \in]0, 1[$, $b = 1$, and $\eta_t \simeq m^{-\frac{2\zeta}{(2\zeta+\gamma)\vee 1}}$ for all $t \in [m^2]$. With probability at least $1 - \delta$, the following holds:*
1) if $2\zeta + \gamma > 1$, $m \ge m_\delta$ and $T^* = \lceil m^{\frac{2\zeta+1}{2\zeta+\gamma}} \rceil$, then we have (18);
2) if $2\zeta + \gamma \le 1$, and for some $\epsilon \in]0, 1[$, $m \ge m_{\delta,\epsilon}$, and $T^* = \lceil m^{1+2\zeta-\epsilon} \rceil$, then we have (19).

Corollary 29 *Under Assumptions 1, 2 and 3, let $\zeta \le 1/2$, $\delta \in]0, 1[$, $b \simeq m^{\frac{2\zeta}{(2\zeta+\gamma)\vee 1}}$, and $\eta_t \simeq \frac{1}{\log m}$ for all $t \in [m]$. With probability at least $1 - \delta$, the following holds:*
1) if $2\zeta + \gamma > 1$, $m \ge m_\delta$ and $T^* = \lceil m^{\frac{1}{2\zeta+\gamma}} \rceil$, then we have (18);
2) if $2\zeta + \gamma \le 1$, and for some $\epsilon \in]0, 1[$, $m \ge m_{\delta,\epsilon}$, and $T^* = \lceil m^{1-\epsilon} \rceil$, then we have (19).

Corollary 30 *Under Assumptions 1, 2 and 3, let $\zeta \le 1/2$, $\delta \in]0, 1[$, $b = m$ and $\eta_t \simeq \frac{1}{\log m}$ for all $t \in [m]$. With probability at least $1 - \delta$, the following holds:*
1) if $2\zeta + \gamma > 1$, $m \ge m_\delta$ and $T^* = \lceil m^{\frac{1}{2\zeta+\gamma}} \rceil$, then we have (18);
2) if $2\zeta + \gamma \le 1$, and for some $\epsilon \in]0, 1[$, $m \ge m_{\delta,\epsilon}$, and $T^* = \lceil m^{1-\epsilon} \rceil$, then we have (19).

C. Proofs for Lemmas

Proof [of Lemma 12] Note that

$$\sum_{k=1}^{t} k^{-\theta} \le 1 + \sum_{k=2}^{t} \int_{k-1}^{k} u^{-\theta} du = 1 + \int_1^t u^{-\theta} du = \frac{t^{1-\theta} - \theta}{1 - \theta},$$

which leads to the first part of the desired result. Similarly,

$$\sum_{k=1}^{t} k^{-\theta} \ge \sum_{k=1}^{t} \int_{k}^{k+1} u^{-\theta} du = \int_1^{t+1} u^{-\theta} du = \frac{(t+1)^{1-\theta} - 1}{1 - \theta},$$

and by mean value theorem, $(t+1)^{1-\theta} - 1 \ge (1-\theta)t(t+1)^{-\theta} \ge (1-\theta)t^{1-\theta}/2$. This proves the second part of the desired result. The proof is complete. ■

Proof [of Lemma 13] Note that

$$\sum_{k=1}^{t} k^{-\theta} = \sum_{k=1}^{t} k^{-1} k^{1-\theta} \le t^{\max(1-\theta, 0)} \sum_{k=1}^{t} k^{-1},$$

and

$$\sum_{k=1}^{t} k^{-1} \le 1 + \sum_{k=2}^{t} \int_{k-1}^{k} u^{-1} du = 1 + \log t.$$

■

Proof [of Lemma 14] Note that

$$\sum_{k=1}^{t-1} \frac{1}{t-k} k^{-q} = \sum_{k=1}^{t-1} \frac{k^{1-q}}{(t-k)k} \le t^{\max(1-q, 0)} \sum_{k=1}^{t-1} \frac{1}{(t-k)k},$$

and that by Lemma 13,

$$\sum_{k=1}^{t-1} \frac{1}{(t-k)k} = \frac{1}{t} \sum_{k=1}^{t-1} \left(\frac{1}{t-k} + \frac{1}{k} \right) = \frac{2}{t} \sum_{k=1}^{t-1} \frac{1}{k} \le \frac{2}{t}(1 + \log t).$$

■

Proof [of Proposition 2] Since μ_{t+1} is given by (25), introducing with (27),

$$\mu_{t+1} = \mu_t - \eta_t (\mathcal{T}_\rho \mu_t - \mathcal{S}_\rho^* f_\mathcal{H}). \tag{78}$$

Thus,

$$\mathcal{S}_\rho \mu_{t+1} = \mathcal{S}_\rho \mu_t - \eta_t \mathcal{S}_\rho (\mathcal{T}_\rho \mu_t - \mathcal{S}_\rho^* f_\mathcal{H}) = \mathcal{S}_\rho \mu_t - \eta_t \mathcal{L}_\rho (\mathcal{S}_\rho \mu_t - f_\mathcal{H}). \tag{79}$$

Subtracting both sides by $f_\mathcal{H}$,

$$\mathcal{S}_\rho \mu_{t+1} - f_\mathcal{H} = (I - \eta_t \mathcal{L}_\rho)(\mathcal{S}_\rho \mu_t - f_\mathcal{H}).$$

Using this equality iteratively, with $\mu_1 = 0$,

$$\mathcal{S}_\rho \mu_{t+1} - f_\mathcal{H} = -\Pi_1^t (\mathcal{L}_\rho) f_\mathcal{H}.$$

Taking the $L^2(H, \rho_X)$-norm, by Assumption 2,

$$\|\mathcal{S}_\rho \mu_{t+1} - f_\mathcal{H}\|_\rho = \|\Pi_1^t (\mathcal{L}_\rho) f_\mathcal{H}\|_\rho \le \|\Pi_1^t (\mathcal{L}_\rho) \mathcal{L}_\rho^\zeta\| R.$$

By applying Lemma 15, we get (33). Combining (33) with Lemma 12, we get (34). The proof is complete. ■

Proof [of Lemma 16] From (78), we have

$$\mu_{t+1} = (I - \eta_t \mathcal{T}_\rho)\mu_t + \eta_t \mathcal{S}_\rho^* f_{\mathcal{H}}.$$

Applying this relationship iteratively, and using $\mu_1 = 0$, we get

$$\mu_{t+1} = \sum_{k=1}^{t} \eta_k \Pi_{k+1}^t(\mathcal{T}_\rho)\mathcal{S}_\rho^* f_{\mathcal{H}} = \sum_{k=1}^{t} \eta_k \mathcal{S}_\rho^* \Pi_{k+1}^t(\mathcal{L}_\rho) f_{\mathcal{H}}.$$

Therefore, using Assumption 2 and spectral theory,

$$\|\mu_{t+1}\|_H \leq \left\| \sum_{k=1}^{t} \eta_k \mathcal{S}_\rho^* \Pi_{k+1}^t(\mathcal{L}_\rho)\mathcal{L}_\rho^\zeta \right\| R \leq R \max_{\sigma \in]0,\kappa^2]} \sigma^{1/2+\zeta} \sum_{k=1}^{t} \eta_k \Pi_{k+1}^t(\sigma).$$

Case $\zeta \geq 1/2$. For any $\sigma \in]0, \kappa^2]$,

$$\sigma^{1/2+\zeta} \sum_{k=1}^{t} \eta_k \Pi_{k+1}^t(\sigma) \leq \kappa^{2\zeta-1}\sigma \sum_{k=1}^{t} \eta_k \Pi_{k+1}^t(\sigma) \leq \kappa^{2\zeta-1},$$

where for the last inequality, we used

$$\sum_{k=1}^{t} \eta_k \sigma \Pi_{k+1}^t(\sigma) = \sum_{k=1}^{t} (1-(1-\eta_k\sigma))\Pi_{k+1}^t(\sigma) = \sum_{k=1}^{t} \Pi_{k+1}^t(\sigma) - \sum_{k=1}^{t} \Pi_k^t(\sigma) = 1 - \Pi_1^t(\sigma). \quad (80)$$

Thus,

$$\|\mu_{t+1}\|_H \leq R\kappa^{2\zeta-1}.$$

Case $\zeta < 1/2$. If $\sum_{k=1}^t \eta_k \leq \kappa^{-2}$, then for any $\sigma \leq \kappa^2$,

$$\sigma^{1/2+\zeta} \sum_{k=1}^{t} \eta_k \Pi_{k+1}^t(\sigma) \leq \sigma^{1/2+\zeta} \sum_{k=1}^{t} \eta_k \leq \kappa^{2\zeta-1}.$$

If $\sum_{k=1}^t \eta_k > \kappa^{-2}$, then for any $\sigma \leq (\sum_{k=1}^t \eta_k)^{-1}$,

$$\sigma^{1/2+\zeta} \sum_{k=1}^{t} \eta_k \Pi_{k+1}^t(\sigma) \leq \sigma^{1/2+\zeta} \sum_{k=1}^{t} \eta_k \leq \left(\sum_{k=1}^{t} \eta_k \right)^{1/2-\zeta},$$

while for $\kappa^2 \geq \sigma \geq (\sum_{k=1}^t \eta_k)^{-1}$, by (80),

$$\sigma^{1/2+\zeta} \sum_{k=1}^{t} \eta_k \Pi_{k+1}^t(\sigma) = \sigma^{\zeta-1/2} \sum_{k=1}^{t} \eta_k \sigma \Pi_{k+1}^t(\sigma) \leq \sigma^{\zeta-1/2} \leq \left(\sum_{k=1}^{t} \eta_k \right)^{1/2-\zeta}.$$

From the above analysis, we get that

$$\max_{\sigma \in]0,\kappa^2]} \sigma^{1/2+\zeta} \sum_{k=1}^{t} \eta_k \Pi_{k+1}^t(\sigma) \leq \kappa^{2\zeta-1} \vee \left(\sum_{k=1}^{t} \eta_k \right)^{1/2-\zeta},$$

and thus

$$\|\mu_{t+1}\|_H \leq R \left\{ \kappa^{2\zeta-1} \vee \left(\sum_{k=1}^{t} \eta_k \right)^{1/2-\zeta} \right\}.$$

The proof is complete. ■

Proof [of Lemma 18 (1)] *Bounding* $\left\| (\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}} \left(\mathcal{S}_\rho^* f_\rho - \mathcal{S}_{\mathbf{x}}^* \mathbf{y} \right) \right\|_H$*:*
For all $i \in [m]$, let $w_i = y_i (\mathcal{T}_\rho + \lambda I)^{-\frac{1}{2}} x_i$. Obviously, from the definitions of f_ρ (see (6)) and $\mathcal{S}_\rho$,

$$\mathbb{E}[w_1] = \mathbb{E}_{x_1}[f_\rho(x_1)(\mathcal{T}_\rho + \lambda I)^{-\frac{1}{2}} x_1] = (\mathcal{T}_\rho + \lambda I)^{-\frac{1}{2}} \mathcal{S}_\rho^* f_\rho.$$

Thus,

$$(\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}} \left(\mathcal{S}_\rho^* f_\rho - \mathcal{S}_{\mathbf{x}}^* \mathbf{y} \right) = \frac{1}{m} \sum_{i=1}^{m} (\mathbb{E}[w_i] - w_i).$$

We next estimate the constants B and $\sigma^2(w_1)$ in (28). Note that for any $l \geq 2$,

$$\mathbb{E}[\|w_1 - \mathbb{E}[w_1]\|_H^l] \leq \mathbb{E}[(\|w_1\|_H + \mathbb{E}[\|w_1\|_H])^l].$$

By using Hölder's inequality twice,

$$\mathbb{E}[\|w_1 - \mathbb{E}[w_1]\|_H^l] \leq 2^{l-1}\mathbb{E}[\|w_1\|_H^l + (\mathbb{E}[\|w_1\|_H])^l] \leq 2^{l-1}\mathbb{E}[\|w_1\|_H^l + \mathbb{E}[\|w_1\|_H^l]].$$

The right-hand side is exactly $2^l \mathbb{E}[\|w_1\|_H^l]$. Therefore, by recalling the definition of w_1 and expanding the integration,

$$\mathbb{E}[\|w_1 - \mathbb{E}[w_1]\|_H^l] \leq 2^l \int_X \|(\mathcal{T}_\rho + \lambda I)^{-\frac{1}{2}} x\|_H^l \int_Y y^l d\rho(y|x) d\rho_X(x). \quad (81)$$

Introducing (42) and (45) into the above inequality, we have

$$\mathbb{E}[\|w_1 - \mathbb{E}[w_1]\|_H^l] \leq l!(2\sqrt{M})^l \sqrt{v} \left(\frac{\kappa}{\sqrt{\lambda}} \right)^{l-2} c_\gamma \lambda^{-\gamma} = \frac{1}{2} l! \left(\frac{2\kappa\sqrt{M}}{\sqrt{\lambda}} \right)^{l-2} 8M\sqrt{v} c_\gamma \lambda^{-\gamma}.$$

Applying Bernstein inequality with $B = \frac{2\kappa\sqrt{M}}{\sqrt{\lambda}}$ and $\sigma = \sqrt{8M\sqrt{v}c_\gamma\lambda^{-\gamma}}$, we get that with probability at least $1 - \frac{\delta_1}{2}$, there holds

$$\left\| (\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}} \left(\mathcal{S}_\rho^* f_\rho - \mathcal{S}_{\mathbf{x}}^* \mathbf{y} \right) \right\|_H = \left\| \frac{1}{m} \sum_{i=1}^{m} (\mathbb{E}[w_i] - w_i) \right\|_H \leq 4\sqrt{M} \left(\frac{\kappa}{m\sqrt{\lambda}} + \frac{\sqrt{2\sqrt{v}c_\gamma}}{\sqrt{m\lambda^\gamma}} \right) \log \frac{4}{\delta_1}. \quad (82)$$

Bounding $\|(\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}})\|$*:*
Let $\xi_i = (\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}} x_i \otimes x_i$, for all $i \in [m]$. It is easy to see that $\mathbb{E}[\xi_i] = (\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}} \mathcal{T}_\rho$, and that $(\mathcal{T}_\rho + \lambda)^{-\frac{1}{2}}(\mathcal{T}_\rho - \mathcal{T}_{\mathbf{x}}) = \frac{1}{m} \sum_{i=1}^m (\mathbb{E}[\xi_i] - \xi_i)$. Denote the Hilbert-Schmidt norm of a bounded operator from H to H by $\|\cdot\|_{HS}$. Note that

$$\|\xi_1\|_{HS}^2 = \|x_1\|_H^2 \mathrm{Trace}((\mathcal{T}_\rho + \lambda)^{-1/2} x_1 \otimes x_1 (\mathcal{T}_\rho + \lambda)^{-1/2}) = \|x_1\|_H^2 \mathrm{Trace}((\mathcal{T}_\rho + \lambda)^{-1} x_1 \otimes x_1).$$

By Assumption (3),

$$\|\xi_1\|_{HS} \leq \sqrt{\kappa^2 \mathrm{Trace}((\mathcal{T}_\rho+\lambda)^{-1}x_1\otimes x_1)} \leq \sqrt{\kappa^2 \mathrm{Trace}(x_1\otimes x_1)/\lambda} \leq \kappa^2/\sqrt{\lambda},$$

and furthermore, by Assumption 3,

$$\mathbb{E}[\|\xi_1\|_{HS}^2] \leq \kappa^2 \mathbb{E}\mathrm{Trace}((\mathcal{T}_\rho+\lambda)^{-1}x_1\otimes x_1) = \kappa^2 \mathrm{Trace}((\mathcal{T}_\rho+\lambda)^{-1}\mathcal{T}_\rho) \leq \kappa^2 c_\gamma \lambda^{-\gamma}.$$

According to Lemma 11, we get that with probability at least $1-\frac{\delta_1}{2}$, there holds

$$\|(\mathcal{T}_\rho+\lambda)^{-\frac{1}{2}}(\mathcal{T}_\rho-\mathcal{T}_{\mathbf{x}})\|_{HS} \leq 2\kappa\left(\frac{2\kappa}{m\sqrt{\lambda}}+\frac{\sqrt{c_\gamma}}{\sqrt{m\lambda^\gamma}}\right)\log\frac{4}{\delta_1}. \tag{83}$$

Finally, using the triangle inequality, we have,

$$\|(\mathcal{T}_\rho+\lambda)^{-\frac{1}{2}}N_k\|_H \leq \|(\mathcal{T}_\rho+\lambda)^{-\frac{1}{2}}(\mathcal{T}_\rho-\mathcal{T}_{\mathbf{x}})\|\|\mu_k\|_H + \left\|(\mathcal{T}_\rho+\lambda)^{-\frac{1}{2}}\left(\mathcal{S}_\rho^* f_\rho - \mathcal{S}_{\mathbf{x}}^*\mathbf{y}\right)\right\|_H.$$

Applying (35) to the above, introducing with (82) and (83), and then noting that $\kappa \geq 1$ and $v \geq 1$, one can prove the first part of the lemma.
■

Proof [of Lemma 27] Obviously, $f_{\mathcal{H}} = \mathcal{S}_\rho \omega^\dagger$ and thus $\mathcal{T}_\rho\omega^\dagger = \mathcal{S}_\rho^* f_{\mathcal{H}}$. Combining with Assumption 2, $\mathcal{T}_\rho\omega^\dagger = \mathcal{S}_\rho^*\mathcal{L}_\rho^\zeta\mathcal{L}_\rho^{-\zeta}f_{\mathcal{H}} = \mathcal{T}_\rho^\zeta\mathcal{S}_\rho^*\mathcal{L}_\rho^{-\zeta}f_{\mathcal{H}}$, and $\omega^\dagger = \mathcal{T}_\rho^\dagger\mathcal{T}_\rho^\zeta\mathcal{S}_\rho^*\mathcal{L}_\rho^{-\zeta}f_{\mathcal{H}}$. Subtracting $\omega^\dagger$ from both sides of (78), and using $\mathcal{S}_\rho^* f_{\mathcal{H}} = \mathcal{T}_\rho\omega^\dagger$, we know that

$$\mu_{t+1}-\omega^\dagger = (I-\eta_t\mathcal{T}_\rho)(\mu_t-\omega^\dagger).$$

Applying this relationship iteratively, with $\mu_1 = 0$,

$$\mu_{t+1}-\omega^\dagger = -\Pi_1^t(\mathcal{T}_\rho)\omega^\dagger = -\Pi_1^t(\mathcal{T}_\rho)\mathcal{T}_\rho^\dagger\mathcal{T}_\rho^\zeta\mathcal{S}_\rho^*\mathcal{L}_\rho^{-\zeta}f_{\mathcal{H}}.$$

Therefore,

$$\|\mu_{t+1}-\omega^\dagger\|_H \leq \|\Pi_1^t(\mathcal{T}_\rho)\mathcal{T}_\rho^\dagger\mathcal{T}_\rho^\zeta\mathcal{S}_\rho^*\|R \leq \|\Pi_1^t(\mathcal{T}_\rho)\mathcal{T}_\rho^{\zeta-1/2}\|R.$$

Applying Lemma 15, one can get the desired result. ■

D. List of Some Notations

Notation	Meaning
H	the hypothesis space
X, Y, Z	the input space, the output space and the sample space ($Z = X\times Y$)
ρ	the fixed probability measure on Z
ρ_X	the induced marginal measure of ρ on X
$\rho(\cdot\|x)$	the conditional probability measure on Y w.r.t. $x\in X$ and ρ
$\mathbf{z}$	the sample $\{z_i=(x_i,y_i)\}_{i=1}^m$ of size $m\in\mathbb{N}$, where each z_i is i.i.d. according to ρ.
m	the sample size of the sample $\mathbf{z}$
$\mathcal{E}$	the expected risk defined by (1)
$\mathcal{E}_{\mathbf{z}}$	the empirical risk w.r.t the sample $\mathbf{z}$ defined by (20)
κ^2	the constant from the bounded assumption (3) on the hypothesis space H
$\{\omega_t\}_t$	the sequence generated by the SGM
θ	the decaying rate on step-sizes
b	the minibatch size of the SGM
T	the maximal number of iterations for the SGM
j_i (j_t etc.)	the random index from the uniform distribution on $[m]$ for the SGM
$\mathbf{J}_t$	the set of random indices at t-th iteration of the SGM
$\mathbf{J}$	the set of all random indices for the SGM after T iterations
$\mathbb{E}_{\mathbf{J}}$	the expectation with respect to the random variables $\mathbf{J}$ (conditional on $\mathbf{z}$)
$\{\eta_t\}_t$	the sequence of step-sizes
M, v	the positive constants from the moment (bounded) assumption on the output
$L^2(H,\rho_X)$	the Hilbert space of square integral functions from H to $\mathbb{R}$ with respect to ρ_X
f_ρ	the regression function defined (6)
H_ρ	$\{f: X\to\mathbb{R}\|\exists\omega\in H$ with $f(x)=\langle\omega,x\rangle_H, \rho_X$-almost surely$\}$
ζ, R	the parameters related to the 'regularity' of $f_{\mathcal{H}}$ (see Assumption 2)
$\omega^\dagger$	the solution of Problem (1) with the minimal norm in the attainable case
γ, c_γ	the parameters related to the effective dimension (see Assumption 3)
$\{\sigma_i\}_i$	the sequence of eigenvalues of $\mathcal{L}_\rho$
$\{\nu_t\}_t$	the sequence generated by the batch GM (10)
$\{\mu_k\}_k$	the sequence defined by the population iteration (21)
$\mathcal{S}_\rho$	the linear map from $H\to L^2(H,\rho_X)$ defined by $\mathcal{S}_\rho\omega = \langle\omega,\cdot\rangle$
$\mathcal{S}_\rho^*$	the adjoint operator of $\mathcal{S}_\rho$, $\mathcal{S}_\rho^* f = \int_X f(x)x d\rho_X(x)$
$\mathcal{L}_\rho$	the operator from $L^2(H,\rho_X)$ to $L^2(H,\rho_X)$, $\mathcal{L}_\rho(f) = \mathcal{S}_\rho\mathcal{S}_\rho^* f = \int_X\langle x,\cdot\rangle_H f(x)\rho_X(x)$
$\mathcal{T}_\rho$	the covariance operator from H to H, $\mathcal{T}_\rho = \mathcal{S}_\rho^*\mathcal{S}_\rho = \int_X\langle\cdot,x\rangle x d\rho_X(x)$
$\mathcal{S}_{\mathbf{x}}$	the sampling operator from H to $\mathbb{R}^m$, $(\mathcal{S}_{\mathbf{x}}\omega)_i = \langle\omega,x_i\rangle_H, i\in[m]$
$\mathcal{S}_{\mathbf{x}}^*$	the adjoint operator of $\mathcal{S}_{\mathbf{x}}$, $\mathcal{S}_{\mathbf{x}}^*\mathbf{y} = \frac{1}{m}\sum_{i=1}^m y_i x_i$
$\mathcal{T}_{\mathbf{x}}$	the empirical covariance operator, $\mathcal{T}_{\mathbf{x}} = \mathcal{S}_{\mathbf{x}}^*\mathcal{S}_{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^m\langle\cdot,x_i\rangle x_i$
$\Pi_{t+1}^T(L)$	$=\Pi_{k=t+1}(I-\eta_k L)$ when $t\in[T-1]$ and $\Pi_{T+1}^T = I$
$\sum_{i=t+1}^t\eta_i$	$=0$
λ	a 'regularization' parameter, $\lambda>0$
$\mathcal{T}_{\rho,\lambda}$,	$\mathcal{T}_{\rho,\lambda} = \mathcal{T}_\rho+\lambda$
$\mathcal{T}_{\mathbf{x},\lambda}$,	$\mathcal{T}_{\mathbf{x},\lambda} = \mathcal{T}_{\mathbf{x}}+\lambda$
$\{N_k\}_k$	the sequence defined by (38).
$M_{k,i}$	defined by (53)

References

Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate O(1/n). In *Advances in Neural Information Processing Systems*, pages 773–781, 2013.

Frank Bauer, Sergei Pereverzev, and Lorenzo Rosasco. On regularization algorithms in learning theory. *Journal of Complexity*, 23(1):52–72, 2007.

Gilles Blanchard and Nicole Mücke. Optimal rates for regularization of statistical inverse learning problems. *arXiv preprint arXiv:1604.04054*, 2016.

Olivier Bousquet and Léon Bottou. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, pages 161–168, 2008.

Stephen Boyd and Almir Mutapcic. Stochastic subgradient methods. Notes for EE364b, Standford University, Winter 2007.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Andrea Caponnetto and Yuan Yao. Cross-validation based adaptation for regularization operators in learning theory. *Analysis and Applications*, 8(02):161–183, 2010.

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004. ISSN 0018-9448.

Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *Advances in Neural Information Processing Systems*, pages 1647–1655, 2011.

Felipe Cucker and Ding-Xuan Zhou. *Learning Theory: an Approximation Theory Viewpoint*, volume 24. Cambridge University Press, 2007.

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(1):165–202, 2012.

Lee H Dicker, Dean P Foster, Daniel Hsu, et al. Kernel ridge vs. principal component regression: Minimax bounds and the qualification of regularization operators. *Electronic Journal of Statistics*, 11(1):1022–1047, 2017.

Aymeric Dieuleveut and Francis Bach. Non-parametric stochastic approximation with large step sizes. *Annals of Statistics*, 44(4):1363–1399, 2016.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, 2016.

Junhong Lin and Lorenzo Rosasco. Optimal learning for multi-pass stochastic gradient methods. In *Advances In Neural Information Processing Systems 29*, pages 4556–4564. 2016.

Junhong Lin, Raffaello Camoriano, and Lorenzo Rosasco. Generalization properties and implicit regularization of multiple passes SGM. In *International Conference on Machine Learning*, 2016a.

Junhong Lin, Lorenzo Rosasco, and Ding-Xuan Zhou. Iterative regularization for learning with convex loss functions. *Journal of Machine Learning Research*, 17(77):1–38, 2016b.

Stanislav Minsker. On some extensions of bernstein's inequality for self-adjoint operators. *arXiv preprint arXiv:1112.5448*, 2011.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Andrew Ng. Machine learning. Coursera, Standford University, 2016.

Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2014.

IF Pinelis and AI Sakhanenko. Remarks on inequalities for large deviation probabilities. *Theory of Probability & Its Applications*, 30(1):143–148, 1986.

Boris T Poljak. *Introduction to Optimization*. Optimization Software, 1987.

Lorenzo Rosasco and Silvia Villa. Learning with incremental iterative regularization. In *Advances in Neural Information Processing Systems*, pages 1621–1629, 2015.

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1648–1656, 2015.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical Programming*, 127(1):3–30, 2011.

Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *International Conference on Machine Learning*, pages 71–79, 2013.

Steve Smale and Ding-Xuan Zhou. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26(2):153–172, 2007.

Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for Machine Learning*. MIT Press, 2012.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science Business Media, 2008.

Ingo Steinwart, Don Hush, and Clint Scovel. Optimal rates for regularized least squares regression. *Conference of Learning Theory*, 2009.

Pierre Tarres and Yuan Yao. Online learning as stochastic approximation of regularization paths: Optimality and almost-sure convergence. *IEEE Transactions on Information Theory*, 60(9):5716–5735, 2014.

Joel A Tropp. User-friendly tools for random matrices: An introduction. Technical report, DTIC Document, 2012.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Yiming Ying and Massimiliano Pontil. Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, 8(5):561–596, 2008.

Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, 17(9):2077–2098, 2005.

Journal of Machine Learning Research 18 (2017) 1-29 Submitted 7/17; Revised 8/17; Published 9/17

Rank Determination for Low-Rank Data Completion

Morteza Ashraphijuo ASHRAPHIJUO@EE.COLUMBIA.EDU
Columbia University
New York, NY 10027, USA

Xiaodong Wang WANGX@EE.COLUMBIA.EDU
Columbia University
New York, NY 10027, USA

Vaneet Aggarwal VANEET@PURDUE.EDU
Purdue University
West Lafayette, IN 47907, USA

Editor: Animashree Anandkumar

Abstract

Recently, fundamental conditions on the sampling patterns have been obtained for finite completability of low-rank matrices or tensors given the corresponding ranks. In this paper, we consider the scenario where the rank is not given and we aim to approximate the unknown rank based on the location of sampled entries and some given completion. We consider a number of data models, including single-view matrix, multi-view matrix, CP tensor, tensor-train tensor and Tucker tensor. For each of these data models, we provide an upper bound on the rank when an arbitrary low-rank completion is given. We characterize these bounds both deterministically, i.e., with probability one given that the sampling pattern satisfies certain combinatorial properties, and probabilistically, i.e., with high probability given that the sampling probability is above some threshold. Moreover, for both single-view matrix and CP tensor, we are able to show that the obtained upper bound is exactly equal to the unknown rank if the lowest-rank completion is given. Furthermore, we provide numerical experiments for the case of single-view matrix, where we use nuclear norm minimization to find a low-rank completion of the sampled data and we observe that in most of the cases the proposed upper bound on the rank is equal to the true rank.

Keywords: Low-rank data completion, rank estimation, tensor, matrix, manifold, Tucker rank, tensor-train rank, CP rank, multi-view matrix.

1. Introduction

Developing methods and algorithms to study large high-dimensional data is becoming more indispensable as hyperspectral images and videos, product ranking datasets and other applications of big datasets are attracting more attention recently. Moreover, in order to guarantee the same level of efficiency in images or videos, a minor increment in dimensionality in the datasets entails a significant increment in the amount of the data, and this fact causes modeling and also computational challenges to analyze big high-dimensional datasets. Consequently, providing a statistically rigorous result requires a massive amount of data that grows exponentially with the dimension. The low-rank data completion problem is concerned with completing a matrix or tensor given a subset of its entries and some rank constraints. Various applications can be found in many fields including image and signal processing (Candès et al., 2013; Ji et al., 2010), data mining (Eldén, 2007), network coding (Harvey et al., 2005), compressed sensing (Lim and Comon, 2010; Sidiropoulos and Kyrillidis, 2012; Ashraphijuo et al., 2016c; Gandy et al., 2011; Ashraphijuo and Wang, 2017c; Ashraphijuo et al., 2015), reconstructing the visual data (Liu et al., 2013), bioinformatics and systems biology (Ogundijo et al., 2017; Ogundijo et al.), fingerprinting (Liu et al., 2016), etc. There is an extensive literature on developing various optimization methods to treat this problem including minimizing a convex relaxation of rank (Candès and Recht, 2009; Candès and Tao, 2010; Cai et al., 2010; Gandy et al., 2011; Ashraphijuo et al., 2016b; Ashraphijuo and Wang, 2017c), non-convex approaches (Recht and Ré, 2013), and alternating minimization (Jain et al., 2013; Ge et al., 2016), etc. More recently, deterministic conditions on the sampling patterns have been studied for subspace clustering in (Pimentel-Alarcón et al., 2016c, 2015, 2016a,b). Moreover, the fundamental conditions on the sampling pattern that lead to different numbers of completion (unique, finite, or infinite) for different data structures given the corresponding rank constraints have been investigated in (Pimentel-Alarcón et al., 2016d; Ashraphijuo et al., 2017c; Ashraphijuo and Wang, 2017b; Ashraphijuo et al., 2016a; Ashraphijuo and Wang, 2017a; Ashraphijuo et al., 2017d,a,b).

However, in many practical low-rank data completion problems, the rank may not be known *a priori*. In this paper, we investigate this problem and we aim to approximate the rank based on the given entries, where it is assumed that the original data is generically chosen from the manifold corresponding to the unknown rank. The only existing work that treats this problem for a single-view matrix data based on the sampling pattern is (Pimentel-Alarcón and Nowak, 2016), which requires some strong assumptions including the existence of a completion whose rank r is a lower bound on the unknown true rank r^*, i.e., $r^* \geq r$. We start by investigating the single-view matrix to provide a new analysis that does not require such assumption and also we can extend our approach to treat the CP rank tensor model. Moreover, we further generalize our approach to treat vector rank data models including the multi-view matrix, the Tucker rank tensor and the tensor-train (TT) rank tensor. For each of these data models, we obtain the upper bound on the scalar rank or component-wise upper bound on the unknown vector rank, deterministically based on the sampling pattern and the rank of a given completion. We also obtain such bound that holds with high probability based on the sampling probability. Moreover, for the single-view matrix, we provide some numerical results to show how tight our probabilistic bounds on the rank are (in terms of the sampling probability). In particular, we used nuclear norm minimization to find a completion and demonstrate our proposed method in obtaining a tight bound on the unknown rank.

In general, providing a completion requires much less samples than recovering the original sampled data. The goal of this paper is to solve the fundamental problem of rank determination for the original sampled data given an arbitrary low-rank data completion. One possible application scenario is to improve upon the low-rank completion obtained by the convex relaxation methods. Specifically, using convex optimization (minimization of nuclear and atomic norms or summation of nuclear norms of matricizations and unfoldings) or any other methods in the literature, we may be able to find a fairly low-rank "completion" of the original data, which is not necessarily equal (or even close) to the original sampled data. Then, under some circumstances, the rank of the obtained completion using any rank independent method can be an upper bound on the rank of the original sampled data (and sometimes the obtained rank can be exactly equal to the rank of the original sampled data).

We take advantage of the geometric analysis on the manifold of the corresponding data which leads to the fundamental conditions on the sampling pattern (independent of the value of entries) (Pimentel-Alarcón et al., 2016d; Ashraphijuo et al., 2017c; Ashraphijuo and Wang, 2017b; Ashraphi-

©2017 Morteza Ashraphijuo, Xiaodong Wang, and Vaneet Aggarwal.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-375.html.

juo et al., 2016a; Ashraphijuo and Wang, 2017a) such that given an arbitrary low-rank completion we can provide a tight upper bound on the rank. To illustrate how such approximation is even possible consider the following example. Assume that an $n_1 \times n_2$ rank-2 matrix is chosen generically from the corresponding manifold. Hence, any 2×2 submatrix of this matrix is full-rank with probability one (due to the genericity assumption). Moreover, note that any 3×3 submatrix of this matrix is not full-rank. As a result, by observing the sampled entries we can find some bounds on the rank. Using the analysis in (Pimentel-Alarcón et al., 2016d; Ashraphijuo et al., 2017c; Ashraphijuo and Wang, 2017b; Ashraphijuo et al., 2016a; Ashraphijuo and Wang, 2017a) on finite completablity of the sampled data (finite number of completions) for different data models, we characterize both deterministic and probablistic bounds on the unknown rank.

The remained of the paper is organized as follows. In Section 2, we introduce the data models and problem statement. In Sections 3 and 4 we characterize our determintic and probablistic bounds for scalar-rank cases (single-view matrix and CP tensor) and vector-rank cases (multi-view matrix, Tucker tensor and TT tensor), respectively. Finally, Section 5 concludes the paper.

2. Data Models and Problem Statement

2.1 Matrix Models

2.1.1 SINGLE-VIEW MATRIX

Assume that the sampled matrix $\mathbf{U}$ is chosen generically from the manifold of the $n_1 \times n_2$ matrices of rank r^*, where r^* is unknown. The matrix $\mathbf{V} \in \mathbb{R}^{n_1 \times r^*}$ is called a basis for $\mathbf{U}$ if each column of $\mathbf{U}$ can be written as a linear combination of the columns of $\mathbf{V}$. Denote $\mathbf{\Omega}$ as the binary sampling pattern matrix that is of the same size as $\mathbf{U}$ and $\mathbf{\Omega}(\vec{x}) = 1$ if $\mathbf{U}(\vec{x})$ is observed and $\mathbf{\Omega}(\vec{x}) = 0$ otherwise, where $\vec{x} = (x_1, x_2)$ represents the entry corresponding to row number x_1 and column number x_2. Moreover, define $\mathbf{U}_{\mathbf{\Omega}}$ as the matrix obtained from sampling $\mathbf{U}$ according to $\mathbf{\Omega}$, i.e.,

$$\mathbf{U}_{\mathbf{\Omega}}(\vec{x}) = \begin{cases} \mathbf{U}(\vec{x}) & \text{if } \mathbf{\Omega}(\vec{x}) = 1, \\ 0 & \text{if } \mathbf{\Omega}(\vec{x}) = 0. \end{cases} \tag{1}$$

2.1.2 MULTI-VIEW MATRIX

The matrix $\mathbf{U} \in \mathbb{R}^{n \times (n_1+n_2)}$ is sampled. Denote a partition of $\mathbf{U}$ as $\mathbf{U} = [\mathbf{U}_1|\mathbf{U}_2]$ where $\mathbf{U}_1 \in \mathbb{R}^{n \times n_1}$ and $\mathbf{U}_2 \in \mathbb{R}^{n \times n_2}$ represent the first and second views of data, respectively. The sampling pattern is defined as $\mathbf{\Omega} = [\mathbf{\Omega}_1|\mathbf{\Omega}_2]$, where $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ represent the sampling patterns corresponding to the first and second views of data, respectively. Assume that $\text{rank}(\mathbf{U}_1) = r_1^*$, $\text{rank}(\mathbf{U}_2) = r_2^*$ and $\text{rank}(\mathbf{U}) = r^*$, and also $\mathbf{U}$ is chosen generically from the manifold structure with above parameters. Denote $\underline{r}^* = (r_1^*, r_2^*, r^*)$ which is assumed unknown.

2.2 Tensor Models

Assume that a d-way tensor $\mathcal{U} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is sampled. For the sake of simplicity in notation, define $N_i \triangleq \left(\Pi_{j=1}^{i}\, n_j\right)$, $\bar{N}_i \triangleq \left(\Pi_{j=i+1}^{d}\, n_j\right)$ and $N_{-i} \triangleq \frac{N_d}{n_i}$. Denote Ω as the binary sampling pattern tensor that is of the same size as $\mathcal{U}$ and $\Omega(\vec{x}) = 1$ if $\mathcal{U}(\vec{x})$ is observed and $\Omega(\vec{x}) = 0$ otherwise, where $\mathcal{U}(\vec{x})$ represents an entry of tensor $\mathcal{U}$ with coordinate $\vec{x} = (x_1, \ldots, x_d)$. Moreover, define $\mathcal{U}_\Omega$ as the tensor obtained from sampling $\mathcal{U}$ according to Ω, i.e.,

$$\mathcal{U}_\Omega(\vec{x}) = \begin{cases} \mathcal{U}(\vec{x}) & \text{if } \Omega(\vec{x}) = 1, \\ 0 & \text{if } \Omega(\vec{x}) = 0. \end{cases} \tag{2}$$

For each subtensor $\mathcal{U}'$ of the tensor $\mathcal{U}$, define $N_\Omega(\mathcal{U}')$ as the number of observed entries in $\mathcal{U}'$ according to the sampling pattern Ω.

Define the matrix $\widetilde{\mathbf{U}}_{(i)} \in \mathbb{R}^{N_i \times \bar{N}_i}$ as the i-th *unfolding* of the tensor $\mathcal{U}$, such that $\mathcal{U}(\vec{x}) = \widetilde{\mathbf{U}}_{(i)}(\widetilde{M}_i(x_1, \ldots, x_i), \widetilde{M}_{-i}(x_{i+1}, \ldots, x_d))$, where $\widetilde{M}_i : (x_1, \ldots, x_i) \to \{1, 2, \ldots, N_i\}$ and $\widetilde{M}_{-i} : (x_{i+1}, \ldots, x_d) \to \{1, 2, \ldots, \bar{N}_i\}$ are two bijective mappings.

Let $\mathbf{U}_{(i)} \in \mathbb{R}^{n_i \times N_{-i}}$ be the i-th *matricization* of the tensor $\mathcal{U}$, such that $\mathcal{U}(\vec{x}) = \mathbf{U}_{(i)}(x_i, M_i(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d))$, where $M_i : (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d) \to \{1, 2, \ldots, N_{-i}\}$ is a bijective mapping. Observe that for any arbitrary tensor $\mathcal{A}$, the first matricization and the first unfolding are the same, i.e., $\mathbf{A}_{(1)} = \widetilde{\mathbf{A}}_{(1)}$.

In what follows, we introduce three different tensor ranks, i.e., the CP rank, Tucker rank and TT rank.

2.2.1 CP DECOMPOSITION

The CP rank of a tensor $\mathcal{U}$, $\text{rank}_{\text{CP}}(\mathcal{U}) = r$, is defined as the minimum number r such that there exist $\mathbf{a}_i^l \in \mathbb{R}^{n_i}$ for $1 \le i \le d$ and $1 \le l \le r$, such that

$$\mathcal{U} = \sum_{l=1}^{r} \mathbf{a}_1^l \otimes \mathbf{a}_2^l \otimes \ldots \otimes \mathbf{a}_d^l, \tag{3}$$

or equivalently,

$$\mathcal{U}(x_1, x_2, \ldots, x_d) = \sum_{l=1}^{r} \mathbf{a}_1^l(x_1)\mathbf{a}_2^l(x_2)\ldots\mathbf{a}_d^l(x_d), \tag{4}$$

where $\otimes$ denotes the tensor product (outer product) and $\mathbf{a}_i^l(x_i)$ denotes the x_i-th entry of vector $\mathbf{a}_i^l$. Note that $\mathbf{a}_1^l \otimes \mathbf{a}_2^l \otimes \ldots \otimes \mathbf{a}_d^l \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a rank-1 tensor, $l = 1, 2, \ldots, r$.

2.2.2 TUCKER DECOMPOSITION

Given $\mathcal{U} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mathbf{X} \in \mathbb{R}^{n_i \times n_i'}$, the product $\mathcal{U}' \triangleq \mathcal{U} \times_i \mathbf{X} \in \mathbb{R}^{n_1 \times \cdots \times n_{i-1} \times n_i' \times n_{i+1} \times \cdots \times n_d}$ is defined as

$$\mathcal{U}'(x_1, \cdots, x_{i-1}, k_i, x_{i+1}, \cdots, x_d) \triangleq \sum_{x_i=1}^{n_i} \mathcal{U}(x_1, \cdots, x_{i-1}, x_i, x_{i+1}, \cdots, x_d)\mathbf{X}(x_i, k_i). \tag{5}$$

The Tucker rank of a tensor $\mathcal{U}$ is defined as $\text{rank}_{\text{Tucker}}(\mathcal{U}) = \underline{r} = (m_1, \ldots, m_d)$ where $m_i = \text{rank}(\mathbf{U}_{(i)})$, i.e., the rank of the i-th matricization, $i = 1, \ldots, d$. The Tucker decomposition of $\mathcal{U}$ is given by

$$\mathcal{U}(\vec{x}) = \sum_{k_1=1}^{m_1} \cdots \sum_{k_d=1}^{m_d} \mathcal{C}(k_1, \ldots, k_d)\mathbf{T}_1(k_1, x_1)\ldots\mathbf{T}_d(k_d, x_d), \tag{6}$$

or in short

$$\mathcal{U} = \mathcal{C} \times_{i=1}^{d} \mathbf{T}_i, \tag{7}$$

where $\mathcal{C} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ is the core tensor and $\mathbf{T}_i \in \mathbb{R}^{m_i \times n_i}$ are d orthogonal matrices.

2.2.3 TT DECOMPOSITION

The separation or TT rank of a tensor is defined as $\text{rank}_{\text{TT}}(\mathcal{U}) = \underline{r} = (u_1, \ldots, u_{d-1})$ where $u_i = \text{rank}(\widetilde{\mathbf{U}}_{(i)})$, i.e., the rank of the i-th unfolding, $i = 1, \ldots, d-1$. Note that $u_i \leq \max\{N_i, \bar{N}_i\}$ in general and also u_1 is simply the conventional matrix rank when $d = 2$. The TT decomposition of a tensor $\mathcal{U}$ is given by

$$\mathcal{U}(\vec{x}) = \sum_{k_1=1}^{u_1} \cdots \sum_{k_{d-1}=1}^{u_{d-1}} \mathcal{U}^{(1)}(x_1, k_1) \left(\prod_{i=2}^{d-1} \mathcal{U}^{(i)}(k_{i-1}, x_i, k_i) \right) \mathcal{U}^{(d)}(k_{d-1}, x_d), \tag{8}$$

or in short

$$\mathcal{U} = \mathcal{U}^{(1)} \ldots \mathcal{U}^{(d)}, \tag{9}$$

where the 3-way tensors $\mathcal{U}^{(i)} \in \mathbb{R}^{u_{i-1} \times n_i \times u_i}$ for $i = 2, \ldots, d-1$ and matrices $\mathcal{U}^{(1)} \in \mathbb{R}^{n_1 \times u_1}$ and $\mathcal{U}^{(d)} \in \mathbb{R}^{u_{d-1} \times n_d}$ are the components of this decomposition.

For each matrix or tensor model, we assume that the true rank of $\mathbf{U}$ or $\mathcal{U}$ is r^* or $\underline{r}^*$ which is unknown, and also $\mathbf{U}$ or $\mathcal{U}$ is chosen generically from the corresponding manifold. The table below represents the mentioned symbols in brief.

Data structure	Sampled data	Rank	Comments
Single-view matrix	$\mathbf{U} \in \mathbb{R}^{n_1 \times n_2}$	r^*	–
Multi-view matrix	$\mathbf{U} = [\mathbf{U}_1 \vert \mathbf{U}_2] \in \mathbb{R}^{n \times (n_1 + n_2)}$	$\underline{r}^* = (r_1^*, r_2^*, r^*)$	$r_i^* = \text{rank}(\mathbf{U}_i)$
CP tensor	$\mathcal{U} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$	r^*	–
Tucker tensor	$\mathcal{U} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$	$\underline{r}^* = (m_1^*, \ldots, m_d^*)$	$m_i^* = \text{rank}(\mathbf{U}_{(i)})$
TT tensor	$\mathcal{U} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$	$\underline{r}^* = (u_1^*, \ldots, u_{d-1}^*)$	$u_i^* = \text{rank}(\widetilde{\mathbf{U}}_{(i)})$

2.3 Problem Statement

In this paper, we assume that there exists a full rank completion of the sampled data (i.e., the data is not over-sampled). For each one of the above data models, we are interested in obtaining the upper bound on the unknown scalar-rank r^* or component-wise upper bound on the unknown vector-rank $\underline{r}^*$, deterministically based on the sampling pattern $\mathbf{\Omega}$ or Ω and the rank of a given completion. Also, we aim to provide such bound that holds with high probability based only on the sampling probability of the entries and the rank of a given completion. Moreover, for the single-view matrix model and CP-rank tensor model, where the rank is a scalar, we provide both deterministic and probabilistic conditions such that the unknown rank can be exactly determined.

3. Scalar-Rank Cases

3.1 Single-View Matrix

Previously, this problem has been treated in (Pimentel-Alarcón and Nowak, 2016), where strong assumptions including the existence of a completion with rank $r \leq r^*$ have been used. In this section, we provide an analysis that does not require such assumption and moreover our analysis can be extended to multi-view data and tensors in the following sections. Furthermore, we show the tightness of our theoretical bounds via numerical examples.

Assume that $\mathbf{U} \in \mathbb{R}^{n_1 \times n_2}$ is the sampled matrix. Let $\mathbb{P}_1$ and $\mathbb{P}_2$ denote the Lebesgue measures on $\mathbb{R}^{n_1 \times r^*}$ and $\mathbb{R}^{r^* \times n_2}$, respectively. In this paper, we assume that the matrix $\mathbf{U}$ is chosen generically from the manifold of $n_1 \times n_2$ matrices of rank r^*, i.e., the entries of $\mathbf{U}$ are drawn independently with respect to Lebesgue measure on the corresponding manifold. Hence, the probability measures of all statements in this subsection are $\mathbb{P}_1 \times \mathbb{P}_2$.

3.1.1 DETERMINISTIC RANK ANALYSIS

The following condition will be used frequently in this subsection.

Condition A_r: Each column of the sampled matrix includes at least r sampled entries.

Consider an arbitrary column of the sampled matrix $\mathbf{U}(:, i)$, where $i \in \{1, \ldots, n_2\}$. Let $l_i = N_{\mathbf{\Omega}}(\mathbf{U}(:, i))$ denote the number of observed entries in the i-th column of $\mathbf{U}$. Condition A_r results that $l_i \geq r$.

We construct a binary valued matrix called **constraint matrix** $\breve{\mathbf{\Omega}}_r$ based on $\mathbf{\Omega}$ and a given number r. Specifically, we construct $l_i - r$ columns with binary entries based on the locations of the observed entries in $\mathbf{U}(:, i)$ such that each column has exactly $r+1$ entries equal to one. Assume that $x_1, \ldots, x_{l_i}$ are the row indices of all observed entries in this column. Let $\mathbf{\Omega}_r^i$ be the corresponding $n_1 \times (l_i - r)$ matrix to this column which is defined as the following: for any $j \in \{1, \ldots, l_i - r\}$, the j-th column has the value 1 in rows $\{x_1, \ldots, x_r, x_{r+j}\}$ and zeros elsewhere. Define the binary constraint matrix as $\breve{\mathbf{\Omega}}_r = \left[\mathbf{\Omega}_r^1 | \mathbf{\Omega}_r^2 \ldots | \mathbf{\Omega}_r^{n_2}\right] \in \mathbb{R}^{n_1 \times K_r}$ (Pimentel-Alarcón et al., 2016d), where $K_r = N_{\mathbf{\Omega}}(\mathbf{U}) - n_2 r$.

Condition B_r: There exists a submatrix[1] $\breve{\mathbf{\Omega}}_r' \in \mathbb{R}^{n_1 \times K}$ of $\breve{\mathbf{\Omega}}_r$ such that $K = n_1 r - r^2$ and for any $K' \in \{1, 2, \ldots, K\}$ and any submatrix $\breve{\mathbf{\Omega}}_r'' \in \mathbb{R}^{n_1 \times K'}$ of $\breve{\mathbf{\Omega}}_r'$ we have

$$r f(\breve{\mathbf{\Omega}}_r'') - r^2 \geq K', \tag{10}$$

where $f(\breve{\mathbf{\Omega}}_r'')$ denotes the number of nonzero rows of $\breve{\mathbf{\Omega}}_r''$.

Note that exhaustive enumeration is needed in order to check whether or not Condition B_r holds. Hence, the deterministic analysis cannot be used in practice for large-scale data. However, it serves as the basis of the subsequent probabilistic analysis that will lead to a simple lower bound on the sampling probability such that Condition B_r holds with high probability, which is of practical value.

In the following, we restate Theorem 1 in (Pimentel-Alarcón et al., 2016d) which will be used later.

Lemma 1 *With probability one, there are finitely many completions of the sampled matrix if and only if Conditions A_{r^*} and B_{r^*} hold.*

1. Specified by a subset of rows and a subset of columns (not necessarily consecutive).

Recall that the true rank r^* is assumed unknown.

Definition 2 *Let $\mathcal{S}_{\mathbf{\Omega}}$ denote the set of all natural numbers r such that both Conditions A_r and B_r hold.*

Lemma 3 *There exists a number $r_{\mathbf{\Omega}}$ such that $\mathcal{S}_{\mathbf{\Omega}} = \{1, 2, \ldots, r_{\mathbf{\Omega}}\}$.*

Proof Assume that $1 < r \leq \min\{n_1, n_2\}$ and $r \in \mathcal{S}_{\mathbf{\Omega}}$. It suffices to show $r - 1 \in \mathcal{S}_{\mathbf{\Omega}}$. By contradiction, assume that $r - 1 \notin \mathcal{S}_{\mathbf{\Omega}}$. Therefore, according to Lemma 1, there exist infinitely many completions of $\mathbf{U}$ of rank $r - 1$ and there exist at most finitely many completions of $\mathbf{U}$ of rank r.

Consider a rank $r - 1$ completion $\mathbf{U}_{r-1}$. Note that changing one single entry (a non-observed entry) of $\mathbf{U}_{r-1}$, for example $\mathbf{U}_{r-1}(1, 1) = x$, to a random number in $y \in \mathbb{R}$ changes the rank of this matrix by at most 1 and basically since we are changing to a random number, it can be easily seen that the rank does not decrease with probability one. Hence, the rank of the modified matrix $\mathbf{U}'_{r-1}$ would be either $r - 1$ or r. Assume that the rank has been increased to r. Then, we show there exist infinitely many completions of rank r, which contradicts the assumption. In fact, for any value of $\mathbf{U}_{r-1}(1, 1)$ except x, this matrix would be a rank r completion. To observe this more clearly, consider the $r \times r$ submatrix of $\mathbf{U}'_{r-1}$ whose determinant is not zero due to changing the value of $\mathbf{U}_{r-1}(1, 1)$. It is easily observed that this submatrix includes $\mathbf{U}'_{r-1}(1, 1)$ and let assume it is $\mathbf{U}'_{r-1}(1 : r, 1 : r)$, and therefore the determinant of $\mathbf{U}'_{r-1}(2 : r, 2 : r)$ is a nonzero number (otherwise the rank would not increase by changing the value of $\mathbf{U}_{r-1}(1, 1)$). Hence, it is easy to see that for any value of $\mathbf{U}_{r-1}(1, 1)$ except x, $\mathbf{U}'_{r-1}$ would be a rank r completion, and therefore there exist infinitely many completions of rank r and proof is complete in this scenario.

Now, assume that changing any of the non-observed entries does not increase the rank of $\mathbf{U}_{r-1}$. Then, this contradicts the assumption that there exists a full rank completion of the sampled data since there does not exist any completion of rank higher than $r - 1$. ■

The following theorem provides a relationship between the unknown rank r^* and $r_{\mathbf{\Omega}}$.

Theorem 4 *With probability one, exactly one of the following statements holds*
(i) $r^* \in \mathcal{S}_{\mathbf{\Omega}} = \{1, 2, \ldots, r_{\mathbf{\Omega}}\}$;
(ii) For any arbitrary completion of the sampled matrix $\mathbf{U}$ of rank r, we have $r \notin \mathcal{S}_{\mathbf{\Omega}}$.

Proof Suppose that there does not exist a completion of the sampled matrix $\mathbf{U}$ of rank r such that $r \in \mathcal{S}_{\mathbf{\Omega}}$. Therefore, it is easily verified that statement (ii) holds and statement (i) does not hold. On the other hand, assume that there exists a completion of the sampled matrix $\mathbf{U}$ of rank r, where $r \in \mathcal{S}_{\mathbf{\Omega}}$. Hence, statement (ii) does not hold and to complete the proof it suffices to show that with probability one, statement (i) holds.

Observe that $r_{\mathbf{\Omega}} \in \mathcal{S}_{\mathbf{\Omega}}$, and therefore Condition $A_{r_{\mathbf{\Omega}}}$ holds. Hence, each column of $\mathbf{U}$ includes at least $r_{\mathbf{\Omega}} + 1$ observed entries. On the other hand, the existence of a completion of the sampled matrix $\mathbf{U}$ of rank $r \in \mathcal{S}_{\mathbf{\Omega}}$ results in the existence of a basis $\mathbf{X} \in \mathbb{R}^{n_1 \times r}$ such that each column of $\mathbf{U}$ is a linear combination of the columns of $\mathbf{X}$, and thus there exists $\mathbf{Y} \in \mathbb{R}^{r \times n_2}$ such that $\mathbf{U}_{\mathbf{\Omega}} = (\mathbf{XY})_{\mathbf{\Omega}}$. Hence, given $\mathbf{X}$, each observed entry $\mathbf{U}(i, j)$ results in a degree-1 polynomial in

terms of the entries of $\mathbf{Y}$ as the following

$$\mathbf{U}(i, j) = \sum_{l=1}^{r} \mathbf{X}(i, l)\mathbf{Y}(l, j). \tag{11}$$

Consider the first column of $\mathbf{U}$ and recall that it includes at least $r_{\mathbf{\Omega}}+1 \geq r+1$ observed entries. The genericity of the coefficients of the above-mentioned polynomials results that using r of the observed entries the first column of $\mathbf{Y}$ can be determined uniquely. This is because there exists a unique solution for a system of r linear equations in r variables that are linearly independent. Then, there exists at least one more observed entry besides these r observed entries in the first column of $\mathbf{U}$ and it can be written as a linear combination of the r observed entries that have been used to obtain the first column of $\mathbf{Y}$. Let $\mathbf{U}(i_1, 1)$, $\ldots$, $\mathbf{U}(i_r, 1)$ denote the r observed entries that have been used to obtain the first column of $\mathbf{Y}$ and $\mathbf{U}(i_{r+1}, 1)$ denote the other observed entry. Hence, the existence of a completion of the sampled matrix $\mathbf{U}$ of rank $r \in \mathcal{S}_{\mathbf{\Omega}}$ results in an equation as the following

$$\mathbf{U}(i_{r+1}, 1) = \sum_{l=1}^{r} t_l \mathbf{U}(i_l, 1), \tag{12}$$

where t_l's are constant scalars, $l = 1, \ldots, r$. Assume that $r^* \notin \mathcal{S}_{\mathbf{\Omega}}$, i.e., statement (i) does not hold. Then, note that $r^* \geq r + 1$ and $\mathbf{U}$ is chosen generically from the manifold of $n_1 \times n_2$ rank-r^* matrices, and therefore an equation of the form of (12) holds with probability zero. Moreover, according to Lemma 1 there exist at most finitely many completions of the sampled matrix of rank r. Therefore, there exist a completion of $\mathbf{U}$ of rank r with probability zero, which contradicts the initial assumption that there exists a completion of the sampled matrix $\mathbf{U}$ of rank r, where $r \in \mathcal{S}_{\mathbf{\Omega}}$. ■

Note that the existing work that treats the similar problem for a single-view matrix data based on the sampling pattern is (Pimentel-Alarcón and Nowak, 2016), which requires some strong assumptions including the existence of a completion whose rank r is a lower bound on the unknown true rank r^*, i.e., $r^* \geq r$. We provide a new analysis that does not require such assumption and also based on our new analysis, we can extend our approach to treat other data structures.

Corollary 5 *Consider an arbitrary number $r' \in \mathcal{S}_{\mathbf{\Omega}}$. Similar to Theorem 4, it follows that with probability one, exactly one of the followings holds*
(i) $r^* \in \{1, 2, \ldots, r'\}$;
(ii) For any arbitrary completion of the sampled matrix $\mathbf{U}$ of rank r, we have $r \notin \{1, 2, \ldots, r'\}$.

As a result of Corollary 5, we have the following.

Corollary 6 *Assuming that there exists a rank-r completion of the sampled matrix $\mathbf{U}$ such that $r \in \mathcal{S}_{\mathbf{\Omega}}$, then with probability one $r^* \leq r$.*

Corollary 7 *Let $\mathbf{U}^*$ denote an optimal solution to the following NP-hard optimization problem*

$$\begin{array}{ll} \text{minimize}_{\mathbf{U}' \in \mathbb{R}^{n_1 \times n_2}} & \text{rank}(\mathbf{U}') \\ \text{subject to} & \mathbf{U}'_{\mathbf{\Omega}} = \mathbf{U}_{\mathbf{\Omega}}. \end{array} \tag{13}$$

Also, let $\hat{\mathbf{U}}$ denote a suboptimal solution to the above optimization problem. Then, Corollary 5 results the following statements:

(i) If $\text{rank}(\mathbf{U}^*) \in \mathcal{S}_{\mathbf{\Omega}}$, *then* $r^* = \text{rank}(\mathbf{U}^*)$ *with probability one.*

(ii) If $\text{rank}(\hat{\mathbf{U}}) \in \mathcal{S}_{\mathbf{\Omega}}$, *then* $r^* \leq \text{rank}(\hat{\mathbf{U}})$ *with probability one.*

Remark 8 *One challenge of applying Corollary 7 or any of the other obtained deterministic results is the computation of $\mathcal{S}_{\mathbf{\Omega}}$, which involves exhaustive enumeration to check Condition B_r. Next, for each number r, we provide a lower bound on the sampling probability in terms of r that ensures $r \in \mathcal{S}_{\mathbf{\Omega}}$ with high probability. Consequently, we do not need to compute $\mathcal{S}_{\mathbf{\Omega}}$ but instead we can certify the above results with high probability.*

3.1.2 PROBABILISTIC RANK ANALYSIS

The following lemma is a re-statement of Theorem 3 in (Pimentel-Alarcón et al., 2016d), which is the probabilistic version of Lemma 1.

Lemma 9 *Suppose $r \leq \frac{n_1}{6}$ and that each* **column** *of the sampled matrix is observed in at least l entries, uniformly at random and independently across entries, where*

$$l > \max\left\{12\ \log\left(\frac{n_1}{\epsilon}\right) + 12, 2r\right\}. \tag{14}$$

Also, assume that $r(n_1 - r) \leq n_2$. Then, with probability at least $1 - \epsilon$, $r \in \mathcal{S}_{\mathbf{\Omega}}$.

The following lemma is taken from (Ashraphijuo et al., 2016a) and will be used to derive a lower bound on the sampling probability that leads to the similar statement as Theorem 4 with high probability.

Lemma 10 *Consider a vector with n entries where each entry is observed with probability p independently from the other entries. If $p > p' = \frac{k}{n} + \frac{1}{\sqrt[4]{n}}$, then with probability at least $\left(1 - \exp(-\frac{\sqrt{n}}{2})\right)$, more than k entries are observed.*

The following proposition characterizes the probabilistic version of Theorem 4.

Proposition 11 *Suppose $r \leq \frac{n_1}{6}$, $r(n_1 - r) \leq n_2$ and that each entry of the sampled matrix is observed uniformly at random and independently across entries with probability p, where*

$$p > \frac{1}{n_1}\max\left\{12\ \log\left(\frac{n_1}{\epsilon}\right) + 12, 2r\right\} + \frac{1}{\sqrt[4]{n_1}}. \tag{15}$$

Then, with probability at least $(1-\epsilon)\left(1 - \exp(-\frac{\sqrt{n_1}}{2})\right)^{n_2}$, we have $r \in \mathcal{S}_{\mathbf{\Omega}}$.

Proof Consider an arbitrary column of $\mathbf{U}$ and note that resulting from Lemma 10 the number of observed entries at this column of $\mathbf{U}$ is greater than $\max\left\{12\ \log\left(\frac{n_1}{\epsilon}\right) + 12, 2r\right\}$ with probability at least $\left(1 - \exp(-\frac{\sqrt{n_1}}{2})\right)$. Therefore, the number of sampled entries at each column satisfies

$$l > \max\left\{12\ \log\left(\frac{n_1}{\epsilon}\right) + 12, 2r\right\}, \tag{16}$$

with probability at least $\left(1 - \exp(-\frac{\sqrt{n_1}}{2})\right)^{n_2}$. Thus, resulting from Lemma 9 with probability at least $(1-\epsilon)\left(1 - \exp(-\frac{\sqrt{n_1}}{2})\right)^{n_2}$, we have $r \in \mathcal{S}_{\mathbf{\Omega}}$. ■

Finally, we have the following probabilistic version of Corollary 7.

Corollary 12 *Assume that* $\text{rank}(\mathbf{U}^*) \leq \frac{n_1}{6}$ *and* $\text{rank}(\mathbf{U}^*)(n_1 - \text{rank}(\mathbf{U}^*)) \leq n_2$ *and* (15) *holds for* $r = \text{rank}(\mathbf{U}^*)$, *where* $\mathbf{U}^*$ *denotes an optimal solution to the optimization problem* (13). *Then, according to Proposition 11 and Corollary 7, with probability at least* $(1-\epsilon)\left(1 - \exp(-\frac{\sqrt{n_1}}{2})\right)^{n_2}$, $r^* = \text{rank}(\mathbf{U}^*)$. *Similarly, assume that* $\text{rank}(\hat{\mathbf{U}}) \leq \frac{n_1}{6}$ *and* $\text{rank}(\hat{\mathbf{U}})(n_1 - \text{rank}(\hat{\mathbf{U}})) \leq n_2$ *and* (15) *holds for* $r = \text{rank}(\hat{\mathbf{U}})$, *where* $\hat{\mathbf{U}}$ *denotes a suboptimal solution to the optimization problem* (13). *Then, with probability at least* $(1-\epsilon)\left(1 - \exp(-\frac{\sqrt{n_1}}{2})\right)^{n_2}$, $r^* \leq \text{rank}(\hat{\mathbf{U}})$.

3.1.3 NUMERICAL RESULTS

In Fig. 1 and Fig. 2, the x-axis represents the sampling probability, and the y-axis denotes the value of r. The color scale represents the lower bound on the probability of event $r \in \mathcal{S}_{\mathbf{\Omega}}$. For example, as we can observe in Fig. 1, for any $r \in \{1, \ldots, 44\}$ we have $r \in \mathcal{S}_{\mathbf{\Omega}}$ with probability at least 0.6 (approximately based on the color scale since the corresponding points are orange) given that $p = 0.54$.

We consider the sampled matrix $\mathbf{U} \in \mathbb{R}^{300\times 15000}$ and $\mathbf{U} \in \mathbb{R}^{1200\times 240000}$ in Fig. 1 and Fig. 2, respectively. In particular, for fixed values of sampling probability p and r, we first find a "small" ϵ that (15) holds by trial-and-error. Then, according to Proposition 11, we conclude that with probability at least $(1-\epsilon)\left(1 - \exp(-\frac{\sqrt{n_1}}{2})\right)^{n_2}$, $r \in \mathcal{S}_{\mathbf{\Omega}}$.

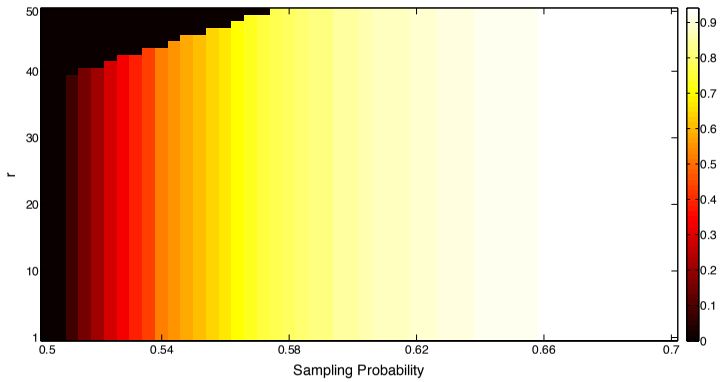


Figure 1: Probability of $r \in \mathcal{S}_{\mathbf{\Omega}}$ as a function of sampling probability for $\mathbf{U} \in \mathbb{R}^{300\times 15000}$.

The purpose of Figs. 3–6 is to show how tight our proposed upper bounds on rank can be. Here, we first generate an $n_1 \times n_2$ random matrix of a given rank r by multiplying a random

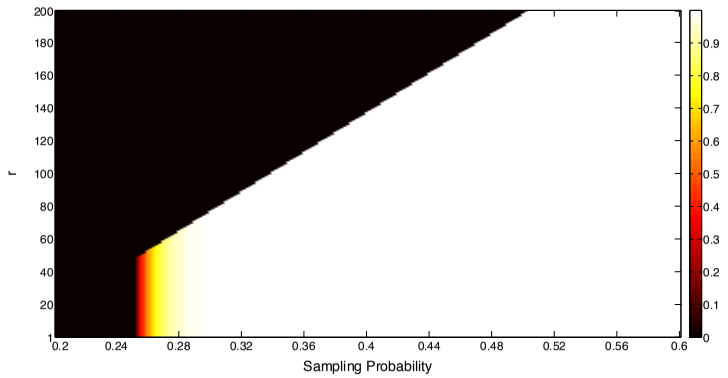


Figure 2: Probability of $r \in \mathcal{S}_{\mathbf{\Omega}}$ as a function of sampling probability for $\mathbf{U} \in \mathbb{R}^{1200\times 240000}$.

(entries are drawn according to a uniform distribution on real numbers within an interval) $n_1 \times r$ matrix and $r \times n_2$ matrix. Then, each entry of the randomly generated matrix is sampled uniformly at random and independently across entries with some sampling probability p. Afterwards, we apply the nuclear norm minimization method proposed in (Candes and Recht, 2012) for matrix completion, where the non-convex objective function in (13) is relaxed by using nuclear norm, which is the convex hull of the rank function, as follows

$$\begin{aligned} &\text{minimize}_{\mathbf{U}'\in\mathbb{R}^{n_1\times n_2}} && \|\mathbf{U}'\|_* \\ &\text{subject to} && \mathbf{U}'_{\mathbf{\Omega}} = \mathbf{U}_{\mathbf{\Omega}}, \end{aligned} \tag{17}$$

where $\|\mathbf{U}'\|_*$ denotes the nuclear norm of $\mathbf{U}'$. Let $\hat{\mathbf{U}}^*$ denote an optimal solution to (17) and recall that $\mathbf{U}^*$ denotes an optimal solution to (13). Since (17) is a convex relaxation to (13), we conclude that $\hat{\mathbf{U}}^*$ is a suboptimal solution to (13), and therefore $\text{rank}(\mathbf{U}^*) \leq \text{rank}(\hat{\mathbf{U}}^*)$. We used the Matlab program found online (Shabat, 2015) to solve (17).

As an example, we generate a random matrix $\mathbf{U} \in \mathbb{R}^{300\times 15000}$ (the same size as the matrix in Fig. 1) of rank r as described above for $r \in \{1, \ldots, 50\}$ and some values of the sampling probability p. Then, we obtain the rank of the completion given by (17) and denote it by r'. Due to the randomness of the sampled matrix, we repeat this procedure 5 times. We calculate the "gap" $r'-r$ in each of these 5 runs and denote the maximum and minimum among these 5 numbers by d_{max} and d_{min}, respectively. Hence, d_{max} and d_{min} represent the loosest (worst) and tightest (best) gaps between the rank obtained by (17) and rank of the original sampled matrix over 5 runs, respectively. In Figs. 3–6, the maximum and minimum gaps are plotted as a function of rank of the matrix, for different sampling probabilities.

We have the following observations.

- According to Fig. 1, for $p = 0.54$ and $p = 0.58$ we can ensure that the rank of any completion is an upper bound on the rank of the sampled matrix or r^* with probability at least 0.6 and 0.8, respectively.

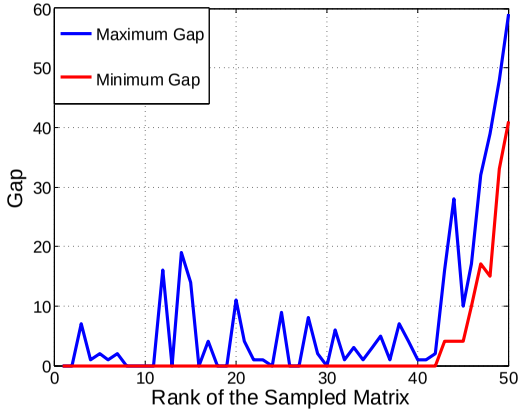


Figure 3: The gaps between the rank of the obtained matrix via (17) and that of the original sampled matrix for $p = 0.46$.

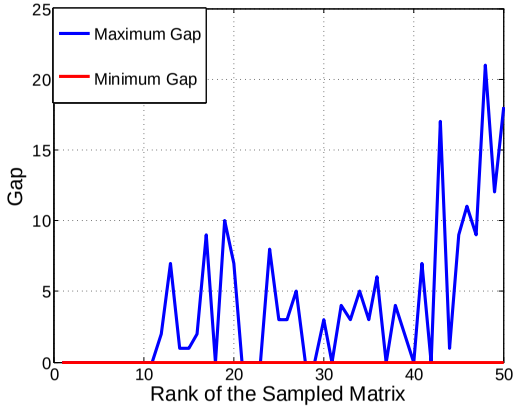


Figure 4: The gaps between the rank of the obtained matrix via (17) and that of the original sampled matrix for $p = 0.50$.

- As we can observe in Figs. 3–6, the defined gap is always a nonnegative number, which is consistent with previous observation that for $p = 0.54$ and $p = 0.58$ we can certify that with high probability (≥ 0.6) the rank of any completion is an upper bound on the rank of the sampled matrix or r^*.

- For $p = 0.54$ and $p = 0.58$ that we have theoretical results (as mentioned in the first observation) the gap obtained by (17) is very close to zero. This phenomenon (that we do not have a rigorous justification for) shows that as soon as we can certify our proposed theoretical

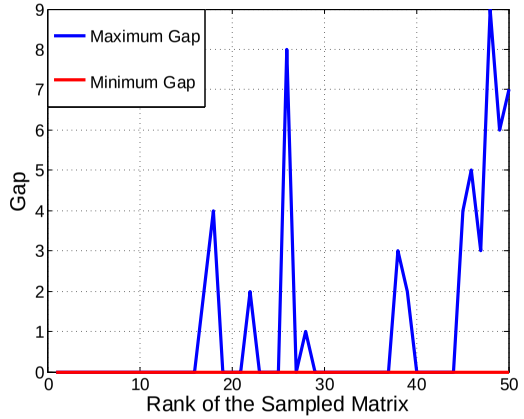


Figure 5: The gaps between the rank of the obtained matrix via (17) and that of the original sampled matrix for $p = 0.54$.

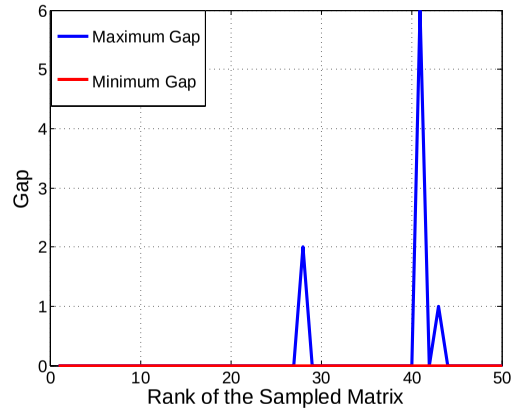


Figure 6: The gaps between the rank of the obtained matrix via (17) and that of the original sampled matrix for $p = 0.58$.

results (i.e., as soon as the rank of a completion provides an upper bound on the rank of the sampled matrix or r^*) by increasing the sampling probability, the upper bound found through (17) becomes very tight; in some cases this bound is exactly equal to r^* (red curves) and in some cases this bound is almost equal to r^* (blue curves). However, these gaps are not small (specially blue curves) for $p = 0.46$ and $p = 0.50$ and note that according to Fig. 1, for these values of p we cannot guarantee the bounds on the value of rank hold with high probability.

3.2 CP-Rank Tensor

Let $\mathbb{P}_i$ denote the Lebesgue measures on $\mathbb{R}^{n_i \times r^*}$, $i = 1, \ldots, d$. In this subsection, we assume that the sampled tensor $\mathcal{U} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is chosen generically from the manifold of tensors of rank $r^* = \text{rank}_{\text{CP}}(\mathcal{U})$ (where r^* is unknown), or in other words, the entries of $\mathcal{U}$ are drawn independently with respect to Lebesgue measure on the corresponding manifold. Hence, the probability measures of all statements in this subsection are $\mathbb{P}_1 \times \mathbb{P}_2 \times \ldots \times \mathbb{P}_d$.

Condition $\mathcal{A}_r$: Each row of the d-th matricization of the sampled tensor, i.e., $\mathbf{U}_{(d)}$ includes at least r observed entries.

We construct a binary valued tensor called **constraint tensor** $\breve{\Omega}_r$ based on Ω and a given number r. Consider any subtensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times 1}$ of the tensor $\mathcal{U}$. The sampled tensor $\mathcal{U}$ includes n_d subtensors that belong to $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times 1}$ and let $\mathcal{Y}_i$ for $1 \leq i \leq n_d$ denote these n_d subtensors. Define a binary valued tensor $\breve{\mathcal{Y}}_i \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times k_i}$, where $k_i = N_\Omega(\mathcal{Y}_i) - r$ and its entries are described as the following. We can look at $\breve{\mathcal{Y}}_i$ as k_i tensors each belongs to $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times 1}$. For each of the mentioned k_i tensors in $\breve{\mathcal{Y}}_i$ we set the entries corresponding to r of the observed entries equal to 1. For each of the other k_i observed entries, we pick one of the k_i tensors of $\breve{\mathcal{Y}}_i$ and set its corresponding entry (the same location as that specific observed entry) equal to 1 and set the rest of the entries equal to 0. In the case that $k_i = 0$ we simply ignore $\breve{\mathcal{Y}}_i$, i.e., $\breve{\mathcal{Y}}_i = \emptyset$

By putting together all n_d tensors in dimension d, we construct a binary valued tensor $\breve{\Omega}_r \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times K}$, where $K = \sum_{i=1}^{n_d} k_i = N_\Omega(\mathcal{U}) - r n_d$ and call it the **constraint tensor**. Observe that each subtensor of $\breve{\Omega}_r$ which belongs to $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times 1}$ includes exactly $r + 1$ nonzero entries. In (Ashraphijuo and Wang, 2017b), an example is given on the construction of $\breve{\Omega}_r$.

Condition $\mathcal{B}_r$: $\breve{\Omega}_r$ consists a subtensor $\breve{\Omega}'_r \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times K}$ such that $K = r(\sum_{i=1}^{d-1} n_i) - r^2 - r(d-2)$ and for any $K' \in \{1, 2, \ldots, K\}$ and any subtensor $\breve{\Omega}''_r \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times K'}$ of $\breve{\Omega}'_r$ we have

$$r\left(\left(\sum_{i=1}^{d-1} f_i(\breve{\Omega}''_r)\right) - \min\left\{\max\left\{f_1(\breve{\Omega}''_r), \ldots, f_{d-1}(\breve{\Omega}''_r)\right\}, r\right\} - (d-2)\right) \geq K', \tag{18}$$

where $f_i(\breve{\Omega}''_r)$ denotes the number of nonzero rows of the i-th matricization of $\breve{\Omega}''_r$.

The following lemma is a re-statement of Theorem 1 in (Ashraphijuo and Wang, 2017b).

Lemma 13 *With probability one, there are only finitely many rank-r^* completions of the sampled tensor if and only if Conditions $\mathcal{A}_{r^*}$ and $\mathcal{B}_{r^*}$ hold.*

Definition 14 *Let $\mathcal{S}_\Omega$ denote the set of all natural numbers r such that both Conditions $\mathcal{A}_r$ and $\mathcal{B}_r$ hold.*

Lemma 15 *There exists a number r_Ω such that $\mathcal{S}_\Omega = \{1, 2, \ldots, r_\Omega\}$.*

Proof The proof is similar to the proof of Lemma 3 since the dimension of the manifold of CP rank-r tensors is $r(\sum_{i=1}^{d} n_i) - r^2 - r(d-1)$, which is an increasing function in r. ■

The following theorem gives an upper bound on the unknown rank r^*.

Theorem 16 *With probability one, exactly one of the following statements holds*
(i) $r^* \in \mathcal{S}_\Omega = \{1, 2, \ldots, r_\Omega\}$;
(ii) For any arbitrary completion of the sampled tensor $\mathcal{U}$ of rank r, we have $r \notin \mathcal{S}_\Omega$.

Proof Similar to the proof of Theorem 4, it suffices to show that the assumption $r^* \notin \mathcal{S}_\Omega$ results that there exists a completion of $\mathcal{U}$ of CP rank r, where $r \in \mathcal{S}_\Omega$, with probability zero. Define $\mathcal{V} = (\mathcal{V}_1, \ldots, \mathcal{V}_r)$ as the basis of the rank-r CP decomposition of $\mathcal{U}$ as in (3), where $\mathcal{V}_l = \mathbf{a}_1^l \otimes \mathbf{a}_2^l \otimes \ldots \otimes \mathbf{a}_{d-1}^l \in \mathbb{R}^{n_1 \times \ldots n_{d-1}}$ is a rank-1 tensor and $\mathbf{a}_i^l$ is defined in (3) for $1 \leq l \leq r$ and $1 \leq i \leq d$. Define $\mathcal{Y} = (\mathbf{a}_d^1, \ldots, \mathbf{a}_d^r)$ and $\mathcal{V} \otimes_d \mathcal{Y} = \sum_{l=1}^r \mathcal{V}_l \otimes \mathbf{a}_d^l$. Observe that $\mathcal{U} = \sum_{l=1}^r \mathcal{V}_l \otimes \mathbf{a}_d^l = \mathcal{V} \otimes_d \mathcal{Y}$.

Observe that each row of $\mathbf{U}_{(d)}$ includes at least $r_\Omega + 1$ observed entries since Condition $\mathcal{A}_{r_\Omega}$ holds. Moreover, the existence of a completion of the sampled tensor $\mathcal{U}$ of rank $r \in \mathcal{S}_\Omega$ results in the existence of a basis $\mathcal{V} = (\mathcal{V}_1, \ldots, \mathcal{V}_r)$ such that there exists $\mathcal{Y} = (\mathbf{a}_d^1, \ldots, \mathbf{a}_d^r)$ and $\mathcal{U}_\Omega = (\mathcal{V} \otimes_d \mathcal{Y})_\Omega$. As a result, given $\mathcal{V}$, each observed entry of $\mathcal{U}$ results in a degree-1 polynomial in terms of the entries of $\mathcal{Y}$ as

$$\mathcal{U}(\vec{x}) = \sum_{l=1}^{r} \mathcal{V}_l(x_1, \ldots, x_{d-1}) \mathbf{a}_d^l(x_d). \tag{19}$$

Note that $r_\Omega \geq r$ and each row of $\mathbf{U}_{(d)}$ includes at least $r_\Omega + 1 \geq r + 1$ observed entries. Consider $r+1$ of the observed entries of the first row of $\mathbf{U}_{(d)}$ and we denote them by $\mathcal{U}(\vec{x}_1), \ldots, \mathcal{U}(\vec{x}_{r+1})$, where the last component of the vector $\vec{x}_i$ is equal to one, $1 \leq i \leq r+1$. Similar to the proof of Theorem 4, genericity of $\mathcal{U}$ results in

$$\mathcal{U}(\vec{x}_{r+1}) = \sum_{l=1}^{r} t_l \mathcal{U}(\vec{x}_i), \tag{20}$$

where t_l's are constant scalars, $l = 1, \ldots, r$. On the other hand, according to Lemma 13 there exist at most finitely many completions of the sampled tensor of rank r. Therefore, there exist a completion of $\mathbf{U}$ of rank r with probability zero. Moreover, an equation of the form of (20) holds with probability zero as $r^* \geq r+1$ and $\mathcal{U}$ is chosen generically from the manifold of tensors of rank-r^*. Therefore, there exists a completion of rank r with probability zero. ■

Corollary 17 *Consider an arbitrary number $r' \in \mathcal{S}_\Omega$. Similar to Theorem 16, it follows that with probability one, exactly one of the followings holds*

(i) $r^* \in \{1, 2, \ldots, r'\}$;

(ii) For any arbitrary completion of the sampled tensor $\mathcal{U}$ of rank r, we have $r \notin \{1, 2, \ldots, r'\}$.

Corollary 18 *Assuming that there exists a CP rank-r completion of the sampled tensor $\mathcal{U}$ such that $r \in \mathcal{S}_\Omega$, we conclude that with probability one $r^* \leq r$.*

Corollary 19 *Let $\mathcal{U}^*$ denote an optimal solution to the following NP-hard optimization problem*

$$\begin{aligned} &\text{minimize}_{\mathcal{U}' \in \mathbb{R}^{n_1 \times \cdots \times n_d}} && \text{rank}_{\text{CP}}(\mathcal{U}') \\ &\text{subject to} && \mathcal{U}'_\Omega = \mathcal{U}_\Omega. \end{aligned} \tag{21}$$

Assume that $\text{rank}_{\text{CP}}(\mathcal{U}^*) \in \mathcal{S}_\Omega$. Then, Corollary 18 results that $r^* = \text{rank}_{\text{CP}}(\mathcal{U}^*)$ with probability one.

The following lemma is Lemma 15 in (Ashraphijuo and Wang, 2017b), which is the probabilistic version of Lemma 13 in terms of the sampling probability.

Lemma 20 *Assume that $n_1 = n_2 = \cdots = n_d = n$, $d > 2$, $n > \max\{200, r(d-2)\}$ and $r \leq \frac{n}{6}$. Moreover, assume that the sampling probability satisfies*

$$p > \frac{1}{n^{d-2}} \max\left\{27 \log\left(\frac{n}{\epsilon}\right) + 9 \log\left(\frac{2r(d-2)}{\epsilon}\right) + 18, 6r\right\} + \frac{1}{\sqrt[4]{n^{d-2}}}. \tag{22}$$

Then, with probability at least $(1-\epsilon)\left(1 - \exp(-\frac{\sqrt{n^{d-2}}}{2})\right)^{n^2}$, we have $r \in \mathcal{S}_\Omega$.

The following corollary is the probabilistic version of Corollaries 18 and 19.

Corollary 21 *Assuming that there exists a CP rank-r completion of the sampled tensor $\mathcal{U}$ such that the conditions given in Lemma 20 hold, with the sampling probability satisfying (22), we conclude that with probability at least $(1-\epsilon)\left(1 - \exp(-\frac{\sqrt{n^{d-2}}}{2})\right)^{n^2}$ we have $r^* \leq r$. Therefore, given that (22) holds for $r = \text{rank}(\mathbf{U}^*)$ and $\mathbf{U}^*$ denotes an optimal solution to the optimization problem (21), with probability at least $(1-\epsilon)\left(1 - \exp(-\frac{\sqrt{n^{d-2}}}{2})\right)^{n^2}$ we have $r^* = \text{rank}(\mathbf{U}^*)$.*

3.2.1 NUMERICAL RESULTS

We generate a random tensor $\mathcal{U} \in \mathbb{R}^{8\times8\times8\times8\times8\times8}$ of CP-rank 2 by adding two random rank-1 tensors. The color scale represents the lower bound on the probability that we can guarantee the rank of a given completion is an upper bound on the true value of rank. Then, we solve the following convex optimization problem for different values of the sampling probability.

$$\begin{aligned} &\text{minimize}_{\mathcal{U}' \in \mathbb{R}^{n_1 \times \cdots \times n_d}} && \|\widetilde{\mathbf{U}}'_{(3)}\|_* \\ &\text{subject to} && \mathcal{U}'_\Omega = \mathcal{U}_\Omega. \end{aligned} \tag{23}$$

Note that rank of any of the unfoldings of a tensor is a lower bound on the CP-rank of that tensor. Hence, we minimize the nuclear norm of the unfolding with the possible maximum rank among all unfoldings as $\widetilde{\mathbf{U}}_{(3)} \in \mathbb{R}^{512\times512}$. Then, we use the Matlab toolbox found online "Tensorlab" to calculate the CP-rank of the obtained tensor via solving convex program (23) (there are other methods to calculate CP decomposition, e.g., (Pimentel-Alarcón, 2016)). In Figure 7, gap represents the CP-rank of the solution of (23) minus the CP-rank of the original sampled tensor.

4. Vector-Rank Cases

4.1 Multi-View Matrix

Let $\mathbb{P}_1$ and $\mathbb{P}_2$ denote the Lebesgue measures on $\mathbb{R}^{n \times r_1^*}$ and $\mathbb{R}^{r_1^* \times n_1}$, respectively. Moreover, let $\mathbb{P}_3$ and $\mathbb{P}_4$ denote the Lebesgue measures on $\mathbb{R}^{n \times (r^* - r_1^*)}$ and $\mathbb{R}^{r_2^* \times n_2}$, respectively. In this paper, we assume that $\mathbf{U}$ is chosen generically from the manifold corresponding to rank vector (r_1^*, r_2^*, r^*), i.e., the entries of $\mathbf{U}$ are drawn independently with respect to Lebesgue measure on the corresponding manifold. Hence, the probability measures of all statements in this subsection are $\mathbb{P}_1 \times \mathbb{P}_2 \times \mathbb{P}_3 \times \mathbb{P}_4$.

The following Conditions will be used frequently in this subsection.

Condition A_{r_1,r_2}: Each column of $\mathbf{U}_1$ and $\mathbf{U}_2$ include at least r_1 and r_2 sampled entries, respectively.

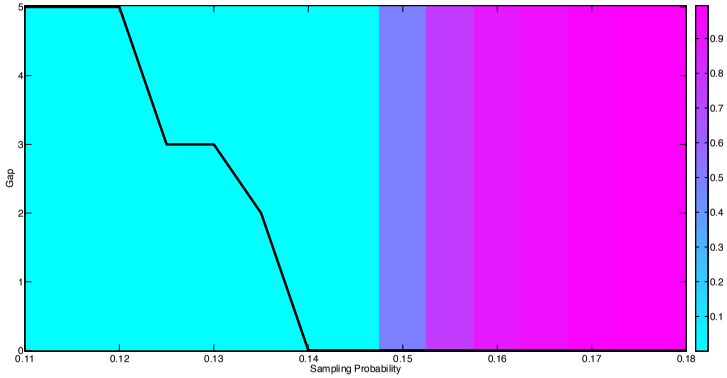


Figure 7: The rank gap as a function of sampling probability for $\mathcal{U} \in \mathbb{R}^{8\times 8\times 8\times 8\times 8\times 8}$ of CP-rank 2.

We construct a binary valued matrix called **constraint matrix** for multi-view matrix $\mathbf{U}$ as $\breve{\mathbf{\Omega}}_{r_1,r_2} = [\breve{\mathbf{\Omega}}_{r_1}|\breve{\mathbf{\Omega}}_{r_2}]$, where $\breve{\mathbf{\Omega}}_{r_1}$ and $\breve{\mathbf{\Omega}}_{r_2}$ represent the constraint matrix for single-view matrices $\mathbf{U}_1$ and $\mathbf{U}_2$ (defined in Section 3.1), respectively.

Condition $B_{r_1,r_2,r}$: $\breve{\mathbf{\Omega}}_{r_1,r_2}$ consists a submatrix $\breve{\mathbf{\Omega}}'_{r_1,r_2} \in \mathbb{R}^{n\times K}$ such that $K = nr - r^2 - r_1^2 - r_2^2 + r(r_1+r_2)$ and for any $K' \in \{1,2,\ldots,K\}$ and any submatrix $\breve{\mathbf{\Omega}}''_{r_1,r_2} \in \mathbb{R}^{n\times K'}$ of $\breve{\mathbf{\Omega}}'_{r_1,r_2}$ we have

$$(r-r_2)\left(f(\breve{\mathbf{\Omega}}''_{r_1}) - r_1\right)^+ + (r-r_1)\left(f(\breve{\mathbf{\Omega}}''_{r_2}) - r_2\right)^+ + (r_1+r_2-r)\left(f(\breve{\mathbf{\Omega}}''_{r_1,r_2}) - (r_1+r_2-r)\right)^+ \geq K', \tag{24}$$

where $f(\mathbf{X})$ denotes the number of nonzero rows of $\mathbf{X}$ for any matrix $\mathbf{X}$ and $\breve{\mathbf{\Omega}}''_{r_1,r_2} = [\breve{\mathbf{\Omega}}''_{r_1}|\breve{\mathbf{\Omega}}''_{r_2}]$, and also $\breve{\mathbf{\Omega}}''_{r_1}$ and $\breve{\mathbf{\Omega}}''_{r_2}$ denote the columns of $\breve{\mathbf{\Omega}}''_{r_1,r_2}$ corresponding to $\breve{\mathbf{\Omega}}_{r_1}$ and $\breve{\mathbf{\Omega}}_{r_2}$, respectively.

The following lemma is a re-statement of Theorem 2 in (Ashraphijuo et al., 2017c).

Lemma 22 *With probability one, there are only finitely many completions of the sampled multi-view data if and only if Conditions $A_{r_1^*,r_2^*}$ and $B_{r_1^*,r_2^*,r^*}$ hold.*

Definition 23 *Denote the rank vector $\underline{r} = (r_1, r_2, r)$. Define the generalized inequality $\underline{r}' \preceq \underline{r}$ as the component-wise set of inequalities, e.g., $r'_1 \leq r_1$, $r'_2 \leq r_2$ and $r' \leq r$.*

Definition 24 *Let $\mathcal{S}_{\mathbf{\Omega}}$ denote the set of all $\underline{r}$ such that both Conditions A_{r_1,r_2} and $B_{r_1,r_2,r}$ hold.*

Lemma 25 *Assume $\underline{r} \in \mathcal{S}_{\mathbf{\Omega}}$. Then, for any $\underline{r}' \preceq \underline{r}$, we have $\underline{r}' \in \mathcal{S}_{\mathbf{\Omega}}$.*

Proof We consider the rank factorization of $\mathbf{U}$ as in (Ashraphijuo et al., 2017c) and similar to the single-view scenario in Lemma 3 each observed entry results in a polynomial in terms of the entries of the components of the decomposition. Note that the dimension of the manifold corresponding to rank vector $\underline{r}$ is equal to $rn+r_1n_1+r_2n_2-r^2-r_1^2-r_2^2+r(r_1+r_2)$ (Ashraphijuo et al., 2017c), and also observe that the fact that $\max\{r_1,r_2\} \leq r \leq r_1+r_2 \leq \min\{2n, n_1+n_2\}$ implies that reducing any of the values r_1, r_2, and r reduces the value of $rn+r_1n_1+r_2n_2-r^2-r_1^2-r_2^2+r(r_1+r_2)$. Hence, the dimension of the manifold corresponding to rank vector $\underline{r}$ is larger than that for rank vector $\underline{r}'$, given $\underline{r}' \preceq \underline{r}$, and thus similar to the proof of Lemma 3, finite completability of data with $\underline{r}$ results finite completability of data with $\underline{r}'$ with probability one. Then, using Lemma 22, the proof is complete. ■

The following theorem provides a relationship between the unknown rank vector $\underline{r}^*$ and $\mathcal{S}_{\mathbf{\Omega}}$.

Theorem 26 *With probability one, exactly one of the following statements holds*
(i) $\underline{r}^* \in \mathcal{S}_{\mathbf{\Omega}}$;
(ii) For any arbitrary completion of the sampled matrix $\mathbf{U}$ of rank vector $\underline{r}$, we have $\underline{r} \notin \mathcal{S}_{\mathbf{\Omega}}$.

Proof Similar to the proof of Theorem 4, suppose that there does not exist a completion of $\mathbf{U}$ of rank vector $\underline{r}$ such that $\underline{r} \in \mathcal{S}_{\mathbf{\Omega}}$. Therefore, it is easily verified that statement (ii) holds and statement (i) does not hold. On the other hand, assume that there exists a completion of $\mathbf{U}$ of rank vector $\underline{r}$, where $\underline{r} \in \mathcal{S}_{\mathbf{\Omega}}$. Hence, statement (ii) does not hold and to complete the proof it suffices to show that with probability one, statement (i) holds. Similar to Theorem 4, we show that assuming $\underline{r}^* \notin \mathcal{S}_{\mathbf{\Omega}}$, there exists a completion of $\mathbf{U}$ of rank vector $\underline{r}$, where $\underline{r} \in \mathcal{S}_{\mathbf{\Omega}}$, with probability zero.

Since $\underline{r}^* \notin \mathcal{S}_{\mathbf{\Omega}}$, according to Lemma 25, for any $\underline{r} \in \mathcal{S}_{\mathbf{\Omega}}$ at least one the following inequalities holds; $r_1 < r_1^*$, $r_2 < r_2^*$ and $r < r^*$. Note that assuming that there exists a completion of $\mathbf{U}_1$ of rank r_1 with probability zero results that there exists a completion of $\mathbf{U}$ of rank vector $\underline{r}$ with probability zero and similar statement holds for r_2 and r. Hence, in any possible scenario ($r_1 < r_1^*$ or $r_2 < r_2^*$ or $r < r^*$) the similar proof as in Theorem 4 (for single-view matrix) results that there exists a completion of $\mathbf{U}$ of rank vector $\underline{r}$, where $\underline{r} \in \mathcal{S}_{\mathbf{\Omega}}$, with probability zero. ■

Corollary 27 *Consider a subset $\mathcal{S}'_{\mathbf{\Omega}}$ of $\mathcal{S}_{\mathbf{\Omega}}$ such that for any two members of $\mathcal{S}_{\mathbf{\Omega}}$ that $\underline{r}' \preceq \underline{r}''$ and $\underline{r}'' \in \mathcal{S}'_{\mathbf{\Omega}}$ we have $\underline{r}' \in \mathcal{S}'_{\mathbf{\Omega}}$. Then, with probability one, exactly one of the followings holds*
(i) $\underline{r}^* \in \mathcal{S}'_{\mathbf{\Omega}}$;
(ii) For any arbitrary completion of $\mathbf{U}$ of rank vector $\underline{r}$, we have $\underline{r} \notin \mathcal{S}'_{\mathbf{\Omega}}$.

Proof Note that the property in the statement of Lemma 25 holds for $\mathcal{S}'_{\mathbf{\Omega}}$ as well as $\mathcal{S}_{\mathbf{\Omega}}$. Moreover, as $\mathcal{S}'_{\mathbf{\Omega}} \subseteq \mathcal{S}_{\mathbf{\Omega}}$, for any $\underline{r} \in \mathcal{S}'_{\mathbf{\Omega}}$ there exists at most finitely many completions of $\mathbf{U}$ of rank vector $\underline{r}$, and therefore the rest of the proof is the same as the proof of Theorem 26. ■

Corollary 28 *Assuming that there exists a completion of $\mathbf{U}$ with rank vector $\underline{r}$ such that $\underline{r} \in \mathcal{S}_{\mathbf{\Omega}}$, then with probability one $\underline{r}^* \preceq \underline{r}$.*

The following lemma which is a re-statement of Theorem 3 in (Ashraphijuo et al., 2017c) gives the number of samples per column that is needed to ensure that Conditions A_{r_1,r_2} and $B_{r_1,r_2,r}$ hold with high probability.

Lemma 29 *Suppose that the following inequalities hold*

$$\begin{aligned} \frac{n}{6} &\geq \max\{r_1, r_2, (r_1+r_2-r)\}, && (25)\\ n_1 &\geq (r-r_2)(n-r_1), && (26)\\ n_2 &\geq (r-r_1)(n-r_2), && (27)\\ n_1+n_2 &\geq (r-r_2)(n-r_1)+(r-r_1)(n-r_2) \\ &+ (r_1+r_2-r)(n-(r_1+r_2-r)). && (28)\end{aligned}$$

Moreover assume that each **column** *of* $\mathbf{U}$ *is observed in at least* l *entries, uniformly at random and independently across entries, where*

$$l > \max\left\{9\ \log\left(\frac{n}{\epsilon}\right) + 3\ \log\left(\frac{3\max\{r-r_2, r-r_1, r_1+r_2-r\}}{\epsilon}\right) + 6, 2r_1, 2r_2\right\}. \quad (29)$$

Then, with probability at least $1-\epsilon$, $\underline{r} \in \mathcal{S}_{\mathbf{\Omega}}$.

The following proposition is the probabilistic version of Theorem 26 in terms of the sampling probability instead of verifying Conditions A_{r_1,r_2} and $B_{r_1,r_2,r}$.

Proposition 30 *Suppose that* (25)-(28) *hold for* $\underline{r}$ *and that each entry of the sampled matrix is observed uniformly at random and independently across entries with probability* p*, where*

$$p > \frac{1}{n}\max\left\{9\ \log\left(\frac{n}{\epsilon}\right) + 3\ \log\left(\frac{3\max\{r-r_2, r-r_1, r_1+r_2-r\}}{\epsilon}\right) + 6, 2r_1, 2r_2\right\} + \frac{1}{\sqrt[4]{n}}.$$

Then, with probability at least $(1-\epsilon)\left(1-\exp(-\frac{\sqrt{n}}{2})\right)^{n_1+n_2}$*, we have* $\underline{r} \in \mathcal{S}_{\mathbf{\Omega}}$.

Proof The proposition is easy to verify using Lemma 29 and Lemma 9 (similar to the proof for Proposition 11). ■

Corollary 31 *Assuming that there exists a completion of* $\mathbf{U}$ *of rank vector* $\underline{r}$ *such that* (25)-(28) *hold and the sampling probability satisfies* (30)*, then with probability at least* $(1-\epsilon)\left(1-\exp(-\frac{\sqrt{n}}{2})\right)^{n_1+n_2}$ *we have* $\underline{r}^* \preceq \underline{r}$.

4.2 Tucker-Rank Tensor

Let $\mathbb{P}_i$ denote the Lebesgue measures on $\mathbb{R}^{n_i \times m_i^*}$, $i = j+1, \ldots, d$, and $\mathbb{P}_0$ denotes the Lebesgue measures on $\mathbb{R}^{m_{j+1}^* \times m_{j+2}^* \times \ldots \times m_d^*}$. In this subsection, we assume that the sampled tensor $\mathcal{U} \in \mathbb{R}^{n_1 \times \ldots \times n_d}$ is chosen generically from the manifold of tensors of rank $\underline{r}^* = \text{rank}_{\text{Tucker}}(\mathcal{U}) = (m_{j+1}^*, \ldots, m_d^*)$ (where $\underline{r}^*$ is unknown), or in other words, the entries of $\mathcal{U}$ are drawn independently with respect to Lebesgue measure on the corresponding manifold. Hence, the probability measures of all statements in this subsection are $\mathbb{P}_0 \times \mathbb{P}_{j+1} \times \mathbb{P}_{j+2} \times \ldots \times \mathbb{P}_d$.

Without loss of generality assume that $m_{j+1}^* \geq \ldots \geq m_d^*$ throughout this subsection. Also, given $\underline{r} = (m_{j+1}, \ldots, m_d)$, define the following function

$$g_{\underline{r}}(x) = \sum_{i=j+1}^{d} \min\left\{r_i, \left(x - \sum_{i'=j+1}^{i-1} r_{i'}\right)^+\right\} r_i. \quad (30)$$

Definition 32 *For any* $i \in \{j+1, \ldots, d\}$ *and* $\mathcal{S}_i \subseteq \{1, \ldots, n_i\}$*, define* $\mathcal{U}^{(\mathcal{S}_i)}$ *as a set containing the entries of* $|\mathcal{S}_i|$ *rows (corresponding to the elements of* $\mathcal{S}_i$*) of* $\mathbf{U}_{(i)}$*. Moreover, define* $\mathcal{U}^{(\mathcal{S}_{j+1}, \ldots, \mathcal{S}_d)} = \mathcal{U}^{(\mathcal{S}_{j+1})} \cup \ldots \cup \mathcal{U}^{(\mathcal{S}_d)}$.

Condition $\mathcal{A}_{\underline{r}}^{\text{Tucker}}$: There exist $\sum_{i=j+1}^{d}(n_i m_i)$ observed entries such that for any $\mathcal{S}_i \subseteq \{1, \ldots, n_i\}$ for $i \in \{j+1, \ldots, d\}$, $\mathcal{U}^{(\mathcal{S}_{j+1}, \ldots, \mathcal{S}_d)}$ includes at most $\sum_{i=j+1}^{d} |\mathcal{S}_i| m_i$ of the mentioned $\sum_{i=j+1}^{d} n_i m_i$ observed entries.

Let $\mathcal{P}$ be a set of $\sum_{i=j+1}^{d}(n_i m_i)$ observed entries such that they satisfy Condition $\mathcal{A}_{\underline{r}}^{\text{Tucker}}$. Now, we construct a $(j+1)^{\text{th}}$-order binary constraint tensor $\breve{\Omega}_{\underline{r}}$ in some sense similar to that in Section 3.2. For any subtensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_j \times 1 \times \cdots \times 1}$ of the tensor $\mathcal{U}$, let $N_\Omega(\mathcal{Y}^{\mathcal{P}})$ denote the number of sampled entries in $\mathcal{Y}$ that belong to $\mathcal{P}$.

The sampled tensor $\mathcal{U}$ includes $n_{j+1} n_{j+2} \cdots n_d$ subtensors that belong to $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_j \times 1 \times \cdots \times 1}$ and we label these subtensors by $\mathcal{Y}_{(t_{j+1}, \ldots, t_d)}$ where $(t_{j+1}, \ldots, t_d)$ represents the coordinate of the subtensor. Define a binary valued tensor $\breve{\mathcal{Y}}_{(t_{j+1}, \cdots, t_d)} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_j \times \overbrace{1 \times \ldots \times 1}^{d-j} \times k}$, where $k = N_\Omega(\mathcal{Y}_{(t_{j+1}, \ldots, t_d)}) - N_\Omega(\mathcal{Y}^{\mathcal{P}}_{(t_{j+1}, \ldots, t_d)})$ and its entries are described as the following. We can look at $\breve{\mathcal{Y}}_{(t_{j+1}, \cdots, t_d)}$ as k tensors each belongs to $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_j \times 1 \times \cdots \times 1}$. For each of the mentioned k tensors in $\breve{\mathcal{Y}}_{(t_{j+1}, \cdots, t_d)}$ we set the entries corresponding to the $N_\Omega(\mathcal{Y}^{\mathcal{P}}_{(t_{j+1}, \ldots, t_d)})$ observed entries that belong to $\mathcal{P}$ equal to 1. For each of the other k observed entries, we pick one of the k tensors of $\breve{\mathcal{Y}}_{(t_{j+1}, \cdots, t_d)}$ and set its corresponding entry (the same location as that specific observed entry) equal to 1 and set the rest of the entries equal to 0.

For the sake of simplicity in notation, we treat tensors $\breve{\mathcal{Y}}_{(t_{j+1}, \cdots, t_d)}$ as a member of $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_j \times k}$ instead of $\mathbb{R}^{n_1 \times n_2 \times \cdots \times n_j \times \overbrace{1 \times \cdots \times 1}^{d-j} \times k}$. Now, by putting together all $n_{j+1} n_{j+2} \cdots n_d$ tensors in dimension $(j+1)$, we construct a binary valued tensor $\breve{\Omega}_{\underline{r}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_j \times K_j}$, where $K_j = N_\Omega(\mathcal{U}) - \sum_{i=j+1}^{d}(n_i m_i)$ and call it the **constraint tensor** (Ashraphijuo et al., 2016a). In (Ashraphijuo et al., 2016a), an example is given on the construction of $\breve{\Omega}_{\underline{r}}$.

Condition $\mathcal{B}_{\underline{r}}^{\text{Tucker}}$: The constraint tensor $\breve{\Omega}_{\underline{r}}$ consists a subtensor $\breve{\Omega}'_{\underline{r}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_j \times K}$ such that $K = \left(\Pi_{i=1}^{j} n_i\right)\left(\Pi_{i=j+1}^{d} m_i\right) - \sum_{i=j+1}^{d} m_i^2$ and for any $K' \in \{1, 2, \ldots, K\}$ and any subtensor $\breve{\Omega}''_{\underline{r}} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times K'}$ of $\breve{\Omega}'_{\underline{r}}$ we have

$$\left(\Pi_{i=j+1}^{d} m_i\right)\left(f_{j+1}(\breve{\Omega}''_{\underline{r}})\right) - g_{\underline{r}}\left(f_{j+1}(\breve{\Omega}''_{\underline{r}})\right) \geq K', \quad (31)$$

where $f_{j+1}(\breve{\Omega}''_{\underline{r}})$ denotes the number of nonzero columns of the $(j+1)$-th matricization of $\breve{\Omega}''_{\underline{r}}$.

The following lemma is a re-statement of Theorem 3 in (Ashraphijuo et al., 2016a).

Lemma 33 *With probability one, there are only finitely many completions of rank* $\underline{r}^*$ *of the sampled tensor if and only if Conditions* $\mathcal{A}_{\underline{r}^*}^{\text{Tucker}}$ *and* $\mathcal{B}_{\underline{r}^*}^{\text{Tucker}}$ *hold.*

Definition 34 *Let* $\mathcal{S}_\Omega$ *denote the set of all rank vectors* $\underline{r}$ *such that both Conditions* $\mathcal{A}_{\underline{r}}^{\text{Tucker}}$ *and* $\mathcal{B}_{\underline{r}}^{\text{Tucker}}$ *hold.*

Lemma 35 *Assume* $\underline{r} \in \mathcal{S}_\Omega$*. Then, for any rank vector* $\underline{r}' \preceq \underline{r}$*, we have* $\underline{r}' \in \mathcal{S}_\Omega$.

Proof Note that the dimension of the manifold corresponding to $\underline{r}$ is $\left(\Pi_{i=1}^{j} n_i\right)\left(\Pi_{i=j+1}^{d} m_i\right) + \sum_{i=j+1}^{d} n_i m_i - \sum_{i=j+1}^{d} m_i^2$, and thus by reducing the value of m_{i_0} by one (for $i_0 \in \{j+1,\dots,d\}$), the value of the mentioned dimension reduces by at least $\left(\Pi_{i=1}^{j} n_i\right) + n_i - 2m_i + 1$, which is greater than zero since $m_i \leq n_i$. The rest of the proof is similar to the proof of Lemma 3. ■

Definition 36 *Define $\mathcal{S}_\Omega(\underline{r})$ as a subset of $\mathcal{S}_\Omega$, which includes all $\underline{r}' \in \mathcal{S}_\Omega$ that $\underline{r}' \preceq \underline{r}$.*

The following theorem gives a relationship between $\underline{r}^*$ and $\mathcal{S}_\Omega$.

Theorem 37 *With probability one, exactly one of the following statements holds*
(i) $\underline{r}^* \in \mathcal{S}_\Omega$;
(ii) For any arbitrary completion of the sampled tensor $\mathcal{U}$ of rank $\underline{r}$, we have $\underline{r} \notin \mathcal{S}_\Omega(\underline{r}^*)$.

Proof Similar to the proof of Theorem 4, to complete the proof it suffices to show that the assumption $\underline{r}^* \notin \mathcal{S}_\Omega$ results that there exists a completion of $\mathcal{U}$ of rank $\underline{r}$, where $\underline{r} \in \mathcal{S}_\Omega(\underline{r}^*)$, with probability zero. Note that $\underline{r} \in \mathcal{S}_\Omega(\underline{r}^*) \subseteq \mathcal{S}_\Omega$ results that Conditions $\mathcal{A}_{\underline{r}}^{\text{Tucker}}$ and $\mathcal{B}_{\underline{r}}^{\text{Tucker}}$ hold. Moreover, note that $\underline{r} \preceq \underline{r}^*$ and since $\underline{r}^* \notin \mathcal{S}_\Omega$ we conclude that there exists $i_0 \in \{j+1,\dots,d\}$ such that $m_{i_0} < m_{i_0}^*$. As a result, $\sum_{i=j+1}^{d} n_i m_i < \sum_{i=j+1}^{d} n_i m_i^*$.

Condition $\mathcal{B}_{\underline{r}}^{\text{Tucker}}$ ensures there exists at least one more observed entry (otherwise the constraint tensor does not exist) besides the $\sum_{i=j+1}^{d} n_i m_i$ mentioned observed entries. Given the basis $\mathcal{C} \in \mathbb{R}^{n_1 \times \dots \times n_j \times m_{j+1} \times \dots \times m_d}$ as in (7), there exist $\sum_{i=j+1}^{d} n_i m_i$ variables in the corresponding Tucker decomposition. However, we have $\sum_{i=j+1}^{d} n_i m_i + 1$ polynomials in terms these $\sum_{i=j+1}^{d} n_i m_i$ variables and therefore the last polynomials can be written as algebraic combination of the other $\sum_{i=j+1}^{d} n_i m_i$ polynomials. This leads to a linear equation in terms of the $\sum_{i=j+1}^{d} n_i m_i + 1$ corresponding observed entries. On the other hand, the $\sum_{i=j+1}^{d} n_i m_i$ observed entries satisfy the property stated as Condition $\mathcal{A}_{\underline{r}}^{\text{Tucker}}$ and it is easily verified that there exist $\sum_{i=j+1}^{d} n_i m_i^*$ entries (observed and non-observed) satisfying Condition $\mathcal{A}_{\underline{r}^*}^{\text{Tucker}}$ such that the union of the mentioned $\sum_{i=j+1}^{d} n_i m_i$ entries with any arbitrary other observed entry be a subset of those $\sum_{i=j+1}^{d} n_i m_i^*$ entries. However, $\mathcal{U}$ is generically chosen from the manifold corresponding to $\underline{r}^*$ and therefore a particular linear equation in terms of the mentioned $\sum_{i=j+1}^{d} n_i m_i^*$ entries holds with probability zero. The rest of the proof is similar to the proof of Theorem 4. ■

Corollary 38 *Assuming that there exists a completion of $\mathcal{U}$ with rank vector $\underline{r}$ such that $\underline{r} \in \mathcal{S}_\Omega$, we conclude that with probability one $\underline{r}^* \preceq \underline{r}$.*

The following lemma is Corollary 2 in (Ashraphijuo et al., 2016a), which ensures that Conditions $\mathcal{A}_{\underline{r}}^{\text{Tucker}}$ and $\mathcal{B}_{\underline{r}}^{\text{Tucker}}$ hold with high probability.

Lemma 39 *Assume that $\sum_{i=j+1}^{d} m_i^2 \leq \Pi_{i=j+1}^{d} m_i$, $\Pi_{i=j+1}^{d} n_i \geq N_j \Pi_{i=j+1}^{d} m_i - \sum_{i=j+1}^{d} m_i^2$, $\Pi_{i=j+1}^{d} m_i \leq N_j$, where $N_j = \Pi_{i=1}^{j} n_i$. Furthermore, assume that we observe each entry of $\mathcal{U}$ with probability p, where*

$$p > \frac{1}{N_j}\left(6\,\log\left(N_j\right) + 2\log\left(\max\left\{\frac{2\sum_{i=j+1}^{d} r_i^2}{\epsilon}, \frac{2\Pi_{i=j+1}^{d} r_i - 2\sum_{i=j+1}^{d} r_i^2}{\epsilon}\right\}\right) + 4\right) + \frac{1}{\sqrt[4]{N_j}}.$$

Then, with probability at least $(1-\epsilon)\left(1-\exp(-\frac{\sqrt{\Pi_{i=1}^{j} n_i}}{2})\right)^{\Pi_{i=j+1}^{d} n_i}$, $\underline{r} \in \mathcal{S}_\Omega$.

The following corollary is the probabilistic version of Theorem 37.

Corollary 40 *Assuming that there exists a completion of the sampled tensor $\mathcal{U}$ of Tucker rank $\underline{r}$ such that the assumptions in Lemma 39 hold and the sampling probability satisfies* (32)*, then with probability at least $(1-\epsilon)\left(1-\exp(-\frac{\sqrt{\Pi_{i=1}^{j} n_i}}{2})\right)^{\Pi_{i=j+1}^{d} n_i}$ we have $\underline{r}^* \preceq \underline{r}$.*

4.2.1 NUMERICAL RESULTS

We generate a random tensor $\mathcal{U} \in \mathbb{R}^{8\times8\times8\times8\times8}$ of Tucker-rank $(1,3,3,2,2)$. The color scale represents the lower bound on the probability that we can guarantee the rank of a given completion is a component-wise upper bound on the true rank. Then, we solve the following convex optimization problem for different values of the sampling probability.

$$\begin{aligned} &\text{minimize}_{\mathcal{U}' \in \mathbb{R}^{n_1 \times \cdots \times n_d}} && \|\sum_{i=1}^{d} \mathbf{U}'_{(i)}\|_* \\ &\text{subject to} && \mathcal{U}'_\Omega = \mathcal{U}_\Omega. \end{aligned} \tag{32}$$

Then, we calculate rank of each matricization of the tensor obtained via solving (32) to find its Tucker-rank. In this scenario, for each component of the Tucker-rank, we find the percentage of error via $\frac{m_i - m_i^*}{n - m_i^*} \times 100\%$, where $n = 8$, m_i and m_i^* are the i-th rank component of the obtained tensor and original tensor, respectively. Hence, 100% error simply means that the corresponding matricization is full rank. In Figure 8, gap represents the average of the defined error over all components of Tucker-rank, i.e., over all matricizations.

4.3 TT-Rank Tensor

Let $\mathbb{P}_i$ denote the Lebesgue measures on $\mathbb{R}^{u_{i-1}^* \times n_i \times u_i^*}$, $i = 1,\dots,d$, where $u_0^* = u_d^* = 1$. In this subsection, we assume that the sampled tensor $\mathcal{U} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ is chosen generically from the manifold of tensors of rank $\underline{r}^* = \text{rank}_{\text{TT}}(\mathcal{U}) = (u_1^*,\dots,u_{d-1}^*)$ (where $\underline{r}^*$ is unknown), or in other words, the entries of $\mathcal{U}$ are drawn independently with respect to Lebesgue measure on the corresponding manifold. Hence, the probability measures of all statements in this subsection are $\mathbb{P}_1 \times \dots \times \mathbb{P}_d$.

Condition $\mathcal{A}_{\underline{r}}^{\text{TT}}$: Each row of the d-th matricization of the sampled tensor, i.e., $\mathbf{U}_{(d)}$ includes at least u_{d-1} observed entries.

We construct the d-way binary valued constraint tensor $\breve{\Omega}_{u_{d-1}}$ similar to that in Section 3.2 as the following. Consider any subtensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_{d-1} \times 1}$ of the tensor $\mathcal{U}$. The sampled tensor

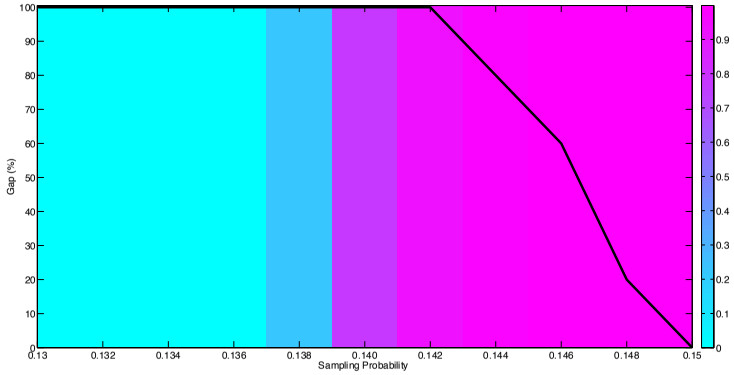


Figure 8: The rank gap as a function of sampling probability for $\mathcal{U} \in \mathbb{R}^{8\times8\times8\times8\times8\times8}$ of Tucker-rank $(1, 3, 3, 2, 2)$.

$\mathcal{U}$ includes n_d subtensors that belong to $\mathbb{R}^{n_1\times n_2\times\cdots\times n_{d-1}\times 1}$ and let $\mathcal{Y}_i$ for $1 \leq i \leq n_d$ denote these n_d subtensors. Define a binary valued tensor $\breve{\mathcal{Y}}_i \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_{d-1}\times k_i}$, where $k_i = N_\Omega(\mathcal{Y}_i) - u_{d-1}$ and its entries are described as the following. We can look at $\breve{\mathcal{Y}}_i$ as k_i tensors each belongs to $\mathbb{R}^{n_1\times n_2\times\cdots\times n_{d-1}\times 1}$. For each of the mentioned k_i tensors in $\breve{\mathcal{Y}}_i$ we set the entries corresponding to u_{d-1} of the observed entries equal to 1. For each of the other k_i observed entries, we pick one of the k_i tensors of $\breve{\mathcal{Y}}_i$ and set its corresponding entry (the same location as that specific observed entry) equal to 1 and set the rest of the entries equal to 0. In the case that $k_i = 0$ we simply ignore $\breve{\mathcal{Y}}_i$, i.e., $\breve{\mathcal{Y}}_i = \emptyset$

By putting together all n_d tensors in dimension d, we construct a binary valued tensor $\breve{\Omega}_{u_{d-1}} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_{d-1}\times K}$, where $K = \sum_{i=1}^{n_d} k_i = N_\Omega(\mathcal{U}) - u_{d-1}n_d$ and call it the **constraint tensor**. Observe that each subtensor of $\breve{\Omega}_{u_{d-1}}$ which belongs to $\mathbb{R}^{n_1\times n_2\times\cdots\times n_{d-1}\times 1}$ includes exactly $u_{d-1}+1$ nonzero entries. In (Ashraphijuo and Wang, 2017a), an example is given on the construction of $\breve{\Omega}_{u_{d-1}}$.

Condition $\mathcal{B}^{\mathrm{TT}}_{\underline{r}}$: $\breve{\Omega}_{u_{d-1}}$ consists a subtensor $\breve{\Omega}'_{u_{d-1}} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_{d-1}\times K}$ such that $K = \sum_{i=1}^{d-1} u_{i-1}n_iu_i - \sum_{i=1}^{d-1} u_i^2$ and for any $K' \in \{1, 2, \ldots, K\}$ and any subtensor $\breve{\Omega}''_{u_{d-1}} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_{d-1}\times K'}$ of $\breve{\Omega}'_{u_{d-1}}$ we have

$$\sum_{i=1}^{d-1} \left(u_{i-1}f_i(\breve{\Omega}''_{u_{d-1}})u_i - u_i^2\right)^+ \geq K', \tag{33}$$

where $f_i(\breve{\Omega}''_{u_{d-1}})$ denotes the number of nonzero rows of the i-th matricization of $\breve{\Omega}''_{u_{d-1}}$.

The following lemma is a re-statement of Theorem 1 in (Ashraphijuo and Wang, 2017a).

Lemma 41 *With probability one, there are only finitely many completions of rank $\underline{r}^*$ of the sampled tensor if and only if Conditions $\mathcal{A}^{TT}_{\underline{r}^*}$ and $\mathcal{B}^{TT}_{\underline{r}^*}$ hold.*

Definition 42 *Let $\mathcal{S}_\Omega$ denote the set of all rank vectors $\underline{r}$ such that both Conditions $\mathcal{A}^{TT}_{\underline{r}}$ and $\mathcal{B}^{TT}_{\underline{r}}$ hold.*

The following lemma will be used in Lemma 44.

Lemma 43 $u_i \leq \min\{u_{i-1}n_i, u_{i+1}n_{i+1}\}$ *for* $1 \leq i \leq d-1$.

Proof We first show that $u_i \leq u_{i-1}n_i$, which is easily verified for $i = 1$ as $\widetilde{\mathbf{U}}_1$ includes n_1 rows and $u_0 = 1$, and therefore assume that $i > 1$. Define the $(d-1)$-way tensor $\mathcal{U}^{l_i} \in \mathbb{R}^{n_1\times\ldots\times n_{i-1}\times n_{i+1}\times\ldots\times n_d}$ such that $\mathcal{U}^{l_i}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d) = \mathcal{U}(x_1, \ldots, x_{i-1}, l_i, x_{i+1}, \ldots, x_d)$ for $1 \leq i \leq d$ and $1 \leq l_i \leq n_i$. Also, recall that $\widetilde{\mathbf{U}}^{l_i}_{(i-1)}$ denotes the $(i-1)$-th unfolding of $\mathcal{U}^{l_i}$. Observe that $\widetilde{\mathbf{U}}^{l_i}_{(i-1)}$ is a subset of columns of matrix $\widetilde{\mathbf{U}}_{(i-1)}$ (those columns that correspond to the entries of $\mathcal{U}$ with the i-th component of the location equal to l_i). Therefore, $\mathrm{rank}(\widetilde{\mathbf{U}}^{l_i}_{(i-1)}) \leq \mathrm{rank}(\widetilde{\mathbf{U}}_{(i-1)}) = u_{i-1}$.

On the other hand, observe that $\widetilde{\mathbf{U}}^{l_i}_{(i-1)}$ is a subset of rows of $\widetilde{\mathbf{U}}_{(i)}$ (those rows that correspond to the entries of $\mathcal{U}$ with the i-th component of the location equal to l_i). Hence, the union of rows of $\widetilde{\mathbf{U}}^{l_i}_{(i-1)}$'s for $1 \leq l_i \leq n_i$ constitute all rows of $\widetilde{\mathbf{U}}_{(i)}$. Therefore, $u_i = \mathrm{rank}(\widetilde{\mathbf{U}}_{(i)}) \leq \sum_{l_i=1}^{n_i} \mathrm{rank}(\widetilde{\mathbf{U}}^{l_i}_{(i-1)}) \leq n_iu_{i-1}$. Similarly, we can show that $u_i \leq u_{i+1}n_{i+1}$ to complete the proof. ■

Lemma 44 *Assume $\underline{r} \in \mathcal{S}_\Omega$. Then, for any $\underline{r}' \preceq \underline{r}$, we have $\underline{r}' \in \mathcal{S}_\Omega$.*

Proof Note that the dimension of the manifold corresponding to $\underline{r}$ is $\sum_{i=1}^{d} u_{i-1}n_iu_i - \sum_{i=1}^{d-1} u_i^2$. If we reduce the value of u_i by one, the value of the mentioned dimension reduces by $u_{i-1}n_i + u_{i+1}n_{i+1} - 2u_i + 1$. According to Lemma 43, $u_{i-1}n_i + u_{i+1}n_{i+1} - 2u_i + 1$ is greater than zero, and therefore $\underline{r}' \preceq \underline{r}$ results that the dimension of the manifold corresponding to $\underline{r}$ is greater than that corresponding to $\underline{r}'$. The rest of the proof is similar to the proof of Lemma 3. ■

Definition 45 *Define $\hat{\mathcal{S}}_\Omega$ as the set of all rank vectors $\underline{r} \in \mathcal{S}_\Omega$ such that there exists a rank vector $\underline{r}' \in \mathcal{S}_\Omega$ with $\underline{r} \preceq \underline{r}'$ and $u_{d-1} < u'_{d-1}$ (instead of $u_{d-1} \leq u'_{d-1}$). Note that $\hat{\mathcal{S}}_\Omega$ also satisfies the property in Lemma 44.*

Theorem 46 *With probability one, exactly one of the following statements holds:*
(i) $\underline{r}^* \in \hat{\mathcal{S}}_\Omega$;
(ii) For any arbitrary completion of the sampled tensor $\mathcal{U}$ of rank $\underline{r}$, we have $\underline{r} \notin \hat{\mathcal{S}}_\Omega$.

Proof Similar to the proof of Theorem 4, to complete the proof it suffices to show that the assumption $\underline{r}^* \notin \hat{\mathcal{S}}_\Omega$ results that there exists a completion of $\mathcal{U}$ of rank $\underline{r}$, where $\underline{r} \in \hat{\mathcal{S}}_\Omega$, with probability zero. Define the multiplication $\mathcal{U}^{(1)}\ldots\mathcal{U}^{(d-1)}$ in (9) as the basis of the rank $\underline{r}$ TT decomposition of $\mathcal{U}$. Then, by considering the $(d-1)$-th unfolding of $\mathcal{U}^{(1)}\ldots\mathcal{U}^{(d-1)}$ in TT decomposition we obtain a matrix factorization of the $(d-1)$-th unfolding of $\mathcal{U}$. The rest of the proof is similar to the proof of Theorem 4. ■

Similar to Theorem 46, we can show the following.

Corollary 47 *Consider a subset $\hat{\mathcal{S}}'_\Omega$ of $\hat{\mathcal{S}}_\Omega$ such that for any two members of $\hat{\mathcal{S}}_\Omega$ that $\underline{r}'' \preceq \underline{r}'$ and $\underline{r}' \in \hat{\mathcal{S}}'_\Omega$ we have $\underline{r}'' \in \hat{\mathcal{S}}'_\Omega$. Then, with probability one, exactly one of the followings holds*
(i) $\underline{r}^* \in \hat{\mathcal{S}}'_\Omega$;
(ii) For any arbitrary completion of $\mathcal{U}$ of rank vector $\underline{r}$, we have $\underline{r} \notin \hat{\mathcal{S}}'_\Omega$.

Corollary 48 *Assuming that there exists a completion of $\mathcal{U}$ with rank vector $\underline{r}$ such that $\underline{r} \in \hat{\mathcal{S}}_\Omega$, we conclude that with probability one $\underline{r}^* \preceq \underline{r}$.*

The following lemma is Lemma 14 in (Ashraphijuo and Wang, 2017a), which ensures that Conditions $\mathcal{A}^{\text{TT}}_{\underline{r}}$ and $\mathcal{B}^{\text{TT}}_{\underline{r}}$ hold with high probability.

Lemma 49 *Define $m = \sum_{k=1}^{d-2} u_{k-1}u_k$, $M = n\sum_{k=1}^{d-2} u_{k-1}u_k - \sum_{k=1}^{d-2} u_k^2$ and $u' = \max\left\{\frac{u_1}{u_0}, \ldots, \frac{u_{d-2}}{u_{d-3}}\right\}$. Assume that $n_1 = n_2 = \cdots = n_d = n$, $n > \max\{m, 200\}$ and $u' \leq \min\{\frac{n}{6}, u_{d-2}\}$ hold. Moreover, assume that the sampling probability satisfies*

$$p > \frac{1}{n^{d-2}} \max\left\{27\ \log\left(\frac{n}{\epsilon}\right) + 9\ \log\left(\frac{2M}{\epsilon}\right) + 18, 6u_{d-2}\right\} + \frac{1}{\sqrt[4]{n^{d-2}}}. \tag{34}$$

Then, with probability at least $(1-\epsilon)\left(1-\exp(-\frac{\sqrt{n^{d-2}}}{2})\right)^{n^2}$, we have $\underline{r} \in \mathcal{S}_\Omega$.

The following corollary is the probabilistic version of Corollary 48.

Corollary 50 *Assuming that there exists a completion of the sampled tensor $\mathcal{U}$ of TT rank $\underline{r}$ such that the assumptions in Lemma 49 hold and the sampling probability satisfies* (34)*, then with probability at least $(1-\epsilon)\left(1-\exp(-\frac{\sqrt{n^{d-2}}}{2})\right)^{n^2}$ we have $\underline{r}^* \preceq \underline{r}$.*

4.3.1 NUMERICAL RESULTS

We generate a random tensor $\mathcal{U} \in \mathbb{R}^{8\times8\times8\times8\times8\times8}$ of TT-rank $(1, 2, 4, 1, 1)$. The color scale represents the lower bound on the probability that we can guarantee the rank of a given completion is a component-wise upper bound on the true rank. Then, we solve the following convex optimization problem for different values of the sampling probability.

$$\begin{aligned} &\text{minimize}_{\mathcal{U}' \in \mathbb{R}^{n_1\times\cdots\times n_d}} && \|\sum_{i=1}^{d-1} \widetilde{\mathbf{U}}'_{(i)}\|_* \\ &\text{subject to} && \mathcal{U}'_\Omega = \mathcal{U}_\Omega. \end{aligned} \tag{35}$$

Then, we calculate rank of each unfolding of the tensor obtained via solving (35) to find its TT-rank. In this scenario, for each component of the TT-rank, we find the percentage of error via $\frac{u_i - u_i^*}{\min\{n^i, n^{d-i}\} - u_i^*} \times 100\%$, where $n = 8$, $d = 6$, u_i and u_i^* are the i-th rank component of the obtained tensor and original tensor, respectively. Hence, 100% error simply means that the corresponding unfolding is full rank. In Figure 9, gap represents the average of the defined error over all components of TT-rank, i.e., over all unfoldings.

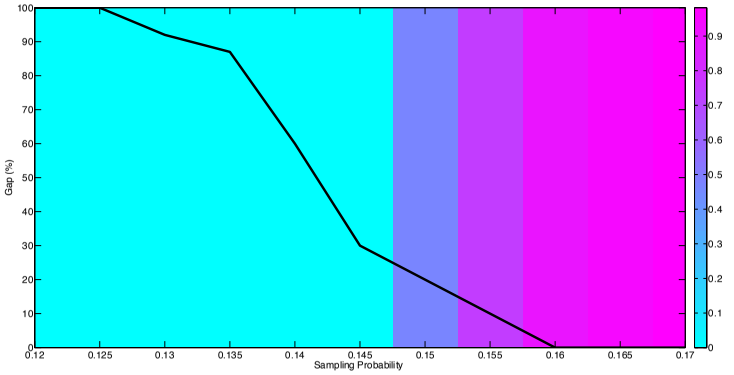


Figure 9: The rank gap as a function of sampling probability for $\mathcal{U} \in \mathbb{R}^{8\times8\times8\times8\times8\times8}$ of TT-rank $(1, 2, 4, 1, 1)$.

5. Conclusions

We make use of the recently developed algebraic geometry analyses that study the fundamental conditions on the sampling patterns for finite completability under a number of low-rank matrix and tensor models to treat the problem of rank approximation for a partially sampled data. Particularly, the goal is to approximate the unknown scalar or vector rank based on the sampling pattern and the rank of a given completion. A number of data models have been treated, including single-view matrix, multi-view matrix, CP tensor, tensor-train tensor and Tucker tensor. First we have provided an upper bound on the unknown scalar rank (for single-view matrix and CP tensor) and an component-wise upper bound on the vector rank (for multi-view matrix, Tucker tensor and TT tensor) with probability one assuming that the sampling pattern satisfies the proposed combinatorial conditions. Moreover, we have also provided probabilistic versions of such bounds that hold with high probability assuming that the sampling probability is above a threshold. In addition, for single-view matrix and CP tensor, these upper bounds can be exactly equal to the unknown scalar rank given the lowest-rank completion. To illustrate how tight our proposed upper bounds are, we have provided some numerical results for the single-view matrix case in which we applied the nuclear norm minimization to find a low-rank completion of the sampled data and observe that the proposed upper bound is almost equal to the true unknown rank.

Acknowledgments

This work was supported in part by the U.S. National Science Foundation (NSF) under grant CIF1064575, and in part by the U.S. Office of Naval Research (ONR) under grant N000141410667.

References

Morteza Ashraphijuo and Xiaodong Wang. Characterization of deterministic and probabilistic sampling patterns for finite completability of low tensor-train rank tensor. *arXiv preprint:1703.07698*,

2017a.

Morteza Ashraphijuo and Xiaodong Wang. Fundamental conditions for low-CP-rank tensor completion. *Journal of Machine Learning Research*, 18 (63):1–29, 2017b.

Morteza Ashraphijuo and Xiaodong Wang. Scaled nuclear norm minimization for low-rank tensor completion. *arXiv preprint:1707.07976*, 2017c.

Morteza Ashraphijuo, Ramtin Madani, and Javad Lavaei. Inverse function theorem for polynomial equations using semidefinite programming. In *IEEE 54th Annual Conference on Decision and Control (CDC)*, pages 6589–6596, 2015.

Morteza Ashraphijuo, Vaneet Aggarwal, and Xiaodong Wang. Deterministic and probabilistic conditions for finite completability of low rank tensor. *arXiv preprint:1612.01597*, 2016a.

Morteza Ashraphijuo, Ramtin Madani, and Javad Lavaei. Characterization of rank-constrained feasibility problems via a finite number of convex programs. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 6544–6550, 2016b.

Morteza Ashraphijuo, Ramtin Madani, and Javad Lavaei. Characterization of rank-constrained feasibility problems via a finite number of convex programs. In *IEEE 55th Conference on Decision and Control (CDC)*, pages 6544–6550, 2016c.

Morteza Ashraphijuo, Vaneet Aggarwal, and Xiaodong Wang. A characterization of sampling patterns for low-Tucker-rank tensor completion problem. In *IEEE International Symposium on Information Theory (ISIT)*, pages 531–535, 2017a.

Morteza Ashraphijuo, Xiaodong Wang, and Vaneet Aggarwal. A characterization of sampling patterns for low-rank multi-view data completion problem. In *IEEE International Symposium on Information Theory (ISIT)*, pages 1147–1151, 2017b.

Morteza Ashraphijuo, Xiaodong Wang, and Vaneet Aggarwal. Deterministic and probabilistic conditions for finite completability of low-rank multi-view data. *arXiv preprint:1701.00737*, 2017c.

Morteza Ashraphijuo, Xiaodong Wang, and Vaneet Aggarwal. An approximation of the CP-rank of a partially sampled tensor. In *55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2017d.

Jian Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Emmanuel Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Emmanuel J Candès, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM Journal on Imaging Sciences*, 6(1):199–225, 2013.

Lars Eldén. *Matrix Methods in Data Mining and Pattern Recognition*, volume 4. Society for Industrial and Applied Mathematics, 2007.

Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):1–19, 2011.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *arXiv preprint:1605.07272*, 2016.

Nicholas JA Harvey, David R Karger, and Kazuo Murota. Deterministic network coding by matrix completion. In *the annual ACM-SIAM symposium on discrete algorithms*, pages 489–498, 2005.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Annual Symposium on the Theory of Computing*, pages 665–674, 2013.

Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu. Robust video denoising using low rank matrix completion. In *CVPR*, pages 1791–1798, 2010.

Lek-Heng Lim and Pierre Comon. Multiarray signal processing: Tensor decomposition meets compressed sensing. *Comptes Rendus Mecanique*, 338(6):311–320, 2010.

Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

X. Y. Liu, S. Aeron, V. Aggarwal, X. Wang, and M. Y. Wu. Adaptive sampling of RF fingerprints for fine-grained indoor localization. *IEEE Transactions on Mobile Computing*, 15(10):2411–2423, 2016.

Oyetunji E Ogundijo, Abdulkadir Elmas, and Xiaodong Wang. Supplemental material for reverse engineering gene regulatory networks from measurement with missing values.

Oyetunji E Ogundijo, Abdulkadir Elmas, and Xiaodong Wang. Reverse engineering gene regulatory networks from measurement with missing values. *EURASIP Journal on Bioinformatics and Systems Biology*, 2017(1):2, 2017.

D Pimentel-Alarcón. A simpler approach to low-rank tensor canonical polyadic decomposition. In *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 474–481. IEEE, 2016.

D Pimentel-Alarcón, Nigel Boston, and Robert D Nowak. Deterministic conditions for subspace identifiability from incomplete sampling. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2191–2195, 2015.

D Pimentel-Alarcón, L Balzano, R Marcia, R Nowak, and R Willett. Group-sparse subspace clustering with missing data. In *IEEE Workshop on Statistical Signal Processing (SSP)*, pages 1–5, 2016a.

D Pimentel-Alarcón, EDU Robert D Nowak, and WISC EDU. The information-theoretic requirements of subspace clustering with missing data. In *International Conference on Machine Learning*, 2016b.

Daniel Pimentel-Alarcón and Robert Nowak. A converse to low-rank matrix completion. In *IEEE International Symposium on Information Theory (ISIT)*, 2016.

Daniel Pimentel-Alarcón, Laura Balzano, and Robert Nowak. Necessary and sufficient conditions for sketched subspace clustering. In *Allerton Conference on Communication, Control, and Computing*, 2016c.

Daniel Pimentel-Alarcón, Nigel Boston, and Robert Nowak. A characterization of deterministic sampling patterns for low-rank matrix completion. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):623–636, 2016d.

Benjamin Recht and Christopher Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Mathematical Programming Computation*, 5(2):201–226, 2013.

Gil Shabat. Matrix completion using nuclear norm, spectral norm or weighted nuclear norm minimization. 2015.

Nicholas D Sidiropoulos and Anastasios Kyrillidis. Multi-way compressed sensing for sparse low-rank tensors. *IEEE Signal Processing Letters*, 19(11):757–760, 2012.

Journal of Machine Learning Research 18 (2017) 1-32 Submitted 1/17; Revised 8/17; Published 10/17

Bayesian Network Learning via Topological Order

Young Woong Park YWPARK@IASTATE.EDU
College of Business
Iowa State University
Ames, IA 50011, USA

Diego Klabjan D-KLABJAN@NORTHWESTERN.EDU
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208, USA

Editor: Zhihua Zhang

Abstract

We propose a mixed integer programming (MIP) model and iterative algorithms based on topological orders to solve optimization problems with acyclic constraints on a directed graph. The proposed MIP model has a significantly lower number of constraints compared to popular MIP models based on cycle elimination constraints and triangular inequalities. The proposed iterative algorithms use gradient descent and iterative reordering approaches, respectively, for searching topological orders. A computational experiment is presented for the Gaussian Bayesian network learning problem, an optimization problem minimizing the sum of squared errors of regression models with L1 penalty over a feature network with application of gene network inference in bioinformatics.

Keywords: Bayesian networks, topological orders, Gaussian Bayesian network, directed acyclic graphs

1. Introduction

Directed graph G is a directed acyclic graph (DAG) or acyclic digraph if G does not contain a directed cycle. In this paper, we consider a generic optimization problem over a directed graph with acyclic constraints, which require the selected subgraph to be a DAG.

Let us consider a complete digraph G. Let m be the number of nodes in digraph G, $Y \in \mathbb{R}^{m\times m}$ a decision variable matrix associated with the arcs, where Y_{jk} is related to arc (j,k), $supp(Y) \in \{0,1\}^{m\times m}$ the 0-1 (adjacency) matrix with $supp(Y)_{jk} = 1$ if $Y_{jk} \neq 0$, $supp(Y)_{jk} = 0$ otherwise, $G(supp(Y))$ the sub-graph of G defined by $supp(Y)$, and let $\mathcal{A}$ be the collection of all acyclic subgraphs of G. Then, we can write the optimization problem with acyclic constraints as

$$\min_{Y} \; F(Y) \quad s.t. \quad G(supp(Y)) \in \mathcal{A}, \tag{1}$$

where F is a function of Y.

Acyclic constraints (or DAG constraints) appear in many network structured problems. The maximum acyclic subgraph problem (MAS) is to find a subgraph of G with maximum cardinality while the subgraph satisfies acyclic constraints. MAS can be written in the form of (1) with $F(Y) = -\|supp(Y)\|_0$. Although exact algorithms were proposed for a superclass of cubic graphs (Fernau and Raible, 2008) and for general directed graphs (Kaas, 1981), most of the works have focused on approximations (Even et al., 1998; Hassin and Rubinstein, 1994) or inapproximability (Guruswami et al., 2008) of either MAS or the minimum feedback arc set problem (FAS). FAS of a directed graph G is a subgraph of G that creates a DAG when the arcs in the feedback arc set are removed from G. Note that MAS is closely related to FAS and is dual to the minimum FAS. Finding a feedback arc set with minimum cardinality is $\mathcal{NP}$-complete in general (Karp, 1972). However, minimum FAS is solvable in polynomial time for some special graphs such as planar graphs (Lucchesi and Younger, 1978) and reducible flow graphs (Ramachandran, 1988), and a polynomial time approximation scheme was developed for a special case of minimum FAS, where exactly one arc exists between any two nodes, called tournament (Kenyon-Mathieu and Schudy, 2007).

DAGs are also extensively studied in Bayesian network learning. Given observational data with m features, the goal is to find the true unknown underlying network of the nodes (features) while the selected arcs (dependency relationship between features) do not create a cycle. In the literature, approaches are classified into three categories: (i) score-based approaches that try to optimize a score function defined to measure fitness, (ii) constraint-based approaches that test conditional independence to check existence of arcs between nodes (iii) and hybrid approaches that use both constraint and score-based approaches. Although there are many approaches based on the constraint-based or hybrid approaches, our focus is solving (1) by means of score-based approaches. For a detailed discussion of constraint-based and hybrid approaches and models for undirected graphs, the reader is referred to Aragam and Zhou (2015) and Han et al. (2016).

For estimating the true network structure by a score-based approach, various functions have been used as different functions give different solutions and behave differently. Many works focus on penalized least squares, where penalty is used to obtain sparse solutions. Popular choices of the penalty term include BIC (Lam and Bacchus, 1994), L_0-penalty (Chickering, 2002; Van de Geer and Bühlmann, 2013), L_1-penalty (Han et al., 2016), and concave penalty (Aragam and Zhou, 2015). Lam and Bacchus (1994) use minimum-description length as a score function, which is equivalent to BIC. Chickering (2002) proposes a two-phase greedy algorithm, called greedy equivalence search, with the L_1 norm penalty. Van de Geer and Bühlmann (2013) study the properties of the L_0 norm penalty and show positive aspects of using L_0 regularization. Raskutti and Uhler Raskutti and Uhler (2013) use a variant of the L_0 norm. They use cardinality of the selected subgraph as the score function where the subgraphs not satisfying the Markov assumption are penalized with a very large penalty. Aragam and Zhou (2015) introduce a generalized penalty, which includes the concave penalty, and develop a coordinate descent algorithm. Han et al. (2016) use the L_1 norm penalty and propose a Tabu search based greedy algorithm for reduced arc sets by neighborhood selection in the pre-processing step.

With any choice of a score function, optimizing the score function is computationally challenging, because the number of possible DAGs of G grows super-exponentially in the number of nodes m and learning Bayesian networks is also shown to be $\mathcal{NP}$-complete (Chickering, 1996). Many heuristic algorithms have been developed based on greedy hill climbing (Chickering, 2002; Heckerman et al., 1995; Han et al., 2016) or coordinate descent (Fu and Zhou, 2013), or enumeration (Raskutti and Uhler, 2013) when the score function

©2017 Young Woong Park and Diego Klabjan.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-033.html.

itself is the main focus. There also exist exact solution approaches based on mathematical programming. One of the natural approaches is based on cycle prevention constraints, which are reviewed in Section 2. The model is covered in Han et al. (2016) as a benchmark for their algorithm, but the MIP based approach does not scale; computational time increases drastically as data size increases and the underlying algorithm cannot solve larger instances. Baharev et al. (2015) studied MIP models for minimum FAS based on triangle inequalities and set covering models. Several works have been focused on the polyhedral study of the acyclic subgraph polytopes (Bolotashvili et al., 1999; Goemans and Hall, 1996; Grötschel et al., 1985; Leung and Lee, 1994). In general, MIP models have gotten relatively less attention due to the scalability issue.

In this paper, we propose an MIP model and iterative algorithms based on the following well-known property of DAGs (Cook et al., 1998).

Property 1 *A directed graph is a DAG if and only if it has a topological order.*

A *topological order* or *topological sort* of a DAG is a linear ordering of all of the nodes in the graph such that the graph contains arc (u, v) if and only if u appears before v in the order (Cormen et al., 2009). Suppose that Z is the adjacency matrix of an acyclic graph. Then, by sorting the nodes of acyclic graph $G(Z)$ based on the topological order, we can create a lower triangular matrix from Z, where row and column indices of the lower triangular matrix are in the topological order. Then, any arc in the lower triangular matrix can be used without creating a cycle. By considering all arcs in the lower triangular matrix, we can optimize F in (1) without worrying to create a cycle. This is an advantage compared to arc-based search, where acyclicity needs to be examined whenever an arc is added. Although the search space of topological orders is very large, a smart search strategy for a topological order may lead to a better algorithm than the existing arc-based search methods. Node orderings are used for Bayesian Network learnings based on Markov chain Monte Carlo methods (Ellis and Wong, 2008; Friedman and Koller, 2003; Niinimäki et al., 2016) as alternatives to network structure-based approaches.

The proposed MIP assigns node orders to all nodes and add constraints to satisfy Property 1. The iterative algorithms search over the topological order space by moving from one topological order to another order. The first algorithm uses the gradient to find a better topological order and the second algorithm uses historical choice of arcs to define the score of the nodes.

With the proposed MIP model and algorithms for (1), we consider a Gaussian Bayesian network learning problem with L_1 penalty for sparsity, which is discussed in detail in Section 4. Out of many possible models in the literature, we pick the L_1-penalized least square model from recently published work of Han et al. (2016), which solves the problem using a Tabu search based greedy algorithm. The algorithm is one of the latest algorithms based on arc search and is shown to be scalable when m is large. Further, their score function, L_1 penalized least squares, is convex and can be solved by standard mathematical optimization packages. Hence, we select the score function from Han et al. (2016) and use their algorithm as a benchmark. In the computational experiment, we compare the performance of the proposed MIP model and algorithms against the algorithm in Han et al. (2016) and other available MIP models for synthetic and real instances.

Our contributions are summarized in the following.

1. We consider a general optimization problem with acyclic constraints and propose an MIP model and iterative algorithms for the problem based on the notion of topological orders.

2. The proposed MIP model has significantly fewer constraints than the other MIP models in the literature, while maintaining the same order of the number of variables. The computational experiment shows that the proposed MIP model outperforms the other MIP models when the subgraph is sparse.

3. The iterative algorithms based on topological orders outperform when the subgraph is dense. They are more scalable than the benchmark algorithm of Han et al. (2016) when the subgraph is dense.

In Section 2, we present the new MIP model along with two MIP models in the literature. In Section 3, we present two iterative algorithms based on different search strategies for topological orders. The Gaussian Bayesian network learning problem with L_1-penalized least square is introduced and computational experiment are presented in Sections 4 and 5, respectively.

In the rest of the paper, we use the following notation.

$J = \{1, \cdots, m\}$ = index set of the nodes
$J^k = J \setminus \{k\}$ = index set of the nodes excluding node k, $k \in J$
$Z = supp(Y) \in \{0, 1\}^{m \times m}$
π = topological order

Given π, we define $\pi_k = q$ to denote that the order of node k is q. For example, given three nodes a, b, c, and topological order $b - c - a$, we have $\pi_a = 3$, $\pi_b = 1$, and $\pi_c = 2$. With this notation, if $\pi_j > \pi_k$, then we can add an arc from j to k.

2. MIP Formulations based on Topological Order

In this section, we present three MIP models for (1). The first and second models, denoted as MIPcp and MIPin, respectively, are models in the literature for similar problems with acyclic constraints. The third model, denoted as MIPto, is the new model we propose based on Property 1.

A popular mathematical programming based approach for solving (1) is the cutting plane algorithm, which is well-known for the traveling salesman problem formulation. Let $\mathcal{C}$ be the set of all possible cycles and $C_l \in \mathcal{C}$ be the set of the arcs defining a cycle. Let $H(supp(Y), C_l)$ be a function that counts the number of selected arcs in $supp(Y)$ from C_l. Then, (1) can be solved by

$$\textbf{MIPcp} \quad \min_{Y} \; F(Y) \quad s.t. \quad H(supp(Y), C_l) \leq |C_l| - 1, C_l \in \mathcal{C}, \tag{2}$$

which can be formulated as an MIP. Note that (2) has exponentially many constraints due to the cardinality of $\mathcal{C}$. Therefore, it is not practical to pass all cycles in $\mathcal{C}$ to a solver. Instead, the cutting plane algorithm starts with an empty active cycle set $\mathcal{C}^A$ and iteratively adds cycles to $\mathcal{C}^A$. That is, the algorithm iteratively solves

$$\min_{Y} \; F(Y) \quad s.t. \quad H(supp(Y), C_l) \leq |C_l| - 1, C_l \in \mathcal{C}^A, \tag{3}$$

with the current active set $\mathcal{C}^A$, detects cycles from the solution, and adds the cycles to $\mathcal{C}^A$. The algorithm terminates when there is no cycle detected from the solution of (3). One of the drawbacks of the cutting plane algorithm based on (3) is that in the worst case we can add all exponentially many constraints. In fact, Han et al. (2016) study the same model and concluded that the cutting plane algorithm does not scale.

Baharev et al. (2015) recently presented MIP models for the minimum feedback arc set problem based on linear ordering and triangular inequalities, where the acyclic constraints presented were previously used for cutting plane algorithms for the linear ordering problem (Grötschel et al., 1984; Mitchell and Borchers, 2000). For any F, we can write the following MIP model based on triangular inequalities presented in Baharev et al. (2015), Grötschel et al. (1984), and Mitchell and Borchers (2000).

$$
\begin{aligned}
\textbf{MIPin} \quad \min \quad & F(Y) && && (4a)\\
s.t. \quad & Z = supp(Y), && && (4b)\\
& Z_{qj} + Z_{jk} - Z_{qk} \leq 1, && 1 \leq q < j < k \leq m, && (4c)\\
& -Z_{qj} - Z_{jk} + Z_{qk} \leq 0, && 1 \leq q < j < k \leq m, && (4d)\\
& Z_{jk} \in \{0,1\}, && 1 \leq j < k \leq m && (4e)
\end{aligned}
$$

Note that Z_{jk} is not defined for all $j \in J^k$ and $k \in J$. Instead of having a full matrix of binary variables, the formulation only uses lower triangle of the matrix using the fact that $Z_{jk} + Z_{kj} = 1$. We can also use this technique to any of the MIP models presented in this paper. However, for ease of explanation, we will use the full matrix, while the computational experiment is done with the reduced number of binary variables. Therefore, the cutting plane algorithm with MIPcp should be more scalable than the implementation in Han et al. (2016), which has twice more binary variables.

Baharev et al. (2015) also provides a set covering based MIP formulation. The idea is similar to MIPcp. In the set covering formulation, each row and column represents a cycle and an arc, respectively. Similar to MIPcp, existence of exponentially many cycles is a drawback of the formulation and Baharev et al. (2015) use the cutting plane algorithm.

Next, we propose an MIP model based on Property 1. Although MIPin uses significantly less constraints than MIPcp, MIPin still has $O(m^3)$ constraints which grows rapidly in m. On the other hand, the MIP model we propose has $O(m^2)$ variables and $O(m^2)$ constraints. In addition to Z, let us define decision variable matrix $O \in \{0,1\}^{m \times m}$.

$$
O_{kq} = \begin{cases} 1 & \text{if } \pi_k = q, \\ 0 & \text{otherwise,} \end{cases} \qquad k \in J, q \in J
$$

Then, we have the following MIP model.

$$
\begin{aligned}
\textbf{MIPto} \quad \min \quad & F(Y) && && (5a)\\
s.t. \quad & Z = supp(Y), && && (5b)\\
& Z_{jk} - mZ_{kj} \leq \sum_{r \in J} r(O_{kr} - O_{jr}), && j \in J^k, k \in J, && (5c)\\
& Z_{jk} + Z_{kj} \leq 1, && j \in J^k, k \in J, && (5d)\\
& \sum_{q \in J^k} O_{kq} = 1, && k \in J, && (5e)\\
& \sum_{k \in J^q} O_{kq} = 1, && q \in J, && (5f)\\
& Z, O \in \{0,1\}^{m \times m}, Y \text{ unrestricted} && && (5g)
\end{aligned}
$$

The key constraint in (5) is (5c). Recall that Z_{jk} indicates which node comes first in the topological order and O_{kr} stores the exact location in the order. With these definitions, (5c) forces correct values of Z_{jk} and Z_{kj} by comparing the order difference. Recall that we can reduce the number of binary variables Z_{jk}'s by plugging $Z_{jk} + Z_{kj} = 1$, but we keep the full matrix notation for ease of explanation. We next show that (5) correctly solves (1).

Proposition 1 *An optimal solution to* (5) *is an optimal solution to* (1).

Proof By Property 1, any DAG has a corresponding topological order. Let π^* be the topological order defined by an optimal solution Y^* for (1). Note that (5e) and (5f) define a topological order. Hence, it suffices to show that (5) gives a DAG given π^*. Note that the right hand side of (5c) measures the difference in the topological order between nodes j and k. If the value is positive, it implies $\pi_k > \pi_j$. Consider (5c) for j_1 and j_2 with $\pi^*_{j_2} > \pi^*_{j_1}$. When $j = j_1$ and $k = j_2$, we have $\sum_{r \in J} r(O_{j_2 r} - O_{j_1 r}) > 0$ in (5c) and at most one of $z_{j_1 j_2}$ and $z_{j_2 j_1}$ can be 1 by (5d). When $j = j_2$ and $k = j_1$, we have $\sum_{r \in J} r(O_{j_1 r} - O_{j_2 r}) < 0$ in (5c) and we must have $z_{j_1 j_2} = 1$ by the left hand side of (5c). Therefore, we have correct value $z_{j_1 j_2} = 1$ when $\pi^*_{j_2} > \pi^*_{j_1}$. This completes the proof. ■

In Table 1, we compare the MIP models introduced in this section. Although all three MIP models have $O(m^2)$ binary variables, MIPto has more binary variables than MIPcp and MIPin due to O_{kq}'s. MIP models MIPin and MIPto have polynomially many constraints, whereas MIPcp has exponentially many constraints. MIPto has the smallest number of constraints among the three MIP models. In the computational experiment, we use a variation of the cutting plane algorithm for MIPcp as it has exponentially many constraints. For MIPin and MIPto, we do not use a cutting plane algorithm.

3. Algorithms based on Topological Order

Although the MIP models introduced in Section 2 guarantee optimality, the execution time for solving an integer programming problem can be exponential in problem size. Further, the execution time could increase drastically in m, as all of the models require at least

Name	Reference	# binary variables	# constraints
MIPcp	(2)	$O(m^2)$	exponential
MIPin	(4)	$O(m^2)$	$O(m^3)$
MIPto	(5)	$O(m^2)$	$O(m^2)$

Table 1: Number of binary variables and constraints of MIP models

$O(m^2)$ binary variables and $O(m^2)$ constraints. In order to deal with larger graphs, we propose iterative algorithms for (1) based on Property 1. Observe that, if we are given a topological order of the nodes, then Z and O are automatically determined in (5). In other words, we can easily obtain a subset of the arcs such that all of the arcs can be used without creating a cycle. Let $\bar{R}$ be the determined adjacency matrix given topological order $\bar{\pi}$. In detail, we set

$$\bar{R}_{jk} = 1 \text{ if } \bar{\pi}_j > \bar{\pi}_k,\ \bar{R}_{jk} = 0 \text{ otherwise.}$$

Let $adj(\bar{\pi})$ be the function generating $\bar{R}$ given input topological order $\bar{\pi}$. If we are given $\bar{\pi}$, then we can generate $\bar{R}$ by $adj(\bar{\pi})$, and solving (1) can be written as

$$\min_{Y} \ F(Y) \quad s.t. \quad \bar{R} \geq supp(Y). \tag{6}$$

Note that (6) has acyclic constraint $\bar{R} \geq supp(Y)$, not $\bar{R} = supp(Y)$. The inequality is needed when we try to obtain a sparse solution, i.e., only a subset of the arcs is selected among all possible arcs implied by $\bar{R}$. As long as we satisfy the inequality, Y forms an acyclic subgraph. Hence, $\bar{R}$ can be different from adjacency matrix $supp(Y)$ in an optimal solution of (6), and any arc (j,k) such that $\bar{R}_{jk} = 1$ can be selected without creating a cycle. For this reason, we call $\bar{R}$ an *adjacency candidate matrix*. The algorithms proposed later in this section solve (6) by providing different $\bar{\pi}$ and $\bar{R} = adj(\bar{\pi})$ in each iteration. In fact, (6) is separable into m sub problems if F is separable. Let Y_k and Z_k be the kth columns of Y and R, respectively, for node k. Then, solving

$$\min_{Y_k} \ F_k(Y_k) \quad s.t. \quad \bar{R}_k \geq supp(Y_k), \tag{7}$$

for all $k \in J$ gives the same solution as solving (6) if F is separable as $F(Y) = \sum_{k \in J} F_k(Y_k)$.

In Section 3.1, a local improvement algorithm for a given topological order is presented. The algorithm swaps pairs of nodes in the order. In both of the iterative algorithms proposed in Sections 3.2 and 3.3, we use the local improvement algorithm presented in the following section.

3.1 Topological Order Swapping Algorithm

Algorithm 1 tries to improve the solution by swapping the topological order. In each iteration, the algorithm determines the nodes to swap that have order s_1 and s_2 in Line 3, where $s_2 = s_1 + 1$ implies that we select two nodes which are neighbors in the current topological order. Then in Line 4, the actual node indices k_1 and k_2 such that $\pi_{k_1} = s_1$ and $\pi_{k_2} = s_2$ are detected. The condition in Line 5 is to avoid meaningless computation when Y^* is sparse. If $|Y^*_{k_2 k_1}| > 0$, we know for sure that $Y^*_{k_2 k_1} = 0$ after swapping the orders of

k_1 and k_2 and thus we will get a different solution. However, if $Y^*_{k_2 k_1} = 0$, we will still have $Y^*_{k_2 k_1} = 0$ after the swap forced by the new order. In Line 6, we create a new topological order $\bar{\pi}$ by swapping nodes k_1 and k_2 in π^*. After obtaining adjacency candidate matrix $\bar{R}$ in Line 7, we solve (6) with $\bar{R}$. It is worth noting that, if F is separable, we only need to solve (7) with $k = k_1$ and k_2 because the values of $\bar{R}$ are the same with R^* except for k_1 and k_2 as the order difference was 1 in π^*. In Line 9, we update the best solution if the new solution is better. The iterations continue until there is no improvement in the past m iterations, which implies that we would swap the same nodes if we proceed after this iteration. Algorithm 1 is illustrated by the following toy example.

Algorithm 1 TOSA (Topological Order Swapping Algorithm)

Require: Y', R', π'
Ensure: Best solution Y^*, R^*, π^*
1: $(Y^*, R^*, \pi^*) \leftarrow (Y', R', \pi')$, $t \leftarrow 0$
2: **While** there is an improvement in the past m iterations
3: $t \leftarrow t+1$, $s_1 \leftarrow (t \bmod (m-1)) + 1$, $s_2 \leftarrow s_1 + 1$
4: $(k_1, k_2) \leftarrow$ node indices satisfying $\pi_{k_1} = s_1$ and $\pi_{k_2} = s_2$
5: **If** $|Y^*_{k_2 k_1}| > 0$ **then**
6: $\bar{\pi} \leftarrow \pi^*$, $\bar{\pi}_{k_1} = s_2$, $\bar{\pi}_{k_2} = s_1$
7: $\bar{R} \leftarrow adj(\bar{\pi})$
8: $\bar{Y} \leftarrow$ solve (6) with $\bar{R}$
9: **If** $F(\bar{Y}) < F(Y^*)$ **then** update (Y^*, R^*, π^*)
10: **End if**
11: **End While**

Example 1 *Consider a graph with $m = 4$ nodes. Let us assume that inputs are $\pi' = (2,3,1,4)$ with corresponding order $3-1-2-4$,*

$$Y' = \begin{bmatrix} 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 \\ 0.4 & 0.8 & 0.1 & 0 \end{bmatrix}, \text{ and } R' = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}.$$

In iteration 1, $t = 1$, $s_1 = 1$, $s_2 = 2$, $k_1 = 3$, and $k_2 = 1$. Hence, we are swapping nodes 3 and 1. Since $|Y^*_{13}| = 0.5 > 0$, $\bar{\pi} = (1,3,2,4)$ is created in Line 6, where the associated order is $1-3-2-4$. If $\bar{\pi}$ gives an improved objective function value, then π^* is updated in Line 9. Let us assume that π^* is not updated. In iteration 2, $t = 2$, $s_1 = 2$, $s_2 = 3$, $k_1 = 1$, and $k_2 = 2$. Since $|Y^*_{21}| = 0$, Lines 6 - 9 are not executed. □

3.2 Iterative Reorering Algorithm

We propose an iterative reordering algorithm based on Property 1, which solves (6) in each iteration aiming to optimize (1). In each iteration of the algorithm, all nodes are sorted based on scores defined by (i) merit scores of the arcs, (ii) historical choice of the arcs (used as weights), (iii) and some random components. Then the sorted node order is directly

used as a topological order. The selected arcs by the topological order give updates on arc weights. Let us first define notation.

ν = uniform random variable on $[\nu_{lb}, \nu_{ub}]$, $\nu_{lb} < 1 < \nu_{ub}$
ρ_{jk} = pre-determined merit score of arc (j,k) for $j \in J, k \in J$
w_{jk} = weight of arc (j,k) for $j \in J, k \in J$
c_k = score of node k, $k \in J$

The range $[\nu_{lb}, \nu_{ub}]$ of the uniform random variable ν balances the randomness and structured scores. Note that ρ_{jk} should be determined based on the data and the characteristic of the problem considered, where larger ρ_{jk} implies that arc (j,k) is attractive. Based on the arc merit scores ρ, the score for node k, $k \in J$, is defined as

$$c_k = \nu \cdot \big(\sum_{j \in J^k} w_{jk}\rho_{jk} \big), \quad k \in J, \tag{8}$$

which can be interpreted as a weighted summation of ρ_{jk}'s multiplied by perturbation random number ν. Hence, nodes with high scores are attractive. Initially, all arcs have equal weights and the weights are updated in each iteration based on the topological order in the iteration. If $\bar{R} = adj(\bar{\pi})$ is the adjacency candidate matrix in the iteration, then, the weights are updated by

$$w_{jk} = w_{jk} + 1, \quad \text{if } \bar{R}_{jk} = 1. \tag{9}$$

The overall algorithmic framework is summarized in Algorithm 2. In Line 1, weights w_{jk}'s are initialized to 1 and $\bar{t}$, which counts the number of iterations without a best solution update, is initialized. Also, a random order π^* of the nodes is generated, and the corresponding solution becomes the best solution. In each iteration, first node scores c_k's are calculated (Line 3), then topological order $\bar{\pi}$ is obtained by sorting the nodes, and finally adjacency candidate matrix $\bar{R}$ is generated (Line 4). Then, in Lines 5 and 6, solution $\bar{Y}$ is obtained by solving (6) with $\bar{R}$ and the best solution is updated if available. In Lines 7 - 10, *TOSA* is executed if the current solution is within a certain percentage α from the best solution. Lines 11 and 12 update $\bar{t}$, and Line 13 updates w_{jk}'s. This ends the iteration and the algorithm continues until $\bar{\pi}$ is converged or there is no update of the best solution in the last t^* iterations. Algorithm 2 is illustrated by the following toy example.

Example 2 *Consider a graph with $m = 3$ nodes. In the current iteration, let us assume that we are given*

$$\rho = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0.2 & 0 & 0.2 \\ 0.3 & 0.3 & 0 \end{bmatrix} \text{ and } w = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}.$$

Note that we have $\sum_{j \in J^1} w_{j1}\rho_{j1} = 0.2 \cdot 1 + 0.3 \cdot 2 = 0.8$, $\sum_{j \in J^2} w_{j2}\rho_{j2} = 0.5 \cdot 1 + 0.3 \cdot 1 = 0.8$, and $\sum_{j \in J^3} w_{j3}\rho_{j3} = 0.4 \cdot 2 + 0.2 \cdot 1 = 1$. If random numbers ($\nu$) are 0.9, 1.1, 0.8 for nodes 1,2, and 3, respectively, then by (8)*, $c_1 = 0.9 \cdot 0.8 = 0.72$, $c_2 = 1.1 \cdot 0.8 = 0.88$, and $c_3 = 0.8 \cdot 1 = 0.8$. Then in Line 4, we obtain $\bar{\pi} = (3,1,2)$, with corresponding order $2-3-1$, and $\bar{R} = [0,1,1;0,0,0;0,1,0]$. After obtaining $\bar{Y}$ and updating the best solution in Lines 5-12, the weights are updated by* (9) *as follows.*

Algorithm 2 IR (Iterative Reordering)

Require: Merit score $\rho \in \mathbb{R}^{m \times m}$, termination parameter t^*, *TOSA* execution parameter α
Ensure: Best solution Y^*, R^*, π^*
1: $w_{jk} \leftarrow 1$, $\pi^* \leftarrow$ a random order, $R^* \leftarrow adj(\pi^*)$, $\bar{\pi} \leftarrow \pi^*$, $Y^* \leftarrow$ solve (6) with R^*, $\bar{t} \leftarrow 0$
2: **While** (i) $\bar{\pi}$ is not convergent or (ii) $\bar{t} < t^*$
3: Calculate score c_k by (8)
4: $\bar{\pi} \leftarrow$ sort nodes with respect to c_k, $\bar{R} \leftarrow adj(\bar{\pi})$
5: $\bar{Y} \leftarrow$ solve (6) with $\bar{R}$
6: **If** $F(\bar{Y}) < F(Y^*)$ **then** update (Y^*, R^*, π^*)
7: **If** $F(\bar{Y}) < F(Y^*) \cdot (1+\alpha)$
8: $(Y', R', \pi') \leftarrow TOSA(\bar{Y}, \bar{R}, \bar{\pi})$,
9: **If** $F(Y') < F(Y^*)$ **then** update (Y^*, R^*, π^*)
10: **End If**
11: **If** (Y^*, R^*, π^*) is updated **then** $\bar{t} \leftarrow 0$
12: **Else** $\bar{t} \leftarrow \bar{t} + 1$
13: Update weights by (9)
14: **End While**

$$w_{new} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix} + \bar{R} = \begin{bmatrix} 0 & 2 & 3 \\ 1 & 0 & 1 \\ 2 & 2 & 0 \end{bmatrix}$$

This ends the current iteration. □

3.3 Gradient Descent Algorithm

In this section, we propose a gradient descent algorithm based on Property 1. The algorithm iteratively executes: (i) moving toward an improving direction by gradients, (ii) DAG structure is recovered and topological order is obtained by a projection step. The algorithm is based on the standard gradient descent framework while the projection step takes care of the acyclicity constraints by generating a topological order from the current (possibly cyclic) solution matrix. In order to distinguish the solutions with and without the acyclicity property, we use the following notation.

$U^t \in \mathbb{R}^{m \times m}$ = decision variable matrix without acyclicity requirement in iteration t
$Y^t \in \mathbb{R}^{m \times m}$ = decision variable matrix satisfying $G(supp(Y^t)) \in \mathcal{A}$ in iteration t

Let γ^t be the step size in iteration t, $\nabla F(Y^t)$ be the derivative of F at Y^t, and $G^t \in \mathbb{R}^{m \times m}$ be a weight matrix that weighs each element. We assume $\|\nabla F(Y^t)\|_\infty \leq M_1$ for a constant M_1, where $\|\cdot\|_\infty$ is the uniform (infinity) norm. The update formula

$$U^t = Y^t - \gamma^t \big[\nabla F(Y^t) \circ G^t \big], \quad t \geq 0, \tag{10}$$

updates Y^t based on the weighted gradient, where $\circ$ represents the entrywise or Hadamard product of the two matrices. Given topological order π^t, we define G^t as

$$G^t_{jk} = \left(1 + \frac{1}{\pi^t_k}\right)^{\pi^t_k}, \quad j \in J^k \tag{11}$$

to balance gradients of the nodes with different orders (small and large values of π^t_k). For nodes k_1 and k_2 with $\pi^t_{k_1} = 1$ and $\pi^t_{k_2} = m$, most of the gradients for node k_1 are zero and most of the gradients for node k_2 are nonzero. Weight (11) tries to adjust this gap. Note that we have $2 \leq G^t_{jk} \leq e$ for any large m. Since U^t may not satisfy acyclic constraints, in order to obtain a DAG, the algorithm needs to solve the projection problem

$$Y^* = \text{argmin}_Y \ \|Y - U^t\|_2^2 \quad s.t. \quad G(supp(Y)) \in \mathcal{A}, \tag{12}$$

where $\|\cdot\|_2$ is the L_2 norm.

Proposition 2 *If U^t is arbitrary, then optimization problem (12) is $\mathcal{NP}$-hard.*

Proof Recall that feedback arc set is $\mathcal{NP}$-complete (Karp, 1972) and maximum acyclic subgraph is the dual of the feedback arc set problem. With $U^t = 1$, (12) becomes the weighted maximum acyclic subgraph problem. Therefore, (12) is $\mathcal{NP}$-complete. ■

Because solving (12) to optimality does not guarantee an optimal solution for (1), we use a greedy strategy to solve (12). The greedy algorithm, presented in Algorithm 3, sequentially determines and fixes the topological order of a node where in each iteration the problem is solved optimally given the currently fixed nodes and corresponding orders. The detailed derivations of the algorithm and the proof that each iteration is optimal, given already fixed node orders, are available in Appendix 6. In other words, we show that Line 3 is 'locally' optimal, i.e., it selects the best next node given that the order $q+1, q+2, \cdots, m$ is fixed. In each iteration, in Line 3, the algorithm first calculates score $\sum_{j\in\bar{J}}(\bar{U}^t_{jk})^2$ for each node k in $\bar{J}$ and picks node k^* with the minimum value. Then, in Line 4, the order of the selected node is fixed to q. The fixed node is then excluded from the active set $\bar{J}$ and iterate q is decreased by 1 in Line 5. At the end of the algorithm, we can determine $\bar{Y}$ based on the order $\bar{\pi}$ determined and (18) in Appendix 6. We illustrate Algorithm 3 by the following example.

Example 3 *Consider a graph with $m = 3$ nodes. Given U^t, the algorithm returns $\bar{Y}$ presented in the following.*

$$U^t = \begin{bmatrix} 0 & 1 & 2 \\ 4 & 0 & 2 \\ 5 & 2 & 0 \end{bmatrix} \quad \bar{Y} = \begin{bmatrix} 0 & 0 & 0 \\ 4 & 0 & 2 \\ 5 & 0 & 0 \end{bmatrix}$$

Algorithm 3 starts with $q = 3$ and $\bar{J} = \{1, 2, 3\}$. In iteration 1, node 2 is selected to have $\pi_2 = 3$ based on argmin$\{4^2 + 5^2, 1^2 + 2^2, 2^2 + 2^2\}$. Then, set $\bar{J}$ and integer q are updated to $\bar{J} = \{1, 3\}$ and $q = 2$. In iteration 2, node 3 is selected to have $\pi_3 = 2$ based on argmin$\{5^2, 2^2\}$. Then, set $\bar{J}$ and integer q are updated to $\bar{J} = \{1\}$ and $q = 1$. In iteration 3, node 1 is selected. Hence, we have node order 1-3-2 and we obtain $\bar{Y}$ presented above with objective function value $\|\bar{Y} - U^t\|_2^2 = 1^2 + 2^2 + 2^2 = 9$. □

Algorithm 3 Greedy

Require: $U^t \in \mathbb{R}^{m\times m}$
Ensure: $\bar{Y}$ feasible to (12), topological order $\bar{\pi}$
1: $q \leftarrow m$, $\bar{J} \leftarrow J$
2: **While** $\bar{J} \neq \emptyset$
3: $\quad k^* = \text{argmin}_{k\in\bar{J}}\left\{\sum_{j\in\bar{J}}(U^t_{jk})^2\right\}$
4: $\quad \bar{\pi}_{k^*} = q$
5: $\quad \bar{J} \leftarrow \bar{J} \setminus \{k^*\}, q \leftarrow q - 1$
6: **End While**
7: Determine $\bar{Y}$ by (18) in Appendix 6

The overall gradient descent algorithm for (1) is presented in Algorithm 4. In Line 1, the algorithm generates a random order π^* and obtain corresponding R^* and Y^* and save them as the best solution. In each iteration of the loop, Lines 3-6 follow the standard gradient descent algorithm. The weighted gradient H^t is calculated in Line 3, and the step size is determined in Line 4 based on the ratio between $\max_{j\in J^k, k\in J}|H^t_{jk}|$ and $\max_{j\in J^k,k\in J}|Y^t_{jk}|$. In Line 5, the solution is updated based on the weighted gradient and, in Line 6, the greedy algorithm is used to obtain the projected solution and the topological order. Observe that we do not directly use the projected solution. This is because the projected solution is not necessarily optimal given π^{t+1}. Hence, in Line 7, a new solution Y^{t+1} is obtained based on π^{t+1}. In Lines 9 - 12, $TOSA$ is executed if the current solution is within a certain percentage from the current best solution. Lines 13 and 14 update $\bar{t}$ and Line 15 copies Y^* to Y^{t+1} if $\bar{t} \geq t^*_2$ in order to focus on the solution space near Y^*. The algorithm continues until Y^t is convergent or $\bar{t} \geq t^*_1$.

In gradient based algorithms, it is common to have γ^t depend only on t, but in our case dependency on H^t and Y^t is justifiable since we multiply the gradient by G^t. We next show the convergence of Y^t in Algorithm 4 when $t^*_1 = t^*_2 = \infty$. This makes the algorithm not to terminate unless Y^t has converged and modification of Y^t in Line 15 is not executed. Further, we assume the following for the analysis.

Assumption 1 *For any non-zero element Y^t_{jk}, $j, k \in J$, of Y^t in any iteration t, we assume $\varepsilon < |Y^t_{jk}| < M_2$, where ε is a small positive number and M_2 is a large enough number.*

Note that Assumption 1 is a mild assumption, as ignoring near-zero values of Y^t happens in practice anyway due to finite precision. For notational convenience, let $L^t = \gamma^t \nabla F(Y^t) \circ G^t = \gamma^t H^t$ be the second term in (10). Then, U^t can be written as $U^t = Y^t - L^t = Y^t - \gamma^t H^t$. In the following lemma, we show that the node orders converge.

Lemma 3 *If t is sufficiently large satisfying $\sqrt{t} > \frac{(M_1 e)^2}{\varepsilon(\sqrt{M_2^2 + \varepsilon^2/m} - M_2)}$, then $\pi^t = \pi^{t+1}$.*

Proof Recall that Y^t is obtained by solving (20) and we know the corresponding node order π^t. Let $k_1, k_2, \cdots, k_m$ be the node indices defined based on π^t. In other words, node k_1 appears first, followed by nodes k_2, k_3, and so on in the topological order π^t. In the proof, we

Algorithm 4 GD (Gradient Descent)

Require: Parameters t_1^* and t_2^*, *TOSA* execution parameter α
Ensure: Best solution Y^*, R^*, π^*
1: $t \leftarrow 1$, $\bar{t} \leftarrow 0$, $\pi^* \leftarrow$ a random order, $R^* \leftarrow adj(\pi^*)$, $Y^* \leftarrow$ solve (6) with R^*
2: **While** (i) Y^t is not convergent or (ii) $\bar{t} < t_1^*$
3: $H^t \leftarrow \nabla F(Y^t) \circ G^t$, G^t defined in (11)
4: $\gamma^t \leftarrow \frac{\|H^t\|_\infty}{\|Y^t\|_\infty}/\sqrt{t}$
5: $U^t \leftarrow Y^t - \gamma^t H^t$
6: $\pi^{t+1} \leftarrow$ *Greedy*(U^t)
7: $Y^{t+1} \leftarrow$ solve (6) with $R^{t+1} = adj(\pi^{t+1})$
8: **If** $F(Y^{t+1}) < F(Y^*)$ **then** $(Y^*, R^*, \pi^*) \leftarrow (Y^{t+1}, R^{t+1}, \pi^{t+1})$
9: **If** $F(\bar{Y}) < F(Y^*) \cdot (1+\alpha)$
10: $(Y', R', \pi') \leftarrow TOSA(Y^{t+1}, R^{t+1}, \pi^t)$,
11: **If** $F(Y') < F(Y^*)$ **then** $(Y^*, R^*, \pi^*) \leftarrow (Y', R', \pi'), (Y^{t+1}, R^{t+1}, \pi^{t+1}) \leftarrow (Y', R', \pi')$
12: **End If**
13: **If** (Y^*, R^*, π^*) is updated **then** $\bar{t} \leftarrow 0$
14: **Else** $\bar{t} \leftarrow \bar{t} + 1$
15: **If** $\bar{t} \geq t_2^*$ **then** $Y^{t+1} \leftarrow Y^*$
16: $t \leftarrow t + 1$
17: **End While**

show that there is no change in the node order when the condition $\sqrt{t} > \frac{(M_1 e)^2}{\varepsilon(\sqrt{M_2^2+\varepsilon^2/m}-M_2)}$ is met, where M_1 and M_2 are the upper bounds for $\|\nabla F(Y^t)\|_\infty$ and $\|Y^t\|_\infty$, respectively, as assumed. We first derive

$$\|L^t\|_\infty = \left\|\gamma^t H^t\right\|_\infty = \left\|\frac{1}{\sqrt{t}}\frac{\|H^t\|_\infty}{\|Y^t\|_\infty}H^t\right\|_\infty \leq \frac{1}{\sqrt{t}}\frac{\left(\|H^t\|_\infty\right)^2}{\|Y^t\|_\infty} < \frac{1}{\sqrt{t}}\frac{(M_1 e)^2}{\varepsilon}, \tag{13}$$

where the last inequality holds since (i) $\|Y^t\|_\infty > \varepsilon$ by Assumption 1, (ii) $\|H^t\|_\infty = \|\nabla F(Y^t) \circ G^t\|_\infty \leq M_1 e$ because $\|\nabla F(Y^t)\|_\infty \leq M_1$ by the assumption and $\|G^t\|_\infty \leq e$, where e is natural number.

Now let us consider $q = \bar{q}$ in Algorithm 3 to decide node order $\bar{q}$ in iteration $t+1$ and assume $\pi_{k_r}^t = \pi_{k_r}^{t+1}$ for $r = m, m-1, \cdots, \bar{q}-1$. Note that we currently have $\bar{J} = \{k_1, k_2, \cdots, k_{\bar{q}}\}$.

1. For $k_{\bar{q}}$, we derive $\sum_{j\in\bar{J}}(U_{jk_{\bar{q}}}^t)^2 = \sum_{j\in\bar{J}}(Y_{jk_{\bar{q}}}^t - L_{jk_{\bar{q}}}^t)^2 = \sum_{j\in\bar{J}}(L_{jk_{\bar{q}}}^t)^2 < m\left[\frac{(M_1 e)^2}{\sqrt{t}\varepsilon}\right]^2$, where the second equality holds since $Y_{jk_{\bar{q}}}^t = 0$ for all $j \in \bar{J}$ since $\pi_{k_{\bar{q}}} = \bar{q}$ and no arc can be used to the nodes in $\bar{J}$, and the last inequality holds due to (13) and $|\bar{J}| \leq m$.

2. For all other nodes $k_r \in \bar{J} \setminus \{k_{\bar{q}}\}$, we derive

$$\begin{aligned}
\textstyle\sum_{j\in\bar{J}}(U_{jk_r}^t)^2 &= \textstyle\sum_{j\in\bar{J}}(Y_{jk_r}^t - L_{jk_r}^t)^2 \\
&= \textstyle\sum_{j\in\bar{J}}(Y_{jk_r}^t)^2 + \sum_{j\in\bar{J}}(L_{jk_r}^t)^2 - 2\sum_{j\in\bar{J}} Y_{jk_r}^t \cdot L_{jk_r}^t \\
&> \textstyle\sum_{j\in\bar{J}}(Y_{jk_r}^t)^2 - 2\sum_{j\in\bar{J}} Y_{jk_r}^t \cdot L_{jk_r}^t \\
&> \varepsilon^2 - 2\textstyle\sum_{j\in\bar{J}} Y_{jk_r}^t \cdot L_{jk_r}^t \\
&\geq \varepsilon^2 - 2\textstyle\sum_{j\in\bar{J}} |Y_{jk_m}^t \cdot L_{jk_r}^t| \\
&> \varepsilon^2 - 2M_2\frac{m(M_1 e)^2}{\sqrt{t}\varepsilon}
\end{aligned}$$

where the fourth line holds due to $|Y_{jk_r}^t| > \varepsilon$ by Assumption 1, and the sixth line holds due to $|Y_{jk}^t| \leq M_2$, $|\bar{J}| \leq m$, and $|L_{jk_r}^t| < \frac{(M_1 e)^2}{\sqrt{t}\varepsilon}$ by (13).

Combining the two results for $k_{\bar{q}}$ and $k_r \in \bar{J} \setminus \{k_{\bar{q}}\}$, we obtain $\sum_{j\in\bar{J}}(U_{jk_{\bar{q}}}^t)^2 < m\left[\frac{(M_1 e)^2}{\sqrt{t}\varepsilon}\right]^2 < \varepsilon^2 - 2M_2\frac{m(M_1 e)^2}{\sqrt{t}\varepsilon} < \sum_{j\in\bar{J}}(U_{jk_r}^t)^2$, for any $r \in \{1, 2, \cdots, \bar{q}-1\}$, where the second inequality holds due to the condition $\sqrt{t} > \frac{(M_1 e)^2}{\varepsilon(\sqrt{M_2^2+\varepsilon^2/m}-M_2)}$. The result implies that we must have $\pi_{k_{\bar{q}}}^{t+1} = \pi_{k_{\bar{q}}}^t = \bar{q}$ by Line 3 in Algorithm 3.

Note that, when $\bar{q} = m$, we have $J = \bar{J}$ and the assumption of $\pi_{k_r}^t = \pi_{k_r}^{t+1}$ for $r = m, m-1, \cdots, \bar{q}-1$ automatically holds. By iteratively applying the above derivation technique from $q = m$ to 1, we can show that $\pi^t = \pi^{t+1}$. ■

When (6) is solved with the identical node orders, the resulting solutions are equivalent. Hence, the following proposition holds.

Proposition 4 *In Algorithm 4, Y^t converges in t.*

4. Estimation of Gaussian Bayesian Networks

In this section, we introduce the Gaussian Bayesian network learning problem, which follows the form of (1). The goal is to learn or estimate unknown structure between the nodes of a graph, where the error is normally distributed. The network can be estimated by optimizing a score function, testing conditional independence, or a mix of the two, as described in Section 1. Among the three categories, we select the score based approach with the L_1-penalized least square function recently studied in Han et al. (2016).

Let $X \in \mathbb{R}^{n\times m}$ be a data set with n observations and m features. Let $I = \{1, \cdots, n\}$ and $J = \{1, \cdots, m\}$ be the index set of observations and features, respectively. For each $k \in J$, we build a regression model in order to explain feature k using a subset of variables in J^k. In other words, we set feature k as the response variable and a sparse subset of J^k as explanatory variables of the regression model for variable k. In order to obtain a subset of J^k, the LASSO penalty function is added. Considering regression models for all $k \in J$ together, the problem can be represented on a graph. Each feature is a node in the graph, and the directed arc from node j to node k represents explanatory and response variable relationship between node j and k. The goal is to minimize the sum of penalized SSE over all regression models for $k \in J$, while the selected arcs do not create a cycle.

Let β_{jk}, $j \in J^k$, $k \in J$, be the coefficient of attribute j for dependent variable k. Then the problem can be written as

$$\min_{\beta} \frac{1}{n} \sum_{k\in J} \sum_{i\in I} (x_{ik} - \sum_{j\in J^k} \beta_{jk} x_{ij})^2 + \lambda \sum_{k\in J} \sum_{j\in J^k} |\beta_{jk}| \quad s.t. \quad G(supp(\beta)) \in \mathcal{A}, \tag{14}$$

which follows the form of (1). In Han et al. (2016), individual weights are used for the penalty term, i.e., $\lambda \sum_{k\in J} \sum_{j\in J^k} w_{jk}|\beta_{jk}|$, however, in the computational experiment, we set all weights equal to 1 for simplicity.

Let $Z_{jk} = 1$ if attribute j is used for dependent variable k and $Z_{jk} = 0$ otherwise. Then, we can formulate MIPto for (14) as

$$\min \quad \frac{1}{n} \sum_{k\in J} \sum_{i\in I} (x_{ik} - \sum_{j\in J^k} \beta_{jk} x_{ij})^2 + \lambda \sum_{k\in J} \sum_{j\in J^k} |\beta_{jk}| \tag{15a}$$

$$s.t. \quad |\beta_{jk}| \le M Z_{jk}, \qquad j \in J^k, k \in J, \tag{15b}$$

$$Z_{jk} - m Z_{kj} \le \sum_{r\in J} r(O_{kr} - O_{jr}), \qquad j \in J^k, k \in J, \tag{15c}$$

$$Z_{jk} + Z_{kj} \le 1, \qquad j \in J^k, k \in J, \tag{15d}$$

$$\sum_{q\in J} O_{kq} = 1, \qquad k \in J, \tag{15e}$$

$$\sum_{k\in J} O_{kq} = 1, \qquad q \in J, \tag{15f}$$

$$Z, O \in \{0,1\}^{m\times m}, \tag{15g}$$

$$\beta_{jk} \text{ not restricted}, \qquad j \in J^k \cup \{0\}, k \in J, \tag{15h}$$

where M is a large constant. Note that (15b) is the linear constraint corresponding to $Z = supp(Y)$ in (5). Similarly, (15a), (15b), (4c) - (4e) and (15h) can be used to formulate MIPin for (14). For MIPcp, (15a), (15b), (15h), and the constraints in (2) can be used for (14).

Note that M in (15b) plays an important role in computational efficiency and optimality. If M is too small, the MIP model cannot guarantee optimality. If M is too large, the solution time can be as large as enumeration. The algorithm for getting a valid value for M in Park and Klabjan (2013) can be used. However, the valid value of big M for multiple linear regression is often too large (Park and Klabjan, 2013). For (15), we observed that a simple heuristic presented in Section 5 works well.

In each iteration of IR (Algorithm 2) and GD (Algorithm 4), we are given topological order $\bar{\pi}$ and matrix $\bar{R} = adj(\bar{\pi})$. Let $S^k = \{j \in J^k | \bar{R}_{jk} = 1\}$ be the set of selected candidate arcs for dependent variable k. Given fixed $\bar{R}$, (14) is separable into m LASSO linear regression problems

$$\min \quad \frac{1}{n} \sum_{i\in I} (x_{ik} - \sum_{j\in S^k} \beta_{jk} x_{ij})^2 + \lambda \sum_{j\in S^k} |\beta_{jk}|, \quad k \in J. \tag{16}$$

5. Computational Experiment

For all computational experiments, a server with two Xeon 2.70GHz CPUs and 24GB RAM is used. Although there are many papers studying Bayesian network learning with various error measures and penalties, here we focus on minimizing the LASSO type objective (SSE and penalty) and we picked one of the latest paper of Han et al. (2016) with the same objective function as a benchmark.

The MIP models MIPcp, MIPin, and MIPto are implemented with CPLEX 12.6 in C#. For MIPcp, instead of implementing the original cutting plane algorithm, we use CPLEX Lazy Callback, which is similar to the cutting plane algorithm. Instead of solving (3) to optimally from scratch in each iteration, we solve (2) with Lazy Callback, which allows updating (adding) constraints (cycle prevention constraints) in the process of the branch and bound algorithm whenever an integer solution with cycles is found. Given a solution with the cycles, we detect all cycles and add cycle prevention constraints for the detected cycles.

For MIPcp, MIPin, and MIPto, we set big M as follows. Given λ, we solve (14) without acyclic constraints. Hence, we are allowed to use all arcs in J^k for each model $k \in J$. Then, we obtain the estimated upper bound for big M by

$$M = 2\Big(\max_{j\in J^k, k\in J} |\beta_{jk}|\Big). \tag{17}$$

We observed that the above formula gives large enough big M for all cases in the following experiment. In Appendix 6, we present comparison of regression coefficients of implanted network (DAG) with big M values by (17). The result shows that the big M value in (17) is always valid for all cases considered.

We compare our algorithms and models with the algorithm in Han et al. (2016), which we denote as DIST here. Their algorithm starts with neighborhood selection (NS), which filters unattractive arcs and removes them from consideration. The procedure is specifically developed for high dimensional variable selection when m is much larger than n. In our experiment, many instances considered are not high dimensional and some have dense solutions. Further, by filtering arcs, there exists a probability that an arc in the optimal solution can be removed. Hence, we deactivated the neighborhood selection step of their original algorithm, where the R script of the original algorithm is available on the journal website.

For GD and IR, the algorithms are written in R (R Core Team, 2016). We use glmnet package (Friedman et al., 2010) function glmnet for solving LASSO linear regression problems in (16). For IR, we use parameters $\alpha = 0.01$, $t^* = 10$, $\nu_{lb} = 0.8$, and $\nu_{ub} = 1.2$. For GD, we use parameters $\alpha = 0.01$, $t_1^* = 10$, and $t_2^* = 5$. Because both GD and IR start with a random solution, they perform different with different random solutions. Further, since we observe that the execution time of GD and IR are much faster than DIST, we decided to run GD and IR with 10 different random seeds and report the best solution. To emphasize the number of different random seeds for GD and IR, we use the notation GD10 and IR10 in the rest of the section.

We first test all algorithms with synthetic instances generated using R package pcalg (Kalisch et al., 2012). Function randomDAG is used to generate a DAG and function rmvDAG is used to generate multivariate data with the standard normal error distribution. First,

a DAG is generated by randomDAG function. Next, the generated DAG and random coefficients are used to create each column (with standard normal error added) by rmvDAG function which uses linear regression as the underlying model. After obtaining the data matrix from the package, we standardize each column to have zero mean with standard deviation equal to one. The DAG used to generate the multivariate data is considered as the true structure or true arc set while it may not be the optimal solution for the score function. The random instances are generated for various parameters described in the following.

m: number of features (nodes)
n: number of observations
s: expected number of true arcs per node
d: expected density of the adjacency matrix of the true arcs

By changing the ranges of the above parameters, three classes of random instances are generated.

Sparse data sets: The expected total number of true arcs is controlled by s and most of the instances have a sparse true arc set. We use $n \in \{100, 200, 300\}$, $m \in \{20, 30, 40, 50\}$, and $s \in \{1, 2, 3\}$ to generate 10 instances for each (n, m, s) triplet. This yields a total of 360 random instances.

Dense data sets: The expected total number of true arcs is controlled by d and most of the instances have a dense true arc set compared with the sparse data sets. We use $n \in \{100, 200, 300\}$, $m \in \{20, 30, 40, 50\}$, and $d \in \{0.1, 0.2, 0.3\}$ to generate 10 instances for each (n, m, d) triplet, and thus we have a total of 360 random instances.

High dimensional data sets: The instances are high dimensional ($m \geq n$) and very sparse. The expected total number of true arcs is controlled by s. We use $n = 100$ and $m \in \{100, 150, 200\}$, and $s \in \{0.5, 1, 1.5\}$ to generate 10 instances for each (m, s) pair, which yields a total of 90 random instances.

We use four λ values differently defined for each data set in order to cover the expected number of arcs with the four λ values. For each sparse instance, we solve (14) with $\lambda \in \{1, 0.5, 0.1, 0.05\}$. For dense data sets, a wide range of λ values are needed to obtain selected arc sets that have similar cardinalities with the true arc sets. Hence, for each dense instance, instead of fixed values over all instances in the set, we use λ values based on expected density d: $\lambda = \lambda_0 \cdot 10^{-(10 \cdot d - 1)}$, where $\lambda_0 \in \{1, 0.1, 0.01, 0.001\}$. For each high dimensional instance, we use $\lambda \in \{1, 0.8, 0.6, 0.4\}$. Observe that the expected densities of the adjacency matrices vary across the three data sets. The sparse instances have expected densities between 0.02 and 0.15, the dense instances have expected densities between 0.1 and 0.3, and the high dimensional instances have expected densities between 0.002 and 0.015. Hence, different ranges of λ values are necessary.

For all of the results presented in this section, we present the average performance by n, m, s, d and λ. For example, the result for $n = 100$ and $m = 20$ are the averages of 120 and 90 instances, respectively. In all of the comparisons, we use the following metrics.

time: computation time in seconds
δ_{sol}: relative gap (%) from the best objective value among the compared algorithms or models. For example, if we compare the three MIP models, δ_{sol} of an MIP model is the relative gap from the best of the three objective function values obtained by the MIP models.
$\|z\|_0$: number of arcs selected (number of nonzero regression coefficients β_{jk}'s)

In comparing the performance metrics, we use plot matrices. In each Figure 1, 2, 3, 5, and 6, multiple bar plots form a matrix. The rows of the plot matrix correspond to performance metrics and the columns stand for parameters used for result aggregation. For example, the left top plot in Figure 1 shows execution times of the algorithms where the results are aggregated by n (the number of observations), because the first row and first column of the plot matrix in Figure 1 are for execution times and n, respectively.

In Section 5.1, we compare the performance of iterative algorithms GD10 and IR10 and the benchmark algorithm DIST. In Section 5.2, we compare the performance of MIP models MIPcp, MIPin, and MIPto. We also compare all models and algorithms with a subset of the synthetic instances in Section 5.2. Finally, in Section 5.4, we solve a popular real instance of Sachs et al. (2005) in the literature.

5.1 Comparison of Iterative Algorithms

In this section, we compare the performance of GD10, IR10, and DIST by *time*, δ_{sol}, and $\|z\|_0$ for each of the three data sets.

In Figure 1, the result for the sparse data sets is presented. The bar plot matrix presents the performance measures aggregated by n, m, s, and λ.

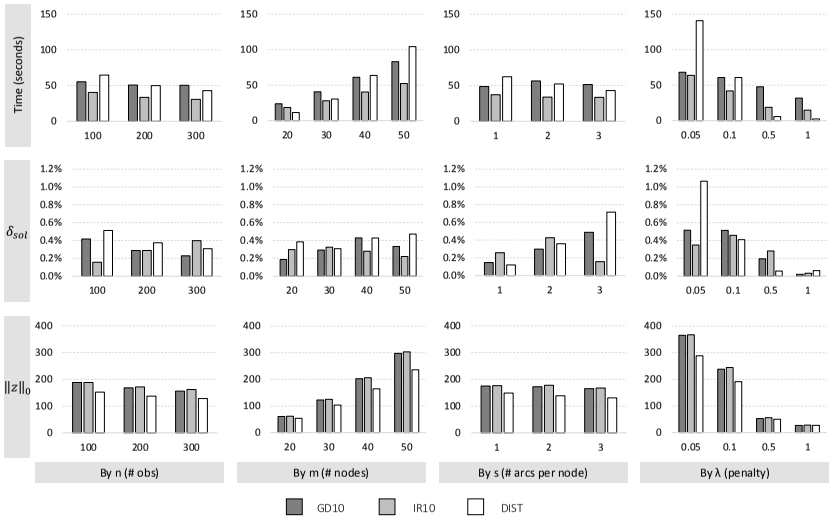


Figure 1: Performance of GD10, IR10, and DIST (sparse data)

The computation time of all three algorithms increases in increasing m and decreasing λ, where the computation time of DIST increases faster than the other two. The computation time of DIST is approximately 10 times faster than the GD10 time when $\lambda = 1$, but 2 times slower when $\lambda = 0.05$. With increasing n, the computation times of all algorithms stay the same or decrease. This can be seen counter-intuitive because larger instances do not increase time. However, a larger number of observations can make predictions more accurate and could reduce search time for unattractive subsets. Especially, the computation time of DIST decreases in increasing n. We think this is because more observations give better local selection in the algorithm when adding and removing arcs. The number of selected arcs ($\|z\|_0$) of GD10 and IR10 is greater than DIST for all cases because the topological order based algorithms are capable of using the maximum number of arcs $\left(\frac{m(m-1)}{2}\right)$, while arc selection based algorithms, such as DIST, are struggling to select many arcs without violating acyclic constraints. In terms of the solution quality, all algorithms have δ_{sol} less than 1.2% and perform good. However, we observe several trends. As λ decreases (required to select more arcs), GD10 and IR10 start to outperform. We also observe that, as the problem requires to select more arcs (increasing m, increasing s, and decreasing λ), GD10 and IR10 perform better. As n increases, δ_{sol} of GD10 and DIST decrease, whereas δ_{sol} of IR10 increases.

The result for the dense data sets is presented in Figure 2. The bar plot matrix presents the performance measures aggregated by n, m, d, and λ_0. Recall that, for the dense data set, we solve (14) with $\lambda = \lambda_0 \cdot 10^{-(10 \cdot d-1)}$ and $\lambda_0 \in \{1, 0.1, 0.01, 0.001\}$. For simplicity of presenting the aggregated result, we use λ_0 in the plot matrix, while λ values are used for actual computation.

The computation time of all three algorithms again increases in increasing m and decreasing λ, where the computation time of DIST increases faster than the other two. Compare to the result for the sparse data sets, the execution times are all larger for the dense data set. The number of selected arcs ($\|z\|_0$) of GD10 and IR10 is again greater than DIST for all cases, where $\|z\|_0$ is twice larger for GD10 and IR10 when m or d is large, or λ is small. In terms of the solution quality, GD10 and IR10 outperform in most of the cases, while δ_{sol} values of DIST increase fast in changing m, d, and λ. The values of δ_{sol} for all algorithms are larger than the sparse data sets result. GD10 and IR10 are better for most of the cases. In general, we again observe that GD10 and IR10 perform better when the problem requires to select more arcs.

The result for the high dimensional data sets is presented in Figure 3. The bar plot matrix presents the performance measures aggregated by m, s, and λ, while n is excluded from the matrix as we fixed n to 100.

The computation time of all three algorithms again increases in increasing m and decreasing λ. However, unlike the previous two sets, the computation times of GD10 and IR10 increase faster than DIST. This is due to the efficiency of the topological order based algorithms. When a very small portion of the arcs should be selected in the solution, topological orders are not informative. For example, consider a graph with three nodes A, B, C and assume that only one arc, (A, B), is selected due to a large penalty. For this case, three topological orders $A-B-C$, $A-C-B$, and $C-A-B$ can represent the selected arc. The third row of Figure 3 shows that $\|z\|_0$ of the three algorithms are very similar, while DIST has much smaller values for the previous two data sets. This implies that arc based search

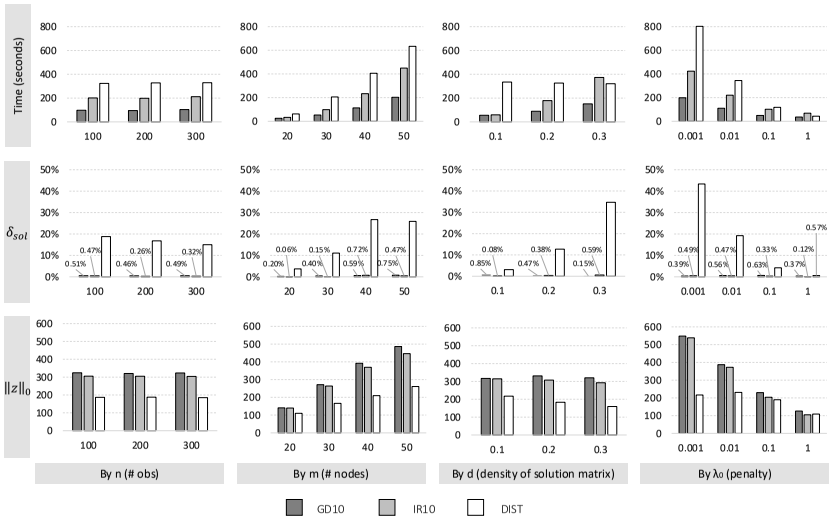


Figure 2: Performance of GD10, IR10, and DIST (dense data)

by DIST does not have difficulties preventing cycles and the algorithm can decide whether to include arcs easier. The comparison of δ_{sol} values also show that arc based search is competitive. Although all algorithms have δ_{sol} values less than 0.5%, we find clear evidence that the performance of IR10 decreases in increasing m and s and decreasing λ. Although the δ_{sol} values of GD10 and DIST are similar, considering the fast computing time of DIST, we recommend to use DIST for very sparse high dimensional data.

In Figure 4, we present combined results of all three data sets by relating solution densities and δ_{sol}. Observe that, for each quadruplet of $n, m, s(\text{or } d), \lambda$, we have results from 10 random instances for each algorithm. Value δ_{sol} is the average of the 10 results for each quadruplet and for each algorithm, $AvgDen$ is the average density of the adjacency matrices of the 10 results and the three algorithms. In Figure 4, we present a scatter plot of $ln(1+AvgDen)$ and $ln(1+100 \cdot \delta_{\text{sol}})$. Each point in the plot is the average of 10 results by an algorithm and each algorithm has 324 points displayed [1]. The numbers in the parenthesis along the axes are the corresponding values of δ_{sol} and $AvgDen$. In the plot, we first observe that the algorithms perform similarly when the solutions are sparse and the δ_{sol} values have large variance when the solutions are dense. When the log transformed solution density is less than 2, the average δ_{sol} values of GD10, IR10, and DIST are 0.06%, 0.15%, and 0.04%, respectively. However, the solution quality of DIST drastically decreases as the solutions become denser. This makes sense because sparse solutions can be efficiently searched by

1. $324 = (3 \cdot 4 \cdot 3 \cdot 4) + (3 \cdot 4 \cdot 3 \cdot 4) + (1 \cdot 3 \cdot 3 \cdot 4)$, where the three terms are for the three data sets and each term is obtained by multiplying the number of parameters $n, m, s(d)$, and λ, respectively.

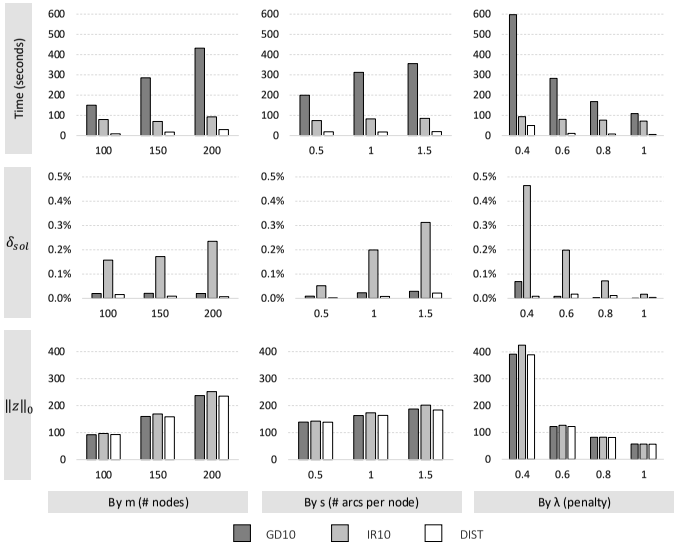


Figure 3: Performance of GD10, IR10, and DIST (high dimensional data)

arc-based search, while dense solutions are not easy to obtain by adding or removing arcs one by one. This also explains the relatively small and large δ_{sol} values for dense and spares solutions, respectively, by the topological order based algorithms. Between GD10 and IR10, we do not observe a big difference.

5.2 Comparison of MIP Models

In this section, we compare the performance of MIPto, MIPin, and MIPcp using *time*, δ_{sol}, and $\|z\|_0$ and the following additional metric.

δ_{IP}: the optimality gap (%) obtained by CPLEX within allowed $15 \cdot m$ seconds

Due to scalability issues of all models, we only use the sparse data with $m = 20, 30, 40$. We also limit $n = 100$. For all instances, we use the $15 \cdot m$ seconds time limit for CPLEX. For example, we have a time limit of 300 seconds for instances with $m = 20$.

The result is presented in Figure 5. Comparing the time of all models with the time limit for CPLEX, we observe that MIPin and MIPcp were able to terminate with optimality for several instances when $m = 20$ and $\lambda = 1$. This implies that MIPin and MIPcp are efficient when the problem is small and the number of selected arcs $\|z\|_0$ is small. However, in general, δ_{IP} values tend to be consistent with different models, while they increase in increasing m and s and in decreasing λ for all three MIP models. The execution times of all models increase in increasing m and s, and in decreasing λ. The same trend can be

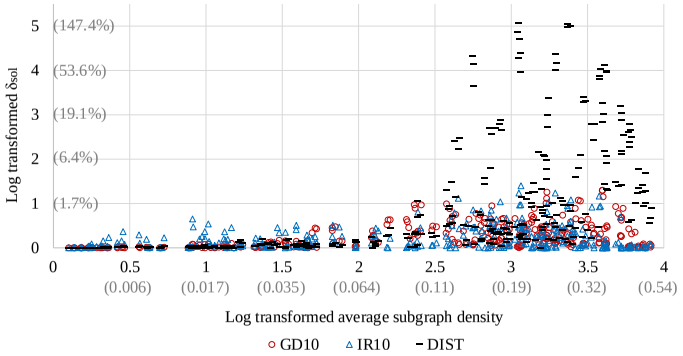


Figure 4: Scatter plot of δ_{sol} and average solution densities

found for δ_{IP} for all models. By comparing δ_{sol} values, we observe that MIPin is best when $m = 20$ and 30. However, the performance of MIPin drops drastically as m and s increase and λ decreases. Actually, MIPin fails to obtain a reasonably good solution within the time limit for several instances. This gives large δ_{sol} values and increases the average. The δ_{sol} values of MIPto are smaller than MIPcp when m and s are small, while MIPcp outperforms when $m = 40$ or $s = 3$.

5.3 Comparison of all MIP Models and Algorithms

In Figure 6, we compare all models and algorithms for selected sparse instances with $n = 100$ and $m \in \{20, 30, 40\}$, which were used to test MIP models. In the plot matrix, we show the average computation time and δ_{sol} (gap from the best objective value among the six models and algorithms) by m, s, and λ. Note that δ_{sol} values of a few MIPin results are not fully displayed in the bar plots due to their large values. Instead, the actual numbers are displayed next to the corresponding bar. The result shows that MIP models spent more time while the solution qualities are inferior in general. The values of δ_{sol} for the MIP models are competitive only when λ is large, which requires sparse solution. However, even for this case, MIP models spend longer time than the algorithms. Hence, ignoring the benefit of knowing and guaranteeing optimality by the MIP models, we conclude that the algorithms perform better for all cases.

The primary reason for the inferior performance of the MIP models is the difficulty of solving integer programming problems. Further, all of the MIP models have at least $O(m^2)$ binary variables and $O(m^2)$ constraints and the problem complexity grows fast. Finally, large values of big M in (15b) make the problem even more difficult. Due to non-tight values of big M, fathoming does not happen frequently in the branch and bound procedure. Hence, at least for sparse Gaussian Bayesian network learning, the MIP models may not be the best option unless other complicated constraints, which cannot be easily dealt with iterative algorithms, are needed.

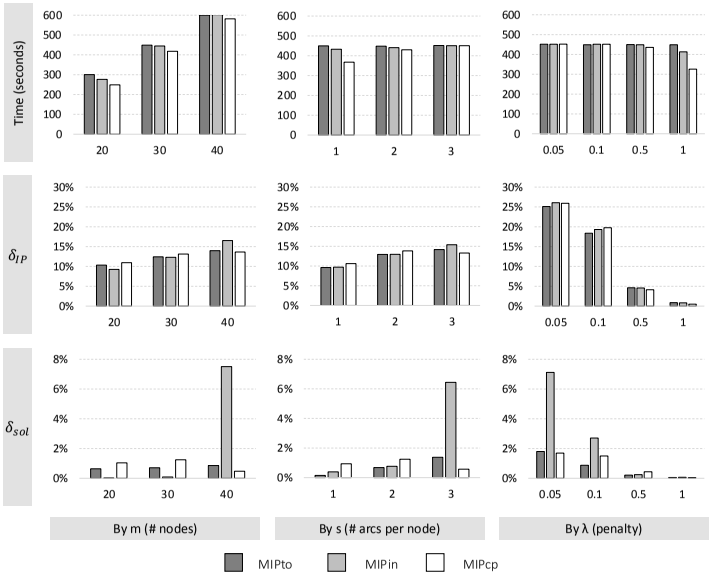


Figure 5: Performance of MIPto, MIPin, and MIPcp (sparse data with $n = 100$ and $m \in \{20, 30, 40\}$)

5.4 Real Data Example

In this section, we study the flow cytometry data set from Sachs et al. (2005) by solving (14). The data set has been studied in many works including Friedman et al. (2008), Shojaie and Michailidis (2010), Fu and Zhou (2013), and Aragam and Zhou (2015). The data set is often used as a benchmark as the casual relationships (underlying DAG) are known. It has $n = 7466$ cells obtained from multiple experiments with $m = 11$ measurements. The known structure contains 20 arcs. For the experiment, we standardize each column to have zero mean with standard deviation equal to one.

In Table 2, we compare the performance of the three algorithms GD10, IR10, and DIST for various values of λ. The MIP models are excluded due to scalability issue [2]. We compare the previously used performance measures execution time, solution cardinalities ($\|z\|_0$), and solution quality (δ_{sol}). In addition, we also compare sensitivities (true positive ratio) of the solutions. By comparing with the known structure with 20 arcs, we calculate directed true

2. Although m is small, with n =7466 observations, the MIP models have at least $7466 \cdot 11 \cdot 2 = 164{,}252$ continuous variables and $7466 \cdot 11 = 82{,}126$ constraints just for the residual terms. Combined with the complexity increment due to the binary variables and acyclicity constraints, it was not feasible to obtain a reasonable solution by any of the MIP models.

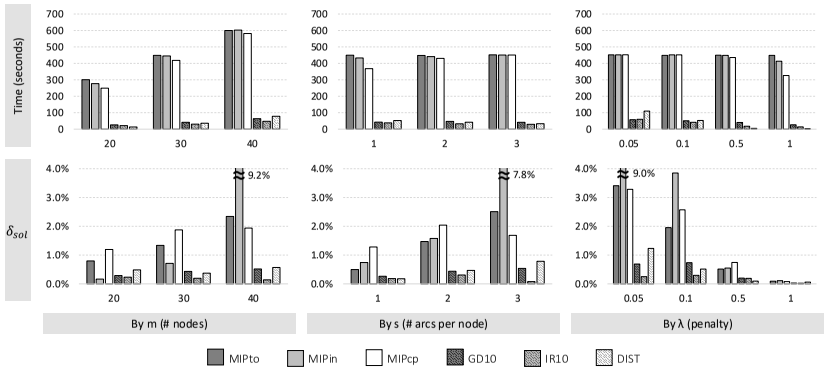


Figure 6: Performance of all models and algorithms (sparse data with $n = 100$ and $m \in \{20, 30, 40\}$)

positive (dTP) and undirected true positive (uTP). If arc (j, k) is in the known structure, dTP counts only if arc (j, k) is in the algorithm's solution, whereas uTP counts either of arcs (j, k) or (k, j) is in the algorithm's solution.

The solution times of the three algorithms are all within a few seconds, as m is small. Also, the solution cardinalities ($\|z\|_0$) are similar. The best δ_{sol} value among the three algorithms in each row is in boldface. We observe that GD10 provides the best solution (smallest δ_{sol}) in most cases and IR10 is the second best. The δ_{sol} values of DIST increase as λ increases. This is consistent with the findings in Section 5.1. Note that the density of the underlying structure is $20/11^2 = 0.165$, which is dense. This explains the good performance of GD10 and IR10. On the other hand, even though GD10 provides the best objective function values for most of the cases, the dTP and uTP values of GD10 are not always the best. The highest value among the three algorithms in each row is in boldface. While δ_{sol} values of DIST are the largest among the three algorithms, dTP and uTP values are the best in some cases. When λ is small, DIST tends to have higher dTP and uTP. However, as λ increases, GD10 gives the best dTP and uTP values. To further improve the prediction power, we may need weighting features or observations.

From Table 2, we observe that a slight change in solution quality δ_{sol} affects the final selection of DAG significantly. Also, the best objective function value does not necessarily give the highest true positive value since the L_1-norm penalized least square (14) may not be the best score function. In Figure 7, we present the graphs of known casual interactions (a), estimated subgraph by GD10 (b), IR10 (c), and DIST (d). All graphs are obtained with $\lambda = 0.25$, but the numbers of arcs are different (see Table 2). In fact, the difference between δ_{sol} values of IR10 and DIST is less than 1% while the subgraphs have only 10 common arcs.

	GD10					IR10					DIST				
λ	*time*	$\|z\|_0$	δ_{sol}	dTP	uTP	*time*	$\|z\|_0$	δ_{sol}	dTP	uTP	*time*	$\|z\|_0$	δ_{sol}	dTP	uTP
0.5	14.6	9	**0.00%**	0.22	**0.56**	10.8	9	0.22%	0.22	**0.56**	5.8	9	0.38%	**0.56**	**0.56**
0.45	13.7	11	**0.00%**	0.27	0.55	10.9	13	0.30%	0.15	0.46	8.5	9	0.49%	**0.56**	**0.56**
0.4	14.3	11	**0.00%**	0.27	**0.55**	14.8	13	0.36%	0.15	0.46	7.0	13	0.61%	**0.38**	0.46
0.35	16.0	13	**0.00%**	0.23	**0.54**	12.4	17	0.42%	0.12	0.41	8.6	15	0.73%	**0.33**	0.47
0.3	15.1	13	**0.00%**	0.23	0.46	14.3	17	0.21%	0.24	0.41	8.0	17	0.84%	**0.29**	**0.47**
0.25	14.4	16	**0.00%**	**0.38**	**0.56**	14.8	20	0.22%	0.25	0.50	6.7	20	0.94%	0.30	0.50
0.2	16.0	16	**0.00%**	0.31	**0.56**	14.6	22	0.31%	0.32	0.55	6.8	21	1.12%	**0.33**	0.52
0.15	16.4	21	**0.00%**	**0.33**	**0.57**	15.6	25	0.32%	0.32	0.52	9.5	23	1.26%	0.30	0.52
0.1	17.4	24	**0.00%**	0.25	**0.50**	11.0	28	0.26%	0.07	0.43	9.1	27	1.45%	**0.26**	0.44
0.05	17.7	28	0.32%	**0.39**	**0.50**	11.5	30	**0.00%**	0.10	0.47	10.1	32	1.87%	0.22	0.41

Table 2: Performance on the real data set from Sachs et al. (2005)

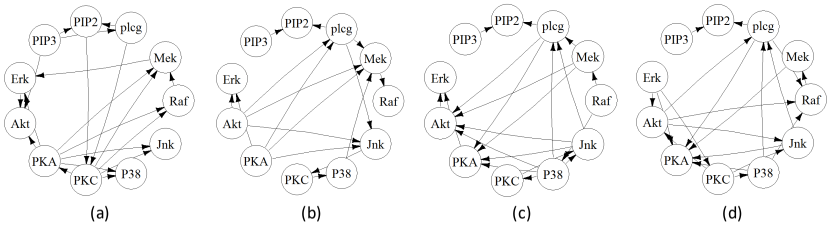


Figure 7: Known DAG (a) and estimated subgraphs with $\lambda = 0.25$ by GD10 (b), IR10 (c), and DIST (d)

6. Conclusion

We propose an MIP model and iterative algorithms based on topological order. Although the computational experiment is conducted for Gaussian Bayesian network learning, all the proposed model and algorithms are applicable for problems following the form in (1). While many MIP models and algorithms are designed based on arc search, using topological order provides some advantages that improve solution quality and algorithm efficiency.

1. DAG constraints (acyclicity constraints) are automatically satisfied when arcs from high order nodes to low order nodes are used.

2. In applying the concept for MIP, a lower number of constraints is needed ($O(m^2)$) whereas arc based modeling can have exponentially many constraints in the worst case.

3. In applying the concept in designing iterative algorithms, one of the biggest merits is the capability of utilizing the maximum number of arcs possible ($\frac{m(m-1)}{2}$), while arc based algorithms struggle with using all possible arcs.

The proposed MIP model has the smallest number of constraints while the number of binary variables is in the same order with the already known MIP models. It performs as good as a cutting plane algorithm. The proposed iterative algorithms get the biggest benefit when the solution matrix is dense. The result presented in Section 5.1 clearly indicates that the topological order based algorithms outperform when the density of the resulting solution is high. On the other hand, arc-based search algorithms, represented by DIST in our experiment, can be efficient when the desired solutions are very sparse.

Comparing all models and algorithms used in the experiment, we observe that the MIP models are not competitive or scale well compared to the heuristic algorithms except for small instances. The experiment shows that the solution times of the MIP models are significantly affected by the number of nodes m. For Gaussian Bayesian network learning, we observe that large n could also decrease the MIP model efficiency even when m is small (Section 5.4). Among the iterative algorithms, we recommend DIST for very sparse high dimensional data and GD10 and IR10 for dense data. Among the two topological order based algorithms, GD10 performs slightly better and is more stable.

Appendix A. Greedy Algorithm for Projection Problem

In this section, we present the detail derivations and proofs of Algorithm 3 (*greedy*). The algorithm sequentially determines topological order by optimizing the projection problem given an already fixed order up to the iteration point. Solving the projection problem is $\mathcal{NP}$-complete and our algorithm may not give a global optimal solution. However, the result of this section shows that Algorithm 3 gives an optimal choice of the next node to have fixed order given pre-fixed orders.

We start by describing some properties of Y^* in the following three lemmas. For the following lemmas, let π_j^* represent the topological order of node j defined by Y^*.

Lemma 5 *For any Y_{jk}^*, we must have either $Y_{jk}^* = 0$ or $Y_{jk}^* = U_{jk}^t$.*

Proof For a contradiction, let us assume that there exist indices q and r such that $Y_{qr}^* \neq 0$ and $Y_{qr}^* \neq U_{qr}^t$. Let us create a new solution $\bar{Y}$ such that $\bar{Y} = Y^*$ except $\bar{Y}_{qr} = U_{qr}^t$. Note that $\bar{Y}$ is a feasible solution to (12) because $supp(\bar{Y}) \leq supp(Y^*)$ element-wise since $Y_{qr}^* \neq 0$. Further, we have $\|Y^* - U^t\|_2 > \|\bar{Y} - U^t\|_2$ because $(Y_{qr}^* - U_{qr}^t)^2 > 0 = (\bar{Y}_{qr} - U_{qr}^t)^2$ and $\bar{Y} = Y^*$ except $Y_{qr}^* \neq \bar{Y}_{qr}$. This contradicts optimality of Y^*. ■

Note that Lemma 5 implies that solving (12) is essentially choosing between 0 and U_{jk}^* for Y_{jk}^*. This selection is also based on the following property.

Lemma 6 *If $\pi_j^* > \pi_k^*$, then $Y_{jk}^* = U_{jk}^t$.*

Proof For a contradiction, let us assume that there exist indices q and r such that $Y_{qr}^* \neq U_{qr}^t$ while $\pi_q^* > \pi_r^*$. Let us create a new solution $\bar{Y}$ such that $\bar{Y} = Y^*$ except $\bar{Y}_{qr} = U_{qr}^t$.

1. If $Y_{qr}^* \neq 0$, then $\bar{Y}$ is a DAG since $supp(\bar{Y}) \leq supp(Y^*)$ element-wise.
2. If $Y_{qr}^* = 0$, then arc (q, r) can be used in the solution without creating a cycle because $\pi_q^* > \pi_r^*$. Hence, $\bar{Y}$ is a DAG.

Therefore, $\bar{Y}$ is a feasible solution to (12). However, it is easy to see that $\|\bar{Y} - U^t\|_2 < \|Y^* - U^*\|_2$ because $(Y_{qr}^* - U_{qr}^t)^2 > 0 = (\bar{Y}_{qr} - U_{qr}^t)^2$. This contradicts optimality of Y^*. ■

Given the topological order by Y^*, let $\hat{J}^k = \{j \in J^k | \pi_j^* > \pi_k^*\}$ be the subset of J^k such that the nodes in $\hat{J}^k$ are earlier than node k. Combining Lemmas 5 and 6, we conclude that Y^* has the following structure.

$$Y_{jk}^* = \begin{cases} U_{jk}^t & \text{if } j \in \hat{J}^k, \\ 0 & \text{if } j \in J \setminus \hat{J}^k, \end{cases} \qquad k \in J \tag{18}$$

Further, we can calculate node k's contribution to the objective function value without explicitly using Y^*.

Lemma 7 *For each node $k \in J$, it contributes*

$$\sum_{j \in J \setminus \hat{J}^k} (U_{jk}^t)^2$$

to the objective function value for (12)*. In other words, the contribution of node k is the squared sum of U_{jk}^t for nodes with $\pi_j^* < \pi_k^*$.*

Proof For node k, we can derive

$$\sum_{j \in J}(Y_{jk}^* - U_{jk}^t)^2 = \sum_{j \in J \setminus \hat{J}^k}(Y_{jk}^* - U_{jk}^t)^2 = \sum_{j \in J \setminus \hat{J}^k}(U_{jk}^t)^2,$$

where both equal signs are due to (18). The first equality is due to $Y_{jk}^* = U_{jk}^t$ for $j \in \hat{J}^k$ and the second equality holds since $Y_{jk}^* = 0$ for $j \in J \setminus \hat{J}^k$. ■

We next detail the derivation of the greedy algorithm presented in Algorithm 3. Let $\bar{J} \subseteq J$ be the index set of yet-to-be-ordered nodes and $\bar{J}^c = J \setminus \bar{J}$ be the index set of the nodes that have already been ordered. The procedure is equivalent to iteratively solving

$$k^* = \text{argmin}_{k \in \bar{J}} \Big\{ \min_{Y_k} \big\{ \sum_{j \in J} (Y_{jk} - U_{jk}^t)^2 \big\} \Big\}, \tag{19}$$

where $Y_k = [Y_{1k}, Y_{2k}, \cdots, Y_{mk}] \in \mathbb{R}^{m \times 1}$ is the column in Y corresponding to node k. Set $\bar{J}$ is updated by $\bar{J} = \bar{J} \setminus \{k^*\}$ and $\pi_{k^*}^* = |\bar{J}|$ after solving (19). We propose an algorithm to solve (19) based on the properties of Y^* described in Lemmas 5 - 7. Given $\bar{J}$, we solve

$$k^* = \text{argmin}_{k \in \bar{J}} \Big\{ \sum_{j \in \bar{J}} (U_{jk}^t)^2 \Big\}. \tag{20}$$

Next, we show that solving (20) gives an optimal solution $\bar{Y}_{jk^*}$, $j \in J^k$ to (19). We can actually replicate the properties of Y^* for $\bar{Y}_{jk^*}$.

Lemma 8 *An optimal solution to* (19) *must have either $\bar{Y}_{jk^*} = 0$ or $\bar{Y}_{jk^*} = U_{jk^*}^t$ for all $j \in J^k$.*

Lemma 9 *An optimal solution to* (19) *must have $\bar{Y}_{jk^*} = U_{jk^*}^t$ for $j \in J \setminus \bar{J}$.*

Lemma 10 *An optimal solution to* (19) *must satisfy $\sum_{j \in J}(\bar{Y}_{jk^*} - U_{jk^*}^t)^2 = \sum_{j \in \bar{J}}(U_{jk^*}^t)^2$*

The proofs are omitted as they are similar to the proofs of Lemmas 5 - 7, respectively. Note that, by Lemma 10, we show the equivalence of $\sum_{j \in J}(Y_{jk^*} - U_{jk^*}^t)^2$ and $\sum_{j \in \bar{J}}(U_{jk^*}^t)^2$. This result also holds for each term, $k \in \bar{J}$, of the argmin function in (19). Hence, the following lemma holds.

Lemma 11 *Solving* (20) *is equivalent to solving* (19)*.*

Observe that (20) is used in Line 3 of the greedy algorithm in Algorithm 3. Hence, by property (19), the algorithm gives an optimal choice of node to be fixed with the pre-fixed topological order.

Appendix B. Summary Statistics for Maximum Coefficients

In this section, we show that the heuristic formula (17) for selecting big M gives reasonable and large enough values. Note that, when creating a synthetic instance, we used a random DAG to generate multivariate data. Although the optimal DAG for the penalized least

squares is unknown, with appropriate penalty constants, we can use the implanted DAG to obtain coefficients estimation. That is, we calculate

$$\hat{B} = \max_{j \in J^k, k \in K} |\hat{\beta}_{jk}|$$

by using the implanted DAG. The $\hat{B}$ value is then compared with the big M value in (17) for all instances used for MIP models (sparse data with $n = 100$ and $m \in \{20, 30, 40\}$). We calculate minimum, average, and maximum of $\hat{B}$ and M by m, s, and λ. The result is presented in Table 3. Note that we have $\hat{B} < M$ for all columns Min, Avg, and Max and for all rows. Although only the summary statistics are presented in Table 3, we observed that M is greater than $\hat{B}$ for all cases considered.

By		$\hat{B}$			M		
Param	Value	Min	Avg	Max	Min	Avg	Max
m	20	0.17	0.65	1.48	0.33	1.34	3.73
	30	0.18	0.62	0.95	0.36	1.32	3.10
	40	0.18	0.63	1.03	0.36	1.33	3.30
s	1	0.17	0.53	0.79	0.33	1.13	2.04
	2	0.23	0.63	0.94	0.45	1.33	3.54
	3	0.32	0.74	1.48	0.63	1.53	3.73
λ	0.05	0.62	0.82	1.48	1.24	1.95	3.73
	0.1	0.60	0.78	0.96	1.19	1.54	2.35
	0.5	0.41	0.59	0.74	0.82	1.14	1.45
	1	0.17	0.35	0.49	0.33	0.69	0.97

Table 3: Comparison of maximum coefficients and big M (sparse data with $n = 100$ and $m \in \{20, 30, 40\}$)

References

Bryon Aragam and Qing Zhou. Concave penalized estimation of sparse Gaussian Bayesian networks. *Journal of Machine Learning Research*, 16:2273–2328, 2015.

Ali Baharev, Hermann Schichl, and Arnold Neumaier. An exact method for the minimum feedback arc set problem. *Technical report*, 2015. Available from http://www.mat.univie.ac.at/∼neum/.

G Bolotashvili, M Kovalev, and Eberhard Girlich. New facets of the linear ordering polytope. *SIAM Journal on Discrete Mathematics*, 12(3):326–336, 1999.

David Maxwell Chickering. Learning Bayesian networks is NP-complete. In *Learning from data*, pages 121–130. Springer, 1996.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

William J. Cook, William H. Cunningham, William R. Pulleyblank, and Alexander Schrijver. *Combinatorial Optimization*. John Wiley & Sons, Inc., New York, NY, USA, 1998.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

Byron Ellis and Wing Hung Wong. Learning causal Bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103(482):778–789, 2008.

Guy Even, J Seffi Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998.

Henning Fernau and Daniel Raible. Exact algorithms for maximum acyclic subgraph on a superclass of cubic graphs. In *International Workshop on Algorithms and Computation*, pages 144–156. Springer, 2008.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9(3):432–441, 2008.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

Nir Friedman and Daphne Koller. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1):95–125, 2003.

Fei Fu and Qing Zhou. Learning sparse causal Gaussian networks with experimental intervention: regularization and coordinate descent. *Journal of the American Statistical Association*, 108(501):288–300, 2013.

Michel X Goemans and Leslie A Hall. The strongest facets of the acyclic subgraph polytope are unknown. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 415–429. Springer, 1996.

Martin Grötschel, Michael Jünger, and Gerhard Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32(6):1195–1220, 1984.

Martin Grötschel, Michael Jünger, and Gerhard Reinelt. On the acyclic subgraph polytope. *Mathematical Programming*, 33(1):28–42, 1985.

Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *49th Annual IEEE Symposium on Foundations of Computer Science*, pages 573–582. IEEE, 2008.

Sung Won Han, Gong Chen, Myun-Seok Cheon, and Hua Zhong. Estimation of directed acyclic graphs through two-stage adaptive Lasso for gene network inference. *Journal of the American Statistical Association*, 111(515):1004–1019, 2016.

Refael Hassin and Shlomi Rubinstein. Approximations for the maximum acyclic subgraph problem. *Information Processing Letters*, 51(3):133–140, 1994.

David Heckerman, Dan Geiger, and David M Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

R Kaas. A branch and bound algorithm for the acyclic subgraph problem. *European Journal of Operational Research*, 8(4):355–362, 1981.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012.

Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 95–103. ACM, 2007.

Wai Lam and Fahiem Bacchus. Learning Bayesian belief networks: An approach based on the mdl principle. *Computational Intelligence*, 10(3):269–293, 1994.

Janny Leung and Jon Lee. More facets from fences for linear ordering and acyclic subgraph polytopes. *Discrete Applied Mathematics*, 50(2):185–200, 1994.

CL Lucchesi and DH Younger. A minimax theorem for directed graphs. *Journal of the London Mathematical Society*, 17(3):369–374, 1978.

John E Mitchell and Brian Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In *High Performance Optimization*, pages 349–366. Springer, 2000.

Teppo Niinimäki, Pekka Parviainen, and Mikko Koivisto. Structure discovery in Bayesian networks by sampling partial orders. *The Journal of Machine Learning Research*, 17(1): 2002–2048, 2016.

Young Woong Park and Diego Klabjan. Subset selection for multiple linear regression via optimization. *Technical report*, 2013. Available from http://www.klabjan.dynresmanagement.com.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL `https://www.R-project.org/`.

Vijaya Ramachandran. Finding a minimum feedback arc set in reducible flow graphs. *Journal of Algorithms*, 9(3):299–313, 1988.

Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graphs based on sparsest permutations. *arXiv preprint arXiv:1307.0366*, 2013.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Ali Shojaie and George Michailidis. Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538, 2010.

Sara Van de Geer and Peter Bühlmann. L0-penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics*, 41(2):536–567, 2013.

Journal of Machine Learning Research 18 (2017) 1-37 Submitted 11/16; Revised 6/17; Published 10/17

Stability of Controllers for Gaussian Process Dynamics

Julia Vinogradska[1,2] JULIA.VINOGRADSKA@DE.BOSCH.COM
Bastian Bischoff[1] BASTIAN.BISCHOFF@DE.BOSCH.COM
Duy Nguyen-Tuong[1] DUY.NGUYEN-TUONG@DE.BOSCH.COM
Jan Peters[2,3] MAIL@JAN-PETERS.NET

[1] *Corporate Research, Robert Bosch GmbH*
Robert-Bosch-Campus 1
71272 Renningen

[2] *Intelligent Autonomous Systems Lab, Technische Universität Darmstadt*
Hochschulstraße 10
64289 Darmstadt

[3] *Max Planck Institute for Intelligent Systems*
Spemannstraße 10
72076 Tübingen

Editor: George Konidaris

Abstract

Learning control has become an appealing alternative to the derivation of control laws based on classic control theory. However, a major shortcoming of learning control is the lack of performance guarantees which prevents its application in many real-world scenarios. As a step towards widespread deployment of learning control, we provide stability analysis tools for controllers acting on dynamics represented by Gaussian processes (GPs). We consider differentiable Markovian control policies and system dynamics given as (i) the mean of a GP, and (ii) the full GP distribution. For both cases, we analyze finite and infinite time horizons. Furthermore, we study the effect of disturbances on the stability results. Empirical evaluations on simulated benchmark problems support our theoretical results.

Keywords: Stability, Reinforcement Learning, Control, Gaussian Process

1. Introduction

Learning control based on Gaussian process (GP) forward models has become a viable approach in the machine learning and control theory communities. Many successful applications impressively demonstrate the efficiency of this approach (Deisenroth et al., 2015; Pan and Theodorou, 2014; Klenske et al., 2013; Maciejowski and Yang, 2013; Nguyen-Tuong and Peters, 2011; Engel et al., 2006; Kocijan et al., 2004). In contrast to classic control theory methods, learning control does not presuppose a detailed understanding of the underlying dynamics but tries to infer the required information from data. Thus, relatively little expert knowledge about the system dynamics is required and fewer assumptions, such as a parametric form and parameter estimates, must be made. Employing Gaussian processes as forward models for learning control is particularly appealing as they incorporate uncertainty about the system dynamics. GPs infer a distribution over all plausible models given the observed data instead of one (possibly erroneous) model and, thus, avoid severe modeling errors. Furthermore, obtaining a task solution by a learning process can significantly decrease the involved manual effort.

Unfortunately, performance guarantees rarely exist for arbitrary system dynamics and policies learned from data. An important and well-established type of performance guarantee is *(asymptotic) stability*. A *stability region* in the state space ensures, that all trajectories starting in this region converge to the target. Classic control theory offers a rich variety of stability analysis, e.g., for linear, nonlinear, and stochastic systems (Khalil, 2014; Khasminskii and Milstein, 2011; Skogestad and Postlethwaite, 2005; Kushner, 1967). In a preliminary study (Vinogradska et al., 2016), we introduced a tool to analyze the stability of learned policies for closed-loop control systems with transition dynamics given as a GP. This tool handled two types of dynamics: dynamics given as (i) the mean of a GP, and (ii) the full GP distribution. While the first case results in a deterministic closed-loop system, uncertainty is present in the second case. As propagating distributions through a GP is analytically intractable, we proposed a novel approach for approximate multi-step-ahead predictions based on numerical quadrature. Finally, we were able to analyze asymptotic stability of the deterministic dynamics in case (i), but in case (ii) only finite time horizons were considered. Furthermore, the results were limited to GPs with squared exponential kernel.

In this paper, we substantially extend our previous work (Vinogradska et al., 2016) and move to a solid foundation for a stability analysis tool. For this detailed study on stability of closed-loop control systems, we consider dynamics given as (i) the mean of a GP, and (ii) the full GP distribution. In both cases, we analyze finite as well as infinite time horizons. For dynamics given as the mean of a GP, we obtain a region of starting points in the state space such that the target state is reached up to a given tolerance at the (finite) time horizon. For infinite time horizons, we construct a stability region in the state space. Furthermore, we study the behavior of dynamics given by the mean of a GP when disturbances are present. Here, we derive criteria for the disturbance such that the closed-loop control system remains stable. For full GP dynamics, we propose an algorithm to find a region of starting points in the state space such that the probability of the state to be in the target region at the (finite) time horizon is at least a given minimum probability. As the GP predictive distribution at any query point is Gaussian, the probability to transition to a point arbitrarily far from the current point is greater than zero and, thus, the system will eventually leave any compact set. Hence, for full GP dynamics the notions of deterministic analysis do not apply. Here, we extend the previous results (Vinogradska et al., 2016) substantially and analyze asymptotic behavior of closed-loop control systems with full GP dynamics. Employing methods from Markov chain theory, we show that for many choices of the prior the system converges to a unique, invariant limiting distribution.

This paper lays a foundation for theoretical stability analysis for control of probabilistic models learned from data. The proposed approaches provide stability guarantees for many

©2017 Julia Vinogradska, Bastian Bischoff, Duy Nguyen-Tuong and Jan Peters.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-590.html.

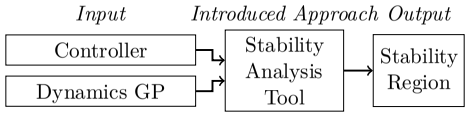


Figure 1: We present a tool to analyze stability for a given controller and Gaussian process (GP) dynamics model. Our tool analytically derives a stability region where convergence to the goal is guaranteed. Thus, we can provide stability guarantees for many of the existing GP based learning control approaches.

existing learning control approaches based on GPs. The paper is organized as follows: first, we briefly review related work and introduce the problem to be addressed. Section 2 provides necessary background for the stability analysis. In Section 3, we analyze closed-loop control systems with dynamics given as the mean of a GP. Subsequently, we study the case when the dynamics are given as the full GP distribution in Section 4. Section 5 provides empirical evaluations on benchmark control tasks. A conclusion is given in Section 6.

1.1 Related Work

To date, only very few special cases of system dynamics and policies learned from data have been analyzed with respect to stability. For example, Perkins and Barto (2003) and Nakanishi et al. (2002) analyze the scenario of an agent switching between given safe controllers. Kim and Ng (2005) monitor stability while learning control for the special case of a linear controller and linear dynamics. Recently, stability properties of autoregressive systems with dynamics given as the mean of a GP were analyzed by Beckers and Hirche (2016), who derive an upper bound on the number of equilibrium points for squared exponential and some special cases of polynomial covariance functions. Furthermore, the authors compute an invariant set and show the boundedness of all system trajectories. However, no control inputs are considered in this work and no asymptotic stability results are given. Furthermore, the full GP distribution is not considered in (Beckers and Hirche, 2016). To the best of our knowledge, stability analysis for closed-loop control with probabilistic models, such as GPs, and arbitrary policies has not been addressed so far.

In classic control theory, the first formal analysis of closed-loop dynamics dates back to the 19th century (Routh, 1877; Hurwitz, 1895; Lyapunov, 1892). Lyapunov's approach allows to analyze stability for nonlinear deterministic systems $\dot{x} = f(x)$. It studies the system behavior around *equilibrium points*, i.e., points, where all derivatives $\dot{x}$ of the state x vanish and the system comes to a hold. An equilibrium point x_e is *stable*, if for every $\varepsilon > 0$ a $\delta > 0$ exists such that $\|x(t) - x_e\| < \epsilon$ for every solution $x(t)$ of the differential equation with $\|x(t_0) - x_e\| < \delta$ and $t \geq t_0$. The equilibrium point is *asymptotically stable*, if it is stable and δ can be chosen such that $\|x(t) - x_e\| \to 0$ as $t \to \infty$, see (Khalil, 2014). One approach to proving stability of x_e is to find a *Lyapunov function*, i.e., a non-negative function, that vanishes only at x_e and decreases along system trajectories. This approach can be applied to nonlinear dynamics. However, finding a Lyapunov function is typically challenging as there exists no general method for this task.

Additionally to the difficulty of finding a suitable Lyapunov function, classical Lyapunov approaches are highly challenging as proving nonnegativity of a nonlinear function is in general intractable. In (Parrilo, 2000), a relaxation of the positive definiteness problem for polynomials was introduced: a polynomial is clearly nonnegative, if it can be written as a sum of squares (SOS) of polynomials. Fortunately, checking whether a polynomial can be written as SOS can be formulated as a semidefinite program (SDP) and can be solved in polynomial time. Thus, for polynomial system dynamics one can search for polynomial Lyapunov functions conveniently by solving the corresponding SDP. This SOS relaxation has been successfully applied to a number of nonlinear stability problems as finding the region of attraction of a system (Chesi, 2004; Majumdar et al., 2014), robust analysis (Topcu et al., 2010b) or controller design (Moore and Tedrake, 2014). While there are some limitations to these methods (Majumdar et al., 2014; Ahmadi and Parrilo, 2011), SOS approaches are of high practical importance. However, SOS methods can only be applied directly to polynomial dynamics. Nonpolynomial dynamics must be recast as polynomial systems by an (automated) procedure (Papachristodoulou and Prajna, 2005). This recasting procedure introduces new variables and constraints for each nonpolynomial term and can, thus, significantly increase the size of the SDP that must be solved. Especially for nonparametric and nonpolynomial models as, e.g., a GP with squared exponential or Matérn kernels, recasting often leads to SDPs that are computationally infeasible (Topcu et al., 2010a).

For consideration of uncertainty in the dynamics, stochastic differential equations (SDEs) have been introduced (Adomian, 1983). SDEs are differential equations where some terms are stochastic processes. In the presence of uncertainty, x_e is *stable in probability*, if for every $\varepsilon > 0$ and $p > 0$, there exists $\delta > 0$ such that $\mathrm{P}\{\|x(t) - x_e\| > \varepsilon\} < p$ for $t > t_0$ and solutions $x(t)$ with $\|x(t_0) - x_e\| < \delta$. Furthermore, x_e is *asymptotically stable in probability*, if it is stable in probability and $\mathrm{P}\{\|x(t) - x_e\| > \varepsilon\} \to 0$ for a suitable δ, see (Khasminskii and Milstein, 2011). Stability follows from the existence of a *supermartingale*, a stochastic Lyapunov function analogue. However, supermartingales exist only if the noise vanishes at the equilibrium point. Relaxing the supermartingale criterion allows for stability statements for a finite time horizon (Steinhardt and Tedrake, 2012; Kushner, 1966). Kushner's (1966) approach requires a supermartingale to be found manually, which is challenging in most cases. Steinhardt and Tedrake (2012) address this difficulty by constructing a supermartingale via SOS programming. This approach is suitable for systems with polynomial dynamics and shares some of the difficulties of SOS approaches as described above.

When controlling a system, uncertainty may arise from modeling inaccuracies, the presence of (external) disturbances and the lack of data, as some system parameters are not known in advance but only during operation. Robustness of nonlinear systems to structured or unstructured uncertainty has been considered in classical control theory, e.g., via the small gain theorem (Zhou and Doyle, 1998). However, these approaches typically lead to an infinite number of nonlinear matrix inequalities and cannot be solved analytically. Another approach is to take uncertainty into account when designing a controller. Thus, a controller must be suitable for a family of systems that is specified, e.g., by bounds for model parameters or for nonparametric uncertainties as bounds for the operator norm of the unknown dynamics. Robust control (Skogestad and Postlethwaite, 2005; Zhou and Doyle, 1998) designs a single controller that is provably stable for all systems in the specified family – often at cost of overall controller performance. Adaptive control (Narendra and Annaswamy, 2012; Tao,

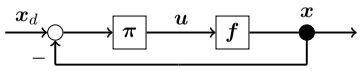


Figure 2: A closed-loop control structure with controller $\boldsymbol{\pi}$ and the system dynamics $\boldsymbol{f}$. We study stability for two types of dynamics: (i) the mean of the GP and (ii) the full GP predictive distribution.

2003) instead adjusts control parameters online in order to achieve prespecified performance, which can be computationally demanding. Both schemes require stability analysis of the dynamics system, e.g., via Lyapunov functions as described above.

1.2 Problem Statement

The goal of this paper is to provide an automated tool to analyze stability of a closed-loop structure when the dynamics model is given as a GP, see Figure 1. As inputs, our tool expects a control policy and a GP dynamics model. It checks the stability of the corresponding closed-loop structure and returns a stability region. Trajectories starting in this region are guaranteed to converge to the target (in probability). Subsequently, we discuss the different components of the tool in more detail.

We consider controllers which depend only on the current state and are differentiable with respect to the state. The dynamics model is given as a GP with stationary covariance function. This covariance function is assumed to have bounded derivatives with respect to all input dimensions and bounded prior mean. We consider a discrete-time system $\boldsymbol{x}^{(t+1)} = \boldsymbol{f}(\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)})$ with $\boldsymbol{x}^{(t)} \in \mathbb{R}^D$, $\boldsymbol{u}^{(t)} \in \mathbb{R}^F$, $t = 1, 2, \ldots$ and a controller $\boldsymbol{\pi}\colon \mathbb{R}^D \to \mathbb{R}^F$, whose objective is to move the system to a desired state $\boldsymbol{x}^d$, see Figure 2. In this paper, we study two possible cases for the dynamics $\boldsymbol{f}$: (i) the mean of a GP and (ii) the full GP predictive distribution. Note, that in the second case, distributions have to be propagated through the GP resulting in non-Gaussian state distributions.

2. Preliminaries

In this section, we introduce basic concepts for the proposed stability analysis. First, we briefly review Gaussian process regression, as GPs are employed to describe the considered dynamics. Second, we recap numerical quadrature, as it is the basis for the uncertainty propagation method proposed in Section 4.

2.1 Gaussian Process Regression

Given noisy observations $\mathscr{D} = \{(\boldsymbol{z}^i, y^i = f(\boldsymbol{z}^i) + \varepsilon^i) \mid 1 \leq i \leq N\}$, where $\varepsilon^i \sim \mathcal{N}(0, \sigma_n^2)$, the prior on the values of f is $\mathcal{N}(0, K(Z,Z) + \sigma_n^2 I)$. The covariance matrix $K(Z,Z)$ is defined by the choice of covariance function k as $[K(Z,Z)]_{ij} = k(\boldsymbol{z}^i, \boldsymbol{z}^j)$. One commonly employed covariance function is the squared exponential

$$k(\boldsymbol{z}, \boldsymbol{w}) = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{w})^\intercal \Lambda^{-1} (\boldsymbol{z} - \boldsymbol{w})\right),$$

with signal variance σ_f^2 and squared lengthscales $\Lambda = \operatorname{diag}(l_1^2, \ldots, l_{D+F}^2)$ for all input dimensions. Given a query point $\boldsymbol{z}_*$, the conditional probability of $f(\boldsymbol{z}_*)$ is

$$f(\boldsymbol{z}_*) \mid \mathscr{D} \sim \mathcal{N}(\boldsymbol{k}(\boldsymbol{z}_*, Z)\boldsymbol{\beta},\ k(\boldsymbol{z}_*, \boldsymbol{z}_*) - \boldsymbol{k}(\boldsymbol{z}_*, Z)(K(Z,Z) + \sigma_n^2 I)^{-1}\boldsymbol{k}(Z, \boldsymbol{z}_*)) \tag{1}$$

with $\boldsymbol{\beta} = (K(Z,Z) + \sigma_n^2 I)^{-1}\boldsymbol{y}$. The hyperparameters, e.g. $\sigma_n^2, \sigma_f^2, \Lambda$ for the squared exponential kernel, are estimated by maximizing the log marginal likelihood of the data (Rasmussen and Williams, 2005).

In this paper, $\boldsymbol{f}$ models system dynamics. It takes state-action pairs $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{u})^\intercal$ and outputs successor states $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})$. As these outputs are multivariate, we train conditionally independent GPs for each output dimension. We write $\sigma_{n,m}^2$, $\sigma_{f,m}^2, \Lambda_m$ for the GP hyperparameters in output dimension m and k_m for the corresponding covariance function.

2.2 Numerical Quadrature

Numerical quadrature approximates the value of an integral

$$\int_a^b f(\boldsymbol{x}) d\boldsymbol{x} \approx \sum_{i=1}^{p} w_i f(\boldsymbol{\xi}_i)$$

given a finite number p of function evaluations. A widely used class of quadrature rules are interpolatory quadrature rules, which integrate all polynomials up to a certain degree exactly. In this paper, we employ Gaussian quadrature rules, where the evaluation points $\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_p$ are chosen to be the roots of certain polynomials from orthogonal polynomial families. They achieve the highest accuracy possible for univariate interpolatory formulæ (Süli and Mayers, 2003). For multivariate integrals, the quadrature problem is significantly harder. While many formulæ for the univariate case can straightforwardly be generalized to multivariate integrals, they often suffer from the curse of dimensionality. However, quadrature methods that scale better and are feasible for up to 20 dimensions have been developed. See (Skrainka and Judd, 2011) for an overview.

3. Stability of GP Mean Dynamics

A wide variety of dynamics can be modeled well by the mean of a GP (Micchelli et al., 2006) due to the universal approximation property of many kernels. Thus, stability of systems with GP mean forward dynamics is an interesting problem. Please note that stability of closed-loop control systems with dynamics given as the mean of a GP is a classical problem from nonlinear control with a particular choice of dynamics function. Classical approaches to solve this problem have been discussed in Section 1.1 and range from direct Lyapunov approaches to the more recent SOS approach to construct polynomial Lyapunov functions. However, for GP mean dynamics, finding a Lyapunov function directly seems even more challenging than for a classical ODE derived from expert knowledge about the physical system. On the other hand, constructing a polynomial Lyapunov function as the solution of an SOS problem is typically computationally infeasible for these dynamics systems, as the nonparametric nature of GP mean functions results in a large number of nonpolynomial terms that significantly increase the complexity of the underlying semidefinite program (see Section 1.1). Thus, we propose an alternative approach to address this problem.

In this section, we provide tools to check the stability of a closed-loop control structure with a given differentiable Markovian control policy π and GP mean dynamics. For this class of forward dynamics, the next state $\boldsymbol{x}^{(t+1)}$ is given as the mean of the GP predictive distribution at the inputs $\boldsymbol{x}^{(t)}, \boldsymbol{u} = \boldsymbol{\pi}(\boldsymbol{x}^{(t)})$. Firstly, we consider infinite time horizons. In this case, we attempt to find a region of starting points in the state space, such that trajectories starting in this region are guaranteed to converge to the desired point $\boldsymbol{x}^d$ as $t \to \infty$. To obtain this result, we analyze the sensitivity of the GP predictive mean to variations in the input space and derive upper bounds for the distance of the predictions at two different points. This analysis can be used to find a region around the target point $\boldsymbol{x}^d$, where the next state is closer to $\boldsymbol{x}^d$ than the point before. It follows straightforwardly, that the full metric ball of maximal radius, which lies completely inside this region, is a stability region. All trajectories that start inside of this metric ball converge to the target point as $t \to \infty$. Furthermore, we will show that in this metric ball, the distance of the current state to the target decreases exponentially. We will exploit this statement to derive some finite time horizon and robustness results. For finite time horizons, we aim to find a region in the state space, such that the state does not deviate more than a given tolerance from the target when the time horizon is reached. The robustness analysis deals with asymptotic stability of the GP mean dynamics when disturbances are present. We derive conditions for these disturbances, such that convergence to the target state $\boldsymbol{x}^d$ is still guaranteed.

In the following, we will briefly review the notion of stability employed for the analysis and derive the main result of this section: an algorithm to find a stability region. After proving correctness of this algorithm, we will derive statements on finite time stability of the GP mean dynamics and on infinite time stability of the disturbed system.

3.1 Stability Notion

If the system dynamics $\boldsymbol{f}$ is given by the mean of a GP, the resulting closed-loop structure is deterministic. To assess the quality of a controller $\boldsymbol{\pi}$, which aims to stabilize the system at the reference point $\boldsymbol{x}^d$, we propose an algorithm to find a stability region of the closed-loop system.

Definition 1 *The reference point $\boldsymbol{x}^d$ is* stable, *if for every $\varepsilon > 0$ there exists $\delta > 0$, such that $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| < \varepsilon$ for $t = 1, 2, \dots$ and $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\| < \delta$. If $\boldsymbol{x}^d$ is stable and there exists a $\delta_0 > 0$ such that $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| \to 0$ for $t \to \infty$, $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\| < \delta_0$, $\boldsymbol{x}^d$ is* asymptotically stable. *A subset X^c of the state space is a* stability region, *if $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| \to 0$ as $t \to \infty$ for all $\boldsymbol{x}^{(0)} \in X^c$.*

Please note that $\boldsymbol{\pi}$ is implicit in this definition, as the states $\boldsymbol{x}^{(t)}$ for $t = 1, 2, \dots$ depend on $\boldsymbol{\pi}$ via $\boldsymbol{x}^{(t+1)} = \boldsymbol{f}(\boldsymbol{x}^{(t)}, \boldsymbol{\pi}(\boldsymbol{x}^{(t)}))$. This definition matches Lyapunov's stability notion (Khalil, 2014). We will present an algorithm that checks asymptotic stability of $\boldsymbol{x}^d$ by constructing a stability region X^c.

When the time horizon considered is finite, we are interested in finding all starting states, such that the distance to the target is at most a given $\varepsilon > 0$.

Definition 2 *A subset X^c of the state space is an (ε, T)-*stability region, *if $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| < \varepsilon$ holds for all $\boldsymbol{x}^0 \in X^c$ and $t \geq T$.*

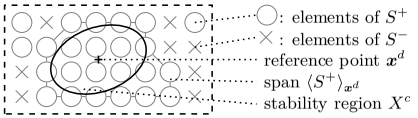


Figure 3: Basic idea of Algorithm 1 to construct a stability region: for a finite set of states S, employ upper bound from Lemma 3 to check whether the successor state is closer to $\boldsymbol{x}^d$, obtaining sets S^+ and S^-. Return metric ball of maximal radius that fits into the span $\langle S^+ \rangle_{\boldsymbol{x}^d}$ as stability region X^c. For $\boldsymbol{x}^{(0)} \in X^c$, the controlled system never leaves X^c and converges to $\boldsymbol{x}^d$ as $t \to \infty$.

In the following, we propose an algorithm to verify asymptotic stability of $\boldsymbol{x}^d$ as in Definition 1 and prove its correctness. Subsequently, we show how to find finite time statements as in Definition 2.

3.2 Algorithm Sketch

In this section, we derive an algorithm, that can find a stability region of a closed-loop control structure with GP mean dynamics. To find a stability region, we analyze how the distance between the current state and the target evolves. The basic idea of the algorithm involves *positively invariant sets*, i.e., sets that once entered, the system will never leave again (Blanchini, 1999). More precisely, we aim to find all $\boldsymbol{x}$, such that

$$\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| < \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| \tag{2}$$

for $\boldsymbol{x}^{(t)} = \boldsymbol{x}$ and a fixed $0 < \gamma < 1$. Assume there is a region containing $\boldsymbol{x}^d$ where Eq. (2) holds. For all states in this region, the distance to the target point decreases in one time step. If it is possible to fit a full metric ball centered at the target point $\boldsymbol{x}^d$ in this region, then all trajectories starting in the ball will never leave it. In addition, the distance of the current state to the reference point $\boldsymbol{x}^d$ will decrease in every step by at least factor γ. Thus, it follows immediately from the existence of such a ball that the target point is asymptotically stable. However, finding a region in closed form where Eq. (2) holds, is usually intractable. Instead, we follow an algorithmic approach to construct a stability region.

The proposed algorithm (Alg. 1) finds a region where Eq. (2) holds if one exists. It employs an upper bound for $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\|$, that depends only on $\boldsymbol{x}^{(t)}$. With this upper bound we can check whether Eq. (2) holds for a (finite) set of points. However, to find a continuous state space region where Eq. (2) holds, this upper bound must allow for generalization from discrete points. More precisely, we need an upper bound of $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\|$ that depends on $\boldsymbol{x}^{(t)}$ smoothly, thus, having bounded gradients, and can be handled conveniently. We employ an upper bound derived from sensitivity analysis of the GP mean to variations in the input space. This bound depends on $\boldsymbol{x}^{(t)}$ in a way that can be exploited to compute a finite set of points S, e.g., a grid that is sufficient to consider as follows: For any grid point, we check if Eq. (2) holds, obtaining the set S^+ with the grid points that fulfill Eq. (2) and the rest in S^-. Let $\langle S^+ \rangle_{\boldsymbol{x}^d}$ be the polygon spanned by the connected component of

Algorithm 1 Stability region X^c for GP mean dynamics

Input: dynamics GP $\boldsymbol{f}$, control policy $\boldsymbol{\pi}$, $\boldsymbol{x}^d$, γ
Output: stability region X^c

```
Construct grid S with Lemma 4, S⁺ ← ∅, X^c ← ∅
for x^(0) ∈ S do
    Compute upper bound C(x^(0), x^d, π) ≥ ‖x^(1) − x^d‖ with Lemma 3
    if C < γ ‖x^(0) − x^d‖ then S⁺ ← S⁺ ∪ {x} fi
od
if ⟨S⁺⟩_{x^d} exists then
    Fit metric ball B(x^d) ⊆ ⟨S⁺⟩_{x^d}
    X^c ← B(x^d) fi
return X^c
```

S^+, which contains $\boldsymbol{x}^d$. We can choose S such that Eq. (2) holds for all points in $\langle S^+\rangle_{\boldsymbol{x}^d}$, see Figure 3. Algorithm 1 gives an overview, Sec. 3.3 provides technical details and proves correctness of the approach.

3.3 Correctness of the Algorithm

In this section, we elaborate on the computation steps of Algorithm 1 and prove that the returned state space region X^c is a stability region of the closed-loop system in the sense of Definition 1. Fundamental to Algorithm 1 is the upper bound for $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\|$, which can be obtained as follows. We denote $\hat{\boldsymbol{x}} \coloneqq (\boldsymbol{x}, \boldsymbol{\pi}(\boldsymbol{x}))^\intercal$ and recall that $\boldsymbol{f}(\hat{\boldsymbol{x}})$ is the GP predictive mean at $\hat{\boldsymbol{x}}$, see Eq. (1).

Lemma 3 *Let $\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^d \in \mathbb{R}^{D+F}$ and $B \in \mathbb{R}^{D\times D}$ be a positive definite matrix. The distance of GP predictive means $\boldsymbol{f}(\hat{\boldsymbol{x}})$ and $\boldsymbol{f}(\hat{\boldsymbol{x}}^d)$ in the metrics induced by B is bounded by*

$$\|\boldsymbol{f}(\hat{\boldsymbol{x}}) - \boldsymbol{f}(\hat{\boldsymbol{x}}^d)\|_B^2 \leq (\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^d)^\intercal M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})(\hat{\boldsymbol{x}} - \hat{\boldsymbol{x}}^d) =: C(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$$

with a symmetric matrix M. This matrix can be constructed explicitly and depends on $\boldsymbol{x}, \boldsymbol{x}^d$ and the policy $\boldsymbol{\pi}$.

Proof This statement is obviously true for $\boldsymbol{x} = \boldsymbol{x}^d$, so let $\boldsymbol{x} \neq \boldsymbol{x}^d$. Evaluating

$$\|\boldsymbol{f}(\hat{\boldsymbol{x}}) - \boldsymbol{f}(\hat{\boldsymbol{x}}^d)\|_B^2 = \sum_{m,m'=1}^{D} \sum_{i,k=1}^{N} b_{mm'}(k_m(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^i) - k_m(\hat{\boldsymbol{x}}^d, \hat{\boldsymbol{x}}^i))(k_{m'}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^k) - k_{m'}(\hat{\boldsymbol{x}}^d, \hat{\boldsymbol{x}}^k))\beta_{mi}\beta_{m'k} \quad (3)$$

we realize the need for an upper bound of $k_m(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^i) - k_m(\hat{\boldsymbol{x}}^d, \hat{\boldsymbol{x}}^i)$. Recall that the covariance functions k_m are differentiable with bounded derivatives with respect to $\boldsymbol{x}$ and also, that the control policy π is differentiable with respect to the state $\boldsymbol{x}$. Thus, we can integrate the gradient field of k_m along a curve $\boldsymbol{\tau}$ from $\hat{\boldsymbol{x}}^d$ to $\hat{\boldsymbol{x}}$. As this path integral does not depend on the particular curve $\boldsymbol{\tau}$, we may choose $\boldsymbol{\tau} = \boldsymbol{\tau}^{D+F} \dots \boldsymbol{\tau}^1$ as the curve along the edges of the hypercube defined by $\hat{\boldsymbol{x}}^d$ and $\hat{\boldsymbol{x}}$, i.e., $\boldsymbol{\tau}_p^j(t) = \hat{x}_p$ if $p \leq j-1$, $\boldsymbol{\tau}_p^j(r) = \hat{x}_p^d + r(\hat{x}_p - \hat{x}_p^d)$ with

$r \in [0;1]$ if $p = j$, and $\boldsymbol{\tau}_p^j(t) = \hat{x}_p^d$ otherwise. This definition yields

$$k_m(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^i) - k_m(\hat{\boldsymbol{x}}^d, \hat{\boldsymbol{x}}^i) = \sum_{j=1}^{D+F} \int_{\hat{x}_j^d}^{\hat{x}_j} \frac{\partial k_m(\boldsymbol{\chi}, \hat{\boldsymbol{x}}^i)}{\partial \chi_j}\bigg|_{\boldsymbol{\chi}=\boldsymbol{\tau}^j} d\chi_j \quad (4)$$

and we compute the partial derivatives $\partial k_m(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}}^d)/\partial \hat{x}_j$ in all state-action space dimensions $1 \leq j \leq D+F$.

We rewrite the sum in Eq. (3) by substituting Eq. (4) and the partial derivatives of the covariance functions. To find an upper bound for this sum, we estimate the occurring integrals by the product of integration interval length and an upper or lower mean value according to the sign of the respective summand. The necessary upper and lower mean values can be obtained via Riemannian upper and lower sums or the upper and lower bounds for the partial derivatives of k with respect to $\boldsymbol{x}$. Sorting the summands by products of integration interval lengths $(\hat{x}_j - \hat{x}_j^d)(\hat{x}_p - \hat{x}_j^d)$, this sum can be rewritten as a quadratic form. The entries of M can be chosen to form a symmetric matrix by making $M_{pj} = M_{jp}$ half of the coefficient of $(\hat{x}_j - \hat{x}_j^d)(\hat{x}_p - \hat{x}_p^d)$. ■

For any point in the state action space, we can find an upper bound for the distance of its prediction and the target point. Note that this distance estimated by Lemma 3 depends heavily on the eigenvalues of $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$. This fact is exploited to compute a grid S, which constitutes the first step of Algorithm 1. As M is constructed as a symmetric matrix, the eigenvalue problem is well conditioned, i.e., when M is perturbed, the change in the eigenvalues of M is at most as large as the perturbation.

Lemma 4 *Let $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^D$ and $M(\boldsymbol{x}, \boldsymbol{x}^d, \pi)$, B be as defined in Lemma 3. If $\|\boldsymbol{f}(\hat{\boldsymbol{x}}) - \boldsymbol{f}(\hat{\boldsymbol{x}}^d)\|_B < c\|\boldsymbol{x} - \boldsymbol{x}^d\|$ for $c < 1$, there exist Δ_j such that $\|\boldsymbol{f}(\hat{\boldsymbol{z}}) - \boldsymbol{f}(\hat{\boldsymbol{x}}^d)\|_B < c\|\boldsymbol{z} - \boldsymbol{x}^d\|$, for all $\boldsymbol{z}$ with $|\hat{z}_j - \hat{x}_j| < \Delta_j$, $1 \leq j \leq D+F$.*

Proof Solving for the eigenvalues of $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$, the set $Q_{\boldsymbol{x}} \coloneqq \{\boldsymbol{v} \in \mathbb{R}^{D+F} \mid (\boldsymbol{v} - \hat{\boldsymbol{x}}^d)^\intercal M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})(\boldsymbol{v} - \hat{\boldsymbol{x}}^d) < a\}$ can be determined. It is symmetric to the axes defined by the eigenvectors of $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$ and meets them at $\pm a^{-1}\sqrt{\lambda}$. For $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{\Delta}$ we want to ensure $\boldsymbol{z} \in Q_{\boldsymbol{z}}$, if $\boldsymbol{x} \in Q_{\boldsymbol{x}}$. Thus, we estimate how much the eigenvalues of $M(\boldsymbol{z}, \boldsymbol{x}^d, \boldsymbol{\pi})$ differ from those of $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$. As $M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})$ is symmetric, the eigenvalue problem has condition $\kappa(\lambda, M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})) = 1$ for any eigenvalue λ. Thus, $|\partial\lambda/\partial\hat{x}_j| \leq \|\partial M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})/\partial\hat{x}_j\|$. Computing $\|\partial M(\boldsymbol{x}, \boldsymbol{x}^d, \boldsymbol{\pi})/\partial\hat{x}_j\|$ or upper bounds for this expression allows solving for all Δ_j. ■

We are now able to compute a grid S, such that it is sufficient to check Eq. (2) for all grid points to retrieve a (continuous) stability region. While the grid width may become small, there is a lower bound for it. As the GP falls back to the prior far away from training data, the entries of M are bounded for bounded mean priors and, being continuous, Lipschitz. Thus, a lower bound exists.

Theorem 5 *The region X^c returned by Algorithm 1 is a stability region. All trajectories starting in X^c move closer to the desired point $\boldsymbol{x}^d$ in each step. Convergence to $\boldsymbol{x}^d$ is guaranteed for all points in X^c as $t \to \infty$.*

Proof We exploit Lemma 4 to compute a grid S as the first step in Algorithm 1. Employing Lemma 3, the algorithm checks for all points in S whether Eq. (2) holds, obtaining the sets S^+ and S^-, respectively. Lemma 4 ensures that Eq. (2) also holds for all points inside the polygon $\langle S^+ \rangle_{\boldsymbol{x}^d}$. For every point in $\langle S^+ \rangle_{\boldsymbol{x}^d}$, the next state is closer to the target point $\boldsymbol{x}^d$. Fitting a full metric ball $\mathcal{B}(\boldsymbol{x}^d)$ in $\langle S^+ \rangle_{\boldsymbol{x}^d}$, we recover an invariant set X^c. More precisely, the trajectories starting in X^c move closer to $\boldsymbol{x}^d$ with every time step. ■

Theorem 5 provides an asymptotic stability result for closed-loop control systems with dynamics given as the mean of a GP and can, thus, can be applied to a broad class of dynamics systems. However, the procedure is based on a grid in the state space and as a consequence suffers from the curse of dimensionality. This limitation is not surprising, as it is well known that estimating stability regions of high degree is NP-hard (Ahmadi et al., 2013). However, several approaches can help to scale the proposed method to higher dimensions. Instead of computing a global grid width $\boldsymbol{\Delta}$ as in Lemma 4, the proof can straightforwardly be adapted to compute local grid widths. Such local grid widths depend on the magnitude of the derivatives of the GP mean with respect to the inputs and, thus, can be substantially larger than the global one, e.g., when moving away from training data or where the dynamics is very slow. With this technique, the number of grid points can be greatly reduced. Furthermore, as we are constructing an inner approximation to the full stability region, the order of the grid points processed by Algorithm 1 can be modified to avoid redundant computations. When starting at the equilibrium point and expanding the considered grid points spherically according to the chosen distance metrics, computation can be stopped as soon as the first grid point is tested to lie outside the stability region. This first negative test point limits the stability region and all grid points with the same or greater distance to $\boldsymbol{x}^d$ need not be considered anymore.

3.4 Finite Time Horizons and Robustness

In this section, we will employ the proposed Algorithm 1 to derive results for finite time horizons and stability in the presence of disturbances.

3.4.1 Finite Time GP Mean Stability

Above, we proposed an algorithm to check asymptotic stability of the target $\boldsymbol{x}^d$ and to find a stability region, if one exists. In this section, we are interested in finding all starting states which deviate from the target by at most a given tolerance $\varepsilon > 0$ after $T < \infty$ timesteps. State space regions that fulfill this criterion were introduced in Definition 2 as (ε, T)-*stability regions*.

The main result of this section follows from the infinite time horizon results of Sec. 3.3. Recall that Algorithm 1 computes a state space region X^c and $\gamma < 1$, such that

$$\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| < \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\|$$

for all $\boldsymbol{x}^{(t)} \in X^c$. We can exploit this result to find a state space region $X^c_{\varepsilon,T}$, such that $\|\boldsymbol{x}^{(T)} - \boldsymbol{x}^d\| < \varepsilon$ for all $\boldsymbol{x}^{(0)} \in X^c_{\varepsilon,T}$.

Lemma 6 *Let $\varepsilon > 0$, $T < \infty$ and $X^c \subseteq X, \gamma < 1$ such that $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| < \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\|$ for all $t = 1, 2, \ldots$ and $\boldsymbol{x}^{(0)} \in X^c$. Then $\|\boldsymbol{x}^{(T)} - \boldsymbol{x}^d\| < \varepsilon$ for all $\boldsymbol{x}^{(0)} \in X^c$ with $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\| < \varepsilon / \gamma^T$.*

Proof Choosing $\boldsymbol{x}^{(0)} \in X^c$ and iteratively employing $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| < \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\|$ we get

$$\|\boldsymbol{x}^{(T)} - \boldsymbol{x}^d\| < \gamma^T \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\| \overset{!}{<} \varepsilon.$$

Solving for ε, we obtain $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\| < \varepsilon / \gamma^T$ and, thus, $X^c_{\varepsilon,T} \coloneqq \mathcal{B}_{\varepsilon/\gamma^T}(\boldsymbol{x}^d)$ is an (ε, T)-stability region. ■

Lemma 6 enables finite time horizon statements. For a given tolerance ε we can compute a region $X^c_{\varepsilon,T}$ such that $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| < \varepsilon$ for all $t \geq T$ if $\boldsymbol{x}^{(0)} \in X^c_{\varepsilon,T}$. Such stability statements are relevant for many applications as controllers are often designed for finite system runtimes.

3.4.2 Robustness of GP Mean Dynamics

In this section, we will analyze the behavior of closed-loop control systems with dynamics given as the mean of a GP when disturbances are present. More precisely, we assume that the next state is given as

$$\boldsymbol{x}^{(t+1)} = \boldsymbol{f}\left(\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)}\right) + \boldsymbol{r}^{(t)} \tag{5}$$

with the disturbance $\boldsymbol{r}^{(t)} < \infty$ at time t, that does not depend on the system state. We aim to find conditions for $\boldsymbol{r}^{(t)} < \infty$ such that $\boldsymbol{x}^{(t)}$ still converges to $\boldsymbol{x}^d$ as $t \to \infty$. Let $R^{(t)} \coloneqq \|\boldsymbol{r}^{(t)}\|$ be the magnitude of the disturbance at time t. Based on Equation (2) we derive an upper bound for $R^{(t)}$. Recall that Algorithm 1 returns a stability region X^c, which is a full metric ball and let ρ be the radius of this ball. Convergence to the target state is guaranteed, as long as the system does not leave X^c and $R^{(t)} \to 0$ as $t \to \infty$.

Theorem 7 *Let X^c be a full metric ball of radius ρ that is a stability region of the undisturbed closed-loop control system with GP mean dynamics. For $\boldsymbol{x}^{(0)} \in X^c$, the system in Eq. (5) converges to the target state $\boldsymbol{x}^d$, if*

$$R^{(t)} \leq \rho - \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| \tag{6}$$

for all t and there exists some T_0 such that

$$R^{(t)} < (c - \gamma) \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| \tag{7}$$

with a constant $c < 1$ holds for all $t \geq T_0$.

Proof Firstly, we will show that $\boldsymbol{x}^{(t+1)} \in X^c$, if $\boldsymbol{x}^{(t)} \in X^c$ and Eq. (6) holds. Employing Equation (5), we get

$$\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| = \|\boldsymbol{f}(\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)}) + \boldsymbol{r}^{(t)} - \boldsymbol{x}^d\| \leq \|\boldsymbol{f}(\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)}) - \boldsymbol{x}^d\| + R^{(t)} \tag{8}$$

and assuming $\boldsymbol{x}^{(t)} \in X^c$, it follows

$$\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| \leq \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| + R^{(t)}. \tag{9}$$

It follows $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| \leq \rho$, if $R^{(t)} \leq \rho - \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\|$ and, thus, $\boldsymbol{x}^{(t+1)} \in X^c$. By induction, $\boldsymbol{x}^{(t)} \in X^c$ for all $t > 0$, if $\boldsymbol{x}^{(0)} \in X^c$ and Equation (6) holds for all t.

If the disturbance $\boldsymbol{r}^{(t)}$ vanishes after a finite number of time steps, it is sufficient to ensure Equation (6) to show convergence to the target state. After this finite number of

disturbed steps, the system state will still lie in X^c and for all time steps to come, the dynamics will resemble the non-disturbed dynamics. Convergence is then guaranteed by Theorem 5.

Next, we will show that $\boldsymbol{x}^{(t)}$ converges to $\boldsymbol{x}^d$, if the disturbance $\boldsymbol{r}^{(t)}$ decreases sufficiently fast, as stated in Equation (7). For $\boldsymbol{x}^{(t)} \in X^c$, Equation (9) holds. To ensure convergence to $\boldsymbol{x}^d$, we aim to decrease the distance to the target state with every time step. Thus,

$$\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| \leq \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| + R^{(t)} \overset{!}{<} c \, \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| \tag{10}$$

for a constant $c < 1$. Solving for $R^{(t)}$, we obtain Eq. (7). ■

This theorem provides a criterion for the disturbance such that the closed-loop control system will still converge to the target state. However, in practice, Equations (6) and (7) may be difficult to verify, as they involve $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\|$ for all t. We derive a closed-form upper bound for the magnitude $R^{(t)}$ of the disturbance $\boldsymbol{r}^{(t)}$.

Lemma 8 *The Inequality (6) holds for all t, if*

$$R^{(t)} \leq \rho - \left(\gamma^t \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\| + \sum_{k=1}^{t-1} \gamma^k R^{(k)} \right). \tag{11}$$

The inequality (7) holds for $t \geq T_0$, if

$$R^{(t)} \leq (c^{t+1} - \gamma c^t) \|\boldsymbol{x}^{T_0} - \boldsymbol{x}^d\|. \tag{12}$$

Proof Both inequalities can be obtained by iteratively applying Eqs. (6) and (7).

$$\begin{aligned}
\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| &\leq \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| + R^{(t)} \\
&\leq \gamma \left(\gamma \|\boldsymbol{x}^{(t-1)} - \boldsymbol{x}^d\| + R^{(t-1)} \right) + R^{(t)} \\
&\leq \gamma^t \|\boldsymbol{x}^{(0)} - \boldsymbol{x}^d\| + \sum_{k=1}^{t-1} \gamma^k R^{(k)} + R^{(t)} \overset{!}{<} \rho
\end{aligned}$$

and analogously

$$\begin{aligned}
\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^d\| &\leq \gamma \|\boldsymbol{x}^{(t)} - \boldsymbol{x}^d\| + R^{(t)} \\
&< \gamma (c \|\boldsymbol{x}^{(t-1)} - \boldsymbol{x}^d\|) + R^{(t)} \\
&< \gamma c^t \|\boldsymbol{x}^{(T_0)} - \boldsymbol{x}^d\| + R^{(t)} \overset{!}{<} c^{t+1} \|\boldsymbol{x}^{(T_0)} - \boldsymbol{x}^d\|.
\end{aligned}$$

■

Employing Lemma 8, we can analyze stability for disturbed system trajectories, if the magnitude of the disturbance (or an upper bound for the magnitude) is known for every time step.

4. Stochastic Stability of GP Dynamics

In this section, we study closed-loop systems with dynamics given as a full GP distribution. For any query point $\boldsymbol{x}^{(t)}$ a GP predicts the next state $\boldsymbol{x}^{(t+1)}$ to be normally distributed. If, however, $\boldsymbol{x}^{(t)}$ is not a point, but a distribution, the integral

$$p\left(\boldsymbol{x}^{(t+1)}\right) = \int_{\mathbb{R}^D} p\left(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}\right) p\left(\boldsymbol{x}^{(t)}\right) d\boldsymbol{x}^{(t)} \tag{13}$$

determines the next state distribution. Note that $p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)})$ is Gaussian with respect to $\boldsymbol{x}^{(t+1)}$. The next state distribution $p(\boldsymbol{x}^{(t+1)})$, however, is not Gaussian, even if $p(\boldsymbol{x}^{(t)})$ is. Generally, $p(\boldsymbol{x}^{(t+1)})$ is analytically intractable and only approximations, e.g., via moment matching (Quiñonero-Candela et al., 2003) or linearization (Ko and Fox, 2008), can be computed. These methods suffer from severe inaccuracies in many cases, e.g., they cannot handle distributions with multiple modes. Also, no pointwise approximation error bounds are available. Thus, these methods are unsuitable for stability analysis. In this paper, we propose numerical quadrature (Sec. 4.2.1) to approximate $p(\boldsymbol{x}^{(t+1)})$, instead. This technique yields significantly better results, e.g., it can handle distributions with multiple modes. In addition, error analysis is readily available (Wasowicz, 2006; Masjed-Jamei, 2014) and can be employed to derive stability guarantees for a finite time horizon.

We will also analyze the closed-loop system behaviour for infinite time horizons. The proposed numerical quadrature approximation converges to a stationary limiting distribution. Unfortunately, the error bounds based on quadrature error analysis do not apply when infinitely many timesteps are taken. However, we will show that closed-loop control systems with GP dynamics expose the same infinite time horizon behavior – they converge to a stationary distribution.

In the following, we will discuss *finite time stochastic stability*, introduce an algorithm to find a stability region based on numerical quadrature, and prove its correctness. Subsequently, we will elaborate on the construction of good quadrature rules for multi-step-ahead predictions. Finally, we will analyze the system behavior for infinite time horizons.

4.1 Stability Notion

Consider a deterministic system, that is locally, but not globally, asymptotically stable. Adding noise to the system may render the target point unstable. Especially when noise is unbounded (e.g., Gaussian), all trajectories will eventually leave any ball around the target point with probability one. For this reason, in the study of SDEs, other stability notions than Lyapunov's, are common. In particular, bounding the probability to drift away from the target over a finite time T is desirable, as in the following definition (Kushner, 1966).

Definition 9 *Let Q_1, Q_2 be subsets of the state space X, with Q_2 open and $Q_1 \subset Q_2 \subset X$. The system is* finite time stable *with respect to $Q_1, Q_2, 1-\lambda, T$, if $\boldsymbol{x}^{(0)} \in Q_1$ implies $P\{\boldsymbol{x}^{(t)} \in Q_2\} \geq 1 - \lambda$ for all $t \leq T$.*

However, we are interested in finding a set Q_s of initial conditions, such that the goal Q is reached within time T with a desired probability (cf. Steinhardt and Tedrake 2012).

Definition 10 *The set Q_s is a stability region with respect to the target region Q, time horizon T and success probability $1-\lambda$, if $P\{\boldsymbol{x}^{(T)} \in Q\} \geq 1 - \lambda$ holds for all $\boldsymbol{x}^{(0)} \in Q_s$ with $\lambda > 0$ and the target region $Q \subset X$.*

Algorithm 2 Stability region for GP dynamics

Input: dynamics GP $\boldsymbol{f}$, control policy $\boldsymbol{\pi}$, time horizon T, target region Q, approximation error tolerance e_{tol}, desired success probability $1-\lambda$

Output: stability region X^c

Construct grid S with Lemma 12
$S^+ \leftarrow \emptyset$
Compute quadrature $\boldsymbol{w}, \boldsymbol{\xi}$ based on Eq. (18) and Lemma 11, such that $\|\epsilon_T\|_{\mathcal{C}(X)} < e_{\text{tol}}$
$\boldsymbol{\phi} \leftarrow (\boldsymbol{f}(\boldsymbol{\xi}_1, \boldsymbol{\pi}(\boldsymbol{\xi}_1)), \ldots, \boldsymbol{f}(\boldsymbol{\xi}_N, \boldsymbol{\pi}(\boldsymbol{\xi}_N)))^\intercal$
$\boldsymbol{m} \leftarrow (\int_Q \phi_1(\boldsymbol{x})d\boldsymbol{x}, \ldots, \int_Q \phi_N(\boldsymbol{x})d\boldsymbol{x})^\intercal$
for $\boldsymbol{x} \in S$ **do**
  $p(\boldsymbol{x}^{(1)}) \leftarrow \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\pi}(\boldsymbol{x}))$
  $\boldsymbol{x}^{(T)} \leftarrow \boldsymbol{\alpha}^{(T)}\boldsymbol{\phi}$
  $P_{min}\{\boldsymbol{x}^{(T)} \in Q\} \leftarrow \boldsymbol{\alpha}^{(T)}\boldsymbol{m} - \text{vol}(Q)\|\epsilon_T\|_{\mathcal{C}(X)}$
  if $P_{min}\{\boldsymbol{x}^{(T)} \in Q\} > 1-\lambda$ **then** $S^+ \leftarrow S^+ \sqcup \{\boldsymbol{x}\}$ **fi**
od
return $X^c \leftarrow \langle S^+ \rangle$

This definition focuses on reaching a target Q after time T, whereas Definition 9 bounds the exit probability from a region Q_2 within time $0 \leq t \leq T$. The methods proposed in this paper can be employed to analyze stability in the sense of both definitions. In the following, we will present an algorithm to find a stability region according to Definition 10.

4.2 Algorithm Sketch

We will now discuss how to find a stability region as in Definition 10 for a closed-loop system with GP dynamics. To analyze system behavior, the capability to compute next state distributions is crucial. As discussed previously, Eq. (13) is analytically intractable and approximation methods must be employed. We propose numerical quadrature to approximately propagate distributions. We will show that numerical quadrature approximates state distributions as Gaussian mixture models. Fortunately, computation of multi-step-ahead predictions becomes convenient and fast even for long time horizons. However, relying on approximations of the state distribution is not sufficient for stability guarantees. The error introduced by approximation and error propagation must be bounded to recover reliable statements on the probability for $\boldsymbol{x}^{(T)}$ to be in a set Q. This error bound $\epsilon_t(\boldsymbol{x})$ at time t can be obtained following one of the available quadrature error analyses, e.g., by Masjed-Jamei (2014).

With numerical quadrature and quadrature error analysis, we can compute a lower bound for the *success probability*, i.e., the probability for $\boldsymbol{x}^{(T)}$ to be in the target set Q. A priori, this enables checking success probabilities for a finite set of points. Fortunately, as in the case of GP mean dynamics, the statement can be generalized to continuous regions. Our algorithm constructs a grid in the state space and computes success probabilities for all grid points. We prove that a stability region can be inferred from these grid point results.

In the following, we elaborate on numerical quadrature as approximate inference method and, studying quadrature error propagation, prove the correctness of the proposed approach. The presented tool is outlined in Algorithm 2.

4.2.1 Numerical Quadrature Uncertainty Propagation

In most applications, especially when the states have physical interpretations, the state space is bounded. For this reason, we assume $X = [a_1, b_1] \times \cdots \times [a_D, b_D]$ and solve

$$F[p(\boldsymbol{x}^{(t)})] := \int_X p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}) p(\boldsymbol{x}^{(t)}) d\boldsymbol{x}^{(t)}. \tag{14}$$

We propose numerical quadrature to approximate this integral. We choose a composed Gaussian product quadrature rule, which will be detailed in Section 4.4. For the rest of this section, it is sufficient to note that our quadrature rule provides a set of evaluation points $\mathbb{X}$ and positive weights w_n for all nodes $\boldsymbol{\xi}^n \in \mathbb{X}$. Integral (14) is then approximated by

$$F[p(\boldsymbol{x}^{(t)})] \approx \sum_{\boldsymbol{\xi}^n \in \mathbb{X}} w_n p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n) p(\boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n), \tag{15}$$

resulting in a weighted sum of Gaussian distributions. The approximate state distribution at time $t+1$ can be given by

$$p(\boldsymbol{x}^{(t+1)}) \approx \boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(t+1)} \tag{16}$$

with $\alpha_n^{(t+1)} := w_n p(\boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n), \phi_n(\boldsymbol{x}) := p(\boldsymbol{x}^{(t+1)} = \boldsymbol{x} \mid \boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n)$. Note that the Gaussian basis functions $\phi_n(\boldsymbol{x})$ do not change over time, so the state distribution at time t is represented by the weight vector $\boldsymbol{\alpha}^{(t)}$. To propagate any distribution multiple steps through the GP, the basis functions ϕ_n must be calculated only once and the task reduces to sequential updates of the weight vector $\boldsymbol{\alpha}$. As $p(\boldsymbol{x}^{(t)}) \approx \boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(t)}$, the weight vector $\boldsymbol{\alpha}^{(t+1)}$ is given by

$$\boldsymbol{\alpha}^{(t+1)} = \text{diag}(\boldsymbol{w})\Phi\boldsymbol{\alpha}^{(t)} = (\text{diag}(\boldsymbol{w})\Phi)^t \boldsymbol{\alpha}^{(1)} \tag{17}$$

with the matrix Φ, $\Phi_{i\,j} = \phi_j(\boldsymbol{\xi}^i)$ with $1 \leq i, j \leq n$, which contains the basis function values at all grid points. In practice, it is helpful to normalize the matrix $\text{diag}(\boldsymbol{w})\Phi$ such that each column sums to 1. In this case, unit vectors will be mapped to unit vectors. This ensures that our approximate state distribution is in fact a probability density, i.e., integrates to 1.

Our algorithm aims to find a stability region Q_s, i.e., a region where the success probability is at least $1-\lambda$ for a given time horizon T, target region Q, and $\lambda > 0$. Computing $\boldsymbol{m} = \int_Q \boldsymbol{\phi}(\boldsymbol{x})d\boldsymbol{x}$, the probability for $\boldsymbol{x}^{(T)}$ to be in Q is approximately $\boldsymbol{m}^\intercal \boldsymbol{\alpha}^{(T)}$.

4.3 Correctness of the Algorithm

We will now show that the region returned by Algorithm 2 is a stability region as in Definition 10. In Section 4.2.1, we proposed numerical quadrature to approximate GP predictions when the input is a distribution. However, to obtain stability guarantees it is not sufficient to consider approximate solutions. For a lower bound on the success probability, additional knowledge about the approximation error and how it is propagated through the dynamics GP is essential.

When approximate inference steps are cascaded for multi-step-ahead predictions, errors are introduced and propagated through the dynamics. Typically, error propagation fortifies initial errors with every time step. The difference of approximation and true distribution after a finite time is bounded. Fortunately, numerical quadrature allows to control this error bound by refining the used quadrature rule.

The derivation of an upper bound for the approximation error after a finite number of time steps relies heavily on quadrature error analysis, i.e., upper bounds for the errors that are made in one time step when numerical quadrature is employed. Several error analyses for numerical quadrature are available (Masjed-Jamei, 2014; Davis et al., 2014; Wasowicz, 2006; Evans, 1994) with the classic analyses relying on higher order derivatives of the function to be integrated. We will employ the error analysis by Masjed-Jamei (2014) that requires first order derivatives only. More precisely (Masjed-Jamei 2014, Theorem 2.1), for a differentiable function f with $\nu_1(x) \leq f'(x) \leq \nu_2(x)$ with continuous functions ν_1, ν_2 the quadrature error is upper and lower bounded

$$m \leq \sum_{i=1}^{N} w_i f(\xi_i) - \int_a^b f(x)dx \leq M \tag{18}$$

with the bounds

$$m = \sum_{i=0}^{N} \int_{\xi_i}^{\xi_{i+1}} \left(\nu_1(t) \frac{v_i(t) + |v_i(t)|}{2} + \nu_2(t) \frac{v_i(t) - |v_i(t)|}{2} \right) dt$$

$$M = \sum_{i=0}^{N} \int_{\xi_i}^{\xi_{i+1}} \left(\nu_1(t) \frac{v_i(t) - |v_i(t)|}{2} + \nu_2(t) \frac{v_i(t) + |v_i(t)|}{2} \right) dt$$

for $\xi_0 = a$, $\xi_{N+1} = b$ and $v_i = t - \left(a + \sum_{j=1}^{i} w_i\right)$, $i = 0, \ldots, n$. In the following, we employ this result to derive an error bound for the pointwise approximation error of the proposed multi-step-ahead predictions based on numerical quadrature.

Lemma 11 *Let $\boldsymbol{x}^{(0)} \in X$ and ϵ_T be the pointwise approximation error $\epsilon_T(\boldsymbol{x}) = p(\boldsymbol{x}^{(T)} = \boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{x})^\intercal \boldsymbol{\alpha}^{(T)}$ at time T. There exists an upper bound for $\|\epsilon_T\|_{\mathcal{C}(X)}$ that can be controlled by the choice of quadrature rule.*

Proof The statement obviously holds for $T = 0$ and for $T = 1$, as $p(\boldsymbol{x}^{(1)})$ is Gaussian and computed exactly. If $T \geq 2$, an error is introduced as $p(\boldsymbol{x}^{(t)})$ cannot be computed analytically for $t \geq 2$. Employing numerical quadrature to compute $p(\boldsymbol{x}^{(2)})$, we obtain $p(\boldsymbol{x}^{(2)}) = \boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(2)} + \epsilon_2$ with our approximation $\boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(2)}$ and the unknown initial quadrature error $\epsilon_2(\boldsymbol{x})$. However, the error analysis by Masjed-Jamei (2014, Theorem 2.1) gives us a bound for $\|\epsilon_2\|_{\mathcal{C}(X)} := \max_{\boldsymbol{x} \in X} |\epsilon_2(\boldsymbol{x})|$. When propagating further, an unknown, bounded error term and an approximation term must be handled. For the approximation term, we get $F[\boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(t)}] = \boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(t+1)} + \boldsymbol{\epsilon}^\intercal \boldsymbol{\alpha}^{(t)}$ with quadrature error bounds $\boldsymbol{\epsilon} = (\epsilon_i(\boldsymbol{x}))_i$ for the integrals (14), when setting $p(\boldsymbol{x}^{(t)}) = \phi_i$. The error term $\epsilon_t(\boldsymbol{x})$ is propagated to $F[\epsilon_t]$ with $\max_{\boldsymbol{x} \in X} |F[\epsilon_t]| \leq C \|\epsilon_t\|_{\mathcal{C}(X)}$ and $C = \max_{\boldsymbol{x} \in X} \int p(\boldsymbol{x}^{(t+1)} \,|\, \boldsymbol{x}^{(t)}) d\boldsymbol{x}^{(t)}$. Finally, the error at time T is bounded by $\|\epsilon_T\|_{\mathcal{C}(X)} \leq \|\boldsymbol{\epsilon}\|^\intercal \boldsymbol{\alpha}^{(T-1)} + \sum_{t=2}^{T-2} C^t \|\boldsymbol{\epsilon}\|^\intercal \boldsymbol{\alpha}^{(t)} + C^{t-1} \|\epsilon_2\|$, where $\boldsymbol{\epsilon}$ and ϵ_2 can

be made arbitrarily small by refining the quadrature rule. ■

Lemma 11 theoretically enables stability guarantees for finite time horizons T. However, as with any approximation of the distribution (14), this error bound grows exponentially with T, while quadrature error decreases polynomially with function evaluations. This drawback limits real-word application of Lemma 11 to small T, but we found that typically, approximation behaves far better than the worst-case bound.

For any starting point, we can now compute the approximate state distribution at time T and a lower bound on the success probability, i.e., the probability for $\boldsymbol{x}^{(T)}$ to be in the target region Q. It remains to show that it is sufficient to compute the success probability for a discrete set of starting states and generalize to the underlying continuous state space region.

Lemma 12 *Let $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^D$ be starting states and $T \in \mathbb{N}$. If $P\{\boldsymbol{x}^{(T)} \in Q\} \geq 1 - \lambda$, there exist Δ_j such that $P\{\boldsymbol{z}^{(T)} \in Q\} \geq 1 - \lambda$ for all $\boldsymbol{z}$ with $|z_j - x_j| < \Delta_j$, $1 \leq j \leq D$.*

Proof Firstly, we note that the absolute error bound does not depend on the starting state. The state distribution $p(\boldsymbol{x}^{(1)})$ is the GP prediction at the query point $\boldsymbol{x}^{(0)}$. Thus, for $T = 1$ the claim follows from Lemma 4. When $t \geq 2$, only an approximation of the state distribution is known. The mixture model weights at time $t = 2$ depend only on the values of $p(\boldsymbol{x}^{(1)})$ at the set of evaluation points $\mathbb{X}$. Thus, the lemma follows for $T = 2$. If $t > 2$, the difference of the approximate distributions for the starting points $\boldsymbol{x}$ and $\boldsymbol{z}$ is

$$\boldsymbol{\phi}^\intercal (\boldsymbol{\alpha}_{\boldsymbol{x}}^{(t)} - \boldsymbol{\alpha}_{\boldsymbol{z}}^{(t)}) = \boldsymbol{\phi}^\intercal (\mathrm{diag}(\boldsymbol{w}) \Phi)^{t-2} (\boldsymbol{\alpha}_{\boldsymbol{x}}^{(2)} - \boldsymbol{\alpha}_{\boldsymbol{z}}^{(2)}).$$

Thus, the difference in approximate probability mass is linear in $(\boldsymbol{\alpha}_{\boldsymbol{x}}^{(2)} - \boldsymbol{\alpha}_{\boldsymbol{z}}^{(2)})$. This fact concludes the proof. ■

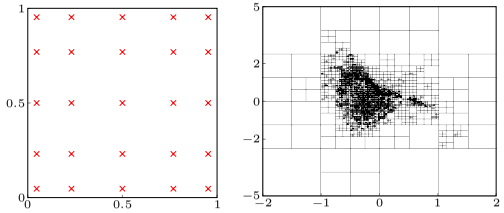
Finally, we are able to prove the main result of this section, the correctness of Algorithm 2.

Theorem 13 *The region X^c returned by Algorithm 2 is a stability region in the sense of Definition 10. For all starting points $\boldsymbol{x}^{(0)} \in X^c$, the probability mass in the target region Q at time T is at least $1 - \lambda$.*

Proof For the first step in Algorithm 2 we exploit Lemma 12 to construct a grid S. For S, stability follows for the region around the grid points S^+ with success probability greater than $1-\lambda$. Secondly, employing Lemma 11 we determine a quadrature rule that ensures the pointwise error at time T to be at most e_{tol}. Finally, we compute approximate success probabilities for all grid points, as in Sec. 4.2.1. Subtracting the maximum error mass $e_{\mathrm{tol}} \,\mathrm{vol}(Q)$ in target region Q, we obtain all grid points S^+, which have a success probability of at least $1-\lambda$. ■

4.4 Construction of Numerical Quadratures for Uncertainty Propagation

In the sections above, we introduced numerical quadrature to approximate GP predictions at uncertain inputs and showed that this approximate inference method can be used to derive finite time horizon stability guarantees. The presented theoretical results are valid for any choice of quadrature rule that allows for quadrature error analysis. However, in practice, the



(a) Product Quadrature (b) Mountain Car

Figure 4: Construction of suitable quadrature rules. Plot (a) shows the nodes of a Gaussian product quadrature rule on the unit square. Here, the quadrature rule for the unit square was constructed as an outer product of the Gaussian quadrature rule with 5 nodes. The state space partition for the mountain car system obtained with Algorithm 3 is shown in (b). For each rectangle in this partition, a Gaussian product quadrature such as (a) is employed.

choice of a suitable quadrature rule is crucial to the applicability of the proposed algorithm. Thus, in the following, we will elaborate on how to find "good" quadrature rules for a given GP dynamics and control policy.

Gaussian product quadrature extends univariate Gaussian quadrature to a multivariate rule using a product grid of evaluation points. This construction has some desirable properties, such as positive quadrature weights and readily available quadrature nodes. However, being an outer product of one-dimensional rules, it suffers from the curse of dimensionality. In this paper, we apply numerical quadrature to integrals as in Equation (14). Typically, the system trajectories are not uniformly spread over the state space. Instead, they are concentrated in a significantly smaller region. We exploit this observation to improve the efficiency of product grid quadrature and cope with the curse of dimensionality. For this purpose, we partition the state space $X = X_1 \sqcup \cdots \sqcup X_L$ and apply a Gaussian product quadrature to each obtained subregion X_l. The next state distribution, cf. Equation (14), can be written as

$$F[p(\boldsymbol{x}^{(t)})] \coloneqq \int_X p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}) p(\boldsymbol{x}^{(t)}) d\boldsymbol{x}^{(t)} = \sum_{l=1}^{L} \int_{X_l} p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)}) p(\boldsymbol{x}^{(t)}) d\boldsymbol{x}^{(t)}$$

and, applying a numerical quadrature rule with nodes $\mathbb{X}_l$ for each integral, we get

$$F[p(\boldsymbol{x}^{(t)})] \approx \sum_{l=1}^{L} \sum_{\boldsymbol{\xi}^n \in \mathbb{X}_l} w_{nl} p(\boldsymbol{x}^{(t+1)} \mid \boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n) p(\boldsymbol{x}^{(t)} = \boldsymbol{\xi}^n).$$

Setting $\mathbb{X} = \mathbb{X}_1 \sqcup \cdots \sqcup \mathbb{X}_L$, we recover Equation (15). Thus, composed quadrature rules can be handled just as a single Gaussian product rule. We exploit this to construct quadrature rules which are efficient for the computation of next state distributions for our particular GP

Algorithm 3 Construction of composed quadrature rules

Input: dynamics GP $\boldsymbol{f}$: $\left(\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)}\right) \mapsto \boldsymbol{x}^{(t+1)}$, control policy $\boldsymbol{\pi}$: $\boldsymbol{x} \mapsto \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{x})$ with parameters $\boldsymbol{\theta}$, state space X, maximum partition size $L_{\max}$
Output: composed quadrature rule with nodes $\mathbb{X}$ and weight vector $\boldsymbol{w}$

```
Initialize partition X = X_1 ⊔ ··· ⊔ X_s and quadrature 𝕏 = 𝕏_1 ⊔ ··· ⊔ 𝕏_s with s < L_max
while s < L_max do
    Sample starting state x^(0), policy parameters θ*
    Compute mean rollout τ_0 = x^(0), ..., τ_H with dynamics GP f and policy π_θ*
    for l = 1, ..., s do
        if vol(X_l) / (|𝕏_l| min_{τ_i ∈ X_l} Var(τ_i)) < 1 then subdivide X_l, add nodes to 𝕏 fi
    od
od
return quadrature nodes 𝕏 and weights w
```

dynamics model and controller. In other words, we aim to find a partition $X = X_1 \sqcup \cdots \sqcup X_L$ of the state space, such that integral (14) is approximated well by the resulting quadrature rule. For this purpose, we maintain a partition of the state space, sample mean trajectories $\boldsymbol{\tau} = \boldsymbol{\tau}_0, \ldots, \boldsymbol{\tau}_H$ and subdivide regions X_l that were visited and fulfill a certain criterion. More precisely, we subdivide X_l if X_l does not contain enough quadrature nodes to integrate predictive distributions from the dynamics GP well. To estimate whether X_l should be divided, we introduce

$$\rho_l(\boldsymbol{\tau}) \coloneqq \frac{\operatorname{vol}(X_l)}{|\mathbb{X}_l| \min_{\boldsymbol{\tau}_i \in X_l} \operatorname{Var}(\boldsymbol{\tau}_i)}, \tag{19}$$

where $\operatorname{Var}(\boldsymbol{\tau}_i)$ denotes the variance of the GP predictive distribution at the point $\boldsymbol{\tau}_i$, and subdivide X_l if $\rho_l(\boldsymbol{\tau})$ is greater than 1. This criterion relates the higher order derivatives of GP predictions which fall inside X_l with the quadrature node density in X_l. Constructing a composed quadrature rule with this approach will concentrate most quadrature nodes in state space regions that are visited frequently when following system trajectories. Algorithm 3 summarizes our approach and Figure 4 illustrates the constructed quadrature rules.

4.5 GP Dynamics with Infinite Time Horizons

In Section 4.2, we proposed numerical quadrature to approximate next state distributions for closed-loop control systems, where the dynamics is given as a full Gaussian process. For this approximate inference, pointwise error bounds are available. Based on this error analysis, we were able to derive stability guarantees for finite time horizons. In this section, we analyze the case when the time horizon is infinite. Unfortunately, error bounds as in Section 4.3 cannot be given in this case. As we will see, the numerical quadrature approximation converges to a stationary limit distribution, i.e., this distribution does not depend on the starting state, as t goes to infinity. As the quadrature error cannot be bounded in this case, there is no guarantee that the system dynamics will converge to the same distribution. However, we will show that closed-loop control systems with GP dynamics also converge to a stationary distribution as $t \to \infty$ for many choices of the prior. We will first analyze

the behavior of the proposed numerical quadrature approximation and subsequently study asymptotic behavior of closed-loop control systems with GP dynamics.

4.5.1 Asymptotic Behavior of NQ Approximated GP Dynamics

In this section we study the properties of numerical quadrature as approximate inference when the time horizon is infinite. Recall that with numerical quadrature the state distribution is approximated by

$$p(\boldsymbol{x}^{(t)}) \approx \boldsymbol{\phi}^\intercal \boldsymbol{\alpha}^{(t)} \tag{16 revisited}$$

with Gaussian basis functions $\boldsymbol{\phi} = (\phi_1, \dots, \phi_N)^\intercal$, that do not change over time, and the weight vector $\boldsymbol{\alpha}^{(t)}$ at time t. As stated in Equation (17), the weight vector $\boldsymbol{\alpha}^{(t)}$ is updated by

$$\boldsymbol{\alpha}^{(t+1)} = \operatorname{diag}(\boldsymbol{w})\Phi\boldsymbol{\alpha}^{(t)} \tag{17 revisited}$$

with the positive matrix $\operatorname{diag}(\boldsymbol{w})\Phi$ as in Section 4.2.1. More precisely, $\operatorname{diag}(\boldsymbol{w})\Phi$ is a left stochastic matrix. Thus, any column vector $\boldsymbol{\beta}$ with norm $\|\boldsymbol{\beta}\|_1 = 1$ is mapped to a unit vector, i.e. $\|\operatorname{diag}(\boldsymbol{w})\Phi\boldsymbol{\beta}\|_1 = 1$. In this case, Equation (17) represents a power iteration which leads to the following theorem.

Theorem 14 *Let $(\mathbb{X}, \boldsymbol{w})$ be a quadrature rule with positive weights $\boldsymbol{w}$ on the state space X and $\boldsymbol{f}\colon (\boldsymbol{x}^{(t)}, \boldsymbol{u}^{(t)}) \mapsto \boldsymbol{x}^{(t+1)}$ be a dynamics GP. The approximate state distribution $p(\boldsymbol{x}^{(t)}) \approx \boldsymbol{\phi}^\intercal\boldsymbol{\alpha}^{(t)}$ obtained by cascading approximate one step ahead predictions with the quadrature $(\mathbb{X}, \boldsymbol{w})$ converges to a unique stationary distribution, independent of the starting distribution, as $t \to \infty$.*

Proof Consider Equation (17). The matrix $\operatorname{diag}(\boldsymbol{w})\Phi$ has norm one, as each of its columns sums to one. Thus, as the starting weights $\boldsymbol{\alpha}^{(0)}$ form a stochastic vector, Equation (17) takes the form of a power iteration, see (Golub and Van Loan, 2013, Section 7.3). By the Perron-Frobenius theorem the largest magnitude eigenvalue λ_1 of $\operatorname{diag}(\boldsymbol{w})\Phi$ is positive and simple. More precisely, the eigenvalue of $\operatorname{diag}(\boldsymbol{w})\Phi$ with the highest absolute value is $\lambda_1 = 1$. The corresponding dominant eigenvector $\boldsymbol{v}_1$ is positive. No other eigenvectors of $\operatorname{diag}(\boldsymbol{w})\Phi$ are non-negative. Thus, the power iteration in Equation (17) converges to the dominant eigenvector, if $\boldsymbol{\alpha}^{(0)}$ has a nonzero projection on the eigenspace of $\lambda_1 = 1$. As $\boldsymbol{\alpha}^{(0)}$ is a non-negative unit vector and $\boldsymbol{v}_1$ is positive, the projection to the dominant eigenspace is ${\boldsymbol{v}_1}^\intercal\boldsymbol{\alpha}^{(0)} > 0$. Thus, the numerical quadrature approximation of $p(\boldsymbol{x}^{(t)})$ converges to $\boldsymbol{\phi}^\intercal\boldsymbol{v}_1$. ■

This theorem shows that the numerical quadrature approximation to the system state converges to a stationary distribution. The limiting distribution corresponds to the dominant eigenvector of the iteration matrix $\operatorname{diag}(\boldsymbol{w})\Phi$ and does not depend on the starting state. However, no bound for the approximation error can be obtained for the limit $t \to \infty$. Thus, in contrast to finite time horizon analysis, no statement about the true system state follows from this result. In the following, we will study asymptotic behavior of closed-loop control systems with GP dynamics to see whether the (exact) system state converges to a limiting distribution as well.

4.5.2 Asymptotic Behavior of GP Dynamics

We have seen that employing the numerical quadrature approximation for next state distributions, closed-loop control systems with GP dynamics converge to a limiting distribution.

In this section, we will show that these closed-loop control systems expose the same behavior even if no approximations are employed. More precisely, a closed-loop control system with dynamics given as a GP will finally converge to a limiting distribution that is independent of the starting state for many common choices of the prior. To obtain this result, we apply Markov Chain theory as in (Meyn and Tweedie, 2009).

Consider a Markov chain $\boldsymbol{\Sigma} = \{\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \dots\}$ with the random variables $\boldsymbol{\Sigma}_t$ taking values in $\mathbb{R}^D$ and a *Markov kernel* $P\colon \mathbb{R}^D \times \mathcal{B}(\mathbb{R}^D) \to \mathbb{R}$, i.e., $P(\cdot, A)$ is a non-negative measurable function on $\mathbb{R}^D$, $P(\boldsymbol{x}, \cdot)$ is a probability measure on $\mathcal{B}(\mathbb{R}^D)$ and

$$\mathbb{P}_\mu[\boldsymbol{\Sigma}_0 \in A_0, \cdots, \boldsymbol{\Sigma}_n \in A_n] = \int_{\boldsymbol{y}_0 \in A_0} \cdots \int_{\boldsymbol{y}_{n-1} \in A_{n-1}} \mu(d\boldsymbol{y}_0) P(\boldsymbol{y}_0, d\boldsymbol{y}_1) \cdots P(\boldsymbol{y}_{n-1}, A_n) \tag{20}$$

for any initial distribution μ and $n \in \mathbb{N}$. We will employ some definitions and results from (Meyn and Tweedie, 2009) to prove our claim.

Definition 15 *The following definitions will help describing Markov chains and the sets they (re-)enter.*

1. *For $A \in \mathcal{B}(\mathbb{R}^D)$ and $\Sigma_0 \in A$, the* first return time *τ_A is given as*

   $$\tau_A \coloneqq \min\{n \geq 1 \colon \boldsymbol{\Sigma}_n \in A\}.$$

   We also define the return time probability *$L(\boldsymbol{x}, A) \coloneqq \mathbb{P}_{\boldsymbol{x}}(\tau_A < \infty)$.*

2. *We call $\boldsymbol{\Sigma}$ φ-*irreducible*, if there exists a measure φ on $\mathcal{B}(\mathbb{R}^D)$, such that $\varphi(A) > 0$ implies $L(\boldsymbol{x}, A) > 0$ for all $\boldsymbol{x} \in X$ and $A \in \mathcal{B}(\mathbb{R}^D)$. ψ is a* maximal irreducibility measure*, if $\boldsymbol{\Sigma}$ is ψ-irreducible and $\varphi \prec \psi$, i.e., $\psi(A) = 0 \Rightarrow \varphi(A) = 0$, for all irreducibility measures φ of $\boldsymbol{\Sigma}$.*

3. *We say that the set $C \in \mathcal{B}(\mathbb{R}^D)$ is* small*, if there exists a probability measure ν, such that $P^n(\boldsymbol{x}, A) \geq \delta\nu(A)$ with some integer n, $\delta > 0$ for all $\boldsymbol{x} \in C$ and all $A \in \mathcal{B}(\mathbb{R}^D)$.*

For finite or countable state spaces, a Markov kernel defines the probability to move from one state to another. The state space partitions into *communication classes*, such that all states in one class can be reached with positive probability from any other state in the same class. If there is only one communication class, the Markov chain is called irreducible – its analysis cannot be further reduced to the consideration of multiple chains with smaller state spaces. The communication structure has an immense effect on the long term behavior of a Markov chain. However, when considering general state spaces (e.g., continuous as in our case), the probability to reach one single state will be zero. The concept of communication is thus generalized to Borel sets via irreducibility measures. Every irreducible Markov chain has maximum irreducibility measures, see Theorem 16. While these are not unique, all maximum irreducibility measures of a Markov chain are equivalent, i.e., they have the same null sets. Small sets play an important role in the analysis of Markov chains on general state spaces, as they behave analogously to single states in discrete state spaces. Sets which consist of one single point, are small. However, it can be shown that for irreducible Markov chains, any set that will be entered with positive probability contains a non-trivial small set (Meyn and Tweedie, 2009, Theorem 5.2.2).

Theorem 16 *Let $\mathbf{\Sigma}$ be φ-irreducible. Then there exists a maximal irreducibility measure ψ, which is equivalent to*

$$\psi'(A) \coloneqq \int_X \varphi(d\boldsymbol{y}) K_{a\frac{1}{2}}(\boldsymbol{y}, A)$$

with the transition kernel $K_{a\frac{1}{2}}(\boldsymbol{y}, A) = \sum_{n=1}^{\infty} P^n(\boldsymbol{x}, A) 2^{-(n+1)}$.

Proof The proof is given in (Meyn and Tweedie, 2009, Proposition 4.2.2). ■

We aim to analyze the asymptotic behavior of Markov chains that are defined by our closed-loop control structure with GP dynamics. One type of asymptotic behavior that is well-known from Markov chains on discrete state spaces is periodicity. The notion of periodicity can be extended to Markov chains on general state spaces as follows.

Definition 17 *Let $\mathbf{\Sigma}$ be irreducible with maximal irreducibility measure ψ. The* period *of $\mathbf{\Sigma}$ is the largest integer $d > 0$, such that there exist non-empty, pairwise disjoint sets $X_1, \cdots, X_d \subseteq \mathbb{R}^D$ with*

$$P(\boldsymbol{x}, X_{i+1}) = 1 \text{ for all } \boldsymbol{x} \in X_i, 1 \leq i \leq d-1 \text{ and } P(\boldsymbol{x}, X_1) = 1 \text{ for } \boldsymbol{x} \in X_d$$

and

$$\psi\left(\bigcup_{i=1}^{d} X_i\right) = 1.$$

If $d = 1$ we call $\mathbf{\Sigma}$ aperiodic*, if $d > 1$ the Markov chain $\mathbf{\Sigma}$ is* periodic*.*

Next, we apply the definitions above to establish some properties of closed-loop control systems with GP dynamics. Fortunately, the topology of $\mathbb{R}^D$ and the Gaussian state predictions obtained from a GP largely facilitate the analysis.

Lemma 18 *The Markov chain $\mathbf{\Sigma}$ on $\mathbb{R}^D$ defined by a GP forward dynamics is irreducible with respect to the Lebesgue measure λ and aperiodic. Also, all compact sets are small.*

Proof The transition probabilities are Gaussian, i.e.,

$$P(\boldsymbol{x}, \cdot) = \mathcal{N}(\boldsymbol{m}(\boldsymbol{x}), S(\boldsymbol{x}))$$

and the Markov kernel has a Gaussian density $q\colon \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$. Thus, $q(\boldsymbol{x}, \boldsymbol{y}) > 0$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^D$ and for any $A \subseteq \mathbb{R}^D$ with $\lambda(A) > 0$ it holds

$$P(\boldsymbol{x}, A) = \int_A q(\boldsymbol{x}, \boldsymbol{y}) \lambda(d\boldsymbol{y}) > 0,$$

so $\mathbf{\Sigma}$ is irreducible with respect to the Lebesgue measure λ.

Assume $\mathbf{\Sigma}$ has period $d > 1$, i.e., $P(\boldsymbol{x}, X_i) = 1$ for $\boldsymbol{x} \in X_{i-1}$, then $1 = P(\boldsymbol{x}, X_i) = \int_{X_i} q(\boldsymbol{x}, \boldsymbol{y}) \lambda(d\boldsymbol{y})$ and, as $q > 0$, it follows that the complement $\overline{X}_i$ of X_i is a λ-null set, i.e., $\lambda(\overline{X}_i) = 0$. However, as X_i, X_j are disjoint for $j \neq i$ it holds $X_j \subseteq \overline{X}_i$. It follows $\lambda(X_j) = 0$ for $j \neq i$. Analogously as for X_i, $\lambda(\overline{X}_j) = 0$. As λ is a non-trivial measure, this is a contradiction. Thus, $d = 1$ and $\mathbf{\Sigma}$ is aperiodic.

Finally, as P has a continuous, positive density, there is a lower bound $\delta > 0$ for this density on any compact set. Thus, all compact sets are small. ■

Above, we showed that the Markov chains considered are aperiodic. The Markov chain can still re-enter sets infinitely often with positive probability. A stronger form of this property is Harris recurrence. As we will see later, Harris recurrence along with some other properties implies the existence and uniqueness of an invariant state distribution.

Definition 19 *Let $A \in \mathcal{B}(\mathbb{R}^D)$ and consider the event*

$$\{\mathbf{\Sigma} \in A \text{ infinitely often}\} \coloneqq \bigcap_{N=1}^{\infty} \bigcup_{k=N}^{\infty} \{\mathbf{\Sigma}_k \in A\}.$$

The set A is called Harris recurrent*, if*

$$Q(\boldsymbol{x}, A) \coloneqq \mathbb{P}_{\boldsymbol{x}}\{\mathbf{\Sigma} \in A \text{ infinitely often}\} = 1.$$

The Markov chain $\mathbf{\Sigma}$ is Harris recurrent, if it is ψ-irreducible and $A \in \mathcal{B}(X)$ is Harris recurrent if $\psi(A) > 0$.

We say that the measure μ is an invariant measure *for the Markov kernel P, if*

$$\mu(A) = \int_X \mu(d\boldsymbol{x}) P(\boldsymbol{x}, A),$$

or shortly $\mu P = \mu$.

The Markov chain $\mathbf{\Sigma}$ is positive Harris recurrent*, if it is Harris recurrent and admits an invariant probability measure.*

Finally, we can state some conditions for the existence and uniqueness of an invariant state distribution as well as convergence of the system state to this distribution (in total variation).

Theorem 20 *If the chain $\mathbf{\Sigma}$ is φ-irreducible, aperiodic and positive Harris recurrent, then it admits a unique invariant probability measure $\mu^*$and for all probability measures ν it holds*

$$\|\nu P^n - \mu^*\|_{TV} \coloneqq \sup_{A \in \mathcal{B}(\mathbb{R}^D)} |\mu^*(A) - \nu(A)| \to 0$$

as $n \to \infty$.

Proof The proof can be found in (Meyn and Tweedie, 2009, Theorem 13.0.1). ■

To employ this theorem to our case of closed-loop control systems with GP dynamics, it remains to establish Harris recurrence. However, it can be challenging to verify the definition directly. Fortunately, drift criteria can be employed. In the following, we will write $Pf \coloneqq \int P(\boldsymbol{x}, d\boldsymbol{y}) f(\boldsymbol{y})$ for any function $f\colon \mathbb{R}^D \to \mathbb{R}$ for notational convenience.

Theorem 21 *Let P be a φ-irreducible and aperiodic Markov kernel. If there exists a function $V\colon \mathbb{R}^D \to [0, \infty)$ with*

$$PV \leq V + c\mathbb{1}_A \tag{21}$$

for a small set A, a constant $c < \infty$, and the sets $\{\boldsymbol{x} \mid V(\boldsymbol{x}) \leq r\}$ are small for all $r \in \mathbb{R}$, then P is Harris recurrent. P is positive Harris recurrent, if Eq. (21) can be modified to

$$PV \leq V - \varepsilon + c\mathbb{1}_A \tag{22}$$

for some $\varepsilon > 0$.

Proof The proof can be found in (Meyn and Tweedie, 2009, Theorems 9.1.8, 11.0.1). ■

Such a function V is called *drift function*. It is the stochastic analogue to a Lyapunov function. Inequality (21) states that whenever the system state is outside of the small set A, V is expected to decrease in the following step. Thus, the system tends towards the minimum of V, which is a recurrent behavior. The speed of convergence can be incorporated in a drift criterion as well.

Definition 22 *The Markov Kernel P has* geometric drift *towards a set A, if there exists a function $V\colon \mathbb{R}^D \to [1,\infty]$, finite for some $\boldsymbol{x}$, and $\beta < 1, c < \infty$, such that*

$$PV \leq \beta V + c\mathbb{1}_A. \tag{23}$$

We say that a φ-irreducible Markov Kernel P with invariant probability measure μ^* is V-geometrically ergodic *on a set A for a function $V\colon \mathbb{R}^D \to [1,\infty]$, if*

$$\sup_{|f|\leq V} \left| \int f d(P^n(\boldsymbol{x},\cdot) - \mu^*) \right| \leq cV(\boldsymbol{x})\rho^n \tag{24}$$

for all $\boldsymbol{x} \in A$, a constant $c < \infty$ and $\rho < 1$ and $V(\boldsymbol{x}) < \infty$ for all $\boldsymbol{x} \in A$.

We will employ geometric drift functions to prove the convergence of Markov chains generated by closed-loop control systems with GP forward dynamics at an exponential convergence speed, irrespective of the starting state. Studying the drift function $V(\boldsymbol{x}) = 1 + \|\boldsymbol{x}\|^2$, the following criterion for geometric ergodicity can be derived.

Theorem 23 *A Markov kernel P with Gaussian transition probabilites has V-geometric drift towards a compact set if and only if*

$$\limsup_{\|\boldsymbol{x}\|\to\infty} \frac{\|\boldsymbol{m}(\boldsymbol{x})\|^2 + \operatorname{tr}(S(\boldsymbol{x}))}{\|\boldsymbol{x}\|^2} < 1 \tag{25}$$

with the mean map $\boldsymbol{m}(\boldsymbol{x})$ and variance map $S(\boldsymbol{x})$. It follows that P has a unique invariant probability measure μ^* and is V-geometrically ergodic.

Proof The proof is given in (Hansen, 2003, Lemma 3.2, Corollary 3.3). ■

Finally, we can prove the main result of this section, the convergence of many closed-loop control systems with GP dynamics towards a unique, invariant limiting distribution that is independent of the starting state.

Theorem 24 *Any discrete-time closed-loop control system with state space $\mathbb{R}^D$, where the forward dynamics is given as GP with zero-mean prior and stationary covariance function, is geometrically ergodic, i.e., the system has a unique invariant probability measure and the Markov chains generated by the system converge to its stationary distribution with exponential speed, see Equation (24).*

Proof We have to verify Eq. (25). As the GP predictions fall back to the zero mean prior far away from the training data, $\|\boldsymbol{m}(\boldsymbol{x})\|^2 + \operatorname{tr}(S(\boldsymbol{x}))$ converges to $\sum_{k=1}^{D} \sigma_{k,f}^2$ as $\|\boldsymbol{x}\| \to \infty$. Thus,

$$\limsup_{\|\boldsymbol{x}\|\to\infty} \frac{\|\boldsymbol{m}(\boldsymbol{x})\|^2 + \operatorname{tr}(S(\boldsymbol{x}))}{\|\boldsymbol{x}\|^2} = 0 < 1,$$

which concludes the proof. ■

This theorem gives us a strong result about the asymptotic behavior of closed-loop control systems with dynamics given as a GP with stationary covariance function and zero mean prior. This result applies to many learning control approaches in the literature.

5. Empirical Evaluation

In this section, we evaluate the previously obtained theoretical results on two benchmark tasks: mountain car and inverted pendulum. Moreover, the performance of the proposed uncertainty propagation is compared to the state-of-the-art moment matching approach and Monte Carlo sampling. We begin with a brief introduction of the test-beds.

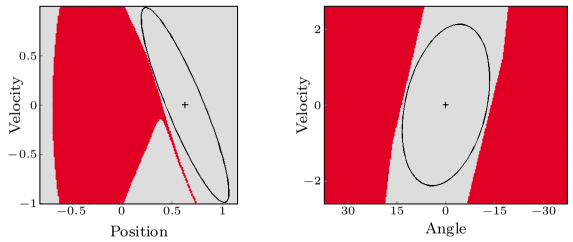
Mountain Car. A car with limited engine power has to reach a desired point in the mountainscape (Sutton and Barto, 1998). The state space has two dimensions: position and velocity of the car. We analyze stability of a PD-controller $\pi((x,\dot{x})^\intercal) = K_p x + K_d \dot{x}$. The gains are chosen as $K_p = 25$ and $K_d = 1$ and the control signal is limited to $u_{\max} = 4$. The GP dynamics model was trained on 250 data points from trajectories with random starting points and control gains.

Inverted Pendulum. In the inverted pendulum task, the goal is to bring the pendulum to an upright position with limited torque (see Doya, 2000) and balance it there. The system state has two dimensions: pendulum angle and velocity. We evaluate stability of a PD-controller with $K_p = 6$, $K_d = 3$ and control limit $u_{\max} = 1.2$. The dynamics GP was trained on 200 points from rollouts with random starting points and control gains.

Cart-Pole. In the cart-pole domain (Deisenroth et al., 2015), a cart with an attached free-swinging pendulum is running on a track of limited length. The goal is to swing the pendulum up and balance it, with the cart in the middle of the track. The state space has four dimensions: position of the cart x, velocity of the cart $\dot{x}$, angle of the pendulum ϑ and the angular velocity $\dot{\vartheta}$. A horizontal force with $u_{\max} = 10$ can be applied to the cart. To demonstrate the proposed approach we analyze stability of the cart-pole system for a PD-controller with randomly chosen gains. The dynamics GP was trained on 250 points from rollouts with sampled starting state and control gains.

5.1 Stability of GP Predictive Mean Dynamics

To evaluate the presented theory on stability of the closed-loop control system with GP mean dynamics, a stability region is determined as described in Section 3. We compare this region to the true stability region, empirically obtained as follows. A grid on the state space is defined and rollouts from every grid point are computed. After a sufficiently long time (1000s), we check whether the state has converged to the target point. Thus, we empirically determine a region of starting points, where the system converges to the desired state. Figure 5 shows the obtained regions for the mountain car and pendulum systems. In both cases the theoretically obtained stability region, which is marked by an ellipsoid, is a subset of the empirically determined region. This effect is due to our analysis yielding a full metric ball centered around the target point, although the full stability region is not necessarily convex. Also, trajectories which first move away from the target point, but finally



(a) Mountain Car (b) Inverted Pendulum

Figure 5: Stability regions for GP mean dynamics on the two benchmark tasks. This figure shows empirically obtained stability regions in gray, points that did not converge in red. The ellipsoid indicates the stability region returned by Algorithm 1.

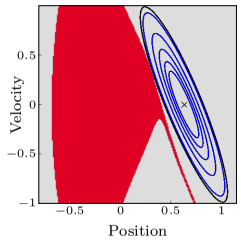


Figure 6: Finite time stability of GP mean dynamics. The black ellipsoid marks the (infinite time) stability region obtained with Algorithm 1. The blue ellipsoids mark the (ε, T)-stability regions with $\varepsilon = 0.05$ and $T = 10, 20, 30, 40, 50$ timesteps (inner to outer ellipsoid). For any starting point inside of one blue ellipsoid, the distance to the target is less than ε after T timesteps.

converge to it, are not considered in the presented theory. Instead, all trajectories starting in the theoretically obtained stability region move towards the target point contractively.

Next, we compute finite time (ε, T)-stability regions as in Lemma 6. Figure 6 shows the obtained regions for $\varepsilon = 0.05$ and different choices of T. By construction, these are full metric balls around the reference point and lie completely inside of the stability region obtained with Algorithm 1.

Finally, we consider robustness of the GP mean dynamics. We introduce disturbances and apply the convergence criteria from Theorem 7 and Lemma 8. We sample the starting time and duration as well as the magnitude of the disturbance. The direction of the disturbance is then sampled independently for every time step where a disturbance is present. The results are shown in Figure 7. Note that robustness results such as Theorem 7 and Lemma 8 must

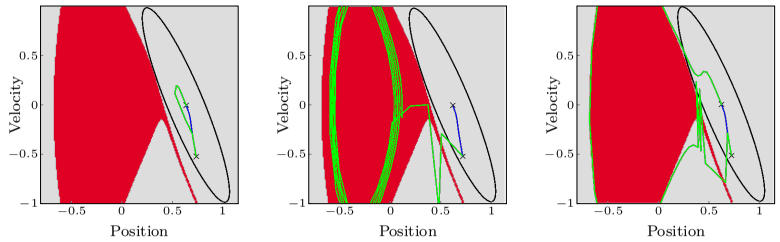
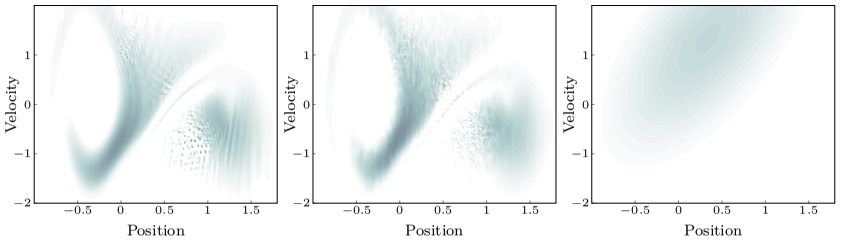


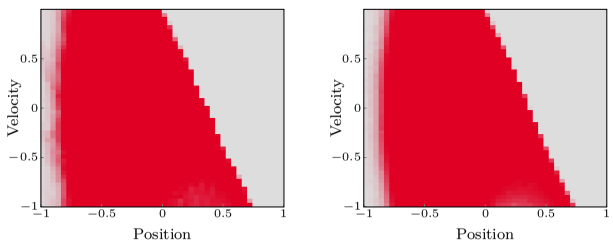
Figure 7: Robustness of GP mean dynamics. Disturbances were sampled for a finite time period. The undisturbed trajectory is colored blue, the distrubed trajectories are colored green. Our robustness criterion from Theorem 7 and Lemma 8 guarantees the convergence of the disturbed trajectory in the left picture. For both other pictures, our criterion cannot guarantee convergence of the disturbed trajectory to the target. As such criteria must be based on worst case estimates of the disturbance, some disturbed trajectories still converge to the target (as in the right picture).



(a) Our Numerical Quadrature (b) Monte Carlo Sampling (c) Moment Matching

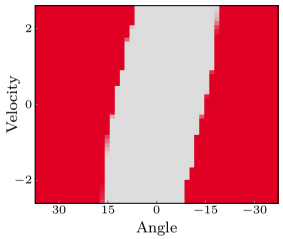
Figure 8: This figure illustrates multi-step-ahead prediction when the input is a distribution. Starting with a normally distributed state centered around the inflection point of the right slope in the mountain car domain, we compute a rollout incorporating uncertainty. The plots show the state distribution after T=30 time steps obtained with our numerical quadrature approach (a), moment matching (c) and the reference Monte Carlo sampling result (b).

consider the worst case scenario. Thus, there are trajectories where the robustness criteria are not able to guarantee convergence, although they converge to the target state.

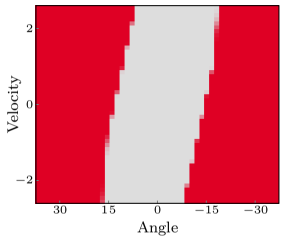


(a) Mountain Car, Our NQ

(b) Mountain Car, MC Sampling



(c) Pendulum, Our NQ



(d) Pendulum, MC Sampling

Figure 9: Success probabilities for dynamics given as GP full distribution and a time horizon of 100s. The color scale encodes the success probability from zero (red) to one (gray). The mountain car ((a) and (b)) and inverted pendulum ((c) and (d)) results were obtained with the proposed numerical quadrature (NQ) and Monte Carlo (MC).

5.2 Numerical Quadrature Uncertainty Propagation

The key to stability analysis for closed-loop systems with dynamics given as full GP distribution is the prediction at uncertain inputs. We compare the performance of the presented approximate inference to the state-of-the-art moment matching (MM) method and Monte Carlo (MC). Consider the following scenario: in the mountain car domain, we position the car on the right slope with some positive velocity. Furthermore, we introduce small Gaussian uncertainty about the starting state. We employ a constant control signal, that is too small to bring the car up directly. We compute rollouts, propagating state distributions through the GP with (i) the presented numerical quadrature (NQ), (ii) MM as in (Deisenroth, 2010), and (iii) MC. The resulting distributions for a time horizon of 3s are shown in Figure 8. The MM approximation differs significantly from MC and NQ results, concentrating most of the probability mass, where the MC approximation has very low probability density. The NQ result closely matches the distribution obtained with MC.

5.3 Stability of Gaussian Process Dynamics

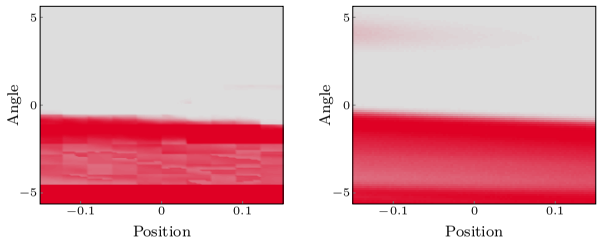
Employing numerical quadrature, we determine success probabilities for the two test-beds and a time horizon of 100s. To obtain the stability region for a particular choice of λ, all pixels whose color corresponds to a value higher then or equal to λ must be selected. As Figure 9 shows, the obtained stability regions match the empirical MC results. The error bound from Lemma 11 can demand for extremely fine quadrature rules to obtain a stability region estimate that matches the true stability region closely. Please note that quadrature rules with less nodes also allow for a guaranteed stability region, however this region can be significantly smaller than the true stability region. For long time horizons, the requirements to obtain a very accurate inner approximation of the stability region can become computationally infeasible. However, we found that the real-world results are substantially better than this worst-case bound. We also experienced computation time ($\approx$ 120s) for NQ to be a fraction of the time required for long time MC predictions. Of course, this will not hold for systems with many state dimensions and our particular setup of product quadrature rules, as these rules suffer from the curse of dimensionality. However, there are various approaches to overcome this drawback of NQ (Heiss and Winschel, 2008; Novak and Ritter, 1996; Xiao and Gimbutas, 2010; Ryu and Boyd, 2015) and our analysis holds for arbitrary quadrature rules.

To show how the proposed approach can be scaled to higher dimensions, we compute success probabilities for the four-dimensional cart-pole system. We employ Algorithm 3 to construct a suitable quadrature rule. However, the quadrature rule CN:3-1 (Stroud, 1971; code from Burkardt, 2014) is applied to each subregion instead of the Gaussian product rule as for the previous examples. This quadrature rule has 8 nodes as opposed to 16 nodes for the Gaussian product rule of the same exactness. Please note also that all necessary computations for the proposed approach can be executed in parallel. Thus, we conduct these computations on a GPU, which leads to a significant speedup and overall computation time comparable to the 2D examples ($\approx$ 140s). The result is shown in Figure 10. As for the other benchmark problems, the obtained stability region closely matches the true stability region of the system.

To evaluate the results on convergence of GP dynamics to a stationary distribution, we considered the dynamics shown in the left plot of Figure 11. The GP mean dynamics has two attractors and their regions of attraction partition the interval $[-1; 1]$. However, when considering the uncertainty, the system has an invariant probability measure (shown in Figure 11 on the right) to which it converges irrespective of the starting point.

6. Conclusion

Gaussian Processes provide a flexible, non-parametric, Bayesian approach to modeling unknown system dynamics from observations. Thus, GP dynamics are a viable approach for learning model-based control. There has been remarkable research in this field that advanced these methods to become an appealing alternative to classic control. For widespread real-world application of learning control based on GP forward dynamics models, performance guarantees are crucial, especially in safety-critical applications. However, stability analysis of closed-loop control with GP forward dynamics learned from data has barely been analyzed so far. In this paper, we laid a foundation for stability analysis of closed-loop control with



(a) Cart-Pole, Our NQ (b) Cart-Pole, MC Sampling

Figure 10: Success probabilities for the cart-pole system. The dynamics are given as GP full distribution and the time horizon is 100s. The color scale encodes the success probability from zero (red) to one (gray). The plots show the results for $\dot{x} = 0, \dot{\theta} = 0$ obtained with numerical quadrature (a) and Monte Carlo sampling (b).

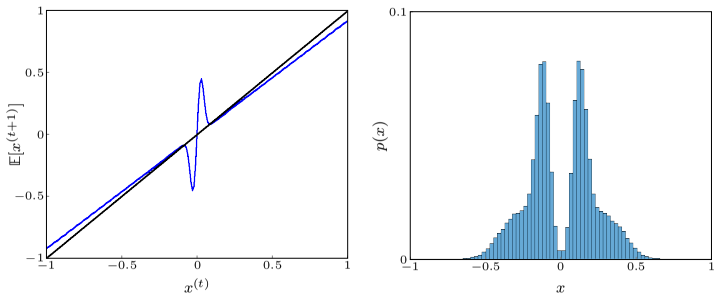
GP dynamics. Subsequently, we conclude the paper with a short summary of the main contributions and a brief outlook on possible future work in this direction.

6.1 Summary of Contributions

In this paper, we analyzed stability of closed-loop control systems with Gaussian process forward model dynamics. We considered two possible types of system dynamics: (i) the mean and (ii) the full GP predictive distribution.

In the first case, we studied asymptotic stability as well as finite time horizons and robustness to disturbances. We presented an algorithm to construct a region in the state space such that trajectories starting inside this region are guaranteed to converge to the target point and stay there for all times. For finite time horizons, we showed how to find a state space region such that the target state will be reached at time horizon up to a certain tolerance. Studying robustness, we derived a criterion for disturbances such that the system remains asymptotically stable. The theoretical results have been evaluated on benchmark problems and compared to empirically obtained results.

In the second case, we introduced a novel approach based on numerical quadrature to approximately propagate uncertainties through a GP. In contrast to other state-of-the-art methods, our approach can model complex distributions with multiple modes. Evaluation results closely match the true distribution approximated by extensive sampling. We used the introduced approximate inference method to derive finite-time stability guarantees based on quadrature error analysis. Empirical Monte Carlo results confirm our theoretical results on the two benchmark problems. Furthermore, we considered the system behavior when the time horizon is infinite. We showed that, applying numerical quadrature to propagate distributions through the GP, the system state converges to a limiting distribution that does not depend on the starting state. Motivated by this result, we studied the system behavior



(a) Forward Dynamics Function (b) Invariant Limiting Distribution

Figure 11: Asymptotic behavior of closed-loop systems with GP dynamics. Plot (a) shows the mean dynamics function with the mean $m(x)$ of the GP prediction at the query point x on the y axis. When considering the mean dynamics, there are two fixpoints $x_1 = -0.1$ and $x_2 = 0.1$. For any starting point $-1 < x < 0$, the mean trajectory converges to x_1 and analogously for all $0 < x < 1$ the trajectories converge to x_2. Now we consider the full GP dynamics and take the uncertainty into account. Plot (b) shows a histogram of the probability density after 10000 time steps when starting at $x = -0.1$, obtained with Monte Carlo. This result matches the histogram obtained when starting at $x = 0.1$ or at any other point. Thus, the system converges to a unique, invariant limiting distribution.

for infinite time horizons without applying any approximate inference steps. We succeeded to show that closed-loop control systems with dynamics given as full GP distribution converge to a unique and invariant limiting distribution that does not depend on the starting state for many choices of the covariance function. Overall, the proposed methods provide stability guarantees for many existing learning control approaches based on GPs.

6.2 Discussion and Next Steps

While the proposed methods apply to many interesting examples of learning control in the literature, the analysis could be extended in several directions. Firstly, for closed-loop control with GP dynamics, we considered GPs with stationary covariance functions and zero mean prior. To include other choices of the prior, e.g., nonstationary covariance functions and unbounded mean priors, novel criteria to ensure Harris recurrence must be found. Another interesting question is how the stability of GP dynamics is affected by disturbances in the input space. While some results could straightforwardly be followed for dynamics given as the mean of a GP, the task seems a lot more challenging when uncertainty is included. Many of the subtle, intricate arguments applied to show convergence of the system to a limiting distribution must be carefully reconsidered when disturbances are present.

Secondly, the obtained results are stability results for closed-loop control systems with dynamics given as a GP. The GP training data have a huge impact on the quality of the GP dynamics model when compared to the physical system the data was collected from. In this direction, criteria to evaluate the quality of a learned model would further support the application of learning control for real-world problems.

Another interesting research direction is encouraged by the presented result on asymptotic stability of closed-loop control systems with GP dynamics. This paper proves the existence of a limiting distribution the system will converge to. This distribution is unique and invariant. However, little more is known about the limiting distribution. Further research in this direction could help for a better understanding of the behavior of learned dynamics and also be directly employed in learning control – possibly by optimizing the limiting distribution itself instead of local approximations as, e.g., system trajectories.

References

G. Adomian. *Stochastic systems*. Mathematics in Science and Engineering. Elsevier Science, 1983.

A. A. Ahmadi and P. A. Parrilo. Converse results on existence of sum of squares lyapunov functions. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 6516–6521, Dec 2011.

A. A. Ahmadi, A. Majumdar, and R. Tedrake. Complexity of ten decision problems in continuous time dynamical systems. In *2013 American Control Conference*, pages 6376–6381, 2013.

T. Beckers and S. Hirche. Stability of gaussian process state space models. In *Proceedings of the European Control Conference (ECC)*, 2016.

F. Blanchini. Set invariance in control. *Automatica*, 35(11):1747 – 1767, 1999.

J. Burkardt. Stroud – numerical integration in m dimensions. `https://people.sc.fsu.edu/~jburkardt/m_src/stroud/stroud.html`, 2014.

G. Chesi. Estimating the domain of attraction for uncertain polynomial systems. *Automatica*, 40(11):1981–1986, 2004.

P.J. Davis, P. Rabinowitz, and W. Rheinbolt. *Methods of Numerical Integration*. Computer Science and Applied Mathematics. Elsevier Science, 2014.

M.P. Deisenroth. *Efficient Reinforcement Learning Using Gaussian Processes*. Karlsruhe series on intelligent sensor actuator systems. KIT Scientific Publ., 2010.

M.P. Deisenroth, D. Fox, and C.E. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(2):408–423, 2015.

K. Doya. Reinforcement learning in continuous time and space. *Neural Computation*, 12: 219–245, 2000.

Y. Engel, P. Szabo, and D. Volkinshtein. Learning to control an octopus arm with gaussian process temporal difference methods. In Y. Weiss, B. Schölkopf, and J.C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 347–354. MIT Press, 2006.

G.A. Evans. The estimation of errors in numerical quadrature. *International Journal of Mathematical Education in Science and Technology*, 25(5):727–744, 1994.

G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013.

N.R. Hansen. Geometric ergodicity of discrete-time approximations to multivariate diffusions. *Bernoulli*, 9(4):725–743, 08 2003.

F. Heiss and V. Winschel. Likelihood approximation by numerical integration on sparse grids. *Journal of Econometrics*, 144(1):62 – 80, 2008.

A. Hurwitz. Ueber die Bedingungen, unter welchen eine Gleichung nur Wurzeln mit negativen reellen Theilen besitzt. *Mathematische Annalen*, 46(2):273–284, 1895.

H.K. Khalil. *Nonlinear control*. Prentice Hall, 2014.

R. Khasminskii and G.N. Milstein. *Stochastic Stability of Differential Equations*. Stochastic Modelling and Applied Probability. Springer Berlin Heidelberg, 2011.

H.J. Kim and A.Y. Ng. Stable adaptive control with online learning. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 977–984. MIT Press, 2005.

E.D. Klenske, M.N. Zeilinger, B. Schölkopf, and P. Hennig. Nonparametric dynamics estimation for time periodic systems. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pages 486–493. IEEE, 2013.

J. Ko and D. Fox. GP-BayesFilters: Bayesian filtering using gaussian process prediction and observation models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008.*, 2008.

J. Kocijan, R. Murray-Smith, C.E. Rasmussen, and A. Girard. Gaussian process model based predictive control. In *American Control Conference, 2004. Proceedings of the 2004*, volume 3, pages 2214–2219. IEEE, 2004.

H.J. Kushner. Finite time stochastic stability and the analysis of tracking systems. *Automatic Control, IEEE Transactions on*, 11(2):219–227, 1966.

H.J. Kushner. *Stochastic Stability and Control*. Mathematics in science and engineering. Academic Press, 1967.

A.M. Lyapunov. *General Problem of the Stability Of Motion*. Doctoral dissertation, Univesity of Kharkov, 1892. Englisch Translation by A.T. Fuller, Taylor & Francis, London 1992.

J.M. Maciejowski and X. Yang. Fault tolerant control using gaussian processes and model predictive control. In *Control and Fault-Tolerant Systems (SysTol), 2013 Conference on*, pages 1–12. IEEE, 2013.

A. Majumdar, A. A. Ahmadi, and R. Tedrake. Control and verification of high-dimensional systems with dsos and sdsos programming. In *53rd IEEE Conference on Decision and Control*, pages 394–401, 2014.

M. Masjed-Jamei. New error bounds for gauss-legendre quadrature rules. *Filomat*, 28(6): 1281–1293, 2014.

S. Meyn and R.L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009.

Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7:2651–2667, December 2006.

J. Moore and R. Tedrake. Adaptive control design for underactuated systems using sums-of-squares optimization. In *2014 American Control Conference*, pages 721–728, June 2014.

J. Nakanishi, J.A. Farrell, and S. Schaal. A locally weighted learning composite adaptive controller with structure adaptation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002.*, pages 882–889 vol.1, 2002.

K.S. Narendra and A.M. Annaswamy. *Stable Adaptive Systems*. Dover Books on Electrical Engineering. Dover Publications, 2012.

D. Nguyen-Tuong and J. Peters. Model learning in robotics: a survey. *Cognitive Processing*, (4), 2011.

E. Novak and K. Ritter. High dimensional integration of smooth functions over cubes. *Numerische Mathematik*, 75(1):79–97, 1996.

Y. Pan and E. Theodorou. Probabilistic differential dynamic programming. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1907–1915. Curran Associates, Inc., 2014.

A. Papachristodoulou and S. Prajna. *Analysis of Non-polynomial Systems Using the Sum of Squares Decomposition*, pages 23–43. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

P.A. Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, 2000.

T.J. Perkins and A.G. Barto. Lyapunov design for safe reinforcement learning. *Journal of Machine Learning Research*, 3:803–832, March 2003.

J. Quiñonero-Candela, A. Girard, J. Larsen, and C.E. Rasmussen. Propagation of uncertainty in bayesian kernel models - application to multiple-step ahead forecasting. In *International Conference on Acoustics, Speech and Signal Processing*, pages 701–704, vol. 2, 2003.

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

E.J. Routh. *A Treatise on the Stability of a Given State of Motion: Particularly Steady Motion*. Macmillan and Company, 1877.

E.K. Ryu and S.P. Boyd. Extensions of gauss quadrature via linear programming. *Foundations of Computational Mathematics*, 15(4):953–971, 2015.

S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, 2005.

B.S. Skrainka and K.L. Judd. High performance quadrature rules: How numerical integration affects a popular model of product differentiation. *Available at SSRN 1870703*, 2011.

J. Steinhardt and R. Tedrake. Finite-time regional verification of stochastic nonlinear systems. In H.F. Durrant-Whyte, N. Roy, and P. Abbeel, editors, *Robotics: Science and Systems VII*, pages 321–328. MIT Press, 2012.

A.H. Stroud. *Approximate calculation of multiple integrals*. Prentice-Hall series in automatic computation. Prentice-Hall, 1971.

E. Süli and D.F. Mayers. *An Introduction to Numerical Analysis*. Cambridge University Press, 2003.

R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

G. Tao. *Adaptive Control Design and Analysis (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

U. Topcu, A. Packard, P. Seiler, and G. Balas. Help on sos [ask the experts]. *IEEE Control Systems*, 30(4):18–23, 2010a.

U. Topcu, A. K. Packard, P. Seiler, and G. J. Balas. Robust region-of-attraction estimation. *IEEE Transactions on Automatic Control*, 55(1):137–142, 2010b.

J. Vinogradska, B. Bischoff, D. Nguyen-Tuong, A. Romer, H. Schmidt, and J. Peters. Stability of controllers for gaussian process forward models. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 545–554, 2016.

S. Wasowicz. On error bounds for gauss-legendre and lobatto quadrature rules. *Journal of Inequalities in Pure & Applied Mathematics*, 7(3):Paper No. 84, 7 p., 2006.

H. Xiao and Z. Gimbutas. A numerical algorithm for the construction of efficient quadrature rules in two and higher dimensions. *Computers & Mathematics with Applications*, 59(2): 663 – 676, 2010.

K. Zhou and J.C. Doyle. *Essentials of Robust Control*. Prentice Hall Modular Series for Eng. Prentice Hall, 1998.

Journal of Machine Learning Research 18 (2017) 1-51 Submitted 7/16; Revised 4/17; Published 10/17

Harder, Better, Faster, Stronger[†] Convergence Rates for Least-Squares Regression

Aymeric Dieuleveut AYMERIC.DIEULEVEUT@ENS.FR
Nicolas Flammarion NICOLAS.FLAMMARION@ENS.FR
Francis Bach FRANCIS.BACH@ENS.FR

INRIA
Département d'Informatique de l'ENS, École normale supérieure, CNRS, PSL Research University Paris, France.

Editor: Alexander Rakhlin

Abstract

We consider the optimization of a quadratic objective function whose gradients are only accessible through a stochastic oracle that returns the gradient at any given point plus a zero-mean finite variance random error. We present the first algorithm that achieves jointly the optimal prediction error rates for least-squares regression, both in terms of forgetting the initial conditions in $O(1/n^2)$, and in terms of dependence on the noise and dimension d of the problem, as $O(d/n)$. Our new algorithm is based on averaged accelerated regularized gradient descent, and may also be analyzed through finer assumptions on initial conditions and the Hessian matrix, leading to dimension-free quantities that may still be small in some distances while the "optimal" terms above are large. In order to characterize the tightness of these new bounds, we consider an application to non-parametric regression and use the known lower bounds on the statistical performance (without computational limits), which happen to match our bounds obtained from a single pass on the data and thus show optimality of our algorithm in a wide variety of particular trade-offs between bias and variance.

Keywords: convex optimization, least-squares regression, stochastic gradient, accelerated gradient, non-parametric estimation

1. Introduction

Many supervised machine learning problems are naturally cast as the minimization of a smooth function defined on a Euclidean space. This includes least-squares regression, logistic regression (see, e.g., Hastie et al., 2009) or generalized linear models (McCullagh and Nelder, 1989). While small problems with few or low-dimensional input features may be solved precisely by many potential optimization algorithms (e.g., Newton method), large-scale problems with many high-dimensional features are typically solved with simple gradient-based iterative techniques whose per-iteration cost is small.

. [†] https://youtu.be/yydNF8tuVmU

©2017 Dieuleveut, Flammarion and Bach.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-335.html.

In this paper, we consider a quadratic objective function f whose gradients are only accessible through a stochastic oracle that returns the gradient at any given point plus a zero-mean finite variance random error. In this stochastic approximation framework (Robbins and Monro, 1951), it is known that two quantities dictate the behavior of various algorithms, namely the covariance matrix V of the noise in the gradients, and the deviation $\theta_0 - \theta_*$ between the initial point of the algorithm θ_0 and any of the global minimizer θ_* of f. This leads to a "bias/variance" decomposition (Bach and Moulines, 2013; Hsu et al., 2014) of the performance of most algorithms as the sum of two terms: (a) the bias term characterizes how fast initial conditions are forgotten and thus is increasing in a well-chosen norm of $\theta_0 - \theta_*$; while (b) the variance term characterizes the effect of the noise in the gradients, independently of the starting point, and with a term that is increasing in the covariance of the noise.

For quadratic functions with (a) a noise covariance matrix V which is proportional (with constant σ^2) to the Hessian of f (a situation which corresponds to least-squares regression) and (b) an initial point characterized by the norm $\|\theta_0 - \theta_*\|^2$, the optimal bias and variance terms are known *separately* from the optimization and statistical theories. On the one hand, the optimal bias dependency after n iterations is proportional to $\frac{L\|\theta_0-\theta_*\|^2}{n^2}$, where L is the largest eigenvalue of the Hessian of f. This rate is achieved by accelerated gradient descent (Nesterov, 1983, 2004), and is known to be optimal if the number of iterations n is less than the dimension d of the underlying predictors, but the algorithm is not robust to random or deterministic noise in the gradients (d'Aspremont, 2008; Schmidt et al., 2011; Devolder et al., 2014). On the other hand, the optimal variance term is proportional to $\frac{\sigma^2 d}{n}$ (Tsybakov, 2008); it is known to be achieved by averaged gradient descent (Bach and Moulines, 2013), for which the bias term only achieves $\frac{L\|\theta_0-\theta_*\|^2}{n}$ instead of $\frac{L\|\theta_0-\theta_*\|^2}{n^2}$.

Our first contribution in this paper is to present a novel algorithm which attains optimal rates for *both the variance and the bias terms*. This algorithm analyzed in Section 4 is averaged accelerated gradient descent; beyond obtaining jointly optimal rates, our result shows that averaging is beneficial for accelerated techniques and provides a provable robustness to noise.

While optimal when measuring performance in terms of the dimension d and the initial distance to optimum $\|\theta_0 - \theta_*\|^2$, these rates are not adapted in many situations where either d is larger than the number of iterations n (i.e., the number of observations for regular stochastic gradient descent) or $L\|\theta_0 - \theta_*\|^2$ is much larger than n^2. Our second contribution is to provide in Section 5 an analysis of a new algorithm (based on some additional regularization) that can adapt our bounds to finer assumptions on $\theta_0 - \theta_*$ and the Hessian of the problem, leading in particular to dimension-free quantities that can thus be extended to the Hilbert space setting (in particular for non-parametric estimation).

In order to characterize the optimality of these new bounds, our third contribution is to consider an application to non-parametric regression in Section 6 and use the known lower bounds on the statistical performance (without computational limits), which happen to match our bounds obtained from a single pass on the data and thus show optimality of our algorithm in a wide variety of particular trade-offs between bias and variance.

Our paper is organized as follows: in Section 2, we present the main problem we tackle, namely least-squares regression, then introduce the two algorithms that we consider in

Section 2.2, as well as the two types of oracles on the gradient in Section 2.3. In Section 3, we present new results for averaged stochastic gradient descent that set the stage for Section 4, where we present our main novel result leading to an accelerated algorithm which is robust to noise. Our tighter analysis of convergence rates based on finer dimension-free quantities is presented in Section 5, and their optimality for kernel-based non-parametric regression is studied in Section 6. Organization of the main results is summarized in the Table 1 bellow.

	Averaged Algo.	**Averaged Accelerated Algo.**
Dimension dependent rates	*Section 3*	*Section 4*
Additive Noise	Lemma 1$^\diamondsuit$	Theorem 3
Multiplicative Noise	Theorem 2$^\diamondsuit$	$\natural$
Dimension independent rates	*Section 5*	*Section 5*
Additive Noise	$\sharp$	Theorem 5
Multiplicative Noise	4th remark after Cor. 6$^\flat$	$\natural$
Kernel regression setting	*Section 6*	*Section 6*
Additive Noise	$\sharp$	Theorem 8
Multiplicative Noise	Theorem 7$^\flat$	$\natural$

Table 1: Organization of the paper. $\diamondsuit$: We extend results from (Bach and Moulines, 2013) to the setting in which extra regularization is added; $\sharp$: apart from Lemma 1 which is useful to develop intuition of the different terms in the upper bound, we do not state result for the averaged algorithm with additive noise, as the most powerful result is for the multiplicative noise; $\flat$: these results recover results from Dieuleveut and Bach (2015) (with the use of an extra regularization); $\natural$: it is still an open problem to get results in the accelerated setting for a multiplicative noise oracle.

2. Least-Squares Regression

In this section, we present our least-squares regression framework, which is risk minimization with the square loss, together with the main assumptions regarding our model and our algorithms. These algorithms will rely on stochastic gradient oracles, which will come in two kinds, an additive noise which does not depend on the current iterate, which will correspond in practice to the full knowledge of the covariance matrix, and a "multiplicative/additive" noise, which corresponds to the regular stochastic gradient obtained from a single pair of observations. This second oracle is much harder to analyze.

2.1 Statistical Assumptions

We consider the following general setting:

- $\mathcal{H}$ is a d-dimensional Euclidean space with $d \geq 1$. The (temporary) restriction to finite dimension will be relaxed in Section 6.

- The observations $(x_n, y_n) \in \mathcal{H} \times \mathbb{R}$, $n \geq 1$, are independent and identically distributed (i.i.d.), and such that $\mathbb{E}\|x_n\|^2$ and $\mathbb{E}y_n^2$ are finite.

- We consider the *least-squares regression* problem, namely the minimization of the expected loss $f(\theta) = \frac{1}{2}\mathbb{E}(\langle x_n, \theta\rangle - y_n)^2$ which is a quadratic function.

We first introduce an assumption on the distribution of x_n.

Covariance matrix. We denote by $\Sigma = \mathbb{E}(x_n \otimes x_n) \in \mathbb{R}^{d\times d}$ the population covariance matrix, which is the Hessian of f at all points. Without loss of generality, we can assume Σ is invertible by reducing $\mathcal{H}$ to the minimal subspace where all x_n, $n \geq 1$, lie almost surely. This implies that all eigenvalues of Σ are strictly positive (but they may be arbitrarily small). Following Bach and Moulines (2013), we assume there exists $R > 0$ such that

$$\mathbb{E}\|x_n\|^2 x_n \otimes x_n \preccurlyeq R^2\Sigma, \tag{$\mathcal{A}_1$}$$

where $A \preccurlyeq B$ means that $B - A$ is positive semi-definite. This assumption implies in particular that (a) $\mathbb{E}\|x_n\|^4$ is finite and (b) $\operatorname{tr}\Sigma = \mathbb{E}\|x_n\|^2 \leq R^2$ since taking the trace of the previous inequality we get $\mathbb{E}\|x_n\|^4 \leq R^2\mathbb{E}\|x_n\|^2$ and using Cauchy-Schwarz inequality we get $\mathbb{E}\|x_n\|^2 \leq \sqrt{\mathbb{E}\|x_n\|^4} \leq R\sqrt{\mathbb{E}\|x_n\|^2}$.

Assumption $(\mathcal{A}_1)$ is satisfied, for example, for least-square regression with almost surely bounded data, since $\|x_n\|^2 \leq R^2$ almost surely implies $\mathbb{E}\|x_n\|^2 x_n \otimes x_n \preccurlyeq \mathbb{E}\left[R^2 x_n \otimes x_n\right] = R^2\Sigma$. This assumption is also true for data with infinite support and a bounded *kurtosis* for the projection of the covariates x_n on any direction $z \in \mathcal{H}$, e.g, for which there exists $\kappa > 0$, such that:

$$\forall z \in \mathcal{H}, \quad \mathbb{E}\langle z, x_n\rangle^4 \leq \kappa\langle z, \Sigma z\rangle^2. \tag{1}$$

Indeed, by Cauchy-Schwarz inequality, Equation (1) implies for all $(z, t) \in \mathcal{H}^2$, the following bound $\mathbb{E}\langle z, x_n\rangle^2\langle t, x_n\rangle^2 \leq \kappa\langle z, \Sigma z\rangle\langle t, \Sigma t\rangle$, which in turn implies that for all positive semi-definite symmetric matrices M, N, we have $\mathbb{E}\langle x_n, Mx_n\rangle\langle x_n, Nx_n\rangle \leq \kappa \operatorname{tr}(M\Sigma)\operatorname{tr}(N\Sigma)$. Equation (1), which is true for Gaussian vectors with $\kappa = 3$, thus implies $(\mathcal{A}_1)$ for $R^2 = \kappa \operatorname{tr}\Sigma = \kappa\mathbb{E}\|x_n\|^2$.

In the next two paragraphs, we introduce some quantities that will be important in the analysis, in order to get tighter bounds.

Eigenvalue decay. Most convergence bounds depend on the dimension d of $\mathcal{H}$. However it is possible to derive dimension-free and often tighter convergence rates by considering bounds depending on the value $\operatorname{tr}\Sigma^b$ for $b \in [0, 1]$. Given b, if we consider the eigenvalues of Σ ordered in decreasing order, which we denote by s_i, then $\operatorname{tr}\Sigma^b = \sum_i s_i^b$, and the eigenvalues decay[1] at least as $\frac{(\operatorname{tr}\Sigma^b)^{1/b}}{i^{1/b}}$. Moreover, it is known that $(\operatorname{tr}\Sigma^b)^{1/b}$ is decreasing in b and thus, the smaller the b, the stronger the assumption. For b going to 0 then $\operatorname{tr}\Sigma^b$ tends to d and we are back in the classical low-dimensional case. When $b = 1$, we simply get $\operatorname{tr}\Sigma = \mathbb{E}\|x_n\|^2$, which will correspond to the weakest assumption in our context.

1. Indeed for any $i \geq 1$, we have $is_i^b \leq \sum_{t=1}^i s_t^b \leq \operatorname{tr}(\Sigma^b)$.

Optimal predictor. In finite dimension the regression function $f(\theta) = \frac{1}{2}\mathbb{E}(\langle x_n, \theta\rangle - y_n)^2$ always admits a global minimum $\theta_* = \Sigma^\dagger \mathbb{E}(y_n x_n)$. When initializing algorithms at $\theta_0 = 0$ or regularizing by the squared norm, rates of convergence generally depend on $\|\theta_*\|$, a quantity which could be arbitrarily large.

However there exists a systematic upper-bound[2] $\|\Sigma^{\frac{1}{2}}\theta_*\| \leq 2\sqrt{\mathbb{E}y_n^2}$. This leads naturally to the consideration of convergence bounds depending on $\|\Sigma^{r/2}\theta_*\|$ for $r \leq 1$. In infinite dimension this will correspond to assuming $\|\Sigma^{r/2}\theta_*\| < \infty$. This new assumption relates the optimal predictor with sources of ill-conditioning (since Σ is the Hessian of the objective function f), the smaller r, the stronger our assumption, with $r = 1$ corresponding to no assumption at all, $r = 0$ to θ_* in $\mathcal{H}$ and $r = -1$ to a convergence of the bias of least-squares regression with averaged stochastic gradient descent in $O\big(\frac{\|\Sigma^{-1/2}\theta_*\|^2}{n^2}\big)$(Dieuleveut and Bach, 2015; Défossez and Bach, 2015). In this paper, we will use arbitrary initial points θ_0 and thus our bounds will depend on $\|\Sigma^{r/2}(\theta_0 - \theta_*)\|$.

Finally , we make an assumption on the joint distribution of (x_n, y_n).

Noise. We denote by $\varepsilon_n = y_n - \langle \theta_*, x_n\rangle$ the residual for which we have $\mathbb{E}[\varepsilon_n x_n] = 0$. Although we do not have $\mathbb{E}[\varepsilon_n | x_n] = 0$ in general unless the model is well-specified, we assume the noise to be a structured process such that there exists $\sigma > 0$ with

$$\mathbb{E}[\varepsilon_n^2 x_n \otimes x_n] \preccurlyeq \sigma^2 \Sigma. \tag{$\mathcal{A}_2$}$$

Assumption $(\mathcal{A}_2)$ is satisfied for example for data almost surely bounded or when the model is well-specified, (e.g., $y_n = \langle \theta_*, x_n\rangle + \varepsilon_n$, with $(\varepsilon_n)_{n\in\mathbb{N}}$ i.i.d. of variance σ^2 and independent of x_n).

2.2 Averaged Gradient Methods and Acceleration

We focus in this paper on stochastic gradient methods with and without acceleration for the least-squares function regularized by $\frac{\lambda}{2}\|\theta - \theta_0\|^2$ for $\lambda \in \mathbb{R}^+$. The regularization will be useful when deriving tighter convergence rates in Section 5, and it has the additional benefit of making the problem λ-strongly-convex. Stochastic gradient descent (referred to from now on as "SGD"), applied to the regularized problem, can be described for $n \geq 1$ as

$$\theta_n = \theta_{n-1} - \gamma f_n'(\theta_{n-1}) - \gamma\lambda(\theta_{n-1} - \theta_0), \tag{2}$$

starting from $\theta_0 \in \mathcal{H}$, where $\gamma > 0$ is either called the step-size in optimization or the learning rate in machine learning, and $f_n'(\theta_{n-1})$ is an unbiased estimate of the gradient of f at θ_{n-1}, that is, its conditional expectation given all other sources of randomness is equal to $f'(\theta_{n-1})$.

Accelerated stochastic gradient descent is defined, for the regularized problem, by an iterative system with two parameters (θ_n, ν_n) satisfying for $n \geq 1$

$$\begin{aligned} \theta_n &= \nu_{n-1} - \gamma f_n'(\nu_{n-1}) - \gamma\lambda(\nu_{n-1} - \theta_0) \\ \nu_n &= \theta_n + \delta\big(\theta_n - \theta_{n-1}\big), \end{aligned} \tag{3}$$

2. Indeed for all $\theta \in \mathbb{R}^d$ and in particular $\theta = 0$, by Minkowski's inequality, $\|\Sigma^{\frac{1}{2}}\theta_*\| - \sqrt{\mathbb{E}y_n^2} = \sqrt{\mathbb{E}\langle\theta_*, x_n\rangle^2} - \sqrt{\mathbb{E}y_n^2} \leq \sqrt{\mathbb{E}(\langle\theta_*, x_n\rangle - y_n)^2} \leq \sqrt{\mathbb{E}(\langle\theta, x_n\rangle - y_n)^2} \leq \sqrt{\mathbb{E}(y_n)^2}$.

starting from $\theta_0 = \nu_0 \in \mathcal{H}$, with $\gamma, \delta \in \mathbb{R}^2$ and $f_n'(\theta_{n-1})$ described as before. It may be reformulated as the following second-order recursion

$$\theta_n = (1 - \gamma\lambda)\big(\theta_{n-1} + \delta(\theta_{n-1} - \theta_{n-2})\big) - \gamma f_n'\big(\theta_{n-1} + \delta(\theta_{n-1} - \theta_{n-2})\big) + \gamma\lambda\theta_0.$$

The *momentum* coefficient $\delta \in \mathbb{R}$ is chosen to accelerate the convergence rate (Nesterov, 1983; Beck and Teboulle, 2009) and has its roots in the heavy-ball algorithm from Polyak (1964). We especially concentrate here, following Polyak and Juditsky (1992), on the average of the sequence

$$\bar{\theta}_n = \frac{1}{n+1}\sum_{i=0}^{n}\theta_n, \tag{4}$$

and we note that it can be computed online as $\bar{\theta}_n = \frac{n}{n+1}\bar{\theta}_{n-1} + \frac{1}{n+1}\theta_n$.

The key ingredient in the algorithms presented above is the unbiased estimate on the gradient $f_n'(\theta)$, which can take two forms that we now describe in our setting.

2.3 Additive versus Multiplicative Stochastic Oracles on the Gradient

We consider the standard stochastic approximation framework (Kushner and Yin, 2003). That is, we let $(\mathcal{F}_n)_{n\geq 0}$ be the increasing family of σ-fields that are generated by all variables (x_i, y_i) for $i \leq n$, and such that for each $\theta \in \mathcal{H}$ the random variable $f_n'(\theta)$ is square-integrable and $\mathcal{F}_n$-measurable with $\mathbb{E}[f_n'(\theta)|\mathcal{F}_{n-1}] = f'(\theta)$, for all $n \geq 0$. Consequently it is of the form

$$f_n'(\theta) = f'(\theta) - \xi_n, \tag{$\mathcal{A}_3$}$$

where the noise process ξ_n is $\mathcal{F}_n$-measurable with $\mathbb{E}[\xi_n|\mathcal{F}_{n-1}] = 0$ and $\mathbb{E}[\|\xi_n\|^2]$ is finite. We will consider two different gradient oracles.

Additive noise. The first oracle is the sum of the true gradient $f'(\theta)$ and an independent zero-mean noise that does not depend on θ. This oracle is equal to

$$f_n'(\theta) = \Sigma\theta - y_n x_n. \tag{5}$$

Since $f'(\theta) = \Sigma\theta - \mathbb{E}y_n x_n$, the oracle above has a noise vector $\xi_n = y_n x_n - \mathbb{E}y_n x_n$ independent of θ and therefore satisfies Assumption $(\mathcal{A}_3)$. Furthermore we also assume that there exists $\tau \in \mathbb{R}$ such that

$$\mathbb{E}[\xi_n \otimes \xi_n] \preccurlyeq \tau^2\Sigma, \tag{$\mathcal{A}_4$}$$

that is, the noise has a particular structure adapted to least-squares regression. For optimal results for unstructured noise, with convergence rate for the noise part in $O(1/\sqrt{n})$, see Lan (2012). Our oracle above with an additive noise which is independent of the current iterate corresponds to the first setting studied in stochastic approximation (Robbins and Monro, 1951; Duflo, 1997; Polyak and Juditsky, 1992). While used by Bach and Moulines (2013) as an artifact of proof, for least-squares regression, such an additive noise corresponds to the situation where the distribution of x is known so that the population covariance matrix is computable, but the distribution of the outputs $(y_n)_{n\in\mathbb{N}}$ remains unknown. Thus it may be seen as an intermediate set-up between regression estimation with fixed and random design (see, e.g., Györfi et al., 2006, Section 1.9).

Assumption $(\mathcal{A}_4)$ will be satisfied, for example if the outputs are almost surely bounded because $\mathbb{E}[\xi_n \otimes \xi_n] \preccurlyeq \mathbb{E}[y_n^2 x_n \otimes x_n] \preccurlyeq \tau^2 \Sigma$ if $y_n^2 \leq \tau^2$ almost surely. But it will also be for data satisfying Equation (1) since we will have

$$\begin{aligned}\mathbb{E}[\xi_n \otimes \xi_n] &\preccurlyeq \mathbb{E}[y_n^2 x_n \otimes x_n] = \mathbb{E}[(\langle \theta_*, x_n\rangle + \varepsilon_n)^2 x_n \otimes x_n] \\ &\preccurlyeq 2\mathbb{E}[\langle \theta_*, x_n\rangle^2 x_n \otimes x_n] + 2\sigma^2 \Sigma \preccurlyeq 2(\kappa \|\Sigma^{1/2}\theta_*\|^2 + \sigma^2)\Sigma \preccurlyeq 2(4\kappa \mathbb{E}[y_n^2] + \sigma^2)\Sigma,\end{aligned}$$

and thus Assumption $(\mathcal{A}_4)$ is satisfied with $\tau^2 = 2(4\kappa \mathbb{E}[y_n^2] + \sigma^2)$.

Stochastic noise ("multiplicative/additive"). This corresponds to:

$$f_n'(\theta) = (\langle x_n, \theta\rangle - y_n) x_n = (\Sigma + \zeta_n)(\theta - \theta_*) - \Xi_n, \tag{6}$$

with $\zeta_n = x_n \otimes x_n - \Sigma$ and $\Xi_n = (y_n - \langle x_n, \theta_*\rangle) x_n = \varepsilon_n x_n$. This oracle corresponds to regular SGD, which is often referred to as the least-mean-square (LMS) algorithm for least-squares regression, where the noise comes from sampling a single pair of observations. While still satisfying Assumption $(\mathcal{A}_3)$, it combines an additive noise Ξ_n independent of θ as in Eq. (5) and a multiplicative noise ζ_n. This multiplicative noise makes this stochastic oracle harder to analyze which explains why it is often approximated by an additive noise oracle. However it is the most widely used and most practical one. Note that for the oracle in Eq. (6), from Equation $(\mathcal{A}_2)$, we have $\mathbb{E}[\Xi_n \otimes \Xi_n] \preccurlyeq \sigma^2 \Sigma$. It has a similar form to Assumption $(\mathcal{A}_4)$ which is valid for the additive noise oracle in Eq. (5): we use different constants σ^2 and τ^2 to highlight the difference between these two oracles.

3. Averaged Stochastic Gradient Descent

In this section, we provide convergence bounds for regularized averaged stochastic gradient descent. The main novelty compared to the work of Bach and Moulines (2013) is (a) the presence of regularization, which will be useful when deriving tighter convergence rates in Section 5 and (b) a much simpler proof. We first consider the additive noise in Section 3.1 before considering the multiplicative/additive noise in Section 3.2.

3.1 Additive Noise

We study here the convergence of the averaged SGD recursion defined by Eq. (2) under the simple oracle defined in Eq. (5). For least-squares regression, it takes the form:

$$\theta_n = \big[I - \gamma\Sigma - \gamma\lambda I\big]\theta_{n-1} + \gamma y_n x_n + \lambda\gamma\theta_0. \tag{7}$$

This is an easy adaptation of the work of Bach and Moulines (2013, Lemma 2) for the regularized case.

Lemma 1 *Assume $(\mathcal{A}_4)$. Consider the recursion in Eq. (7) with any regularization parameter $\lambda \in \mathbb{R}_+$ and any constant step-size γ such that $\gamma(\Sigma + \lambda I) \preccurlyeq I$. Then*

$$\mathbb{E} f(\bar{\theta}_n) - f(\theta_*) \leq \Big(\lambda + \frac{1}{\gamma n}\Big)^2 \|\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\|^2 + \frac{\tau^2 \operatorname{tr}\big[\Sigma^2(\Sigma + \lambda I)^{-2}\big]}{n}. \tag{8}$$

We can make the following observations:

- The proof (see Appendix A) relies on the fact that $\theta_n - \theta_*$ is obtainable in closed form since the cost function is quadratic and thus the recursions are linear, and follows from Polyak and Juditsky (1992).
- The constraint on the step-size γ is equivalent to $\gamma(L + \lambda) \leqslant 1$ where L is the largest eigenvalue of Σ and we thus recover the usual step-size from deterministic gradient descent (Nesterov, 2004).
- When n tends to infinity, the algorithm converges to the minimum of $f(\theta) + \frac{\lambda}{2}\|\theta - \theta_0\|^2$ and our performance guarantee becomes $\lambda^2 \|\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\|^2$. This is the standard "bias term" from regularized ridge regression (Hsu et al., 2014) which we naturally recover here. The term $\frac{\tau^2}{n} \operatorname{tr}\big[\Sigma^2(\Sigma + \lambda I)^{-2}\big]$ is usually referred to as the "variance term" (Hsu et al., 2014), and is equal to $\frac{\tau^2}{n}$ times the quantity $\operatorname{tr}\big[\Sigma^2(\Sigma + \lambda I)^{-2}\big]$, which is often called the degrees of freedom of the ridge regression problem (Gu, 2013).
- For finite n, the first term in Eq. (8) is the usual bias term which depends on the distance from the initial point θ_0 to the objective point θ_* with an appropriate norm. It includes a regularization-based component which is proportional to λ^2 and optimization-based component which depends on $(\gamma n)^{-2}$. The regularization-based bias appears because the algorithm tends to minimize the regularized function instead of the true function f.
- Given Eq. (8), it is natural to set $\lambda\gamma = \frac{1}{n}$, and the two components of the bias term are exactly of the same order leading to $\frac{4}{\gamma^2 n^2}\|\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\|^2$. It corresponds up to a constant factor to the bias term of regularized least-squares (Hsu et al., 2014), but it is achieved by an algorithm accessing only n stochastic gradients. Note that when λ or γ depend on n, this term is not necessarily of order $O(n^{-2})$, as the numerator might be arbitrarily large. Note also that here as in the rest of the paper, we only prove results in the finite horizon setting, meaning that the number of samples is known in advance and the parameters γ, λ may be chosen as functions of n, but remain constant along the iterations (when λ or γ depend on n, our bounds only hold for the last iterate).
- Note that the bias term can also be bounded by $\frac{1}{\gamma n}\|\Sigma^{1/2}(\Sigma + \lambda I)^{-1/2}(\theta_0 - \theta_*)\|^2$ only when $\|\theta_0 - \theta_*\|$ is finite (note the difference in the powers of n and $(\Sigma + \lambda I)^{-1}$). See the proof in Appendix A.2 for details.
- The second term in Eq. (8) is the variance term. It depends on the noise in the gradient. When this one is not structured the variance turns to be also bounded by $\gamma \operatorname{tr}\big(\Sigma(\Sigma + \lambda I)^{-1}\mathbb{E}[\xi_n \otimes \xi_n]\big)$ (see Appendix A.3) and we recover for $\gamma = O(1/\sqrt{n})$, the usual rate of $\frac{1}{\sqrt{n}}$ for SGD in the smooth case (Shalev-Shwartz et al., 2009).
- Overall we get the same performance as the empirical risk minimizer with fixed design, but with an algorithm that performs a single pass over the data.
- When $\lambda = 0$ we recover Lemma 2 of Bach and Moulines (2013). In this case the variance term $\frac{\tau^2 d}{n}$ is optimal over all estimators in $\mathcal{H}$ (Tsybakov, 2008) even without

computational limits, in the sense that no estimator that uses the same information can improve upon this rate.

3.2 Multiplicative/Additive Noise

When the general stochastic oracle in Eq. (6) is considered, the regularized LMS algorithm defined by Eq. (2) takes the form:

$$\theta_n = \big[I - \gamma x_n \otimes x_n - \gamma\lambda I\big]\theta_{n-1} + \gamma y_n x_n + \lambda\gamma\theta_0. \tag{9}$$

We have a very similar result with an additional corrective term (second line below) compared to Lemma 1.

Theorem 2 *Assume* $(\mathcal{A}_{1,2})$*. Consider the recursion in Eq.* (9)*. For any regularization parameter* $\lambda \in \mathbb{R}^+$ *and for any constant step-size* γ *such that* $2\gamma(R^2+2\lambda) \leq 1$ *we have:*

$$\begin{aligned}\mathbb{E}f(\bar\theta_n) - f(\theta_*) \;\leqslant\; & 3\Big(2\lambda + \frac{1}{\gamma n}\Big)^2 \|\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\|^2 + \frac{6\sigma^2}{n+1}\operatorname{tr}\big[\Sigma^2(\Sigma+\lambda I)^{-2}\big] \\ & + 3\frac{\big\|(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\big\|^2 \operatorname{tr}(\Sigma(\Sigma+\lambda I)^{-1})}{\gamma^2(n+1)^2}.\end{aligned}$$

We can make the following remarks:

- The proof (see Appendix B) relies on a bias-variance decomposition, each term being treated separately. We adapt a proof technique from Bach and Moulines (2013) which considers the difference between the recursions in Eq. (9) and in Eq. (7).
- As in Lemma 1, the bias term can also be bounded by $\frac{1}{\gamma n}\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\|^2$ and the variance term by $\gamma\operatorname{tr}[\Sigma(\Sigma+\lambda I)^{-1}\xi_n\otimes\xi_n]$ (see proof in Appendices B.4 and B.5). This is useful in particular when considering unstructured noise.
- The variance term is the same as in the previous case. However there is a residual term that now appears when we go to the fully stochastic oracle (second line). This term will go to zero when γ tends to zero and can be compared to the corrective term which also appears when Hsu et al. (2014) go from fixed to random design. Nevertheless our bounds are more concise than theirs, making significantly fewer assumptions and relying on an efficient single-pass algorithm.
- In this setting, the step-size may not exceed $1/(2(R^2+2\lambda))$, whereas with an additive noise in Lemma 1 the condition is $\gamma \leq 1/(L+\lambda)$, a quantity which can be much bigger than $1/(2(R^2+2\lambda))$, as L is the spectral radius of Σ whereas R^2 is of the order of $\operatorname{tr}(\Sigma)$. Note that in practice, computing L is as hard as computing θ_* so that the step-size $\gamma \propto 1/R^2$ is a good practical choice. See Défossez and Bach (2015) for larger allowed step-sizes that require more information.
- For $\lambda = 0$ the error is bounded by $\frac{3(1+d)}{(\gamma n)^2}\|\Sigma^{-1/2}(\theta_0-\theta_*)\|^2 + \frac{6\sigma^2 d}{n+1}$. We recover results from Défossez and Bach (2015) with a non-asymptotic bound but we lose the advantage of having an asymptotic equivalent (i.e., a limit rather than an upper-bound). We note that the assumption $(\mathcal{A}_{1,2})$ are close to the minimal assumptions required to obtain the optimal rate of convergence of $\sigma^2 d/n$ (Lecué and Mendelson, 2016; Oliveira, 2016)

4. Accelerated Stochastic Averaged Gradient Descent

We study the convergence under the stochastic oracle from Eq. (5) of averaged *accelerated* stochastic gradient descent defined by Eq. (3) which can be rewritten for the least-squares function f as a second-order iterative system with constant coefficients:

$$\theta_n = \big[I - \gamma\Sigma - \gamma\lambda I\big]\big[\theta_{n-1} + \delta(\theta_{n-1}-\theta_{n-2})\big] + \gamma y_n x_n + \gamma\lambda\theta_0. \tag{10}$$

When using averaging, we refer to this algorithm as "averaged-accelerated-SGD".

Theorem 3 *Assume* $(\mathcal{A}_4)$*. For any regularization parameter* $\lambda \in \mathbb{R}_+$ *and for any constant step-size* $\gamma(\Sigma+\lambda I) \preccurlyeq I$*, we have for any* $\delta \in \big[\frac{1-\sqrt{\gamma\lambda}}{1+\sqrt{\gamma\lambda}}, 1\big]$*, for the recursion in Eq. (10):*

$$\mathbb{E}f(\bar\theta_n) - f(\theta_*) \leq 2\Big(\lambda + \frac{36}{\gamma(n+1)^2}\Big)\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\|^2 + 8\tau^2\frac{\operatorname{tr}\big[\Sigma^2(\Sigma+\lambda I)^{-2}\big]}{n+1}.$$

The numerical constants are partially artifacts of the proof (see Appendices C and E). Thanks to a wise use of tight inequalities, the bound is independent of δ and valid for all $\lambda \in \mathbb{R}_+$. This results in the simple following corollary for $\lambda = 0$, which corresponds to the particularly simple recursion (with averaging to obtain $\bar\theta_n$):

$$\theta_n = \big[I - \gamma\Sigma\big](2\theta_{n-1} - \theta_{n-2}) + \gamma y_n x_n. \tag{11}$$

Corollary 4 *Assume* $(\mathcal{A}_4)$*. For any constant step-size* $\gamma\Sigma \preccurlyeq I$*, we have for* $\delta = 1$*,*

$$\mathbb{E}f(\bar\theta_n) - f(\theta_*) \leq 36\frac{\|\theta_0-\theta_*\|^2}{\gamma(n+1)^2} + 8\frac{\tau^2 d}{n+1}. \tag{12}$$

We can make the following observations:

- The proof technique relies on direct moment computations in each eigensubspace obtained by O'Donoghue and Candès (2013) in the deterministic case. Indeed as Σ is a symmetric matrix, the space can be decomposed on an orthonormal eigenbasis of Σ, and the iterations are decoupled in such an eigenbasis. Although we only provide an upper-bound, this is in fact an equality plus other exponentially small terms as shown in the proof which relies on linear algebra, with difficulties arising from the fact that this second-order system can be expressed as a linear stochastic dynamical system with non-symmetric matrices. We only provide a result for additive noise.
- The first bound $\frac{1}{\gamma n^2}\|\theta_0-\theta_*\|^2$ in Eq. (12) corresponds to the usual accelerated rate. It has been shown by Nesterov (2004) to be the optimal rate of convergence for optimizing a quadratic function with a first-order method that can access only to sequences of gradients when $n \leq d$. We recover by averaging an algorithm dedicated to strongly-convex function the traditional convergence rate for non-strongly convex functions. Even if it seems surprising, the algorithm works also for $\lambda = 0$ and $\delta = 1$ (see also simulations in Section 7).

- The second bound in Eq. (12) also matches the optimal statistical performance $\frac{\tau^2 d}{n}$ described in the observations following Lemma 1. Accordingly this algorithm achieves joint bias/variance optimality (when measured in terms of τ^2 and $\|\theta_0 - \theta_*\|^2$).

- We have the same rate of convergence for the bias when compared to the regular Nesterov acceleration without averaging studied by Flammarion and Bach (2015), which corresponds to choosing $\delta_n = 1 - 2/n$ for all n. However if the problem is μ-strongly convex, this latter was shown to also converge at the linear rate $O\big((1-\gamma\mu)^n\big)$ and thus is adaptive to hidden strong-convexity (since the algorithm does not need to know μ to run). This explains that it ends up converging faster for quadratic function since for large n the convergence at rate $1/n^2$ becomes slower than the one at rate $(1-\gamma\mu)^n$ even for very small μ. This is confirmed in our experiments in Section 7. Thanks to this adaptivity, we can also show using the same tools and considering its weighted average $\tilde{\theta}_n = \frac{2}{n(n+1)}\sum_{k=0}^n k\theta_k$ that the bias term of $\mathbb{E}f(\tilde{\theta}_n) - f(\theta_*)$ has a convergence rate of order $\big(\lambda^2 + \frac{1}{\gamma^2(n+1)^4}\big)\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\|^2$ without any change in the variance term. This has to be compared to the bias of averaged SGD $\big(\lambda + \frac{1}{\gamma(n+1)^2}\big)\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\|^2$ in Section 3 and may lead to faster convergence for the bias in presence of hidden strong-convexity.

- Overall, the bias term is improved whereas the variance term is not degraded and acceleration is thus robust to noise in the gradients. Thereby, while second-order finite difference methods for optimizing quadratic functions in the singular case, such as conjugate gradient (Polyak, 1987, Section 6.1) are notoriously highly sensitive to noise, we are able to propose a version which is robust to stochastic noise.

- Note that when there is no assumption on the covariance of the noise we still have the variance bounded by $\frac{\gamma n}{2}\operatorname{tr}\big[\Sigma(\Sigma+\lambda I)^{-1}V\big]$; setting $\gamma = 1/n^{3/2}$ and $\lambda = 0$ leads to the bound $\frac{\|\theta_0-\theta_*\|^2}{\sqrt{n}} + \frac{\operatorname{tr} V}{\sqrt{n}}$. We recover the usual rate for accelerated stochastic gradient in the non-strongly-convex case (Xiao, 2010). When the values of the bias and the variance are known, we can achieve the optimal trade-off of Lan (2012) $\frac{R^2\|\theta_0-\theta_*\|^2}{n^2} + \frac{\|\theta_0-\theta_*\|\sqrt{\operatorname{tr} V}}{\sqrt{n}}$ for $\gamma = \min\Big\{1/R^2, \frac{\|\theta_0-\theta_*\|}{\sqrt{\operatorname{tr} V}n^{3/2}}\Big\}$.

5. Tighter Dimension-Independent Convergence Rates

We have seen in Corollary 4 above that the averaged accelerated gradient algorithm matches the lower bounds $\tau^2 d/n$ and $\frac{L}{n^2}\|\theta_0-\theta_*\|^2$ for the prediction error. However the algorithm performs better in almost all cases except the worst-case scenarios corresponding to the lower bounds. For example the algorithm may still predict well when the dimension d is much bigger than n. Similarly the norm of the optimal predictor $\|\theta_*\|^2$ may be huge and the prediction still good, as gradients algorithms happen to be adaptive to the difficulty of the problem: indeed, if the problem is simpler, the convergence rate of the gradient algorithm will be improved. In this section, we provide such a theoretical guarantee.

The following bound stands for the averaged *accelerated* algorithm. It extends previously known bounds in the kernel least-mean-squares setting (Dieuleveut and Bach, 2015).

Theorem 5 *Assume ($\mathcal{A}_4$); for any regularization parameter $\lambda \in \mathbb{R}_+$ and for any constant step-size such that $\gamma(\Sigma+\lambda I) \preccurlyeq I$ we have for $\delta \in \big[\frac{1-\sqrt{\gamma\lambda}}{1+\sqrt{\gamma\lambda}}, 1\big]$, for the recursion in Eq. (10):*

$$\mathbb{E}f(\bar{\theta}_n) - f(\theta_*) \leq \min_{r\in[0,1],\ b\in[0,1]} \left[2\|\Sigma^{r/2}(\theta_0-\theta_*)\|^2\ \lambda^{-r}\left(\frac{36}{\gamma(n+1)^2} + \lambda\right) + 8\frac{\tau^2\operatorname{tr}(\Sigma^b)\lambda^{-b}}{n+1}\right].$$

The proof is straightforward by upper bounding the terms coming from regularization, depending on $\Sigma(\Sigma+\lambda I)^{-1}$, by a power of λ times the considered quantities. More precisely, the quantity $\operatorname{tr}(\Sigma(\Sigma+\lambda I)^{-1})$ can be seen as an effective dimension of the problem (Gu, 2013), and is upper bounded by $\lambda^{-b}\operatorname{tr}(\Sigma^b)$ for any $b \in [0;1]$. Similarly, $\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1/2}\theta_*\|^2$ can be upper bounded by $\lambda^{-r}\|\Sigma^{r/2}(\theta_0-\theta_*)\|^2$. A detailed proof of these results is given in Appendix D.

In order to benefit from the acceleration, we choose $\lambda = (\gamma n^2)^{-1}$. With such a choice we have the following corollary:

Corollary 6 *Assume ($\mathcal{A}_4$), for any constant step-size $\gamma(\Sigma+\lambda I) \preccurlyeq I$, we have for $\lambda = \frac{1}{\gamma(n+1)^2}$ and $\delta \in \big[1-\frac{2}{n+2}, 1\big]$, for the recursion in Eq. (10):*

$$\mathbb{E}f(\bar{\theta}_n) - f(\theta_*) \leq \min_{r\in[0,1],\ b\in[0,1]} \left[74\ \frac{\|\Sigma^{r/2}(\theta_0-\theta_*)\|^2}{\gamma^{1-r}(n+1)^{2(1-r)}} + 8\frac{\tau^2\gamma^b\operatorname{tr}(\Sigma^b)}{(n+1)^{1-2b}}\right].$$

We can make the following observations:

- The algorithm is independent of r and b, thus all the bounds for different values of (r,b) are valid. This is a strong property of the algorithm, which is indeed adaptative to the regularity and the effective dimension of the problem (once γ is chosen). In situations in which either d is larger than n or $L\|\theta_0-\theta_*\|^2$ is larger than n^2, the algorithm can still enjoy good convergence properties, by adapting to the best values of b and r.

- For $b = 0$ we recover the variance term of Corollary 4, but for $b > 0$ and fast decays of eigenvalues of Σ, the bound may be much smaller; note that we lose in the dependency in n, but typically, for large d, this can be advantageous.

- For $r = 0$ we recover the bias term of Corollary 4 and for $r = 1$ (no assumption at all) the bias is bounded by $\|\Sigma^{1/2}\theta_*\|^2 \leq 4R^2$, which is not going to zero. The smaller r is, the stronger the decrease of the bias with respect to n is (which is coherent with the fact that we have a stronger assumption). Moreover, r is only considered between 0 and 1: indeed, if $r < 0$, the constant$\|(\gamma\Sigma)^{r/2}(\theta_0-\theta_*)\|$ is bigger than $\|\theta_0-\theta_*\|$, but the dependence on n cannot improve beyond $(\gamma n^2)^{-1}$. This is a classical phenomenon called "saturation" (Engl et al., 1996). It is linked with the uniform averaging scheme: here, the bias term cannot forget the initial condition faster than n^{-2}.

- A similar result happens to hold, for averaged gradient descent, with $\lambda = (\gamma n)^{-1}$:

$$\mathbb{E}f(\bar{\theta}_n) - f(\theta_*) \leq \min_{\substack{r\in[-1,1]\\ b\in[0,1]}} \left[(18 + \operatorname{Res}(b,r,n,\gamma))\ \frac{\|\Sigma^{r/2}(\theta_0-\theta_*)\|^2}{\gamma^{1-r}(n+1)^{(1-r)}} + 6\frac{\sigma^2\gamma^b\operatorname{tr}(\Sigma^b)}{(n+1)^{1-b}}\right], \quad (13)$$

where $\mathrm{Res}(b, r, n, \gamma))$ corresponds to a residual term, which is smaller than $\mathrm{tr}(\Sigma^b) n^b \gamma^{1+b}$ if $r \geq 0$ and does not exist otherwise. The bias term's dependence on n is degraded, thus the "saturation" limit is logically pushed down to $r = -1$, which explains the $[-1; 1]$ interval for r. The choice $\lambda = (\gamma n)^{-1}$ arises from Th. 2, in order to balance both components of the bias term $\lambda + (\gamma n)^{-1}$. This result is proved in Appendix D. This recovers the result of Dieuleveut and Bach (2015).

- Considering a non-uniform averaging, as proposed after Theorem 2 the $\min_{0 \leq r \leq 1}$ in Th. 5 and Corollary 6 can be extended to $\min_{-1 \leq r \leq 1}$. Indeed, considering a non-uniform averaging allows to have a faster decreasing bias, pushing the saturation limit observed below.

In finite dimension these bounds for the bias and the variance cannot be said to be optimal independently in any sense we are aware of. Indeed, in finite dimension, the asymptotic rate of convergence for the bias (respectively the variance), when n goes to ∞ is governed by $L\|\theta_0 - \theta_*\|^2/n^2$ (resp. $\tau^2 d/n$). However, we show in the next section that in the setting of non parametric learning in kernel spaces, these bounds lead to the optimal statistical rate of convergence among all estimators (independently of their computational cost). Moving to the infinite-dimensional setting allows to characterize the optimality of the bounds by showing that they achieve the statistical rate when optimizing the bias/variance tradeoff in Corollary 6.

6. Rates of Convergence for Kernel Regression

Computational convergence rates give the speed at which an objective function can decrease depending on the amount of computation which is allowed. Typically, they show how the error decreases with respect to the number of iterations, as in Theorem 2. Statistical rates, however, show how close one can get to some objective given some amount of information which is provided. Statistical rates do not depend on some chosen algorithm: these bounds do not involve computation, on the contrary, they state the best performance that no algorithm can beat, given the information, and without computational limits. In particular, any lower bound on the statistical rate implies a lower bound on the computational rates, if each iteration corresponds to access to some new information, here pairs of observations. Interestingly, many algorithms for the past few years have proved to match, with minimal computations (in general one pass through the data), the statistical rate, emphasizing the importance of carrying together optimization and approximation in large scale learning, as described by Bottou and Bousquet (2008). In a similar flavor, it also appears that regularization can be accomplished through early stopping (Yao et al., 2007; Rudi et al., 2015), highlighting this interplay between computation and statistics.

To characterize the optimality of our bounds, we will show that averaged-accelerated-SGD matches the statistical lower bound in the context of non-parametric estimation. Even if it may be computationally hard or impossible to implement averaged-accelerated-SGD with additive noise in the kernel-based framework below (see remarks following Theorem 8), it leads to the optimal statistical rate for a broader class of problems than averaged-SGD, showing that for a wider set of trade-offs, acceleration is optimal.

A natural extension of the finite-dimensional analysis is the non-parametric setting, especially with reproducing kernel Hilbert spaces. In the setting of non-parametric regression, we consider a probability space $\mathcal{X} \times \mathbb{R}$ with probability distribution ρ, and assume that we are given an i.i.d. sample $(x_i, y_i)_{i=1,\dots,n} \sim \rho^{\otimes n}$, and denote by ρ_X the marginal distribution of x_n in $\mathcal{X}$; the aim of non-parametric least-squares regression is to find a function $g : \mathcal{X} \to \mathbb{R}$, which minimizes the expected risk:

$$f(g) = \frac{1}{2}\mathbb{E}_\rho[(g(x_n) - y_n)^2]. \tag{14}$$

The optimal function g is the conditional expectation $g(x) = \mathbb{E}_\rho(y_n|x)$. In the kernel regression setting, we consider as hypothesis space a reproducing kernel Hilbert space (Aronszajn, 1950; Steinwart and Christmann, 2008; Schölkopf and Smola, 2002) associated with a kernel function K. The space $\mathcal{H}$ is a subspace of the space of squared integrable functions $L^2_{\rho_X}$. We look for a function $g_{\mathcal{H}}$ which satisfies: $f(g_{\mathcal{H}}) = \inf_{g \in \mathcal{H}} f(g)$, and $g_{\mathcal{H}}$ belongs to the closure $\bar{\mathcal{H}}$ of $\mathcal{H}$ (meaning that there exists a sequence of function $g_n \in \mathcal{H}$ such that $\|g_n - g_H\|_{L^2_{\rho_X}} \to 0$). When $\mathcal{H}$ is dense, the minimum is attained for the regression function defined above. This function however *is not* in $\mathcal{H}$ in general. Moreover there exists an operator $\Sigma : \mathcal{H} \to \mathcal{H}$, which extends the finite-dimensional population covariance matrix, that will allow the characterization of the smoothness of $g_{\mathcal{H}}$. This operator is known to be trace class when $\mathbb{E}_{\rho_X}[K(x_n, x_n)] < \infty$.

Data points x_i are mapped into the RKHS, via the feature map: $x \mapsto K_x$, where $K_x : \mathcal{H} \to \mathbb{R}$ is a function in the RKHS, such that $K_x : y \mapsto K(x, y)$. The reproducing property[3] allows to express the minimization problem (14) as a least-squares linear regression problem: for any $g \in \mathcal{H}$, $f(g) = \frac{1}{2}\mathbb{E}_\rho[(\langle g, K_{x_n}\rangle_{\mathcal{H}} - y_n)^2]$, and can thus be seen as an extension to the infinite-dimensional setting of linear least-squares regression.

However, in such a setting, both quantities $\|\Sigma^{r/2}\theta_*\|_{\mathcal{H}}$ (where $\|\cdot\|_{\mathcal{H}}$ stands for the norm associated with the inner product in the Hilbert space $\mathcal{H}$) and $\mathrm{tr}(\Sigma^b)$ *may exist or not*. It thus arises as a natural *assumption* to consider the smaller $r \in [-1; 1]$ and the smaller $b \in [0; 1]$ such that

- $\|\Sigma^{r/2}\theta_*\|_{\mathcal{H}} < \infty$ (meaning that $\Sigma^{r/2}\theta_* \in \mathcal{H}$), $(\mathcal{A}_5)$
- $\mathrm{tr}(\Sigma^b) < \infty$. $(\mathcal{A}_6)$

The quantities considered in Sections 2 and 5 are the natural finite-dimensional twins of these assumptions. However in infinite dimension a quantity may exist or not and it is thus an assumption to consider its existence, whereas it can only be characterized by its value, big or small, in finite dimension. The first assumption is generally called the "source condition", the second one the "capacity condition".

In the last decade, De Vito et al. (2005); Cucker and Smale (2002) studied non-parametric least-squares regression in the RKHS framework. These works were extended to derive rates of convergence depending on assumption $(\mathcal{A}_5)$: Ying and Pontil (2008) studied unregularized stochastic gradient descent and derived asymptotic rate of convergence $O(n^{-\frac{1-r}{2-r}})$, for $r \leq 1$ and proved that one could derive similar rates of convergence for $0 \leq r \leq 1$ from

3. It states that for any function $g \in \mathcal{H}$, $\langle g, K_x\rangle_{\mathcal{H}} = g(x)$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the scalar product in the Hilbert space.

Zhang (2004), who studies stochastic gradient descent with averaging; whereas Tarrès and Yao (2011) give similar performance for $-1 \leq r \leq 0$. Interestingly, Ying and Pontil (2008) do not have saturation, meaning that the rate still improves for r smaller than -1. As it will appear, any algorithm based on a uniform averaging scheme faces a saturation issue : one cannot forget initial conditions faster than n^{-2}, which makes the algorithm sub-optimal in situations in which the optimal predictor is very smooth ($\mathcal{A}_5$ holds with $r \leq -1$). However, these papers only prove rates in the capacity-independent setting, meaning without assumption on the spectrum of the covariance matrix. Although the rate $O(n^{-\frac{1-r}{2-r}})$ is optimal in this setting, it comes from a worst-case analysis. Considering the capacity-dependent setting is more challenging, but allows to derive tighter and more realistic rates (a capacity condition always stands under the trace class assumption that is made). Moreover, the capacity-independent setting also does not allow to recover finite-dimensional rates. Up to our knowledge, there is no one pass stochastic gradient algorithm which does not have saturation while getting the minimax rate under both the capacity condition and source condition. In a recent work, Lin and Rosasco (2016) achieves optimality without saturation with multiple passes. We show in the next paragraphs that we can derive a tighter and optimal rate for both averaged-SGD (recovering results from Dieuleveut and Bach (2015)) and averaged-accelerated-SGD, for a larger class of kernels for the latter. Note that the averaging scheme for the RKHS setting was originally considered by Yao (2006).

We will first describe results for averaged-SGD, then increase the validity region of these rates (which depends on r, b) using averaged accelerated SGD. We show that the derived rates match statistical rates for our setting and thus our algorithms reach the optimal prediction performance for certain b and r.

6.1 Averaged SGD

We have the following result, proved in Appendix D and following from Theorem 2: for some fixed b, r, we choose the best step-size γ, that optimizes the bias-variance trade-off, while still satisfying the constraint $\gamma \leq 1/(2R^2)$. We get a result for the stochastic oracle (multiplicative/additive noise).

Theorem 7 *With $\lambda = \frac{1}{\gamma n}$, we have, if $r \leq b$, under Assumptions $(\mathcal{A}_{1,2,5,6})$ and the stochastic oracle Eq. (6), for any constant step-size γ such that $2\gamma(R^2 + 2\lambda) \leq 1$, with $\gamma \propto n^{\frac{-b+r}{b+1-r}}$, for the recursion in Eq. (9):*

$$\mathbb{E} f(\bar{\theta}_n) - f(\theta_*) \quad \leq \quad \left((27 + o(1)) \left\| \Sigma^{r/2} (\theta_0 - \theta_*) \right\|^2 + 6\sigma^2 \operatorname{tr}(\Sigma^b) \right) n^{-\frac{1-r}{b+1-r}}.$$

We can make the following remarks:

- The term $o(1)$ stands for a quantity which is decreasing to 0 when $n \to \infty$. More specifically, this constant is smaller than $3 \operatorname{tr}(\Sigma^b)$ divided by n^χ, where χ is bigger than 0 (see Appendix D). The result comes from Eq. (13) (which follows from Theorem 5), with the choice of the optimal step-size.
- We recover the same errors bounds as in Dieuleveut and Bach (2015), but with a simpler analysis resulting from the consideration of the regularized version of the problem associated with a choice of λ. However, we only recover rates in the finite horizon setting.
- This result shows that we get the optimal rate of convergence under Assumptions $(\mathcal{A}_{5,6})$, for $r \leq b$. This point will be discussed in more details after Theorem 8.

We now turn to the averaged accelerated SGD algorithm. We prove that it enjoys the optimal rate of convergence for a larger class of problems, but only for the additive noise which corresponds to knowing the distribution of x_n.

6.2 Averaged-Accelerated SGD

Similarly, choosing the best step-size γ, it comes from Theorem 5, that in the RKHS setting, under additional Assumptions $(\mathcal{A}_{5,6})$, we have for the the averaged accelerated algorithm the following result:

Theorem 8 *With $\lambda = \frac{1}{\gamma n^2}$, we have, if $r \leq b + 1/2$, under Assumptions $(\mathcal{A}_{4,5,6})$, for any constant step-size $\gamma \leq \frac{1}{L+\lambda}$, with $\gamma \propto n^{\frac{-2b+2r-1}{b+1-r}}$, for the recursion in Eq. (10):*

$$\mathbb{E} f(\bar{\theta}_n) - f(\theta_*) \quad \leq \quad \left(74 \left\| \Sigma^{r/2} (\theta_0 - \theta_*) \right\|^2 + 8\tau^2 \operatorname{tr}(\Sigma^b) \right) n^{-\frac{1-r}{b+1-r}}.$$

We can make the following remarks:

- The rate $\frac{1-r}{b+1-r}$ is always between 0 and 1, and improves when our assumptions gets stronger (r getting smaller, b getting smaller). Ultimately, with $b \to 0$, and $r \to -1$, we recover the finite-dimensional n^{-1} rate.
- We can achieve this optimal rate when $r \leq b + 1/2$. Beyond, if $r > b + 1/2$, the rate is only $n^{-2(1-r)}$. Indeed, the bias term cannot decrease faster than $n^{-2(1-r)}$, as γ is compelled to be upper bounded.
- The same phenomenon appears in the un-accelerated averaged situation, as shown by Theorem 7, but the critical value was then $r \leq b$. There is thus a region (precisely $b < r \leq b + 1/2$) in which only the accelerated algorithm gets the optimal rate of convergence. Note that we increase the optimality region towards optimization problems which are more ill-conditioned, naturally benefiting from acceleration.
- This algorithm cannot be computed in practice (at least with computational limits). Indeed, without any further assumption on the kernel K, it is not possible to compute images of vectors by the covariance operator Σ in the RKHS. However, as explained in the following remark, this is enough to show some form of optimality of our algorithm.

  Note that the easy computability is a great advantage of the multiplicative/additive noise variant of the algorithms, for which the current point θ_n can always be expressed as a finite sum of features $\theta_n = \sum_{i=1}^n \alpha_i K_{x_i}$, with $\alpha_i \in \mathbb{R}$, leading to a tractable algorithm. An accelerated variant of SGD naturally arises from our algorithm, when considering this stochastic oracle from Eq. (6). Such a variant can be implemented but does not behave similarly for large step sizes, say, $\gamma \simeq 1/(2R^2)$. It is an open problem to prove convergence results for averaged accelerated gradient under this multiplicative/additive noise.

- These rates happen to be optimal from a statistical perspective, meaning that no algorithm which is given access to the sample points and the distribution of x_n can perform better for all functions that satisfy assumption $(\mathcal{A}_6)$, for a kernel satisfying $(\mathcal{A}_5)$. Indeed it is equivalent to assuming that the function lives in some ellipsoid in the space of squared integrable functions. Note that the statistical minimization problem (and thus the lower bound) does not depend on the kernel, and is valid without computational limits. The case of learning with kernels is studied by Caponnetto and De Vito (2007) which shows these minimax convergence rates under $(\mathcal{A}_{5,6})$, under assumption that $-1 \leq r \leq 0$ (but state that it can be easily extended to $0 \leq r \leq 1$). They do not assume knowledge of the distribution of the inputs; however, Massart (2007) and Tsybakov (2008) discuss optimal rates on ellipsoids, and Györfi et al. (2006) proves similar results for certain class of functions under a known distribution for the input data, showing that the knowledge of the distribution does not make any difference. This minimax statistical rate stands without computational limits and is thus valid for both algorithms (additive noise that corresponds to knowing Σ, and multiplicative/additive noise). The optimal tradeoff is derived for an extended region of b, r (namely $r \leq b + 1/2$ instead of $r \leq b$) in the accelerated case which shows the improvement upon non-accelerated averaged SGD.

- The choice of the optimal γ is difficult in practice, as the parameters b, r are unknown, and this remains an open problem in general (see, e.g., Birgé, 2001, for some methods for non-parametric regression), even if in the capacity-independent setting, Orabona (2014) has proposed an algorithm that adapts to the unknown parameter r.

- Note that we do not give rates in terms of norm in the RHKS (i.e., an upper bound on $\|\bar{\theta}_n - \theta_*\|_{\mathcal{H}}$), because we mainly aim at extending optimality of prediction error rate to ill-conditioned cases (i.e. situations for which $r \geq b \geq 0$). In such a situation, Hilbert spaces norm bounds would not be relevant as the optimal estimator does not even live in the RKHS.

7. Experiments

We illustrate now our theoretical results on synthetic examples. For $d = 25$ we consider normally distributed inputs x_n with random covariance matrix Σ which has eigenvalues $1/i^3$, for $i = 1, \dots, d$, and random optimum θ_* and starting point θ_0 such that $\|\theta_0 - \theta_*\| = 1$. The outputs y_n are generated from a linear function with homoscedastic noise with unit signal to noise-ratio ($\sigma^2 = 1$), we take $R^2 = \operatorname{tr} \Sigma$ the average radius of the data and a step-size $\gamma = 1/R^2$ and $\lambda = 0$. The additive noise oracle is used. We show results averaged over 10 replications.

We compare the performance of averaged SGD (AvSGD), AccSGD (usual Nesterov acceleration for convex functions) and our novel averaged accelerated SGD from Section 4 AvAccSGD (which is not the averaging of AccSGD because the momentum term is proportional to $1 - 3/n$ for AccSGD instead of being equal to 1 for AvAccSGD), on two different problems: one deterministic ($\|\theta_0 - \theta_*\| = 1$, $\sigma^2 = 0$) which will illustrate how the bias term behaves, and one purely stochastic ($\|\theta_0 - \theta_*\| = 0$, $\sigma^2 = 1$) which will illustrate how the variance term behaves. For the bias (left plot of Figure 1), AvSGD converges at

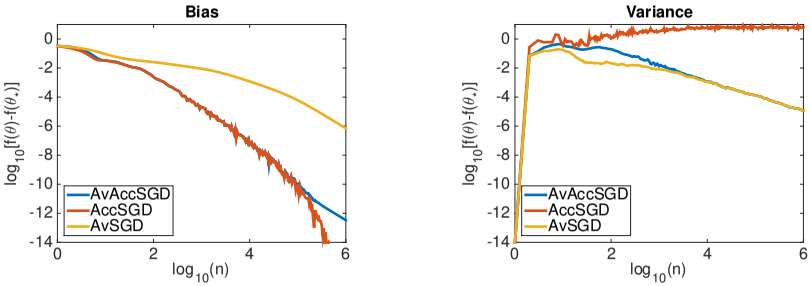


Figure 1: Synthetic problem ($d = 25$) and $\gamma = 1/R^2$. Left: Bias. Right: Variance.

speed $O(1/n)$, while AvAccSGD and AccSGD converge both at speed $O(1/n^2)$. However, as mentioned in the observations following Corollary 4, AccSGD takes advantage of the hidden strong convexity of the least-squares function and starts converging linearly at the end. For the variance (right plot of Figure 1), AccSGD is not converging to the optimum and keeps oscillating whereas AvSGD and AvAccSGD both converge to the optimum at a speed $O(1/n)$. However AvSGD remains slightly faster in the beginning.

Note that for small n, or when the bias $L\|\theta_0 - \theta_*\|^2/n^2$ is much bigger than the variance $\sigma^2 d/n$, the bias may have a stronger effect, although asymptotically, the variance always dominates. It is thus essential to have an algorithm which is optimal in both regimes; this is achieved by AvAccSGD.

8. Conclusion

In this paper, we showed that stochastic *averaged* accelerated gradient descent was robust to structured noise in the gradients present in least-squares regression. Beyond being the first algorithm which is jointly optimal in terms of both bias and finite-dimensional variance, it is also adapted to finer assumptions such as fast decays of the covariance matrices or optimal predictors with large norms.

Our current analysis is performed for least-squares regression. While it could be directly extended to smooth losses through efficient online Newton methods (Bach and Moulines, 2013), an extension to all smooth or self-concordant-like functions (Bach, 2014) would widen its applicability. Moreover, our accelerated gradient analysis is performed for additive noise (i.e., for least-squares regression, with knowledge of the population covariance matrix) and it would be interesting to study the robustness of our results in the contexts of least-mean squares and online learning. Finally, our analysis relies on single observations per iteration and could be made finer by using mini-batches (Cotter et al., 2011; Dekel et al., 2012), which should not change the variance term but could impact the bias term.

Acknowledgments

The authors would like to thank Damien Garreau for interesting discussions and the reviewers for their constructive and helpful comments.

Appendix A. Proofs of Section 3

We first prove the results of Section 3.

A.1 Proof of Lemma 1

We proof here Lemma 1 which is the extension of Lemma 2 of Bach and Moulines (2013) for the regularized case. The proof technique relies on the fact that recursions in Eq. (7) are linear since the cost function is quadratic which allows us to obtain $\theta_n - \theta_*$ in closed form.

For any regularization parameter $\lambda \in \mathbb{R}_+$ and any constant step-size $\gamma(\Sigma + \lambda I) \preccurlyeq I$ we may rewrite the regularized stochastic gradient recursion in Eq. (7) as:

$$\theta_n - \theta_* = \big[I - \gamma\Sigma - \gamma\lambda I\big](\theta_{n-1} - \theta_*) + \gamma\xi_n + \lambda\gamma(\theta_0 - \theta_*).$$

We thus get for $n \geq 1$ the expansion

$$\begin{aligned}
\theta_n - \theta_* &= (I - \gamma\Sigma - \gamma\lambda I)^n(\theta_0 - \theta_*) + \gamma\sum_{k=1}^{n}(I - \gamma\Sigma - \gamma\lambda I)^{n-k}\xi_k \\
&\quad + \gamma\lambda\sum_{k=1}^{n}(I - \gamma\Sigma - \gamma\lambda I)^{n-k}(\theta_0 - \theta_*) \\
&= (I - \gamma\Sigma - \gamma\lambda I)^n(\theta_0 - \theta_*) + \gamma\sum_{k=1}^{n}(I - \gamma\Sigma - \gamma\lambda I)^{n-k}\xi_k \\
&\quad + \lambda\big[I - (I - \gamma\Sigma - \gamma\lambda I)^n\big](\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*) \\
&= (I - \gamma\Sigma - \gamma\lambda I)^n[I - \lambda(\Sigma + \lambda I)^{-1}](\theta_0 - \theta_*) + \gamma\sum_{k=1}^{n}(I - \gamma\Sigma - \gamma\lambda I)^{n-k}\xi_k \\
&\quad + \lambda(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*).
\end{aligned}$$

We then have using the definition of the average

$$\begin{aligned}
n(\bar{\theta}_{n-1} - \theta_*) &= \sum_{j=0}^{n-1}(\theta_j - \theta_*) \\
&= \sum_{j=0}^{n-1}(I - \gamma\Sigma - \gamma\lambda I)^j[I - \lambda(\Sigma + \lambda I)^{-1}](\theta_0 - \theta_*) + \gamma\sum_{j=0}^{n-1}\sum_{k=1}^{j}(I - \gamma\Sigma - \gamma\lambda I)^{n-k}\xi_k \\
&\quad + n\lambda(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*).
\end{aligned}$$

For which we will compute the two sums separately

$$\begin{aligned}
&\sum_{j=0}^{n-1}(I - \gamma\Sigma - \gamma\lambda I)^j[I - \lambda(\Sigma + \lambda I)^{-1}](\theta_0 - \theta_*) \\
&\qquad = \frac{1}{\gamma}\big[I - (I - \gamma\Sigma - \gamma\lambda I)^n\big](\Sigma + \lambda I)^{-1}[I - \lambda(\Sigma + \lambda I)^{-1}](\theta_0 - \theta_*),
\end{aligned}$$

and

$$\begin{aligned}
\gamma\sum_{j=0}^{n-1}\sum_{k=1}^{j}(I - \gamma\Sigma - \gamma\lambda I)^{j-k}\xi_k &= \gamma\sum_{k=1}^{n-1}\Big(\sum_{j=k}^{n-1}(I - \gamma\Sigma - \gamma\lambda I)^{j-k}\Big)\xi_k \\
&= \gamma\sum_{k=1}^{n-1}\Big(\sum_{j=0}^{n-1-k}(I - \gamma\Sigma - \gamma\lambda I)^{j}\Big)\xi_k \\
&= \sum_{k=1}^{n-1}\big[I - (I - \gamma\Sigma - \gamma\lambda I)^{n-k}\big](\Sigma + \lambda I)^{-1}\xi_k.
\end{aligned}$$

Gathering the three terms together, we thus have

$$\begin{aligned}
n(\bar{\theta}_{n-1} - \theta_*) &= \frac{1}{\gamma}\big[I - (I - \gamma\Sigma - \gamma\lambda I)^n\big](\Sigma + \lambda I)^{-1}[I - \lambda(\Sigma + \lambda I)^{-1}](\theta_0 - \theta_*) \\
&\quad + \sum_{k=1}^{n-1}\big[I - (I - \gamma\Sigma - \gamma\lambda I)^{n-k}\big](\Sigma + \lambda I)^{-1}\xi_k + n\lambda(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*) \\
&= \Big[\frac{1}{\gamma}\big[I - (I - \gamma\Sigma - \gamma\lambda I)^n\big][I - \lambda(\Sigma + \lambda I)^{-1}] + n\lambda I\Big](\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*) \\
&\quad + \sum_{k=1}^{n-1}\big[I - (I - \gamma\Sigma - \gamma\lambda I)^{n-k}\big](\Sigma + \lambda I)^{-1}\xi_k.
\end{aligned}$$

Using standard martingale square moment inequalities which amount to consider ξ_i, $i = 1, \ldots, n$ independent, the variance of the sum is the sum of variances and we have for $V = \mathbb{E}\xi_n \otimes \xi_n$

$$\begin{aligned}
&n^2\mathbb{E}\|\Sigma^{1/2}(\bar{\theta}_{n-1} - \theta_*)\|^2 = \textstyle\sum_{k=1}^{n-1}\operatorname{tr}\big[I - (I - \gamma\Sigma - \gamma\lambda I)^{n-k}\big]^2\Sigma(\Sigma + \lambda I)^{-2}V \\
&+ \Big\|\Big[\frac{1}{\gamma}\big[I - (I - \gamma\Sigma - \gamma\lambda I)^n\big][I - \lambda(\Sigma + \lambda I)^{-1}] + n\lambda I\Big]\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\Big\|^2. \quad (15)
\end{aligned}$$

Since all the matrices in this equality are symmetric positive-definite we are allowed to bound

$$\begin{aligned}
\Big[\frac{1}{\gamma}\big[I - (I - \gamma\Sigma - \gamma\lambda I)^n\big][I - \lambda(\Sigma + \lambda I)^{-1}] + n\lambda I\Big] &\preccurlyeq \Big(\frac{1}{\gamma} + n\lambda\Big)I \quad (16) \\
\big[I - (I - \gamma\Sigma - \gamma\lambda I)^{n-k}\big]^2 &\preccurlyeq I.
\end{aligned}$$

This concludes proof of the Lemma 1

$$\mathbb{E}\|\Sigma^{1/2}(\bar{\theta}_{n-1}-\theta_*)\|^2 \leq \Big(\frac{1}{n\gamma}+\lambda\Big)^2\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\|^2 + \frac{1}{n}\operatorname{tr}\Sigma(\Sigma+\lambda I)^{-2}V. \quad (17)$$

A.2 Proof When Only $\|\theta_0-\theta_*\|$ Is Finite

Unfortunately $\|\Sigma^{-1}(\theta_0-\theta_*)\|$ may not be finite. However we can use that for all $u\in[0,1]$ we have $\frac{1-(1-u)^n}{nu}\leq 1$[4] and have therefore the bound

$$\begin{aligned}
&\Big[\frac{1}{\gamma}\big[I-(I-\gamma\Sigma-\gamma\lambda I)^n\big][I-\lambda(\Sigma+\lambda I)^{-1}]+n\lambda I\Big][\Sigma+\lambda I]^{-1}\\
&\preccurlyeq \Big[\frac{1}{\gamma}\big[I-(I-\gamma\Sigma-\gamma\lambda I)^n\big]+n\lambda I\Big][\Sigma+\lambda I]^{-1}\\
&\preccurlyeq \Big[\frac{1}{\gamma}\big[I-(I-\gamma\Sigma-\gamma\lambda I)^n\big][\Sigma+\lambda I]^{-1}+n\lambda[\Sigma+\lambda I]^{-1}\Big]\\
&\preccurlyeq I+nI.
\end{aligned}$$

Combining with Eq. (16) we have

$$\Big\|\Big[\frac{1}{\gamma}\big[I-(I-\gamma\Sigma-\gamma\lambda I)^n\big][I-\lambda(\Sigma+\lambda I)^{-1}]+n\lambda I\Big]\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\Big\|^2 \leq (n+1)\Big(\frac{1}{\gamma}+n\lambda\Big)\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\|^2,$$

which implies that

$$\mathbb{E}\|\Sigma^{1/2}(\bar{\theta}_{n-1}-\theta_*)\|^2 \leq 2\Big(\frac{1}{n\gamma}+\lambda\Big)\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\|^2 + \frac{1}{n}\operatorname{tr}\Sigma(\Sigma+\lambda I)^{-2}V, \quad (18)$$

which is interesting when only $\|\theta_0-\theta_*\|$ is finite.

A.3 Proof When the Noise Is Not Structured

The bound in Eq. (17) becomes less interesting when the noise is not structured. However using the same technique we have that $\big[I-(I-\gamma\Sigma-\gamma\lambda I)^{n-k}\big]^2(\Sigma+\lambda I)^{-1}\preccurlyeq(n-k)\gamma I$ and we get the following upper-bound on the variance

$$\begin{aligned}
\sum_{k=1}^n \operatorname{tr}\big[I-(I-\gamma\Sigma-\gamma\lambda I)^{n-k}\big]^2\Sigma(\Sigma+\lambda I)^{-2}V &\leq \gamma\sum_{k=1}^n(n-k)\operatorname{tr}\Sigma(\Sigma+\lambda I)^{-1}V\\
&\leq \gamma\frac{n(n+1)}{2}\operatorname{tr}\Sigma(\Sigma+\lambda I)^{-1}V.
\end{aligned}$$

4. since $\frac{1-(1-u)^n}{u}=\sum_{k=0}^n(1-u)^k\leq n$

Therefore we get

$$\mathbb{E}\|\Sigma^{1/2}(\bar{\theta}_{n-1}-\theta_*)\|^2 \leq \Big(\frac{1}{n\gamma}+\lambda\Big)^2\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\|^2 + \gamma\operatorname{tr}\Sigma(\Sigma+\lambda I)^{-1}V, \quad (19)$$

which is meaningful when the noise is not structured.

Appendix B. Proof of Theorem 2

In this section, we will prove Theorem 2. The proof relies on a decomposition of the error as the sum of three main terms which will be studied separately. We state decomposition in Section B.1 then prove upper bounds for the different terms in Sections B.2 and B.3.

B.1 Expansion of the Recursion

We may rewrite the regularized stochastic gradient recursion as:

$$\begin{aligned}
\theta_n &= \big[I-\gamma x_n\otimes x_n-\gamma\lambda I\big]\theta_{n-1}+\gamma\varepsilon_n x_n+\gamma\langle x_n,\theta_*\rangle x_n+\lambda\gamma\theta_0\\
\theta_n-\theta_* &= \big[I-\gamma x_n\otimes x_n-\gamma\lambda I\big](\theta_{n-1}-\theta_*)+\gamma\varepsilon_n x_n+\lambda\gamma(\theta_0-\theta_*).
\end{aligned}$$

For $i\geqslant k$, let

$$M(i,k)=\big[I-\gamma x_i\otimes x_i-\gamma\lambda I\big]\cdots\big[I-\gamma x_k\otimes x_k-\gamma\lambda I\big]$$

be an operator from $\mathcal{H}$ to $\mathcal{H}$. We have the expansion

$$\theta_n-\theta_*=M(n,1)(\theta_0-\theta_*)+\gamma\sum_{k=1}^n M(n,k+1)\varepsilon_k x_k+\gamma\sum_{k=1}^n M(n,k+1)\lambda(\theta_0-\theta_*).$$

Our goal is to study these three terms separately and bound $\|\Sigma^{1/2}(\bar{\theta}_n-\theta_*)\|$ for each of them.

B.2 Regularization-Based Bias Term

This is the term: $\theta_n-\theta_*=\gamma\sum_{k=1}^n M(n,k+1)\lambda(\theta_0-\theta_*)$, which corresponds to the recursion

$$\theta_n-\theta_*=\big(I-\gamma x_n\otimes x_n-\gamma\lambda I\big)(\theta_{n-1}-\theta_*)+\lambda\gamma(\theta_0-\theta_*), \quad (20)$$

initialized with $\theta_0=\theta_*$, and no noise.

Following the proof technique of Bach and Moulines (2013), we are going to consider a related recursion by replacing in Equation (20) the operator $x_n\otimes x_n$ by its expectation Σ. Thus, we consider η_n defined as

$$\eta_n-\theta_*=\gamma\sum_{k=1}^n(I-\gamma\Sigma-\lambda\gamma I)^{n-k}\lambda(\theta_0-\theta_*),$$

which satisfies the recursion (with initialization $\eta_0=\theta_*$) and

$$\eta_n-\theta_*=\big[I-\gamma\Sigma-\lambda\gamma I\big](\eta_{n-1}-\theta_*)+\lambda\gamma(\theta_0-\theta_*).$$

In order to bound $\|\Sigma^{1/2}(\theta_n-\theta_*)\|$, we will independently bound $\|\Sigma^{1/2}(\eta_n-\theta_*)\|$ and $\|\Sigma^{1/2}(\theta_n-\eta_n)\|$ using Minkowski's inequality.

Bounding $\|\Sigma^{1/2}(\theta_n - \eta_n)\|$. We have $\theta_0 - \eta_0 = 0$, and

$$\theta_n - \eta_n = \big[I - \gamma x_n \otimes x_n - \lambda\gamma I\big](\theta_{n-1} - \eta_{n-1}) + \gamma\big[\Sigma - x_n \otimes x_n\big](\eta_{n-1} - \theta_*).$$

We can now bound the recursion for $\theta_n - \eta_n$ as follows, using standard online learning proofs (Nemirovski et al., 2009):

$$\begin{aligned}\|\theta_n - \eta_n\|^2 \leqslant\ & \|\theta_{n-1} - \eta_{n-1}\|^2 - 2\gamma\big\langle \theta_{n-1} - \eta_{n-1}, (x_n \otimes x_n + \lambda I)(\theta_{n-1} - \eta_{n-1})\big\rangle \\ & + 2\gamma\big\langle \theta_{n-1} - \eta_{n-1}, \big[\Sigma - x_n \otimes x_n\big](\eta_{n-1} - \theta_*)\big\rangle \\ & + \gamma^2\big\| \big[x_n \otimes x_n + \lambda I\big](\theta_{n-1} - \eta_{n-1}) - \big[\Sigma - x_n \otimes x_n\big](\eta_{n-1} - \theta_*)\big\|^2.\end{aligned}$$

By taking conditional expectations given $\mathcal{F}_{n-1}$, we get, using first the fact that $\mathbb{E}(\Sigma - x_n \otimes x_n | \mathcal{F}_{n-1}) = 0$ and the inequality $(a+b)^2 \leq 2(a^2+b^2)$, then developing and using $\mathbb{E}[(x_n \otimes x_n)^2] \leq R^2\Sigma$, which is assumption $\mathcal{A}_1$.

$$\begin{aligned}\mathbb{E}\big(\|\theta_n - \eta_n\|^2 | \mathcal{F}_{n-1}\big) \leqslant\ & \|\theta_{n-1} - \eta_{n-1}\|^2 - 2\gamma\big\langle \theta_{n-1} - \eta_{n-1}, (\Sigma + \lambda I)(\theta_{n-1} - \eta_{n-1})\big\rangle \\ & + 2\gamma^2\mathbb{E}\big(\big\| \big[x_n \otimes x_n + \lambda I\big](\theta_{n-1} - \eta_{n-1})\big\|^2 | \mathcal{F}_{n-1}\big) \\ & + 2\gamma^2\mathbb{E}\big(\big\| \big[\Sigma - x_n \otimes x_n\big](\eta_{n-1} - \theta_*)\big\|^2 | \mathcal{F}_{n-1}\big) \\ \leqslant\ & \|\theta_{n-1} - \eta_{n-1}\|^2 - 2\gamma\big\langle \theta_{n-1} - \eta_{n-1}, (\Sigma + \lambda I)(\theta_{n-1} - \eta_{n-1})\big\rangle \\ & + 2\gamma^2\big\langle \theta_{n-1} - \eta_{n-1}, (R^2\Sigma + \lambda^2 I + 2\lambda\Sigma)(\theta_{n-1} - \eta_{n-1})\big\rangle \\ & + 2\gamma^2 R^2\langle \eta_{n-1} - \theta_*, \Sigma\rangle \\ \leqslant\ & \|\theta_{n-1} - \eta_{n-1}\|^2 - 2\gamma\big[1 - \gamma(R^2 + 2\lambda)\big]\big\langle \theta_{n-1} - \eta_{n-1}, \Sigma(\theta_{n-1} - \eta_{n-1})\big\rangle \\ & + 2\gamma^2 R^2\langle \eta_{n-1} - \theta_*, \Sigma(\eta_{n-1} - \theta_*)\rangle.\end{aligned}$$

This leads by taking full expectations and moving terms to

$$\begin{aligned}\mathbb{E}\big\langle \theta_{n-1} - \eta_{n-1}, \Sigma(\theta_{n-1} - \eta_{n-1})\big\rangle \leqslant\ & \frac{1}{2\gamma\big[1 - \gamma(R^2 + 2\lambda)\big]}\big[\mathbb{E}\|\theta_{n-1} - \eta_{n-1}\|^2 - \mathbb{E}\|\theta_n - \eta_n\|^2\big] \\ & + \frac{\gamma R^2}{1 - \gamma(R^2 + 2\lambda)}\langle \eta_{n-1} - \theta_*, \Sigma(\eta_{n-1} - \theta_*)\rangle.\end{aligned}$$

Thus, if $\gamma(R^2 + 2\lambda) \leqslant \frac{1}{2}$

$$\begin{aligned}\mathbb{E}\big\langle \theta_{n-1} - \eta_{n-1}, \Sigma(\theta_{n-1} - \eta_{n-1})\big\rangle \leqslant\ & \frac{1}{\gamma}\big[\mathbb{E}\|\theta_{n-1} - \eta_{n-1}\|^2 - \mathbb{E}\|\theta_n - \eta_n\|^2\big] \\ & + 2\gamma R^2\mathbb{E}\langle \eta_{n-1} - \theta_*, \Sigma(\eta_{n-1} - \theta_*)\rangle.\end{aligned}$$

This leads to, summing and using initial conditions $\theta_0 - \eta_0 = 0$, then using convexity to upper bound $\big\langle \bar\theta_n - \bar\eta_n, \Sigma(\bar\theta_n - \bar\eta_n)\big\rangle \leq \frac{1}{n+1}\sum_{k=0}^n \big\langle \theta_k - \eta_k, \Sigma(\theta_k - \eta_k)\big\rangle$,

$$\mathbb{E}\big\langle \bar\theta_n - \bar\eta_n, \Sigma(\bar\theta_n - \bar\eta_n)\big\rangle \leqslant \frac{2\gamma R^2}{n+1}\sum_{k=0}^n \langle \eta_k - \theta_*, \Sigma(\eta_k - \theta_*)\rangle.$$

Bounding $\|\Sigma^{1/2}(\eta_n - \theta_*)\|$. Moreover we have:

$$\begin{aligned}\eta_n - \theta_* &= \lambda(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*) - (I - \gamma\Sigma - \lambda\gamma I)^n\big[\lambda(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\big] \\ \bar\eta_n - \theta_* &= \lambda(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*) - \frac{1}{n+1}\sum_{k=0}^n (I - \gamma\Sigma - \lambda\gamma I)^k\big[\lambda(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\big] \\ &= \lambda(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*) \\ &\qquad - \frac{1}{n+1}\gamma^{-1}(\Sigma + \lambda I)^{-1}\big[I - (I - \gamma\Sigma - \lambda\gamma I)^{n+1}\big]\big[\lambda(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\big].\end{aligned}$$

This leads using Minkowski inequality to

$$\begin{aligned}\big(\mathbb{E}\|\Sigma^{1/2}(\eta_n - \theta_*)\|^2\big)^{1/2} &\leqslant \|\lambda\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\| \\ \big(\mathbb{E}\|\Sigma^{1/2}(\bar\eta_n - \theta_*)\|^2\big)^{1/2} &\leqslant \|\lambda\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\|.\end{aligned}$$

Thus this part is such that

$$\begin{aligned}\big(\mathbb{E}\|\Sigma^{1/2}(\bar\theta_n - \theta_*)\|^2\big)^{1/2} &\leqslant \|\lambda\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\| \\ &\quad + \Big(2\gamma R^2\|\lambda\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\|^2\Big)^{1/2} \\ &\leqslant \|\lambda\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\|\big(1 + \sqrt{2\gamma R^2}\big),\end{aligned}$$

that gives the first bound on the regularization-based bias

$$\mathbb{E}\|\Sigma^{1/2}(\bar\theta_n - \theta_*)\|^2 \leqslant \|\lambda\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\|^2\big(1 + \sqrt{2\gamma R^2}\big)^2. \tag{21}$$

B.3 Expansion without the Regularization Term

We will follow here the outline of the proof of Györfi and Walk (1996) which considers a full expansion of the function value $\|\Sigma^{1/2}(\bar\theta_n - \theta_*)\|^2$. This corresponds to

$$\theta_n - \theta_* = M(n,1)(\theta_0 - \theta_*) + \gamma\sum_{k=1}^n M(n,k+1)\varepsilon_k x_k.$$

We have

$$\mathbb{E}\sum_{i=0}^n\sum_{j=0}^n \langle \theta_i - \theta_*, \Sigma(\theta_j - \theta_*)\rangle = \mathbb{E}\sum_{i=0}^n \langle \theta_i - \theta_*, \Sigma(\theta_i - \theta_*)\rangle + 2\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^n \langle \theta_i - \theta_*, \Sigma(\theta_j - \theta_*)\rangle.$$

Moreover,

$$
\begin{aligned}
&\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^{n}\langle\theta_i-\theta_*,\Sigma(\theta_j-\theta_*)\rangle\\
&=\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^{n}\Big\langle\theta_i-\theta_*,\Sigma\Big[M(j,i+1)(\theta_i-\theta_*)+\sum_{k=i+1}^{j}M(j,k+1)\gamma\varepsilon_k x_k\Big]\Big\rangle\\
&=\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^{n}\langle\theta_i-\theta_*,\Sigma M(j,i+1)(\theta_i-\theta_*)\rangle \text{ because } \varepsilon_k x_k \text{ and } \theta_i \text{ are independent,}\\
&=\mathbb{E}\sum_{i=0}^{n-1}\sum_{j=i+1}^{n}\langle\theta_i-\theta_*,\Sigma(I-\gamma\Sigma-\gamma\lambda I)^{j-i}(\theta_i-\theta_*)\rangle \text{ as } M(j,i+1) \text{ and } \theta_i \text{ are independent,}\\
&=\mathbb{E}\sum_{i=0}^{n-1}\Big\langle\theta_i-\theta_*,\gamma^{-1}\Sigma(\Sigma+\lambda I)^{-1}\big[(I-\gamma\Sigma-\gamma\lambda I)-(I-\gamma\Sigma-\gamma\lambda I)^{n-i+1}\big](\theta_i-\theta_*)\Big\rangle\\
&\leqslant\mathbb{E}\sum_{i=0}^{n}\Big\langle\theta_i-\theta_*,\gamma^{-1}\Sigma(\Sigma+\lambda I)^{-1}(I-\gamma\Sigma-\gamma\lambda I)(\theta_i-\theta_*)\Big\rangle \text{ using } (\Sigma+\lambda I)\preccurlyeq I,\\
&=\gamma^{-1}\mathbb{E}\sum_{i=0}^{n}\langle\theta_i-\theta_*,\Sigma(\Sigma+\lambda I)^{-1}(\theta_i-\theta_*)\rangle-\mathbb{E}\sum_{i=0}^{n}\langle\theta_i-\theta_*,\Sigma(\theta_i-\theta_*)\rangle.
\end{aligned}
$$

We thus simply need to bound $\gamma^{-1}\mathbb{E}\sum_{i=0}^{n}\langle\theta_i-\theta_*,\Sigma(\Sigma+\lambda I)^{-1}(\theta_i-\theta_*)\rangle$, to get a bound on $n^2\mathbb{E}\|\Sigma^{1/2}(\bar\theta_n-\theta_*)\|^2$.

Recursion on operators. We have:

$$
\begin{aligned}
\mathbb{E}\big[M(i,k)\Sigma(\Sigma+\lambda I)^{-1}M(i,k)^*\big] &= \mathbb{E}\Big[M(i,k+1)\big[I-\gamma x_k\otimes x_k-\gamma\lambda I\big]\Sigma(\Sigma+\lambda I)^{-1}\\
&\quad\big[I-\gamma x_k\otimes x_k-\gamma\lambda I\big]M(i,k+1)^*\Big]\\
&= \mathbb{E}\Big[M(i,k+1)\Big(\Sigma(\Sigma+\lambda I)^{-1}-2\gamma\Sigma+\gamma^2\big[x_k\otimes x_k\\
&\quad+\lambda I\big]\Sigma(\Sigma+\lambda I)^{-1}\big[x_k\otimes x_k+\lambda I\big]\Big)M(i,k+1)^*\Big]\\
&\preccurlyeq \mathbb{E}\Big[M(i,k+1)\big[\Sigma(\Sigma+\lambda I)^{-1}-2\gamma\Sigma\\
&\quad+\gamma^2(R^2+2\lambda)\Sigma\big]M(i,k+1)^*\Big]\\
&= \mathbb{E}\Big[M(i,k+1)\Sigma(\Sigma+\lambda I)^{-1}M(i,k+1)^*\Big]\\
&\quad-\gamma(2-\gamma(R^2+2\lambda))\mathbb{E}\Big[M(i,k+1)\Sigma M(i,k+1)^*\Big],
\end{aligned}
$$

which leads to

$$
\begin{aligned}
\mathbb{E}\Big[M(i,k+1)\Sigma M(i,k+1)^*\Big] &\preccurlyeq \frac{1}{\gamma(2-\gamma(R^2+2\lambda))}\Big(E\Big[M(i,k+1)\Sigma(\Sigma+\lambda I)^{-1}M(i,k+1)^*\Big]\\
&\quad-E\Big[M(i,k)\Sigma(\Sigma+\lambda I)^{-1}M(i,k)^*\Big]\Big). \qquad (22)
\end{aligned}
$$

Using the operator T on matrices defined below, this corresponds to showing

$$
(I-\gamma T)\big[\Sigma(\Sigma+\lambda I)\big]\preccurlyeq\Sigma(\Sigma+\lambda I)-\gamma\Sigma.
$$

Noise term. For $\theta_0-\theta_*=0$, we have:

$$
\begin{aligned}
&\mathbb{E}\langle\theta_i-\theta_*,\Sigma(\Sigma+\lambda I)^{-1}(\theta_i-\theta_*)\rangle\\
=\;&\gamma^2\mathbb{E}\sum_{k=1}^{i}\sum_{j=1}^{i}\varepsilon_j x_j^*M(i,j+1)^*\Sigma(\Sigma+\lambda I)^{-1}M(i,k+1)\varepsilon_k x_k \text{ by expanding all terms,}\\
=\;&\gamma^2\mathbb{E}\sum_{k=1}^{i}\varepsilon_k x_k^*M(i,k+1)^*\Sigma(\Sigma+\lambda I)^{-1}M(i,k+1)\varepsilon_k x_k \text{ using independence,}\\
=\;&\gamma^2\operatorname{tr}\Big(\sum_{k=1}^{i}\mathbb{E}\varepsilon_k^2x_kx_k^*\mathbb{E}M(i,k+1)^*\Sigma(\Sigma+\lambda I)^{-1}M(i,k+1)\Big)\\
\leqslant\;&\gamma^2\sigma^2\operatorname{tr}\Big(\sum_{k=1}^{i}\mathbb{E}M(i,k+1)\Sigma M(i,k+1)^*\Sigma(\Sigma+\lambda I)^{-1}\Big)\\
&\text{using our assumption regarding the noise.}
\end{aligned}
$$

Using the recurrence between operators

$$
\begin{aligned}
&\mathbb{E}\langle\theta_i-\theta_*,\Sigma(\Sigma+\lambda I)^{-1}(\theta_i-\theta_*)\rangle\\
\leqslant\;&\frac{\gamma\sigma^2}{2-\gamma(R^2+2\lambda)}\operatorname{tr}\sum_{k=1}^{i}\Big(E\Big[M(i,k+1)\Sigma(\Sigma+\lambda I)^{-1}M(i,k+1)^*\Sigma(\Sigma+\lambda I)^{-1}\Big]\\
&-E\Big[M(i,k)\Sigma(\Sigma+\lambda I)^{-1}M(i,k)^*\Sigma(\Sigma+\lambda I)^{-1}\Big]\Big)\\
\leqslant\;&\frac{\gamma\sigma^2}{2-\gamma(R^2+2\lambda)}\operatorname{tr}\Big(E\Big[M(i,i+1)\Sigma(\Sigma+\lambda I)^{-1}M(i,i+1)^*\Sigma(\Sigma+\lambda I)^{-1}\Big]\\
&-E\Big[M(i,1)\Sigma(\Sigma+\lambda I)^{-1}M(i,1)^*\Sigma(\Sigma+\lambda I)^{-1}\Big]\Big) \text{ by summing,}\\
\leqslant\;&\frac{\gamma\sigma^2}{2-\gamma(R^2+2\lambda)}\operatorname{tr}\Sigma^2(\Sigma+\lambda I)^{-2}.
\end{aligned}
$$

This implies that for the noise process

$$
\mathbb{E}\|\Sigma^{1/2}(\bar\theta_n-\theta_*)\|^2\leqslant\Big(\frac{\sigma^2}{n+1}\operatorname{tr}\big[\Sigma^2(\Sigma+\lambda I)^{-2}\big]\Big)\frac{1}{1-\gamma(R^2/2+\lambda)}.
$$

Note that when γ tends to zero, we recover the optimal variance term.

Noiseless term. Without noise, we then need to bound:

$$
\gamma^{-1}\mathbb{E}\sum_{i=0}^{n}\langle\theta_i-\theta_*,\Sigma(\Sigma+\lambda I)^{-1}(\theta_i-\theta_*)\rangle,
$$

with $\theta_i - \theta_* = M(i,1)(\theta_0 - \theta_*)$, that is

$$\gamma^{-1}\mathbb{E}\sum_{i=0}^{n} \operatorname{tr}\Big[M(i,1)^*\Sigma(\Sigma+\lambda I)^{-1}M(i,1)(\theta_0-\theta_*)(\theta_0-\theta_*)^*\Big].$$

We follow here the proof of Défossez and Bach (2015) and consider the operator T from symmetric matrices to symmetric matrices defined as

$$TA = (\Sigma+\lambda I)A + A(\Sigma+\lambda I) - \gamma E\big[(x_n\otimes x_n+\lambda I)A(x_n\otimes x_n+\lambda I)\big].$$

of the form $TA = (\Sigma+\lambda I)A + (\Sigma+\lambda I)A - \gamma SA$.

The operator S is self-adjoint and positive. Moreover:

$$\begin{aligned}
\langle A, SA\rangle &= \mathbb{E}\operatorname{tr}\big[A(x_n\otimes x_n+\lambda I)A(x_n\otimes x_n+\lambda I)\big]\\
&= \operatorname{tr}\big[2A^2\lambda\Sigma+\lambda^2A^2\big] + \mathbb{E}\operatorname{tr}\big[\langle x_n, Ax_n\rangle^2\big]\\
&\leqslant \operatorname{tr}\big[2A^2\lambda\Sigma+\lambda^2A^2\big] + \mathbb{E}\operatorname{tr}\big[\|x_n\|^2x_n\otimes x_n, A^2\big] \text{ using Cauchy-Schwarz inequality,}\\
&\leqslant \operatorname{tr}\big[2A^2\lambda\Sigma+\lambda^2A^2\big] + R^2\operatorname{tr}\Sigma A^2\\
&\leqslant (R^2+2\lambda)\operatorname{tr}\big[\Sigma+\lambda I]A^2.
\end{aligned}$$

We have for any symmetric matrix A:

$$\mathbb{E}M(i,1)^*AM(i,1) = (I-\gamma T)^iA.$$

Thus,

$$\gamma^{-1}\mathbb{E}\sum_{i=0}^{n}\operatorname{tr}\Big[M(i,1)^*\Sigma(\Sigma+\lambda I)^{-1}M(i,1)(\theta_0-\theta_*)(\theta_0-\theta_*)^*\Big] = \gamma^{-1}\mathbb{E}\sum_{i=0}^{n}\langle\langle(I-\gamma T)^iA, E_0\rangle\rangle$$

with $E_0 = (\theta_0-\theta_*)(\theta_0-\theta_*)^*$ and $A = \Sigma(\Sigma+\lambda I)^{-1}$. This leads to

$$\gamma^{-1}\mathbb{E}\langle\langle\gamma^{-1}T^{-1}(I-(I-\gamma T)^{n+1})A, E_0\rangle\rangle,$$

where $\langle\langle\cdot,\cdot\rangle\rangle$ denote the dot-product between self-adjoint operators.

The sum is less than its limit for $n\to\infty$, and thus, we can get rid of the term $(I-\gamma T)^{n+1}$, and we need to bound

$$\gamma^{-2}\langle\langle M, E_0\rangle\rangle = \gamma^{-2}\langle\langle T^{-1}(\Sigma(\Sigma+\lambda I)^{-1}), E_0\rangle\rangle,$$

with $M := T^{-1}\big[\Sigma(\Sigma+\lambda I)^{-1}\big]$, i.e., such that

$$\begin{aligned}
\Sigma(\Sigma+\lambda I)^{-1} &= (\Sigma+\lambda I)M + M(\Sigma+\lambda I) - \gamma\mathbb{E}(x_n\otimes x_n+\lambda I)M(x_n\otimes x_n+\lambda I)\\
&= (\Sigma+\lambda I)M + M(\Sigma+\lambda I) - \gamma SM. \qquad (23)
\end{aligned}$$

So that :

$$\begin{aligned}
M &= \big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}\big[\Sigma(\Sigma+\lambda I)^{-1}\big] + \gamma\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}SM\\
&= \frac{1}{2}\Sigma(\Sigma+\lambda I)^{-2} + \gamma\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}SM.
\end{aligned}$$

The operator $(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)$ is self adjoint, and so is its inverse, thus:

$$\begin{aligned}
\gamma^{-2}\langle\langle M, E_0\rangle\rangle &= \gamma^{-2}\langle\langle\frac{1}{2}\Sigma(\Sigma+\lambda I)^{-2} + \gamma\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}SM, E_0\rangle\rangle\\
&= \frac{1}{2}\gamma^{-2}\langle\langle\Sigma(\Sigma+\lambda I)^{-2}, E_0\rangle\rangle + \gamma^{-1}\langle\langle SM, \big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}E_0\rangle\rangle\\
&= \frac{1}{2}\gamma^{-2}\operatorname{tr}(\Sigma(\Sigma+\lambda I)^{-2}E_0) + \gamma^{-1}\langle\langle SM, \big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}E_0\rangle\rangle.
\end{aligned}$$

Moreover,

$$\begin{aligned}
E_0 &= (\theta_0-\theta_*)(\theta_0-\theta_*)^*\\
&= (\Sigma+\lambda I)^{1/2}(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)(\theta_0-\theta_*)^*(\Sigma+\lambda I)^{-1/2}(\Sigma+\lambda I)^{+1/2}\\
&\preccurlyeq [(\theta_0-\theta_*)^*(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)]\ \ (\Sigma+\lambda I),\\
&\qquad \text{as } (\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)(\theta_0-\theta_*)^*(\Sigma+\lambda I)^{-1/2} \preccurlyeq (\theta_0-\theta_*)^*(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)I.
\end{aligned}$$

Thus, as $[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)]^{-1}$ is an non-decreasing operator on $(S_n(\mathbb{R}), \preccurlyeq)$ (see technical Lemma 15 in Appendix E):

$$\begin{aligned}
&\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}E_0\\
\preccurlyeq\ &\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}\big([(\theta_0-\theta_*)^*(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)](\Sigma+\lambda I)\big)\\
=\ &\frac{(\theta_0-\theta_*)^*(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)}{2}I.
\end{aligned}$$

Thus as SM is positive :

$$\gamma^{-2}\langle\langle M, E_0\rangle\rangle \leq \frac{1}{2\gamma^2}\operatorname{tr}(\Sigma(\Sigma+\lambda I)^{-2}E_0) + \frac{(\theta_0-\theta_*)^*(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)}{2\gamma}\operatorname{tr}(SM).$$

Moreover we can upper bound $\operatorname{tr}(SM)$: using Equation (23) we have

$$\operatorname{tr}(\Sigma(\Sigma+\lambda I)^{-1}) = 2\operatorname{tr}(\Sigma+\lambda I)M - \gamma\operatorname{tr}\mathbb{E}(x_n\otimes x_n+\lambda I)M(x_n\otimes x_n+\lambda I)$$

then, using Assumption $(\mathcal{A}_1)$:

$$\operatorname{tr}\mathbb{E}(x_n\otimes x_n+\lambda I)M(x_n\otimes x_n+\lambda I) \leqslant R^2\operatorname{tr}M\Sigma + 2\operatorname{tr}M\Sigma\lambda + \lambda^2\operatorname{tr}M \leqslant (R^2+2\lambda)\operatorname{tr}M(\Sigma+\lambda I).$$

This implies

$$\begin{aligned}
\operatorname{tr}\big[\Sigma(\Sigma+\lambda I)^{-1}\big] &\geqslant \big(\frac{2}{R^2+2\lambda}-\gamma\big)\operatorname{tr}\mathbb{E}(x_n\otimes x_n+\lambda I)M(x_n\otimes x_n+\lambda I),\\
&\geqslant \frac{1}{R^2+2\lambda}\operatorname{tr}\mathbb{E}(x_n\otimes x_n+\lambda I)M(x_n\otimes x_n+\lambda I) \text{ since } \gamma(R^2+2\lambda)\leqslant 1,\\
&\geqslant \frac{1}{R^2+2\lambda}\operatorname{tr}SM.
\end{aligned}$$

Thus finally:

$$\begin{aligned}
\gamma^{-2}\langle\langle M, E_0\rangle\rangle \leq\ &\frac{1}{2\gamma^2}\operatorname{tr}E_0\Sigma(\Sigma+\lambda I)^{-2}\\
&+ \frac{(\theta_0-\theta_*)^*(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)}{2\gamma}(R^2+2\lambda)\operatorname{tr}(\Sigma(\Sigma+\lambda I)^{-1}),
\end{aligned}$$

which leads to the desired error term.

B.4 Proof When Only $\|\theta_0 - \theta_*\|$ Is Finite

When $\lambda = 0$, without noise, we then need to bound:

$$\gamma^{-1}\mathbb{E}\sum_{i=0}^{n}\langle\theta_i - \theta_*, \theta_i - \theta_*\rangle,$$

with $\theta_i - \theta_* = M(i,1)(\theta_0 - \theta_*)$, that is

$$\gamma^{-1}\mathbb{E}\sum_{i=0}^{n}\operatorname{tr}\Big[M(i,1)^*M(i,1)(\theta_0 - \theta_*)(\theta_0 - \theta_*)^*\Big].$$

By definition of $M(i,1)$ we have that $\mathbb{E}M(i,1)^*M(i,1) \preccurlyeq I$ leading to

$$\gamma^{-1}\mathbb{E}\sum_{i=0}^{n}\langle\theta_i - \theta_*, \theta_i - \theta_*\rangle \leq \frac{(n+1)\|\theta_0 - \theta_*\|^2}{\gamma}.$$

For the regularization-based bias we also have

$$\|\lambda\Sigma^{1/2}(\Sigma + \lambda I)^{-1}(\theta_0 - \theta_*)\|^2 \leq \lambda\|\Sigma^{1/2}(\Sigma + \lambda I)^{-1/2}(\theta_0 - \theta_*)\|^2.$$

B.5 Proof When the Noise Is Not Structured

For $\|\theta_0 - \theta_*\| = 0$ we have $\theta_n - \theta_* = \gamma\sum_{k=1}^{n} M(n,k+1)\varepsilon_k x_k$ which leads to

$$\mathbb{E}\|\Sigma^{1/2}(\theta_n - \theta_*)\|^2 = \gamma^2\sum_{k=1}^{n}\operatorname{tr}\mathbb{E}M(n,k+1)^*\Sigma M(n,k+1)V,$$

where $V = \mathbb{E}\varepsilon_k^2 x_k x_k^*$. And using the recursion on operators in Eq. (22) by changing order of elements we have

$$\begin{aligned}\mathbb{E}\Big[M(n,k+1)^*\Sigma M(n,k+1)\Big] \preccurlyeq \frac{1}{\gamma(2-\gamma(R^2+2\lambda))}\Big(&E\Big[M(n,k+1)^*\Sigma(\Sigma+\lambda I)^{-1}M(n,k+1)\Big]\\ &- E\Big[M(n,k)^*\Sigma(\Sigma+\lambda I)^{-1}M(n,k)\Big]\Big).\end{aligned}$$

And by adding the terms

$$\mathbb{E}\|\Sigma^{1/2}(\theta_n - \theta_*)\|^2 \preccurlyeq \frac{\gamma^2}{\gamma(2-\gamma(R^2+2\lambda))}\operatorname{tr}\Sigma(\Sigma+\lambda I)^{-1}V,$$

We conclude by convexity

$$\mathbb{E}\|\Sigma^{1/2}(\bar{\theta}_n - \theta_*)\|^2 \preccurlyeq \frac{\gamma^2}{\gamma(2-\gamma(R^2+2\lambda))}\operatorname{tr}\Sigma(\Sigma+\lambda I)^{-1}V.$$

Appendix C. Convergence of Accelerated Averaged Stochastic Gradient Descent

We now prove Theorem 3. We thus consider iterates satisfying Eq. (10), under Assumptions $(\mathcal{A}_3)$, $(\mathcal{A}_4)$. We consider a fixed step size γ such that $\gamma(\Sigma + \lambda I) \preccurlyeq I$. Seeing Eq. (10) as a linear second order for θ_n, we will derive from exact calculations a decomposition of the errors a sum of three terms that will be studied independently. The proof is organized as follows: in Section C.1, we state the formulation as a second order linear system and derive the three main terms that have to be studied (see Lemma 9). Section C.2 studies asymptotic behaviors of the three terms, ignoring some exponentially decreasing terms, in order to give insight of how they behave. This section is not necessary for the proof, indeed a direct and exact calculation in the eigenbasis of Σ, following O'Donoghue and Candès (2013), is provided in Section C.3. Results are summed up in Section C.4.

C.1 General Expansion

We study the regularized stochastic accelerated gradient descent recursion defined for $n \geq 1$ by

$$\begin{aligned}\theta_n &= \nu_{n-1} - \gamma f'(\nu_{n-1}) - \gamma\lambda(\nu_n - \theta_0) + \gamma\xi_n\\ \nu_n &= \theta_n + \delta(\theta_n - \theta_{n-1}),\end{aligned}$$

starting from $\theta_0 = \nu_0 \in \mathcal{H}$. We may rewrite it for a quadratic function $f : \theta \mapsto \frac{1}{2}\langle\theta - \theta_*, \Sigma(\theta - \theta_*)\rangle$ for $n \geq 2$ as

$$\theta_n = \big[I - \gamma\Sigma - \gamma\lambda I\big]\Big[\theta_{n-1} + \delta(\theta_{n-1} - \theta_{n-2})\Big] + \gamma\xi_n + \gamma\lambda\theta_0 + \gamma\Sigma\theta_*,$$

with $\theta_0 \in \mathcal{H}$ and $\theta_1 = \big[I - \gamma\Sigma - \gamma\lambda I\big]\theta_0 + \gamma\xi_1 + \gamma\lambda\theta_0 + \gamma\Sigma\theta_*$.

And by centering around the optimum, we get:

$$\theta_n - \theta_* = \big[I - \gamma\Sigma - \gamma\lambda I\big]\big[\theta_{n-1} - \theta_* + \delta(\theta_{n-1} - \theta_* - \theta_{n-2} + \theta_*)\big] + \gamma\xi_n + \lambda\gamma(\theta_0 - \theta_*).$$

Thus this is a second order iterative system which is standard to cast in a linear form

$$\Theta_n = F\Theta_{n-1} + \gamma\Xi_n + \gamma\lambda\Theta_\lambda, \tag{24}$$

with $T = I - \gamma\Sigma - \gamma\lambda I$, $F = \begin{pmatrix}(1+\delta)T & -\delta T\\ I & 0\end{pmatrix}$, $\Theta_n = \begin{pmatrix}\theta_n - \theta_*\\ \theta_{n-1} - \theta_*\end{pmatrix}$, $\Theta_0 = \begin{pmatrix}\theta_0 - \theta_*\\ \theta_0 - \theta_*\end{pmatrix}$, $\Xi_n = \begin{pmatrix}\xi_n\\ 0\end{pmatrix}$ and $\Theta_\lambda = \begin{pmatrix}\theta_0 - \theta_*\\ 0\end{pmatrix}$.

We are interested in the behavior of the average $\bar{\Theta}_n = \frac{1}{n+1}\sum_{k=0}^{n}\Theta_k$ for which we have the following general convergence result:

Lemma 9 *For all $\lambda \in \mathbb{R}_+$ and γ such that $\gamma(\Sigma + \lambda I) \preccurlyeq I$ and any matrix C the average of the iterates Θ_n defined by Eq. (24) satisfy for $P_k \overset{(def)}{=} C^{1/2}(I - F^k)(I - F)^{-1}$, with*

$\tilde{\Theta}_0 = \Theta_0 - \gamma\lambda(I-F)^{-1}\Theta_\lambda$,

$$\begin{aligned}\mathbb{E}\langle \bar{\Theta}_n, C\bar{\Theta}_n\rangle &\leq 2\,(\gamma\lambda)^2\,\|C^{1/2}(I-F)^{-1}\Theta_\lambda\|^2 + \frac{2}{(n+1)^2}\|P_{n+1}\tilde{\Theta}_0\|^2 \\ &\quad + \frac{\gamma^2}{(n+1)^2}\sum_{j=1}^n \operatorname{tr} P_j V P_j^\top.\end{aligned}$$

The error thus decomposes as the sum of three main terms:

- the two first ones are bias terms, one arising from the regularization (the first one), and one arising computation (the second one),
- a variance term. which is the last one.

We remark that as we have assumed that Σ is invertible, the matrix $I-F$ can be shown to be invertible for all the considered δ.

The regularization-based term will be studied directly whereas the two others will be studied in two stages. First a heuristic will lead to an asymptotic bound then an exact computation will give a non-asymptotic bound. Then using $C = H = \begin{pmatrix}\Sigma & 0\\ 0 & 0\end{pmatrix}$ would give a convergence result on the function value and $C = \begin{pmatrix}I & 0\\ 0 & 0\end{pmatrix}$ a result on the iterate. The end of the section is devoted to the proof of this lemma.

Proof The sequence Θ_n satisfies a linear recursion, from which we get, for all $n \geq 1$:

$$\begin{aligned}\Theta_n &= F^n\Theta_0 + \gamma\sum_{k=1}^n F^{n-k}\Xi_k + \gamma\lambda\sum_{k=1}^n F^{n-k}\Theta_\lambda \\ &= F^n\Theta_0 + \gamma\sum_{k=1}^n F^{n-k}\Xi_k + \gamma\lambda(I-F^n)(I-F)^{-1}\Theta_\lambda.\end{aligned}$$

We study the averaged sequence: $\bar{\Theta}_n = \frac{1}{n+1}\sum_{k=0}^n \Theta_k$. Using the identity $\sum_{k=0}^{n-1} F^k = (I-F^n)(I-F)^{-1}$, we get

$$\bar{\Theta}_n = \frac{1}{n+1}\sum_{k=0}^n F^k\Theta_0 + \frac{\gamma}{n+1}\sum_{k=1}^n\sum_{j=1}^k F^{k-j}\Xi_j + \frac{\gamma\lambda}{n+1}\sum_{k=1}^n (I-F^k)(I-F)^{-1}\Theta_\lambda.$$

With

$$\tilde{\Theta}_0 = \Theta_0 - \gamma\lambda(I-F)^{-1}\Theta_\lambda,$$

and $\sum_{k=1}^n (I-F^k) = \sum_{k=0}^n (I-F^k) = [n+1-(I-F^{n+1})(I-F)^{-1}]$.

Using summation formulas for geometric series, we derive:

$$\begin{aligned}\bar{\Theta}_n &= \frac{1}{n+1}(I-F^{n+1})(I-F)^{-1}\tilde{\Theta}_0 + \frac{\gamma}{n+1}\sum_{k=1}^n\sum_{j=1}^k F^{k-j}\Xi_j + \gamma\lambda(I-F)^{-1}\Theta_\lambda \\ &= \frac{1}{n+1}(I-F^{n+1})(I-F)^{-1}\tilde{\Theta}_0 + \frac{\gamma}{n+1}\sum_{j=1}^n\Big(\sum_{k=j}^n F^{k-j}\Big)\Xi_j + \gamma\lambda(I-F)^{-1}\Theta_\lambda \\ &= \frac{1}{n+1}(I-F^{n+1})(I-F)^{-1}\tilde{\Theta}_0 + \frac{\gamma}{n+1}\sum_{j=1}^n\Big(\sum_{k=0}^{n-j} F^{k}\Big)\Xi_j + \gamma\lambda(I-F)^{-1}\Theta_\lambda \\ &= \frac{1}{n+1}(I-F^{n+1})(I-F)^{-1}\tilde{\Theta}_0 + \frac{\gamma}{n+1}\sum_{j=1}^n(I-F^{n+1-j})(I-F)^{-1}\Xi_j + \gamma\lambda(I-F)^{-1}\Theta_\lambda \\ &= \frac{1}{n+1}(I-F^{n+1})(I-F)^{-1}\tilde{\Theta}_0 + \frac{\gamma}{n+1}\sum_{j=1}^n(I-F^{j})(I-F)^{-1}\Xi_{n+1-j} + \gamma\lambda(I-F)^{-1}\Theta_\lambda.\end{aligned}$$

Using martingale square moment inequalities which amount to consider $\Xi_i, i = 1, ..., n$ independent, so that the variance of the sum is the sum of variances, and denoting by $V = \mathbb{E}[\Xi_n \otimes \Xi_n]$ we have for any positive semi-definite C,

$$\begin{aligned}\mathbb{E}\langle \bar{\Theta}_n, C\bar{\Theta}_n\rangle &= \Big\|C^{1/2}\Big(\frac{1}{n+1}(I-F^{n+1})(I-F)^{-1}\tilde{\Theta}_0 + \gamma\lambda(I-F)^{-1}\Theta_\lambda\Big)\Big\|^2 \\ &\quad + \frac{\gamma^2}{(n+1)^2}\sum_{j=1}^n \operatorname{tr}(I-F^j)(I-F)^{-1}V(I-F^\top)^{-1}(I-F^j)^\top C,\end{aligned}$$

where $C^{1/2}$ denotes a symmetric square root of C. Define $P_k \overset{(def)}{=} C^{1/2}(I-F^k)(I-F)^{-1}$, we have, Using Minkowski's inequality and inequality $(a+b)^2 \leq 2(a^2+b^2)$ for any $a, b \in \mathbb{R}$,

$$\begin{aligned}\mathbb{E}\langle \bar{\Theta}_n, C\bar{\Theta}_n\rangle &= \Big\|\frac{1}{n+1}P_{n+1}\tilde{\Theta}_0 + \gamma\lambda C^{1/2}(I-F)^{-1}\Theta_\lambda\Big\|^2 + \frac{\gamma^2}{(n+1)^2}\sum_{j=1}^n \operatorname{tr} P_j V P_j^\top \\ &\leq 2\,(\gamma\lambda)^2\,\|C^{1/2}(I-F)^{-1}\Theta_\lambda\|^2 + \frac{2\|P_{n+1}\tilde{\Theta}_0\|^2}{(n+1)^2} + \frac{\gamma^2}{(n+1)^2}\sum_{j=1}^n \operatorname{tr} P_j V P_j^\top.\end{aligned}$$

This concludes proof of Lemma 9. ■

C.2 Asymptotic Expansion

To give the main terms that we expect, we first provide an asymptotic analysis, which shall only be understood as an insight and is not necessary for the proof. Operator F will have only eigenvalues smaller than 1, thus $\|\|F^j\|\|$ will decrease exponentially to 0 as $j \to \infty$ (even if $\|\|F\|\|$[5] might be bigger than 1). The asymptotic analysis relies on ignoring all terms in

5. $\|\|F\|\|$ denotes the operator norm of F, i.e., $\sup_{\|x\|\leq 1}\|Fx\|$.

which F^j appears. We thus approximately have:

$$\begin{aligned}\mathbb{E}\langle\bar{\Theta}_n, C\bar{\Theta}_n\rangle &\leq 2(\gamma\lambda)^2\|C^{1/2}(I-F)^{-1}\Theta_\lambda\|^2 + 2\Big\|C^{1/2}\frac{1}{n+1}(I-F^{n+1})(I-F)^{-1}\tilde{\Theta}_0\Big\|^2 \\ &\quad + \frac{\gamma^2}{(n+1)^2}\sum_{j=1}^n \operatorname{tr}(I-F^j)(I-F)^{-1}V(I-F^\top)^{-1}(I-F^j)^\top C \\ &\approx 2(\gamma\lambda)^2\|C^{1/2}(I-F)^{-1}\Theta_\lambda\|^2 + 2\Big\|C^{1/2}\frac{1}{n+1}(I-F)^{-1}\tilde{\Theta}_0\Big\|^2 \\ &\quad + \frac{\gamma^2}{(n+1)^2}\sum_{j=1}^n \operatorname{tr}(I-F)^{-1}V(I-F^\top)^{-1}C,\end{aligned}$$

where, as it has been explained $\approx$ stands for an equality up to terms that will decay exponentially. However, these terms have to be studied very carefully, what will be done in the Section C.3.

Using the matrix inversion lemma we have for $C = \begin{pmatrix} c & 0 \\ 0 & 0\end{pmatrix}$,

$$\begin{aligned} I-F &= \begin{pmatrix}(1+\delta)(\gamma\Sigma+\gamma\lambda I) - \delta I & \delta(I-(\gamma\Sigma+\gamma\lambda I)) \\ -I & I\end{pmatrix} \\ (I-F)^{-1} &= \begin{pmatrix}(\gamma\Sigma+\gamma\lambda I)^{-1} & \delta\big(I-(\gamma\Sigma+\gamma\lambda I)^{-1}\big) \\ (\gamma\Sigma+\gamma\lambda I)^{-1} & (1+\delta)I - \delta(\gamma\Sigma+\gamma\lambda I)^{-1}\end{pmatrix} \\ C^{1/2}(I-F)^{-1} &= \begin{pmatrix}c^{1/2}(\gamma\Sigma+\gamma\lambda I)^{-1} & \delta c^{1/2}\big(I-(\gamma\Sigma+\gamma\lambda I)^{-1}\big) \\ 0 & 0\end{pmatrix}.\end{aligned} \tag{25}$$

Regularization based term. This gives for the regularization based term

$$\begin{aligned}\Big\|C^{1/2}(I-F)^{-1}\Theta_\lambda\Big\|^2 &= \left\|\begin{pmatrix}c^{1/2}(\gamma\Sigma+\gamma\lambda I)^{-1} & \delta c^{1/2}\big(I-(\gamma\Sigma+\gamma\lambda I)^{-1}\big) \\ 0 & 0\end{pmatrix}\begin{pmatrix}\theta_0-\theta_* \\ 0\end{pmatrix}\right\|^2 \\ &= \left(\frac{1}{\gamma}\right)^2\|(c^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*))\|^2.\end{aligned} \tag{26}$$

The computation of this term is exact (not asymptotic).

Bias term. For the bias term we have

$$\begin{aligned}\tilde{\Theta}_0 &= \Theta_0 - \gamma\lambda(I-F)^{-1}\Theta_\lambda \\ &= \begin{pmatrix}\theta_0-\theta_* \\ \theta_0-\theta_*\end{pmatrix} - \gamma\lambda\begin{pmatrix}(\gamma\Sigma+\gamma\lambda I)^{-1} & \delta\big(I-(\gamma\Sigma+\gamma\lambda I)^{-1}\big) \\ (\gamma\Sigma+\gamma\lambda I)^{-1} & (1+\delta)I-\delta(\gamma\Sigma+\gamma\lambda I)^{-1}\end{pmatrix}\begin{pmatrix}\theta_0-\theta_* \\ 0\end{pmatrix} \\ &= \begin{pmatrix}\theta_0-\theta_* \\ \theta_0-\theta_*\end{pmatrix} - \gamma\lambda\begin{pmatrix}(\gamma\Sigma+\gamma\lambda I)^{-1}(\theta_0-\theta_*) \\ (\gamma\Sigma+\gamma\lambda I)^{-1}(\theta_0-\theta_*)\end{pmatrix} \\ &= \begin{pmatrix}[I-\lambda(\Sigma+\lambda I)^{-1}](\theta_0-\theta_*) \\ [I-\lambda(\Sigma+\lambda I)^{-1}](\theta_0-\theta_*)\end{pmatrix}.\end{aligned}$$

Thus this gives for the dominant term

$$\begin{aligned}\Big\|C^{1/2}(I-F)^{-1}\tilde{\Theta}_0\Big\|^2 &= \left\|\begin{pmatrix}c^{1/2}(\gamma\Sigma+\gamma\lambda I)^{-1} & \delta c^{1/2}\big(I-(\gamma\Sigma+\gamma\lambda I)^{-1}\big) \\ 0 & 0\end{pmatrix}\tilde{\Theta}_0\right\|^2 \\ &= \|(c^{1/2}[(1-\delta)(\gamma\Sigma+\gamma\lambda I)^{-1}+\delta I][I-\lambda(\Sigma+\lambda I)^{-1}](\theta_0-\theta_*)\|^2.\end{aligned}$$

And if c commutes with Σ we have the bound for $\delta \in [\frac{1-\sqrt{\gamma\lambda}}{1+\sqrt{\gamma\lambda}}, 1]$

$$\begin{aligned}\Big\|C^{1/2}(I-F)^{-1}\tilde{\Theta}_0\Big\|^2 &\leq (\frac{(1-\delta)}{\gamma\lambda}+\delta)\|(c^{1/2}[I-\lambda(\Sigma+\lambda I)^{-1}](\theta_0-\theta_*)\|^2 \\ &\leq (\frac{2}{\sqrt{\gamma\lambda}}+1)\|(c^{1/2}[I-\lambda(\Sigma+\lambda I)^{-1}](\theta_0-\theta_*)\|^2.\end{aligned}$$

Variance term. And for the variance term with $V = \begin{pmatrix} v & 0 \\ 0 & 0\end{pmatrix}$, we have $C^{1/2}(I-F)^{-1}V^{1/2} = \begin{pmatrix}c^{1/2}(\gamma\Sigma+\gamma\lambda I)^{-1}v^{1/2} & 0 \\ 0 & 0\end{pmatrix}$, and

$$\operatorname{tr} C^{1/2}(I-F)^{-1}V(I-F^\top)^{-1}C^{1/2} = \operatorname{tr} c(\gamma\Sigma+\gamma\lambda I)^{-1}v(\gamma\Sigma+\gamma\lambda I)^{-1}.$$

This gives the three dominant terms. However in order to control the remainders we have to compute the eigenvalues more carefully, as done in the next section.

C.3 Direct Computation without the Regularization Based Term

We derive now direct computation both the bias and variance terms. This is not required for the regularization based term whose previous expression in Eq. (26) is already non-asymptotic. Following O'Donoghue and Candès (2013) we consider an eigen-decomposition of the matrix F, in order to study independently the recursion on eigenspaces. We assume Σ has eigenvalues (s_i) and we decompose vectors in an eigenvector basis of Σ we denote by (p_i), with $\theta_n^i = p_i^\top\theta_n$ and $\xi_n^i = p_i^\top\xi_n$ and we have the reduced equation:

$$\Theta_{n+1}^i = F_i\Theta_n^i + \gamma\Xi_{n+1}^i.$$

with $\Theta_0^i = \tilde{\Theta}_0^i$, $F_i = \begin{pmatrix}(1+\delta)T_i & -\delta T_i \\ 1 & 0\end{pmatrix}$, with $T_i = 1-\gamma s_i - \gamma\lambda$.

Computing initial point $\tilde{\Theta}_0^i$. $\tilde{\Theta}_0^i = \Theta_0^i - \gamma\lambda(I-F_i)^{-1}\Theta_\lambda^i$, with $\Theta_0^i = \begin{pmatrix}\theta_0^i-\theta_*^i \\ \theta_0^i-\theta_*^i\end{pmatrix}$, $\Theta_\lambda^i = \begin{pmatrix}\theta_0^i-\theta_*^i \\ 0\end{pmatrix}$ and $(I-F_i)^{-1}$ given in Eq. (25). Thus

$$\begin{aligned}\tilde{\Theta}_0^i &= \begin{pmatrix}\theta_0^i-\theta_*^i \\ \theta_0^i-\theta_*^i\end{pmatrix} - \frac{\gamma\lambda}{(\gamma s_i+\gamma\lambda)}\begin{pmatrix}1 & \delta((\gamma s_i+\gamma\lambda)-1) \\ 1 & (1+\delta)(\gamma s_i+\gamma\lambda)-\delta\end{pmatrix}\begin{pmatrix}\theta_0^i-\theta_*^i \\ 0\end{pmatrix} \\ &= \begin{pmatrix}(1-\frac{\lambda}{\lambda+s_i})(\theta_0^i-\theta_*^i) \\ (1-\frac{\lambda}{\lambda+s_i})(\theta_0^i-\theta_*^i)\end{pmatrix}.\end{aligned} \tag{27}$$

Study of spectrum of F_i. Depending on δ, F_i may have two distinct complex eigenvalues of same modulus, only one (double) eigenvalue, or two real eigenvalues. We only consider the two former cases, which we detail below.

Indeed, the characteristic polynomial

$$\chi_{F_i}(X) \stackrel{def}{=} \det(XI - F_i) = X^2 - (1+\delta)(1-\gamma(s_i+\lambda))X + \delta(1-\gamma(s_i+\lambda))$$

has discriminant $\Delta_i = (1-\gamma(s_i+\lambda))((1+\delta)^2(1-\gamma(s_i+\lambda)) - 4\delta)$ which is non positive as far as $\delta \in [\delta_-;\delta_+]$, with $\delta_- = \frac{1-\sqrt{\gamma(s_i+\lambda)}}{1+\sqrt{\gamma(s_i+\lambda)}}$, $\delta_+ = \frac{1+\sqrt{\gamma(s_i+\lambda)}}{1-\sqrt{\gamma(s_i+\lambda)}}$.

C.3.1 Two Distinct Eigenvalues

We first assume that F_i has two distinct complex eigenvalues $r_\pm = \frac{(1+\delta)(1-\gamma(s_i+\lambda))\pm\sqrt{-1}\sqrt{-\Delta_i}}{2}$ which are conjugate. Thus the roots are of the form $\rho_i e^{\pm i\omega_i}$ with $\rho_i = \sqrt{\delta(1-\gamma(s_i+\lambda))}$, $\cos(\omega_i) = \frac{(1+\delta)(1-\gamma(s_i+\lambda))}{2\rho_i}$, $\omega_i \in [-\pi/2;\pi/2]$ and $\sin(\omega_i) = \frac{\sqrt{-\Delta_i}}{2\rho_i}$.

Let $Q_i = \begin{pmatrix} r_i^- & r_i^+ \\ 1 & 1 \end{pmatrix}$ be the transfer matrix into an eigenbasis of F_i, i.e., $F_i = Q_i D_i Q_i^{-1}$ with $D_i = \begin{pmatrix} r_i^- & 0 \\ 0 & r_i^+ \end{pmatrix}$ and $Q_i^{-1} = \frac{1}{r_i^- - r_i^+}\begin{pmatrix} 1 & -r_i^+ \\ -1 & r_i^- \end{pmatrix}$.

Computing $P_{i,k}$. We first compute the matrix $P_{i,k}$: With

$$C_i^{1/2} = \begin{pmatrix} \sqrt{c_i} & 0 \\ 0 & 0 \end{pmatrix}, C_i^{1/2} Q_i = \begin{pmatrix} r_i^-\sqrt{c_i} & r_i^+\sqrt{c_i} \\ 0 & 0 \end{pmatrix}$$

we have

$$C_i^{1/2} Q_i (I - D_i^k)(I - D_i)^{-1} = \sqrt{c_i}\begin{pmatrix} \frac{1-(r_i^-)^k}{1-r_i^-} r_i^- & \frac{1-(r_i^+)^k}{1-r_i^+} r_i^+ \\ 0 & 0 \end{pmatrix},$$

and, when developing and regrouping terms which depend on k, we get :

$$\begin{aligned} P_{i,k} &= C_i^{1/2} Q_i (I - D_i^k)(I - D_i)^{-1} Q_i^{-1} \\ &= \frac{\sqrt{c_i}}{r_i^- - r_i^+}\begin{pmatrix} \frac{1-(r_i^-)^k}{1-r_i^-} r_i^- - \frac{1-(r_i^+)^k}{1-r_i^+} r_i^+ & \frac{1-(r_i^+)^k}{1-r_i^+} r_i^- r_i^+ - \frac{1-(r_i^-)^k}{1-r_i^-} r_i^+ r_i^- \\ 0 & 0 \end{pmatrix} \\ &= \sqrt{c_i}\begin{pmatrix} \frac{1}{(1-r_i^-)(1-r_i^+)} & \frac{-r_i^+ r_i^-}{(1-r_i^-)(1-r_i^+)} \\ 0 & 0 \end{pmatrix} \\ &\quad - \frac{\sqrt{c_i}}{r_i^- - r_i^+}\begin{pmatrix} \frac{(r_i^-)^{k+1}}{1-r_i^-} - \frac{(r_i^+)^{k+1}}{1-r_i^+} & \frac{(r_i^+)^{k+1}}{1-r_i^+} r_i^- - \frac{(r_i^-)^{k+1}}{1-r_i^-} r_i^+ \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

We also have $P_{i,k} = C_i^{1/2} Q_i (I - D_i^k)(I - D_i)^{-1} Q_i^{-1} = \sum_{j=0}^{k-1} R_{i,j}$ with

$$\begin{aligned} R_{i,j} &= C_i^{1/2} Q_i D_i^j Q_i^{-1} \\ &= \sqrt{c_i}\begin{pmatrix} (r_i^-)^{j+1} & (r_i^+)^{j+1} \\ 0 & 0 \end{pmatrix} Q_i^{-1} \\ &= \frac{\sqrt{s_i}}{r_i^- - r_i^+}\begin{pmatrix} (r_i^-)^{j+1} - (r_i^+)^{j+1} & -r_i^+(r_i^-)^{j+1} + r_i^-(r_i^+)^{j+1} \\ 0 & 0 \end{pmatrix}, \end{aligned}$$

but computing error terms based in $R_{i,j}$ before summing these errors gives a looser error bound than a tight calculation using $P_{i,k}$. More precisely, if we use $P_{i,k}\Theta_0^i = \sum_{j=0}^{k-1} R_{i,j}\Theta_0^i$ to upper bound $\|P_{i,k}\Theta_0^i\| \leq \sum_{j=0}^{k-1}\|R_{i,j}\Theta_0^i\|$, we end up with a worse bound.

Bias term. Thus, for the bias term:

$$\begin{aligned} P_{i,k}\Theta_0^i &= \sqrt{c_i}\theta_0^i \frac{1 - r_i^+ r_i^-}{(1-r_i^-)(1-r_i^+)} - \frac{\sqrt{c_i}\theta_0^i}{r_i^- - r_i^+}\begin{pmatrix} \left[(r_i^-)^{k+1}\frac{1-r_i^+}{1-r_i^-} - (r_i^+)^{k+1}\frac{1-r_i^-}{1-r_i^+}\right] \\ 0 \end{pmatrix} \\ &= \frac{\sqrt{c_i}\theta_0^i}{\sqrt{(1-r_i^-)(1-r_i^+)}}\begin{pmatrix} \frac{\left[(1-r_i^+r_i^-) - \rho_i^k A_1\right]}{\sqrt{(1-r_i^-)(1-r_i^+)}} \\ 0 \end{pmatrix}, \end{aligned}$$

where

$$\rho_i^k A_1 = \frac{(r_i^-)^{k+1}(1-r_i^+)^2 - (r_i^+)^{k+1}(1-r_i^-)^2}{r_i^- - r_i^+}.$$

This can be bound with the following lemma

Lemma 10 *For all $\rho \in (0,1)$ and $\omega \in [-\pi/2;\pi/2]$ and $r^\pm = \rho(\cos(\omega) \pm \sqrt{-1}\sin(\omega))$ we have:*

$$\left|\frac{1 - r^+r^- - \rho^k|A_1|}{|1-r^+|}\right| \leq 3 + 3\rho^k \leq 6 \tag{28}$$

We note that the exact constant seems empirically to be 2. This lemma is proved as Lemma 16 in Appendix E. This gives for the bias term

$$\begin{aligned} \|P_{i,k}\Theta_0^i\| &= \frac{\sqrt{c_i}(\theta_0^i)}{\sqrt{(1-r_i^-)(1-r_i^+)}}\Big[\frac{1}{\sqrt{(1-r_i^-)(1-r_i^+)}}\Big((1-r_i^+r_i^-) - \rho_i^k A_1\Big)\Big] \\ &\leq 6\frac{\sqrt{c_i}(\theta_0^i)}{\sqrt{\gamma(s_i+\lambda)}}, \end{aligned}$$

since:

$$\begin{aligned} (1-r_i^-)(1-r_i^+) &= 1 - 2\mathfrak{Re}\,(r_i^+) + |r_i^+|^2 \\ &= 1 - (1+\delta)(1-\gamma(s_i+\lambda)) + \delta(1-\gamma(s_i+\lambda)) \\ &= \gamma(s_i+\lambda). \end{aligned}$$

We also have a looser bound using $P_{i,k}\Theta_0^i = \sum_{j=0}^{k-1} R_{i,j}\Theta_0^i$.

$$\begin{aligned}
R_{i,j}\Theta_0^i &= \frac{\sqrt{c_i}\theta_0^i}{r_i^- - r_i^+}\big((1-r_i^+)(r_i^-)^{j+1} - (1-r_i^-)(r_i^+)^{j+1}\big) \\
&= \sqrt{c_i}\theta_0^i\bigg(\frac{(r_i^-)^{j+1} - (r_i^+)^{j+1}}{r_i^- - r_i^+} - \frac{r_i^+(r_i^-)^{j+1} - r_i^-(r_i^+)^{j+1}}{r_i^- - r_i^+}\bigg) \text{ using De Moivre's formula,} \\
&= \sqrt{c_i}\theta_0^i\bigg(\frac{\rho_i^{j+1}\sin(\omega_i(j+1))}{\rho_i\sin(\omega_i)} - \frac{\rho_i e^{i\omega_i}\rho_i^{j+1}e^{-i\omega_i(j+1)} - \rho_i e^{-i\omega_i}\rho_i^{j+1}e^{+i\omega_i(j+1)}}{\rho_i e^{-i\omega_i} - \rho_i e^{i\omega_i}}\bigg) \\
&= \sqrt{c_i}\theta_0^i\bigg(\frac{\rho_i^{j+1}\sin(\omega_i(j+1))}{\rho_i\sin(\omega_i)} - \rho_i^{j+1}\frac{e^{-i\omega_i j} - e^{+i\omega_i j}}{e^{-i\omega_i} - e^{i\omega_i}}\bigg) \\
&= \sqrt{c_i}\theta_0^i\bigg(\frac{\rho_i^{j}\sin(\omega_i(j+1))}{\sin(\omega_i)} - \rho_i^{j+1}\frac{\sin(\omega_i j)}{\sin(\omega_i)}\bigg) \\
&\leq (1+e^{-1})\sqrt{c_i}\theta_0^i \quad \text{using Lemma 17 (see proof in Appendix E),}
\end{aligned}$$

which also gives for the bias term

$$\|P_{i,k}\Theta_0^i\| \leq (1+e^{-1})\sqrt{c_i}\theta_0^i k.$$

Thus we have the final bound:

$$\|P_{i,k}\Theta_0^i\|^2 \leq \min\left\{36\frac{c_i(\theta_0^i)^2}{\gamma(s_i+\lambda)}, 6n(1+e^{-1})\frac{c_i(\theta_0^i)^2}{\sqrt{\gamma(s_i+\lambda)}}, n^2(1+e^{-1})^2 c_i(\theta_0^i)^2\right\}. \tag{29}$$

Variance term. As for the variance term, with $V_i = \begin{pmatrix} v_i & 0 \\ 0 & 0\end{pmatrix}$, we have $\operatorname{tr} P_{i,k}V_iP_{i,k} = \left\|P_{i,k}\begin{pmatrix}\sqrt{v_i}\\0\end{pmatrix}\right\|^2$.

$$\begin{aligned}
\left\|P_{i,k}\begin{pmatrix}\sqrt{v_i}\\0\end{pmatrix}\right\| &= \frac{\sqrt{v_i c_i}}{(1-r_i^-)(1-r_i^+)}\left[1 + \frac{(r_i^-)^{k+1}(1-r_i^+) - (r_i^+)^{k+1}(1-r_i^-)}{r_i^+ - r_i^-}\right] \\
&= \frac{\sqrt{v_i c_i}}{\gamma(s_i+\lambda)}\Big[1 - \rho_i^k B_{i,k}\Big],
\end{aligned}$$

where

$$\rho_i^k B_{i,k} = -\frac{(r_i^-)^{k+1}(1-r_i^+) - (r_i^+)^{k+1}(1-r_i^-)}{r_i^+ - r_i^-},$$

which we can bound using the following Lemma:

Lemma 11 *For all $\rho \in (0,1)$ and $\omega \in [-\pi/2;\pi/2]$ and $r^\pm = \rho(\cos(\omega) \pm \sqrt{-1}\sin(\omega))$ we have:*

$$\Big|\rho^k B_k\Big| \leq 1.75.$$

Where we note that the exact majoration seems to be 1.3. This Lemma is proved as Lemma 18 in Appendix E.

We can also have a looser bound using $P_{i,k}\begin{pmatrix}v_i^{1/2}\\0\end{pmatrix} = \sum_{j=0}^{k-1} R_{i,j}\begin{pmatrix}v_i^{1/2}\\0\end{pmatrix}$ and

$$\begin{aligned}
R_{i,j}\begin{pmatrix}v_i^{1/2}\\0\end{pmatrix} &= \frac{\sqrt{c_i v_i}}{r_i^- - r_i^+}\ \big((r_i^-)^{j+1} - (r_i^+)^{j+1}\big) \\
&= \sqrt{c_i v_i}\ \frac{\rho_i^{j+1}\sin(\omega_i(j+1))}{\rho_i\sin(\omega_i)} \\
&\leq (j+1)\sqrt{c_i v_i}, \text{ using the inequality } |\sin(k\omega_i)| \leq k|\sin(\omega_i)|
\end{aligned}$$

and $\left\|P_{i,k}\begin{pmatrix}v_i^{1/2}\\0\end{pmatrix}\right\| \leq \frac{\sqrt{c_i v_i}(k+1)k}{2}$.

This gives for the Variance term

$$\begin{aligned}
\sum_{k=1}^n \operatorname{tr} P_{i,k}V_iP_{i,k} &\leq v_i c_i \sum_{k=1}^n \min\left\{\frac{\big[1-\rho_i^k B_{1,k}\big]^2}{\gamma^2(s_i+\lambda)^2}, \frac{\big[1-\rho_i^k B_{1,k}\big]k(k+1)}{2\gamma(s_i+\lambda)}, \frac{k^2(k+1)^2}{4}\right\} \\
&\leq v_i c_i \min\left\{\frac{8n}{\gamma^2(s_i+\lambda)^2}, \frac{(n+1)^3}{2\gamma(s_i+\lambda)}, \frac{(n+1)^5}{20}\right\}.
\end{aligned} \tag{30}$$

C.3.2 One Coalescent Eigenvalue

We now turn to the case where F has two coalescent eigenvalues, which happens when the discriminant $\Delta = 0$. We assume that F_i has one coalescent eigenvalue $r_i = \frac{(1+\delta)(1-\gamma(s_i+\lambda))}{2}$. Then, with $\delta = \frac{1-\sqrt{\gamma(s_i+\lambda)}}{1+\sqrt{\gamma(s_i+\lambda)}}$, $r_i = \frac{(1+\delta)(1-\gamma(s_i+\lambda))}{2} = 1 - \sqrt{\gamma(s_i+\lambda)}$. Then F_i can be trigonalized as $F_i = Q_i D_i Q_i^{-1}$ with $Q_i = \begin{pmatrix} r_i & 1\\ 1 & 0\end{pmatrix}$, $D_i = \begin{pmatrix} r_i & 1\\ 0 & r_i\end{pmatrix}$ and $Q_i^{-1} = \begin{pmatrix}0 & 1\\ 1 & -r_i\end{pmatrix}$. We note that for all $k \geq 0$, then $D_i^k = r_i^{k-1}\begin{pmatrix} r_i & k\\ 0 & r_i\end{pmatrix}$.

Computing $P_{i,k}$. We first compute $P_{i,k}$:

$$(I_2 - D_i)^{-1} = \begin{pmatrix}\frac{1}{1-r_i} & \frac{1}{(1-r_i)^2}\\ 0 & \frac{1}{1-r_i}\end{pmatrix}$$

and

$$(I_2 - D_i^k)(I_2 - D_i)^{-1} = \begin{pmatrix}\frac{1-r_i^k}{1-r_i} & \frac{1-r_i^k}{(1-r_i)^2} - \frac{kr_i^{k-1}}{1-r_i}\\ 0 & \frac{1-r_i^k}{1-r_i}\end{pmatrix}.$$

Thus with $C_i^{1/2}Q_i = \begin{pmatrix}\sqrt{c_i}r_i & \sqrt{c_i}\\ 0 & 0\end{pmatrix}$ we have

$$C_i^{1/2}Q_i(I_2 - D_i^k)(I_2 - D_i)^{-1} = \sqrt{c_i}\begin{pmatrix}\frac{1-r_i^k}{1-r_i}r_i & \frac{1-r_i^k}{(1-r_i)^2} - \frac{kr_i^k}{1-r_i}\\ 0 & 0\end{pmatrix}.$$

And, computing as previously the matrices products, we derive:

$$\begin{aligned}
P_{i,k} &= C_i^{1/2} Q_i (I_2 - D_i^k)(I_2 - D_i)^{-1} Q_i^{-1} \\
&= \sqrt{c_i} \begin{pmatrix} \frac{1-r_i^k}{(1-r_i)^2} - \frac{kr_i^k}{1-r_i} & \frac{1-r_i^k}{1-r_i} r_i - (\frac{1-r_i^k}{(1-r_i)^2} - \frac{kr_i^k}{1-r_i}) r_i \\ 0 & 0 \end{pmatrix} \\
&= \sqrt{c_i} \begin{pmatrix} \frac{1-r_i^k}{(1-r_i)^2} - \frac{kr_i^k}{1-r_i} & \frac{1-r_i^k}{(1-r_i)^2}(r_i)^2 + \frac{kr_i^{k+1}}{1-r_i} \\ 0 & 0 \end{pmatrix} \\
&= \frac{\sqrt{c_i}}{1-r_i} \begin{pmatrix} \frac{1-r_i^k}{1-r_i} - kr_i^k & -\frac{1-r_i^k}{1-r_i}(r_i)^2 + kr_i^{k+1} \\ 0 & 0 \end{pmatrix}.
\end{aligned}$$

Bias term. We thus have:

$$\begin{aligned}
P_{i,k}\Theta_0^i &= \frac{\sqrt{c_i}}{1-r_i} \begin{pmatrix} \frac{1-r_i^k}{1-r_i} - kr_i^k & -\frac{1-r_i^k}{1-r_i}(r_i)^2 + kr_i^{k+1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \theta_0^i \\ \theta_0^i \end{pmatrix} \\
&= \theta_0^i \sqrt{c_i} \begin{pmatrix} (1-r_i^k)\frac{1+r_i}{1-r_i} - kr_i^k \\ 0 \end{pmatrix},
\end{aligned}$$

and this gives for the bias term:

$$\begin{aligned}
&\|P_{i,k}\Theta_0^i\|^2 \\
=&(\theta_0^i)^2 c_i \Big[(1-r_i^k)\frac{1+r_i}{1-r_i} - kr_i^k\Big]^2 \\
=&(\theta_0^i)^2 c_i \Big[\frac{1+r_i}{1-r_i} - \Big(k + \frac{1+r_i}{1-r_i}\Big) r_i^k\Big]^2 \text{ developing the product, then using formulas for } r_i \\
=&(\theta_0^i)^2 c_i \Big[\frac{2-\sqrt{\gamma(s_i+\lambda)}}{\sqrt{\gamma(s_i+\lambda)}} - \Big(k + \frac{2-\sqrt{\gamma(s_i+\lambda)}}{\sqrt{\gamma(c_i+\lambda)}}\Big)(1-\sqrt{\gamma(s_i+\lambda)})^k\Big]^2 \\
=&\frac{(\theta_0^i)^2 c_i}{\gamma(s_i+\lambda)}\Big[2 - \sqrt{\gamma(s_i+\lambda)} - \big(k\sqrt{\gamma(s_i+\lambda)} + 2 - \sqrt{\gamma(s_i+\lambda)}\big)(1-\sqrt{\gamma(s_i+\lambda)})^k\Big]^2 \\
=&\frac{(\theta_0^i)^2 c_i}{\gamma(s_i+\lambda)}\Big[2 - \sqrt{\gamma(s_i+\lambda)} - \big(2 + (k-1)\sqrt{\gamma(s_i+\lambda)}\big)(1-\sqrt{\gamma(s_i+\lambda)})^k\Big]^2 \\
\leq& 4\frac{(\theta_0^i)^2 c_i}{\gamma(s_i+\lambda)}, \text{ using Lemma 19 in Appendix E.}
\end{aligned}$$

Variance term. With $V = \begin{pmatrix} v_i & 0 \\ 0 & 0 \end{pmatrix}$,

$$\begin{aligned}
&\operatorname{tr} P_{i,k} V P_{i,k} \\
=&\frac{s_i}{(1-r_i)^2} \begin{pmatrix} \frac{1-r_i^k}{1-r_i} - kr_i^k & -\frac{1-r_i^k}{1-r_i}(r_i)^2 + kr_i^{k+1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} v_i & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1-r_i^k}{1-r_i} - kr_i^k & -\frac{1-r_i^k}{1-r_i}(r_i)^2 + kr_i^{k+1} \\ 0 & 0 \end{pmatrix}^\top \\
=&\frac{s_i v_i}{(1-r_i)^2}\Big[\frac{1-r_i^k}{1-r_i} - kr_i^k\Big]^2 \\
=&\frac{v_i h_i}{\gamma(s_i+\lambda)}\Big[\frac{1-r_i^k}{1-r_i} - kr_i^k\Big]^2 \\
=&\frac{v_i h_i}{\gamma(s_i+\lambda)(1-r_i)^2}\Big[1 - r_i^k - (1-r_i)kr_i^k\Big]^2 \\
=&\frac{v_i h_i}{\gamma^2(s_i+\lambda)^2}\Big[1 - (1 + k\sqrt{\gamma(s_i+\lambda)})(1-\sqrt{\gamma(s_i+\lambda)})^k\Big]^2,
\end{aligned}$$

and

$$\begin{aligned}
\sum_{k=1}^n \operatorname{tr} P_{i,k} V P_{i,k} &= \frac{v_i s_i}{\gamma^2(s_i+\lambda)^2} \sum_{k=1}^n \Big[1 - (1 + k\sqrt{\gamma(s_i+\lambda)})(1-\sqrt{\gamma(s_i+\lambda)})^k\Big]^2 \\
&\leq n\frac{v_i s_i}{\gamma^2(s_i+\lambda)^2} \text{ using Lemma 19 in Appendix E.} \qquad (31)
\end{aligned}$$

Alternative bounds for the bias and the variance term, as in Equations(26), (29) may be derived as well. Combining all these results, we are now able to state Theorem 3.

C.4 Conclusion

Combining results from Lemma 9, and Equations (26), (29), (30), with $c = \Sigma$, and using the following simple facts:

- For the least squares regression function, with $c = \Sigma$, $\mathbb{E}\langle \bar{\Theta}_n, C\bar{\Theta}_n\rangle = \mathbb{E}f(\bar{\theta}_n) - f(\theta_*)$.
- Under assumption $\mathcal{A}_3$, $\mathcal{A}_4$, we have $V \preccurlyeq \tau^2\Sigma$.
- The squared norm of a vector is the sum of its squared components on the orthonormal eigenbasis. For example $\|P_{n+1}\Theta_0\|^2 = \sum_{i=1}^d \|P_{i,n+1}\Theta_0^i\|^2$.
- For any regularization parameter $\lambda \in \mathbb{R}_+$ and for any constant step-size $\gamma(\Sigma+\lambda I) \preccurlyeq I$, for any $\delta \in \big[\frac{1-\sqrt{\gamma\lambda}}{1+\sqrt{\gamma\lambda}}, 1\big]$, matrix F will have only two distinct complex eigenvalues or two coalescent eigenvalues.

Proposition 12 *Under ($\mathcal{A}_{4,5}$), for any regularization parameter $\lambda \in \mathbb{R}_+$ and for any constant step-size $\gamma(\Sigma+\lambda I) \preccurlyeq I$ we have for any $\delta \in \left[\frac{1-\sqrt{\gamma\lambda}}{1+\sqrt{\gamma\lambda}}, 1\right]$, for the recursion in Eq. (10):*

$$\begin{aligned}\mathbb{E}f(\bar{\theta}_n) - f(\theta_*) \leq\ & 2\lambda\|\lambda^{1/2}\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\|^2 \\ & + \sum_{i=1}^d \frac{2}{(n+1)^2}\min\left\{36\frac{c_i(\tilde{\theta}_0^i)^2}{\gamma(s_i+\lambda)}, 6n(1+e^{-1})\frac{c_i(\tilde{\theta}_0^i)^2}{\sqrt{\gamma(s_i+\lambda)}}, n^2(1+e^{-1})^2 c_i(\tilde{\theta}_0^i)^2\right\} \\ & + \sum_{i=1}^d \frac{\gamma^2}{(n+1)^2} v_i c_i \min\left\{\frac{8n}{\gamma^2(s_i+\lambda)^2}, \frac{(n+1)^3}{2\gamma(s_i+\lambda)}, \frac{(n+1)^5}{20}\right\}.\end{aligned}$$

This implies, using the Equation (27) for the initial point, using $c_i = \sigma_i$ and regrouping sums as traces or norms:

$$\begin{aligned}\mathbb{E}f(\bar{\theta}_n) - f(\theta_*) \ \leq\ & 2\lambda\|\lambda^{1/2}\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\|^2 \\ & + \ 2\min\left\{\frac{36\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\|^2}{\gamma(n+1)^2}, (1+e^{-1})^2\|\Sigma^{1/2}(\theta_0-\theta_*)\|^2\right\} \\ & + \ \min\left\{\frac{8\operatorname{tr}(V\Sigma(\Sigma+\lambda I)^{-2})}{n+1}, n\gamma\operatorname{tr}(V\Sigma(\Sigma+\lambda I)^{-1})\right\},\end{aligned}$$

which gives exactly Theorem 3 using $V \preccurlyeq \tau^2\Sigma$ in the Variance term, and $\lambda^{1/2}(\Sigma+\lambda I)^{-1/2} \preccurlyeq I$ in the first term.

Appendix D. Tighter Bounds

D.1 Simple Upper-Bounds

In this section, we chow how tighter bounds naturally appear from the regularized quantities appearing in Theorems. It only relies on simple algebraic majorations, even if one has to be careful with the allowed intervals for r, b.

Lemma 13 *For any $\lambda \geq 0$, for any $b \in [0;1]$, if $\operatorname{tr}(\Sigma^b)$ exists, we have :*

$$\begin{aligned}\operatorname{tr}(\Sigma(\Sigma+\lambda I)^{-1}) &\leq \frac{\operatorname{tr}(\Sigma^b)}{\lambda^b} \\ \operatorname{tr}(\Sigma^{-2}(\Sigma+\lambda I)^{-2}) &\leq \frac{\operatorname{tr}(\Sigma^b)}{\lambda^b}.\end{aligned}$$

Proof As all operators can be diagonalized in a same eigenbasis with positive eigenvalues, we have,

$$\begin{aligned}\operatorname{tr}(\Sigma(\Sigma+\lambda I)^{-1}) &\leq \left|\left|\left|\Sigma^{1-b}(\Sigma+\lambda I)^{-1}\right|\right|\right| \operatorname{tr}(\Sigma^b) \\ |||\Sigma^{1-b}(\Sigma+\lambda I)^{-1}||| &\leq \sup_{0\leq x} \frac{x^{1-b}}{(x+\lambda)} \\ &\leq \sup_{0\leq x} x^{1-b}\left(\frac{1}{\lambda}\wedge\frac{1}{x}\right) \\ &\leq \sup_{0\leq x} x^{1-b}\left(\frac{1}{\lambda}\right)^b\left(\frac{1}{x}\right)^{1-b} = \lambda^{-b}.\end{aligned}$$

The calculations are exactly the same for $\operatorname{tr}(\Sigma^{-2}(\Sigma+\lambda I)^{-2}) \leq \frac{\operatorname{tr}(\Sigma^b)}{\lambda^b}$. ■

As for the bias term, we need to bound the following quantities :

Lemma 14 *For any $\lambda \geq 0$, for any $r \in [-1;1]$, we have :*

$$\left\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\right\|^2 \ \leq\ \lambda^{-(1+r)}\left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\|^2.$$

For any $\lambda \geq 0$, for any $r \in [-1;0]$, we have :

$$\left\|(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\right\|^2 \ \leq\ \lambda^{-(1+r)}\left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\|^2.$$

For any $\lambda \geq 0$, for any $r \in [0;1]$, we have :

$$\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\|^2 \ \leq\ \lambda^{-r}\left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\|^2$$

(No result when $r \leq 0$ because of saturation effect).

Proof Proof relies of simple following calculations:

$$\begin{aligned}\left\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1}(\theta_0-\theta_*)\right\| &\leq \left|\left|\left|\Sigma^{1/2-r/2}(\Sigma+\lambda I)^{-1}\right|\right|\right| \left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\| \\ &\leq \left(\frac{1}{\lambda}\right)^{1-(1/2-r/2)}\left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\| \\ &\leq \lambda^{-\frac{1+r}{2}}\left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\|\end{aligned}$$

$$\begin{aligned}\left\|(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\right\| &\leq \left|\left|\left|\Sigma^{-r/2}(\Sigma+\lambda I)^{-1/2}\right|\right|\right| \left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\| \\ &\leq \left(\frac{1}{\lambda}\right)^{\frac{1+r}{2}}\left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\| \\ &\leq \lambda^{-\frac{1+r}{2}}\left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\|\end{aligned}$$

$$\begin{aligned}\|\Sigma^{1/2}(\Sigma+\lambda I)^{-1/2}(\theta_0-\theta_*)\| &\leq \left|\left|\left|\Sigma^{1/2-r/2}(\Sigma+\lambda I)^{-1/2}\right|\right|\right| \left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\| \\ &\leq \left(\frac{1}{\lambda}\right)^{\frac{1-(1-r)}{2}}\left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\| \\ &\leq \lambda^{-\frac{r}{2}}\left\|\Sigma^{r/2}(\theta_0-\theta_*)\right\|.\end{aligned}$$

■

D.2 Theorem 5 and Equation (13)

Theorem 5 and Equation (13) are directly derived from Theorem 2 and Theorem 3, using Lemmas 13 and 14.

To derive corollaries for the optimal γ, one has to find the γ that balances the bias and variance term and to compute the products for such a step size.

D.2.1 Equation (13)

We derive from Theorem 2, when choosing $\gamma = (\lambda n)^{-1}$, and using Lemmas 13 and 14, the following bound, under assumptions of Theorem 2 :

$$\mathbb{E}f(\bar{\theta}_n) - f(\theta_*) \leq \frac{(18 + \mathrm{Res}(n,b,r,\gamma))\|\Sigma^{r/2}(\theta_0 - \theta_*)\|^2}{(\gamma n)^{\frac{1-r}{2}}} + \frac{6\sigma^2 \operatorname{tr}(\Sigma^b)\gamma^b}{n^{1-b}}.$$

Where $\mathrm{Res}(n,b,r,\gamma) := 3\gamma^{1+b}n^b \operatorname{tr}(\Sigma^b)$ if $-1 \leq r \leq 0$ and $\mathrm{Res}(n,b,r,\gamma) := 0$ if $0 \leq r \leq 1$. When choosing the optimal $\gamma \propto n^{\frac{-b+r}{b+1-r}}$, we have that $\gamma^{1+b}n^b = n^{-1+\frac{1+b}{1+b-r}} = n^{\chi}$, with $\chi = \frac{-r}{1+b-r} \geq 0$ if $r \leq 0$. Thus the residual term is always vanishing for $r \leq 0$ and does not exist for $r \geq 0$.

D.2.2 Theorem 5

Theorem 5 directly follows from Lemmas 13 and 14 and the choice of $\gamma \propto n^{\frac{-2b+2r-1}{b+1-r}}$.

Appendix E. Technical Lemmas

The following sequence of Lemmas appear in the proof. They are mostly independent and rely on simple calculations.

Lemma 15 *The operator* $\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}$ *is a non-decreasing operator on* $(S_n, \preccurlyeq)$.

Proof Lemma means that for two matrices $M, N \in S_n(\mathbb{R})$ such that $M \preccurlyeq N$, then

$$\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1} M \preccurlyeq \big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1} N.$$

It is equivalent to show that for any symmetric positive matrix $A \in S_n^+$,

$$\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1} A \in S_n^+(\mathbb{R}).$$

We consider a matrix $A \in S_n^+(\mathbb{R})$. A can be decomposed as a sum of (at most) n rank one matrices $A = \sum_{i=1}^n \omega_i \omega_i^\top$, with $\omega_i \in \mathbb{R}^n$. We thus just have to prove that for some $\omega \in \mathbb{R}^n$, $\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}\omega\omega^\top \in S_n^+(\mathbb{R})$.

Let $\Sigma = \sum_{i\geqslant 0}\mu_i e_i \otimes e_i$ is the eigenvalue decomposition of Σ, then

$$\big[(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)\big]^{-1}\omega\omega^\top = \sum_{i,j\geqslant 0}\frac{\langle\omega,e_i\rangle\langle\omega,e_j\rangle}{\mu_i+\mu_j+2\lambda}e_i\otimes e_j.$$

Thus, in the orthonormal basis of eigenvectors, this is thus Hadamard product between

$$\sum_{i,j\geqslant 0}\langle\omega,e_i\rangle\langle\omega,e_j\rangle e_i\otimes e_j = \omega\omega^\top$$

and the matrix $C = \Big(\big(\frac{1}{\mu_i+\mu_j+2\lambda}\big)_{i,j\geqslant 0}\Big)$. Matrix C is a Cauchy matrix and is thus positive. Moreover the Hadamard product of two positive matrices is positive, which concludes the proof. ■

Remark: surprisingly, the inverse operator $(\Sigma+\lambda I)\otimes I + I\otimes(\Sigma+\lambda I)$ is not non-decreasing. Indeed, $\preccurlyeq$ is not a total order on S_n so we may have that an operator is non-decreasing and its inverse is not.

Lemma 16 *For all* $\rho \in (0,1)$ *and* $\omega \in [-\pi/2;\pi/2]$ *and* $r^\pm = \rho(\cos(\omega) \pm \sqrt{-1}\sin(\omega))$ *we have:*

$$\left|\frac{1 - r^+r^- - \rho^k|A_1|}{|1-r^+|}\right| \leq \min\{1+\rho+e^{-1}+4\rho^k, 2+\rho+\sqrt{5}\rho^{k+1}\} \leq 6. \tag{32}$$

Proof We note that $\rho_i^k A_1$ is a real number as is is a quotient of pure complex numbers, which come from the difference between a complex and its conjugate. We first write A_1 as a combination of sine and cosine functions:

$$\begin{aligned}
\rho_i^k A_1 &= \frac{(r_i^-)^{k+1}(1-r_i^+)^2 - (r_i^+)^{k+1}(1-r_i^-)^2}{r_i^- - r_i^+} \\
&= -\frac{(r_i^-)^{k+1} - (r_i^+)^{k+1} - 2r_i^-r_i^+((r_i^-)^k - (r_i^+)^k) + (r_i^-r_i^+)(r_i^-)^{k-1} - (r_i^+)^{k-1})}{\rho_i \sin\omega_i} \\
&= -\frac{\rho_i^{k+1}\sin((k+1)\omega_i) - 2\rho_i^{k+2}\sin(k\omega_i) + \rho_i^{k+3}\sin((k-1)\omega_i)}{\rho_i\sin\omega_i}.
\end{aligned}$$

This quantity can be simplified when $\rho \to 1$ or $\omega \to 0$. We thus modify the expression of A_1 to make these dependencies clearer:

$$\begin{aligned}
-A_1 &= \frac{\sin((k+1)\omega_i) - 2\rho_i\sin(k\omega_i) + \rho_i^2\sin((k-1)\omega_i)}{\sin\omega_i} \\
&= \frac{(\cos(\omega)-\rho)(\sin(k\omega) - \rho\sin((k-1)\omega)) + \cos(k\omega)\sin(\omega) - \rho\cos((k-1)\omega)\sin(\omega)}{\sin\omega_i} \\
&\text{developing } \sin(a+b) = \sin(a)\cos(b) + \cos(a)\sin(b) \text{ and regrouping terms,} \\
&= \frac{(\cos(\omega)-\rho)^2\sin((k-1)\omega) + (\cos(\omega)-\rho)\sin(\omega)\cos((k-1)\omega) + \cos(k\omega)\sin(\omega)}{\sin\omega_i} \\
&\qquad + \frac{-\rho\cos((k-1)\omega)\sin(\omega)}{\sin\omega_i} \\
&= \frac{(\cos(\omega)-\rho)^2\sin((k-1)\omega)}{\sin\omega_i} + (\cos(\omega)-\rho)\cos((k-1)\omega) + \cos(k\omega) - \rho\cos((k-1)\omega) \\
&\text{simplifying expression, then developing the cosine,} \\
&= \frac{(\cos(\omega)-\rho)^2\sin((k-1)\omega)}{\sin\omega_i} + 2(\cos(\omega)-\rho)\cos((k-1)\omega) + \sin(\omega)\sin((k-1)\omega). \qquad (33)
\end{aligned}$$

So that in that final expression all the terms behave relatively simply when $\rho \to 1$ or $\omega \to 0$. We want to upper bound:

$$\left| \frac{1 - r^+ r^- - \rho^k |A_1|}{|1 - r^+|} \right|.$$

We thus consider separately the first and second term.

$$\frac{1 - r_i^+ r_i^-}{|1 - r_i^+|} = \frac{1 - \rho^2}{|1 - r_i^+|} \leq 1 + \rho \qquad \text{(exact if } \omega = 0\text{)}.$$

Then, using Equation (33):

$$\frac{-\rho_i^k |A_1|}{|1 - r_i^+|} = \rho^k \frac{\frac{(\cos(\omega) - \rho)^2 \sin((k-1)\omega)}{\sin \omega_i} + 2(\cos(\omega) - \rho)\cos((k-1)\omega) + \sin(\omega)\sin((k-1)\omega)}{\sqrt{(1 - \rho\cos\omega)^2 + \rho^2 \sin^2(\omega)}}.$$

Considering separately the three terms in the numerator, using numerous times that for any $a, b \in [0;1]$, $|a - b| \leq 1 - ab$:

$$\begin{aligned}
\left| \frac{\rho^k \frac{(\cos(\omega) - \rho)^2 \sin((k-1)\omega)}{\sin \omega_i}}{\sqrt{(1 - \rho\cos\omega)^2 + \rho^2 \sin^2(\omega)}} \right| &\leq \rho^k \frac{(\cos(\omega) - \rho)\sin((k-1)\omega)}{\sin \omega_i} \\
&\quad \text{as } |(\cos(\omega) - \rho)| \leq 1 - \rho\cos(\omega), \\
&\leq \rho^k \frac{(\cos(\omega) - 1)\sin((k-1)\omega)}{\sin \omega_i} + \rho^k \frac{(1-\rho)\sin((k-1)\omega)}{\sin \omega_i} \\
&\quad \text{writing } \cos(\omega) - \rho = \cos(\omega) - 1 + 1 - \rho \\
&\leq \rho^k (1-\rho)(k-1) + \rho^k \frac{(\cos(\omega) - 1)\sin((k-1)\omega)}{\sin \omega_i} \\
&\quad \text{as } |\sin((k-1)\omega)| \leq |(k-1)\sin(\omega)|, \\
&\leq \rho^k (1-\rho)k - (1-\rho)\rho^k + \rho^k \frac{(\cos(\omega) - 1)\sin((k-1)\omega)}{\sin \omega_i} \\
&\quad \text{writing } \cos(\omega) - 1 = 2\sin^2(\omega/2), \\
&\leq \rho^k (1 + (1-\rho))^k - \rho^k - (1-\rho)\rho^k + \rho^k \frac{2\sin^2(\omega/2)}{\sin \omega_i} \\
&\quad \text{using } 1 + (1-\rho)k \leq (1 + (1-\rho))^k, \\
&\leq \rho^k (1 + (1-\rho))^k - \rho^k - (1-\rho)\rho^k + \rho^k \tan(\omega/2) \\
&\quad \text{and as } \tan(\omega/2) \leq 1 \text{ for } |\omega| \leq \pi/2, \\
&\leq 1 - (1-\rho)\rho^k \\
&\quad \text{using } \rho^k (1 + (1-\rho))^k = (1 - (1-\rho)^2)^k \leq 1,
\end{aligned}$$

And for the second and third term:

$$\begin{aligned}
2\left| \rho^k \frac{(\cos(\omega) - \rho)\cos((k-1)\omega)}{\sqrt{(1 - \rho\cos\omega)^2 + \rho^2 \sin^2(\omega)}} \right| &\leq 2\rho^k, \\
\left| \rho^k \frac{+\sin(\omega)\sin((k-1)\omega)}{\sqrt{(1 - \rho\cos\omega)^2 + \rho^2 \sin^2(\omega)}} \right| &\leq \rho^k.
\end{aligned}$$

Thus:

$$\left| \frac{1 - r_i^+ r_i^- - \rho_i^k |A_1|}{|1 - r_i^+|} \right| \leq 1 + \rho + 1 + 3\rho^k.$$

We also have

$$\begin{aligned}
|\rho^k \frac{\frac{(\cos(\omega) - \rho)^2 \sin((k-1)\omega)}{\sin \omega_i}}{\sqrt{(1 - \rho\cos\omega)^2 + \rho^2 \sin^2(\omega)}}| &\leq \rho^k \frac{(\cos(\omega) - \rho)\sin((k-1)\omega)}{\sin \omega_i} \\
&\leq \rho^k (1-\rho)(k-1) + \rho^k \frac{(\cos(\omega) - 1)\sin((k-1)\omega)}{\sin \omega_i} \\
&\leq (1 - \frac{1}{k+1})^{k+1} - (1-\rho)\rho^k + \rho^k \frac{(\cos(\omega) - 1)\sin((k-1)\omega)}{\sin \omega_i} \\
&\leq e^{-1} - (1-\rho)\rho^k + \rho^k \frac{\sin^2(\omega/2)}{\sin \omega_i}.
\end{aligned}$$

Using that

$$k \sup_{x \in [0;1]} x^k (1-x) = k \frac{1}{k+1}(1 - \frac{1}{k+1})^k = (1 - \frac{1}{k+1})^{k+1} = \exp((k+1)\ln((1 - \frac{1}{k+1})) \leq e^{-1}, \quad (34)$$

we get

$$\left| \frac{1 - r_i^+ r_i^- - \rho_i^k |A_1|}{|1 - r_i^+|} \right| \leq 1 + \rho + e^{-1} + 4\rho^k$$

We can also change $3\rho^k$ into $\sqrt{5}\rho^k$ We have used that $|(\rho - \cos(\omega))| \leq (1 - \rho\cos(\omega))$. ■

Lemma 17 *For any $\rho_i \in (0;1)$, for any $\omega_i \in [-\pi/2; \pi/2]$*

$$\frac{\rho_i^j \sin(\omega_i(j+1))}{\sin(\omega_i)} - \rho_i^{j+1} \frac{\sin(\omega_i j)}{\sin(\omega_i)} \leq 1 + e^{-1}.$$

Proof

$$\begin{aligned}
\frac{\rho_i^j \sin(\omega_i(j+1))}{\sin(\omega_i)} - \rho_i^{j+1} \frac{\sin(\omega_i j)}{\sin(\omega_i)} &= \rho_i^j \left(\frac{\sin(\omega_i(j+1)) - \rho_i \sin(\omega_i j)}{\sin(\omega_i)} \right) \\
&= \rho_i^j \left(\frac{(\cos(\omega_i) - \rho_i)\sin(\omega_i j)}{\sin(\omega_i)} + \cos(j\omega_i) \right) \\
&\leq \rho_i^j \left((1 - \rho_i) j + 1 \right) \\
&\leq 1 + e^{-1} \text{ using (34)}.
\end{aligned}$$

Lemma 18 *For all $\rho \in (0,1)$ and $\omega \in [-\pi/2;\pi/2]$ and $r^{\pm} = \rho(\cos(\omega) \pm \sqrt{-1}\sin(\omega))$ we have:*

$$\left|\rho_i^k B_{1,k}\right| \leq 1.75 \tag{35}$$

Proof Once again, as the considered quantity is real, we first express it as a combination of sine and cosine functions. We then use some simple trigonometric trics to upper bound the quantity.

$$\begin{aligned}
\rho_i^k B_{1,k} &= -\frac{(r_i^-)^{k+1}(1-r_i^+) - (r_i^+)^{k+1}(1-r_i^-)}{r_i^+ - r_i^-} \\
&= -\frac{2\mathfrak{Im}\big[(r_i^-)^{k+1}(1-r_i^+)\big]}{\sqrt{-\Delta_i}} \text{ as it is the difference between a complex and its conjugate,} \\
&= -\frac{\mathfrak{Im}\big[\rho_i^k e^{-(k+1)i\omega_i}(1-\rho_i\cos(\omega_i) - i\rho_i\sin(\omega_i))\big]}{\sin\omega_i\rho_i} \text{ developing the product,} \\
&= \rho_i^k \frac{\cos((k+1)\omega_i)\sin(\omega_i)\rho_i + \sin((k+1)\omega_i)(1-\rho_i\cos(\omega_i))}{\sin\omega_i\rho_i} \\
&= \rho_i^k\Big[\rho_i\cos((k+1)\omega_i) + (1-\rho_i\cos(\omega_i))\frac{\sin((k+1)\omega_i)}{\sin\omega_i}\Big] \text{ and simplifying.}
\end{aligned}$$

Let's turn our interest to the second part of the quantity:

$$\begin{aligned}
\left|\rho_i^k(1-\rho_i\cos(\omega_i))\frac{\sin((k+1)\omega_i)}{\sin\omega_i}\right| &= \left|\rho_i^k(1-\rho_i+\rho_i(1-\cos(\omega_i)))\frac{\sin((k+1)\omega_i)}{\sin\omega_i}\right| \\
&\text{introducing an artificial } +\rho_i - \rho_i, \\
&\leq \rho_i^k\left|(1-\rho_i)\frac{\sin((k+1)\omega_i)}{\sin\omega_i}\right| + \rho_i^k\left|\rho_i(1-\cos(\omega_i))\frac{\sin((k+1)\omega_i)}{\sin\omega_i}\right| \\
&\text{by triangular inequality,} \\
&\leq \rho_i^k\left|(1-\rho_i)(k+1)\right| + \rho_i^k\left|\rho_i\sin^2(\tfrac{\omega}{2})\frac{1}{2\cos(\frac{\omega}{2})\sin(\frac{\omega}{2})}\right| \\
&\text{using } 1-\cos(\omega_i) = 2\sin^2(\frac{\omega}{2}) \\
&\leq \rho_i^k(1-\rho_i)k + \rho_i^k(1-\rho) + \rho_i^k\left|\rho_i\sin^2(\tfrac{\omega}{2})\frac{1}{2\cos(\frac{\omega}{2})\sin(\frac{\omega}{2})}\right| \\
&\leq (1-(1-\rho_i))^k(1+(1-\rho_i))^k - \rho_i^k + \frac{1}{2(k+1)} + \rho_i^k\left|\frac{\rho_i}{2}\tan(\frac{\omega}{2})\right| \\
&\leq (1-(1-\rho_i)^2)^k + \frac{1}{4} + \frac{1}{2} \leq 1 + \frac{1}{4} + \frac{1}{2} - \rho_i^k.
\end{aligned}$$

Thus

$$\left|\rho_i^k B_{1,k}\right| = \rho_i^k + 1 + \frac{1}{4} + \frac{1}{2} - \rho_i^k \leq 1 + \frac{1}{4} + \frac{1}{2} = 1.75.$$

Lemma 19 *For any $s_i, \gamma, \lambda \in \mathbb{R}_+^3$ such that $\gamma(s_i+\lambda) \leq 1$, for any $k \in \mathbb{N}$, we have the two following highly related identities:*

$$\begin{aligned}
0 \leq 2 - \sqrt{\gamma(s_i+\lambda)} - \big(2 + (k-1)\sqrt{\gamma(s_i+\lambda)}\big)(1-\sqrt{\gamma(s_i+\lambda)})^k &\leq 2 \\
0 \leq 1 - (1 + k\sqrt{\gamma(s_i+\lambda)})(1-\sqrt{\gamma(s_i+\lambda)})^k &\leq 1.
\end{aligned}$$

Proof Proof relies on the trick, for any $\alpha \in \mathbb{R}, n \in \mathbb{N}$: $1 + n\alpha \leq (1+\alpha)^n$. For the first one:

$$\begin{aligned}
&\sqrt{\gamma(s_i+\lambda)} + \big(2 + (k-1)\sqrt{\gamma(s_i+\lambda)}\big)(1-\sqrt{\gamma(s_i+\lambda)})^k = \\
=\ &\sqrt{\gamma(s_i+\lambda)} + (1-\sqrt{\gamma(s_i+\lambda)})^k + \big(1 + (k-1)\sqrt{\gamma(s_i+\lambda)}\big)(1-\sqrt{\gamma(s_i+\lambda)})^k \\
\leq\ &\sqrt{\gamma(s_i+\lambda)} + (1-\sqrt{\gamma(s_i+\lambda)}) + \big(1 + (k-1)\sqrt{\gamma(s_i+\lambda)}\big)(1-\sqrt{\gamma(s_i+\lambda)})^{k-1} \\
\leq\ &1 + (1-\gamma(s_i+\lambda))^{k-1} \leq 2.
\end{aligned}$$

For the second one:

$$0 \leq (1 + k\sqrt{\gamma(s_i+\lambda)})(1-\sqrt{\gamma(s_i+\lambda)})^k \leq (1-\gamma(s_i+\lambda))^k \leq 1.$$

References

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

F. Bach. Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression. *J. Mach. Learn. Res.*, 15(1):595–627, 2014.

F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

L. Birgé. An Alternative Point of View on Lepski's Method. *Lecture Notes-Monograph Series*, 36:113–133, 2001.

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

A. Cotter, O. Shamir, N. Srebro, and K. Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *Advances in Neural Information Processing Systems (NIPS)*. 2011.

F. Cucker and S. Smale. Best choices for regularization parameters in learning theory: on the bias-variance problem. *Found. Comput. Math.*, 2:413–418, 2002.

A. d'Aspremont. Smooth optimization with approximate gradient. *SIAM J. Optim.*, 19(3): 1171–1183, 2008.

E. De Vito, A. Caponetto, and L. Rosasco. Model selection for regularized least-squares algorithm in learning theory. *Found. Comput. Math.*, 5:59–85, 2005.

A. Défossez and F. Bach. Averaged least-mean-squares: bias-variance trade-offs and optimal sampling distributions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics, (AISTATS)*, 2015.

O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao. Optimal distributed online prediction using mini-batches. *J. Mach. Learn. Res.*, 13:165–202, 2012.

O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Math. Program.*, 146(1-2, Ser. A):37–75, 2014.

A. Dieuleveut and F. Bach. Non-parametric stochastic approximation with large step sizes. *Annals of Statistics*, 44(4):1363–1399, 2015.

M. Duflo. *Random Iterative Models*. Springer, 1997.

H. W. Engl, M. Hanke, and Neubauer A. Regularization of Inverse Problems. *Klüwer Academic Publishers*, 1996.

N. Flammarion and F. Bach. From averaging to acceleration, there is only a step-size. In *Proceedings of the International Conference on Learning Theory (COLT)*, 2015.

C. Gu. *Smoothing Spline ANOVA Models*. Springer, 2013.

L. Györfi and H. Walk. On the averaged stochastic approximation for linear regression. *SIAM J. Optim.*, 34(1):31–61, 1996.

L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, 2006.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, second edition, 2009.

D. Hsu, S. M. Kakade, and T. Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014.

H. Kushner and G G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

G. Lan. An optimal method for stochastic composite optimization. *Math. Program.*, 133 (1-2, Ser. A):365–397, 2012.

G. Lecué and S. Mendelson. Performance of empirical risk minimization in linear aggregation. *Bernoulli*, 22(3):1520–1534, 2016.

J. Lin and L. Rosasco. Optimal learning for multi-pass stochastic gradient methods. *arXiv preprint arXiv:1605.08882*, 2016.

P. Massart. *Concentration Inequalities and Model Selection*. Lecture Notes in Mathematics. Springer, 2007.

P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Monographs on Statistics and Applied Probability. Chapman & Hall, London, second edition, 1989.

A. S. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19(4):1574–1609, 2009.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Y. Nesterov. *Introductory Lectures on Convex Optimization*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004. A basic course.

B. O'Donoghue and E. Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, pages 1–18, 2013.

R. I. Oliveira. The lower tail of random quadratic forms with applications to ordinary least squares. *Probab. Theory Related Fields*, 166(3-4):1175–1194, 2016.

F. Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems (NIPS)*. 2014.

B. T. Polyak. Some methods of speeding up the convergence of iteration methods. {*USSR*} *Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

B. T. Polyak. *Introduction to Optimization*. Translations Series in Mathematics and Engineering. Optimization Software, Inc., Publications Division, New York, 1987.

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, 1992.

H. Robbins and S. Monro. A stochastic approxiation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

A. Rudi, R. Camoriano, and L. Rosasco. Less is More: Nyström Computational Regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Stochastic convex optimization. In *Proceedings of the International Conference on Learning Theory (COLT)*, 2009.

I. Steinwart and A. Christmann. *Support Vector Machines*. Springer Series in Information Science and Statistics. Springer, 2008.

P. Tarrès and Y. Yao. Online learning as stochastic approximation of regularization paths. *EEE Transactions in Information Theory*, (99):5716–5735, 2011.

A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2008.

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.*, 11:2543–2596, 2010.

Y. Yao. *A Dynamic Theory of Learning*. PhD thesis, University of California at Berkeley, 2006.

Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Y. Ying and M. Pontil. Online gradient descent learning algorithms. *Foundations of Computational Mathematics*, 2008.

T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. *Proceedings of the Conference on Machine Learning (ICML)*, 2004.

Journal of Machine Learning Research 18 (2017) 1-28 Submitted 11/16; Revised 9/17; Published 10/17

Confidence Sets with Expected Sizes for Multiclass Classification

Christophe Denis Christophe.Denis@u-pem.fr
LAMA UMR-CNRS 8050
Université Paris-Est Marne-la-Vallée
5 Bd Descartes, 77454 Marne-la-Vallée cedex 2, France

Mohamed Hebiri Mohamed.Hebiri@u-pem.fr
LAMA UMR-CNRS 8050
Université Paris-Est Marne-la-Vallée
5 Bd Descartes, 77454 Marne-la-Vallée cedex 2, France

Editor: Gábor Lugosi

Abstract

Multiclass classification problems such as image annotation can involve a large number of classes. In this context, confusion between classes can occur, and single label classification may be misleading. We provide in the present paper a general device that, given an unlabeled dataset and a score function defined as the minimizer of some empirical and convex risk, outputs a set of class labels, instead of a single one. Interestingly, this procedure does not require that the unlabeled dataset explores the whole classes. Even more, the method is calibrated to control the expected size of the output set while minimizing the classification risk. We show the statistical optimality of the procedure and establish rates of convergence under the Tsybakov margin condition. It turns out that these rates are linear on the number of labels. We apply our methodology to convex aggregation of confidence sets based on the V-fold cross validation principle also known as the superlearning principle (van der Laan et al., 2007). We illustrate the numerical performance of the procedure on real data and demonstrate in particular that with moderate expected size, w.r.t. the number of labels, the procedure provides significant improvement of the classification risk.

Keywords: Multiclass Classification, Confidence Sets, Empirical Risk Minimization, Convex Loss, Superlearning

1. Introduction

The advent of high-throughput technology has generated tremendous amounts of large and high-dimensional classification data. This allows classification at unprecedented scales with hundreds or even more classes. The standard approach to classification in the multiclass setting is to use a classification rule for assigning a single label. More specifically, it consists in assigning a single label $Y \in \mathcal{Y}$, with $\mathcal{Y} = \{1, \dots, K\}$, to a given input example $X \in \mathcal{X}$ among a collection of labels. However, while a large number of classes can lead to precise characterizations of data points, similarities among classes also bear the risk of confusion and misclassification. Hence, assigning a single label can lead to wrong or ambiguous results.

In this paper, we address this problem by introducing an approach that yields sets of labels as outputs, namely, confidence sets. A confidence set Γ is a function that maps $\mathcal{X}$ onto $2^{\mathcal{Y}}$. A natural way to obtain a set of labels is to use ranked outputs of the classification rule. For example, one could take the classes that correspond to the top-level conditional probabilities $\mathbb{P}(Y = \cdot | X = x)$. Here, we provide a more general approach where we control the *expected* size of the confidence sets. For a confidence set Γ, the expected size of Γ is defined as $\mathbb{E}[|\Gamma(X)|]$, where $|\cdot|$ stands for the cardinality. For a sample X and given an expected set size β, we provide an algorithm that outputs a set $\hat{\Gamma}(X)$ such that $\mathbb{E}[|\hat{\Gamma}(X)|] \approx \beta$. Furthermore, the procedure aims at minimizing the classification error given by

$$\mathbb{P}\left(Y \notin \hat{\Gamma}(X)\right) \approx \min_{\Gamma\,:\,\mathbb{E}[|\Gamma(X)|]=\beta} \mathbb{P}\left(Y \notin \Gamma(X)\right) = \mathcal{R}^*_\beta.$$

We establish a close formula of the oracle confidence set $\Gamma^* = \operatorname{argmin}_{\Gamma\,:\,\mathbb{E}[|\Gamma(X)|]=\beta} \mathbb{P}\left(Y \notin \Gamma(X)\right)$ that involves the cumulative distribution functions of the conditional probabilities. Besides, we formulate a data-driven counterpart of Γ^* based on the minimization of the empirical risk. However, the natural risk function in the multiclass setting is non convex, and then minimizing it is proving to be a computational issue. As a remedy, convex surrogates are often used in machine learning, and more specifically in classification as reflected by the popularity of methods such as Boosting (Freund and Schapire, 1997), logistic regression (Friedman et al., 2000) and support vector machine (Vapnik, 1998). In our problem this translates by considering some convex surrogate of the 0/1-loss in $\mathbb{P}\left(Y \notin \Gamma(X)\right)$; we introduce a new convex risk function which enables us to emphasize specific aspects of confidence sets. That convex risk function is partly inspired by the works by Zhang (2004), Bartlett et al. (2006) and Yuan and Wegkamp (2010), that deal with binary classification.

Our approach displays two main features. First, our method can be implemented in a semi-supervised way (Vapnik, 1998). More precisely, the method is a two-steps procedure that requires the estimation of score functions (as minimizers of some convex risk function) in a first step and the estimation of the cumulative distribution of these scores in a second one. Hence, the first step requires labeled data whereas the second one involves only unlabeled samples. Notably, the unlabeled sample does not necessary consists of examples coming from the whole classes. This aspects is fondamental when we deal with a large number of classes, some of which been sparsely represented. Second, from the theoretical point of view, we provide an oracle inequality satisfied by $\hat{\Gamma}$, the empirical confidence set that results from the minimization of an empirical convex risk. The obtained oracle inequality shall enable us to derive rates of convergence on a particular class of confidence sets under the Tsybakov noise assumption on the data generating distribution. These rates are linear in the number of classes K and also depend on the regularity of the cumulative distribution functions of the conditional probabilities.

An obvious benefit of considering convex risk minimization is when we deal with aggregation. However, aggregating confidence sets is no simple matter. Another contribution of the present paper is about applying the above methodology to aggregation of confidence sets. More specifically, we provide a generalization of the superlearning algorithm (van der Laan et al., 2007) initially introduced in the context of regression and binary classification. This algorithm relies on the V-fold cross-validation principle. We prove the consistency of this aggregation procedure and illustrate its relevance on real datasets. Let us point out that any arbitrary library of machine learning algorithms may be used in this aggregation procedure; we propose in the present paper to exploit support vector machines, random for-

©2017 Christophe Denis and Mohamed Hebiri.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-596.html.

est procedures or softmax regression since these methods are popular in machine learning and that each of them is associated to a different shape.

We end up this discussion by highlighting in two words what we perceive as being the main contributions of the present paper. We describe an optimal strategy for building confidence sets in multiclass classification setting, and we derive a new aggregation procedure for confidence sets that still allows controlling the expected size of the resulting confidence set.

Related works: The closest learning task to the present work is *classification with reject option* which is a particular setting in binary classification. Several papers fall within the scope of this area (Chow, 1970; Herbei and Wegkamp, 2006; Yuan and Wegkamp, 2010; Wegkamp and Yuan, 2011; Lei, 2014; Denis and Hebiri, 2015) and differ from each other by the goal they consider. Among these references, our procedure is partially inspired by Denis and Hebiri (2015) that also consider a semi-supervised approach to build confidence sets invoking some cumulative distribution functions (of the conditional probabilities themselves in their case). The similarity is however limited to the definition of oracle confidence sets, the oracle confidence in the present paper being an extension of the one defined in the paper by Denis and Hebiri (2015) to the multiclass setting. On the other hand, all the data-driven considerations are completely different (in particular, Denis and Hebiri (2015) focus on plug-in rules) and importantly, we develop here new probabilistic results on sums of cumulative distribution functions of random variables, that are of own interest.

Assigning a set of labels instead of a single one for an input example is not new (Vovk et al., 2005; Wu et al., 2004; del Coz et al., 2009; Lei et al., 2013; Choromanska et al., 2016). One of the most popular methods is based on *Conformal Prediction* approach (Vovk et al., 1999; Vovk, 2002; Vovk et al., 2005). In the multiclass classification framework, the goal of this algorithm is to build the smallest set of labels such that its classification error is below a pre-specified level. Since our procedure aims at minimizing the classification error while keeping under control the size of the set, *Conformal Prediction* can be seen as a dual of our method. It is worth mentioning that conformal predictors approaches need two labeled datasets where we only need one labeled dataset, the second being unlabeled. We refer to the very interesting statistical study of *Conformal Predictors* in the binary case in the paper by Lei (2014).

Notation: First, we state general notation. Let $\mathcal{Y} = \{1, \dots, K\}$, with $K \geq 2$ being an integer. Let (X, Y) be the generic data-structure taking its values in $\mathcal{X} \times \mathcal{Y}$ with distribution $\mathbb{P}$. The goal in classification is to predict the label Y given an observation of X. This is performed based on a classifier (or classification rule) s which is a function mapping $\mathcal{X}$ onto $\mathcal{Y}$. Let $\mathcal{S}$ be the set of all classifiers. The misclassification risk R associated with $s \in \mathcal{S}$ is defined as

$$R(s) = \mathbb{P}(s(X) \neq Y).$$

Moreover, the minimizer of R over $\mathcal{S}$ is the Bayes classifier, denoted by s^*, and is characterized by

$$s^*(\cdot) = \underset{k \in \mathcal{Y}}{\operatorname{argmax}}\, p_k(\cdot),$$

where $p_k(x) = \mathbb{P}(Y = k | X = x)$ for $x \in \mathcal{X}$ and $k \in \mathcal{Y}$.
Let us now consider more specific notation related to the multiclass confidence set setting.

Let a confidence set be any measurable function that maps $\mathcal{X}$ onto $2^{\mathcal{Y}}$. Let Γ be a confidence set. This confidence set is characterized by two attributes. The first one is the risk associated to the confidence set

$$\mathcal{R}(\Gamma) = \mathbb{P}(Y \notin \Gamma(X)), \tag{1}$$

and is related to its accuracy. The second attribute is linked to the information given by the confidence set. It is defined as

$$\mathcal{I}(\Gamma) = \mathbb{E}(|\Gamma(X)|), \tag{2}$$

where $|\cdot|$ stands for the cardinality. Moreover, for some $\beta \in [1, K]$, we say that, for two confidence sets Γ and Γ' such that $\mathcal{I}(\Gamma) = \mathcal{I}(\Gamma') = \beta$, the confidence set Γ is "better" than Γ' if $\mathcal{R}(\Gamma) \leq \mathcal{R}(\Gamma')$.

Organization of the paper: The rest of the paper is organized as follows. Next section is devoted to the definition and the main properties of the oracle confidence set for multiclass classification. The empirical risk minimization procedure is provided in Section 3. Rates of convergence for the confidence set that results from this minimization can also be found in this section. We present an application of our procedure to aggregation of confidence sets in Section 4. We finally draw some conclusions and present perspectives of our work in Section 5. Proofs of our results are postponed to the Appendix.

2. Confidence set for multiclass classification

In the present section, we define a class of confidence sets that are suitable for multiclass classification and referred as *Oracle β-sets*. For some $\beta \in (0, K)$, these sets are shown to be optimal according to the risk (1) with an information (2) equal to β. Moreover, basic but fondamental properties of Oracle β-sets can be found in Proposition 1, while Proposition 7 provides another interpretation of these sets.

2.1 Notation and definition

First of all, we introduce in this section a class of confidence sets that specifies oracle confidence sets. Let $\beta \in (0, K)$ be a desired information level. The so-called *Oracle β-sets* are optimal according to the risk (1) among all the confidence sets Γ such that $\mathcal{I}(\Gamma) = \beta$. Throughout the paper we make the following assumption

(A1) *For all $k \in \{1, \dots, K\}$, the cumulative distribution function F_{p_k} of $p_k(X)$ is continuous.*

The definition of the *Oracle β-set* relies on the continuous and decreasing function G defined for any $t \in [0, 1]$ by

$$G(t) = \sum_{k=1}^{K} \bar{F}_{p_k}(t),$$

where for any $k \in \{1, \dots, K\}$, we denote by $\bar{F}_{p_k}$ the tail distribution function of $p_k(X)$, that is, $\bar{F}_{p_k} = 1 - F_{p_k}$ with F_{p_k} being the cumulative distribution function (c.d.f.) of $p_k(X)$. The generalized inverse G^{-1} of G is given by (see (van der Vaart, 1998)):

$$G^{-1}(\beta) = \inf\{t \in [0, 1] : \; G(t) \leq \beta\}, \;\; \forall \beta \in (0, K).$$

The functions G and G^{-1} are central in the construction of the Oracle β-sets. We then provide some of their useful properties in the following proposition.

Proposition 1 *The following properties on G hold*

i) For every $t \in (0,1)$ and $\beta \in (0,K)$, $G^{-1}(\beta) \leq t \Leftrightarrow \beta \geq G(t)$.

ii) For every $\beta \in (0,K)$, $G(G^{-1}(\beta)) = \beta$.

iii) Let ε be a random variable, independent of X, and distributed from a uniform distribution on $\{1,\dots,K\}$ and let U be uniformly distributed on $[0,K]$. Define

$$Z = \sum_{k=1}^{K} p_k(X)\mathbf{1}_{\{\varepsilon=k\}}.$$

If the function G is continuous, then $G(Z) \overset{\mathcal{L}}{=} U$ and $G^{-1}(U) \overset{\mathcal{L}}{=} Z$.

The proof of Proposition 1 relies on Lemma 1 in the Appendix. Now, we are able to defined the Oracle β-set:

Definition 2 *Let $\beta \in (0,K)$, the Oracle β-set is given by*

$$\begin{aligned}\Gamma^*_\beta(X) &= \{k \in \{1,\dots,K\} :\ G(p_k(X)) \leq \beta\} \\ &= \left\{k \in \{1,\dots,K\} :\ p_k(X) \geq G^{-1}(\beta)\right\}.\end{aligned}$$

This definition of the Oracle β-set is intuitive and can be related to the binary classification with reject option setting (Chow, 1970; Herbei and Wegkamp, 2006; Denis and Hebiri, 2015) in the following way: a label k is assigned to the Oracle β-set if the probability $p_k(X)$ is large enough. It is worth noting that the function G plays the same role as the c.d.f. of the score function used by Denis and Hebiri (2015). As emphasized by Proposition 1, their introduction allows to control exactly the information (2). Indeed, it follows from the definition of the Oracle β-set that for each $\beta \in (0,K)$

$$|\Gamma^*_\beta(X)| = \sum_{k=1}^{K} \mathbf{1}_{\{p_k(X) \geq G^{-1}(\beta)\}},$$

and then $\mathcal{I}(\Gamma^*_\beta) = \mathbb{E}\left[|\Gamma^*_\beta(X)|\right] = G(G^{-1}(\beta))$. Therefore, Proposition 1 ensures that

$$\mathcal{I}(\Gamma^*_\beta) = \beta.$$

This last display points out that the Oracle β-sets are indeed β-level (that is, its information equals β). In the next section, we focus on the study of the risk of these oracle confidence sets.

Remark 3 *Naturally, the definition of Oracle β-sets can be extended to any $\beta \in [0,K]$. However, the limit cases $\beta = 0$ and $\beta = K$ are of limited interest and are completely trivial. We then exclude these two limit cases from the present study.*

2.2 Properties of the oracle confidence sets

Let us first state the optimality of the Oracle β-set:

Proposition 4 *Let Assumption (A1) be satisfied. For any $\beta \in (0,K)$, we have both:*

1. *The Oracle β-set Γ^*_β satisfies the following property:*

$$\mathcal{R}\left(\Gamma^*_\beta\right) = \inf_{\Gamma} \mathcal{R}\left(\Gamma\right),$$

where the infimum is taken over all β-level confidence sets.

2. *For any β-level confidence set Γ, the following holds*

$$0 \leq \mathcal{R}\left(\Gamma\right) - \mathcal{R}\left(\Gamma^*_\beta\right) = \mathbb{E}\left[\sum_{k \in (\Gamma^*_\beta(X)\,\Delta\,\Gamma(X))} \left|p_k(X) - G^{-1}(\beta)\right|\right], \tag{3}$$

where the symbol Δ stands for the symmetric difference of two sets, that is, for two subsets A and B of $\{1,\dots,K\}$, we write $A\,\Delta\,B = (A \setminus B) \cup (B \setminus A)$.

Several remarks can be made from Proposition 4. First, for $\beta \in (0,K)$, the Oracle β-set is optimal for the misclassification risk, over the class of β-level confidence sets. Moreover, the excess risk of any β-level confidence set relies on the behavior of the score functions p_k around the threshold $G^{-1}(\beta)$. Finally, we can note that if $K = 2$ and $\beta = 1$, which implies that $G^{-1}(\beta) = 1/2$, Equation (3) reduces to the misclassification excess risk in binary classification.

Remark 5 *One way to build a confidence set Γ with information β is to set Γ as the β top levels conditional probabilities. In the sequel this method is referred as the* `max` *procedure. This strategy is natural but actually suboptimal as shown by the first point of Proposition 4. As an illustration, we consider a simulation scheme with $K = 10$ classes. We generate (X,Y) according to a mixture model. More precisely,*

i) the label Y is distributed from a uniform distribution on $\{1,\dots,K\}$;

ii) conditional on $Y = k$, the feature X is generated according to a multivariate gaussian distribution with mean parameter $\mu_k \in \mathbb{R}^{10}$ and identity covariance matrix. For each $k = 1,\dots,K$, the vectors μ_k are i.i.d realizations of uniform distribution on $[0,4]$.

For $\beta = 2$ we evaluate the risks of the Oracle β-set and the `max` *procedure and obtain respectively 0.05 and 0.09 (with very small variance).*

Remark 6 *An important motivation behind the introduction of confidence sets and in particular of Oracle β-sets is that they might outperform the Bayes rule which can be seen as the Oracle β-set associated to $\beta = 1$. This gap in performance is even larger when the number of classes K is large and there is a big confusion between classes. Such improvement will be illustrated in the numerical study (see Section 4.3).*

We end up this section by providing another characterization of the Oracle β-set.

Proposition 7 *For $t \in [0,1]$, and Γ a confidence set, we define*

$$L_t(\Gamma) = \mathbb{P}\left(Y \notin \Gamma(X)\right) + t\mathcal{I}(\Gamma).$$

For $\beta \in [0,K]$, the following equality holds:

$$L_{G^{-1}(\beta)}(\Gamma^*_\beta) = \min_{\Gamma} L_{G^{-1}(\beta)}(\Gamma).$$

The proof of this proposition relies on the same arguments as those of Proposition 4. It is then omitted. Proposition 7 states that the Oracle β-set is defined as the minimizer, over all confidence sets Γ, of the risk function L_t when the tuning parameter t is set equal to $G^{-1}(\beta)$. Note moreover that the risk function L_t is a trade-off, controlled by the parameter t, between the risk of a confidence set on the one hand, and the information provided by this confidence set on the other hand. Hence, the risk function L_t can be viewed as a generalization to the multiclass case of the risk function provided by Chow (1970) and Herbei and Wegkamp (2006) for binary classification with reject option setting.

3. Empirical risk minimization

In this section we introduce and study confidence sets which rely on the minimization of convex risks. Their definitions and main properties are given in Sections 3.1-3.2. As a consequence, we deduce a data-driven procedure described in Section 3.3 with several theoretical properties, such as rates of convergence, that we demonstrate in Section 3.4.

3.1 ϕ-risk

Let $f = (f_1, \ldots, f_K) : \mathcal{X} \to \mathbb{R}^K$ be a score function and $G_f(.) = \sum_{k=1}^K \bar{F}_{f_k}(.)$. Assuming that the function G_f is continuous and given an information level $\beta \in (0,K)$, there exists $\delta \in \mathbb{R}$, such that $G_f(-\delta) = \beta$. Given this simple but important fact, we define the confidence set $\Gamma_{f,\delta}$ associated to f and δ as

$$\Gamma_{f,\delta}(X) = \{k \,:\, f_k(X) \geq -\delta\}. \tag{4}$$

In this way, the confidence set $\Gamma_{f,\delta}$ consists of top scores, and the threshold δ is fixed so that $\mathcal{I}\left(\Gamma_{f,\delta}\right) = \beta$. As a consequence, we naturally aim at solving the problem

$$\min_{f \in \mathcal{F}} \mathcal{R}(\Gamma_{f,\delta}),$$

where $\mathcal{F}$ is a class of functions. Due to computational issues, it is convenient to focus on a convex surrogate of the previous minimization problem. To this end, let $\phi : \mathbb{R} \to \mathbb{R}$ be a convex function. We define the ϕ-risk of f by

$$R_\phi\left(f\right) = \mathbb{E}\left[\sum_{k=1}^K \phi(Z_k f_k(X))\right], \tag{5}$$

where $Z_k = 2\,\mathbf{1}_{\{Y=k\}} - 1$ for all $k = 1, \ldots, K$. Therefore, our target score becomes

$$\bar{f} \in \operatorname*{argmin}_{f \in \mathcal{F}} R_\phi\left(f\right),$$

for the purpose of building the confidence $\Gamma_{\bar{f},\delta}$. In the sequel, we also introduce f^*, the minimizer over the class of all measurable functions, of the ϕ-risk. The notation suppresses the dependence on ϕ. It is worth mentioning at this point that the definition of the risk function R_ϕ is dictated by Equation (3) and suits for confidence sets. Moreover, this function differs from the classical risk function used in the multiclass setting (see (Tewari and Bartlett, 2007)). The reason behind this is that building a confidence set is closer to K binary classification problems.

3.2 Classification calibration for confidence sets

Convexification of the risk in classification is a standard technique. In this section we adapt classical results and tools to confidence sets in the multiclass setting. We refer the reader to earlier papers for interesting developments (Zhang, 2004; Bartlett et al., 2006; Yuan and Wegkamp, 2010).

One of the main important concept when we deal with convexification of the risk is the notion of calibration of the loss function ϕ. This property permits to connect confidence sets deduced from the convex risk and from the classification risk.

Definition 8 *We say that the function ϕ is confidence set calibrated if for all $\beta > 0$, there exists $\delta^* \in \mathbb{R}$ such that*

$$\Gamma_{f^*,\delta^*} = \Gamma^*_\beta,$$

with f^* being the minimizer of the ϕ-risk

$$f^* \in \operatorname*{argmin}_f R_\phi\left(f\right),$$

where the infimum is taken over the class of all measurable functions. Hence, the confidence set based on f^* coincides with the Bayes confidence set.

Given this, we can state the following proposition that gives a characterization of the confidence set calibration property in terms of the function G.

Proposition 9 *The function ϕ is confidence set calibrated if and only if for all $\beta \in (0,K)$, there exists $\delta^* \in \mathbb{R}$ such that $\phi'(\delta^*)$ and $\phi'(-\delta^*)$ both exists, $\phi'(\delta^*) < 0$, $\phi'(-\delta^*) < 0$ and*

$$G^{-1}(\beta) = \frac{\phi'(\delta^*)}{\phi'(\delta^*) + \phi'(-\delta^*)}\,,$$

where ϕ' denotes the derivative of ϕ.

The proof of the proposition follows the lines of Theorem 1 in the paper by Yuan and Wegkamp (2010) with minor modifications. These characterizations of calibration for confidence sets generalize the notion of calibration in the classification setting as well as the necessary and sufficient condition for the minimizer of the ϕ-risk to be calibrated. Indeed, if we pick $\delta = 0$ in Definition 8 and Proposition 9 we exactly come back to the calibration property in the classical classification setting (see (Bartlett et al., 2006)). Note that commonly used loss functions as boosting ($x \mapsto \exp(-x)$), least squares ($x \mapsto (x-1)^2$) and logistic ($x \mapsto \log(1+\exp(-x))$) are examples of calibrated losses (see for instance (Bartlett

et al., 2006; Wegkamp and Yuan, 2011)). Now, for some score function f and some real number δ such that $G_f(-\delta) = \beta$, we define the excess risk

$$\Delta\mathcal{R}(\Gamma_{f,\delta}) = \mathcal{R}(\Gamma_{f,\delta}) - \mathcal{R}(\Gamma^*_\beta),$$

and the excess ϕ-risk

$$\Delta R_\phi(f) = R_\phi(f) - R_\phi(f^*).$$

We also introduce the marginal conditional excess ϕ-risk on $f = (f_1, \ldots, f_K)$ as

$$\Delta R^k_\phi(f(X)) = p_k(X)(\phi(f_k(X)) - \phi(f^*_k(X))) + (1 - p_k(X))(\phi(-f_k(X)) - \phi(-f^*_k(X))),$$

for $k = 1, \ldots, K$. The following theorem shows that the consistency in terms of ϕ-risk implies the consistency in terms of classification risk $\mathcal{R}$.

Theorem 10 *Assume that ϕ is confidence set calibrated and assume that there exists constants $C > 0$ and $s \geq 1$ such that*[1]

$$|p_k(X) - G^{-1}(\beta)|^s \leq C\Delta R^k_\phi(-\delta^*). \tag{6}$$

Let $\hat{f}_n$ be a sequence of scores. We assume that for each n, the function $G_{\hat{f}_n}$ is continuous. Let $\delta_n \in \mathbb{R}$ be such that $G_{\hat{f}_n}(-\delta_n) = \beta$, then

$$\Delta R_\phi(\hat{f}_n) \xrightarrow{P} 0 \qquad \Rightarrow \qquad \Delta\mathcal{R}(\Gamma_{\hat{f}_n,\delta_n}) \xrightarrow{P} 0.$$

The theorem ensures that the convergence in terms of ϕ risk implies the convergence in terms of risk $\mathcal{R}$. This convergence is made possible since we manage, in the proof, to bound the excess risk by (a power of) the excess ϕ-risk. The assumption needed in this theorem is also standard and is for instance satisfied for the boosting, least square and logistic losses with the parameter s being equal to 2 (see (Bartlett et al., 2006)).

3.3 Data-driven procedure

In this section we provide the steps of the construction of our empirical confidence set that is deduced from the empirical risk minimization. Before going into further details, let us first mention that our procedure is semi-supervised in the sense that it requires two datasets, one of which being unlabeled. Hence we introduce a first data set $\mathcal{D}_n = \{(X_i, Y_i),\ i = 1, \ldots, n\}$, which consists of independent copies of (X, Y). We define the empirical ϕ-risk associated to a score function f (which is the empirical counterpart of R_ϕ given in (5)):

$$\hat{R}_\phi(f) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\phi(Z^i_k f_k(X_i)), \tag{7}$$

where $Z^i_k = 2\,\mathbf{1}_{\{Y_i=k\}} - 1$ for all $k = 1, \ldots, K$. We also define the empirical risk minimizer over $\mathcal{F}$, a convex set of score functions, as

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \hat{R}_\phi(f).$$

1. With abuse of notation, we write $\Delta R^k_\phi(-\delta^*)$ instead of $\Delta R^k_\phi((-\delta^*, \ldots, -\delta^*))$ since no confusion can occur.

At this point, we have in hands the optimal score function $\hat{f}$ and need to build the corresponding confidence set with the right expected size. However, let us before introduce an intermediate confidence set that would help comprehension since it mimics the oracle β-set Γ^* with regard to its construction using $\hat{f}$ instead of f^*. For this purpose, we define

$$F_{\hat{f}_k}(t) = \mathbb{P}_X\left(\hat{f}_k(X) \leq t | \mathcal{D}_n\right),$$

for $t \in \mathbb{R}$, where $\mathbb{P}_X$ is the marginal distribution of X. As for the c.d.f. of p_k and f_k, we make the following assumption:

(A2) *The cumulative distribution functions $F_{\hat{f}_k}$ with $k = 1, \ldots, K$ are continuous.*

At this point, we are able to define an empirical approximation of the Oracle β-set for $\beta \in (0, K)$:

$$\widetilde{\Gamma}_\beta(X) = \left\{k \in \{1, \ldots, K\} :\ \widetilde{G}(\hat{f}_k(X)) \leq \beta\right\}, \tag{8}$$

where for $t \in \mathbb{R}$

$$\widetilde{G}(t) = \sum_{k=1}^{K} \bar{F}_{\hat{f}_k}(t), \tag{9}$$

with $\bar{F}_{\hat{f}_k} = 1 - F_{\hat{f}_k}$. Since the function $\widetilde{G}$ depends on the unknown distribution of X, we consider a second but interestingly *unlabeled* dataset $\mathcal{D}_N = \{X_i, i = 1, \ldots, N\}$, independent of $\mathcal{D}_n$ in order to compute the empirical versions of the $\bar{F}_{\hat{f}_k}$'s. By now, we can define the empirical β-set based on $\hat{f}$:

Definition 11 *Let $\hat{f}$ be the minimizer of the empirical ϕ-risk given in (7) based on $\mathcal{D}_n$, and consider the unlabeled dataset $\mathcal{D}_N$. Let $\beta \in (0, K)$. The empirical β-set is given by*

$$\hat{\Gamma}_\beta(X) = \left\{k \in \{1, \ldots, K\} :\ \hat{G}(\hat{f}_k(X)) \leq \beta\right\}, \tag{10}$$

where

$$\hat{G}(.) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\mathbf{1}_{\{\hat{f}_k(X_i)\geq .\}}.$$

The most important remark about the construction of this data-driven confidence set is that it is made in a semi-supervised way. Indeed, the estimation of the tail distribution functions of $\hat{f}_k$ requires only a set of *unlabeled* observations. This is particularly attractive in applications where the number of label observations is small (because labelling examples is time consuming or may be costly) and where one has in hand several unlabeled features that can be used to make prediction more accurate. As an important consequence is that the estimation of the tail distribution functions of $\hat{f}_k$ does not depend on the number of observations in each class label. That is to say, this unlabeled dataset can be unbalanced with respect to the classes, that can often occur in multiclass classification settings where the number of classes K is quite large. Next section deals with the theoretical performance of the empirical β-sets.

3.4 Rates of convergence

In this Section, we provide rates of convergence for the empirical confidence sets defined in Section 3.3. First, we state some additional notation. In the sequel, the symbols $\mathbf{P}$ and $\mathbf{E}$ stand for generic probability and expectation, respectively. Moreover, given the empirical β-set $\hat{\Gamma}_\beta$ from Definition 11, we consider the risk $\mathbf{R}\left(\hat{\Gamma}_\beta\right) = \mathbf{P}\left(Y \notin \hat{\Gamma}_\beta(X)\right)$ and the information $\mathbf{I}\left(\hat{\Gamma}_\beta\right) = \mathbf{E}\left[|\hat{\Gamma}_\beta(X)|\right]$. Our first result ensures that $\hat{\Gamma}_\beta$ is of level β up to a term of order $K/\sqrt{N}$.

Proposition 12 *For each $\beta \in (0, K)$, the following equality holds*

$$\mathbf{I}\left(\hat{\Gamma}_\beta\right) = \beta + O\left(\frac{K}{\sqrt{N}}\right).$$

In order to precise rates of convergence for the risk, we need to formulate several assumptions. First, we consider loss functions ϕ which have in common that the modulus of convexity of their underlying risk function R_ϕ, defined by

$$\delta(\varepsilon) = \inf\left\{\frac{R_\phi(f) + R_\phi(g)}{2} - R_\phi\left(\frac{f+g}{2}\right),\ \sum_{k=1}^{K}\mathbb{E}_X\left[(f_k - g_k)^2(X)\right] \geq \varepsilon^2\right\}, \qquad (11)$$

satisfies $\delta(\varepsilon) \geq c_1\epsilon^2$ for some $c_1 > 0$ (we refer to (Bartlett et al., 2006; Bartlett and Mendelson, 2006) for more details on modulus of convexity for classification risk). Moreover, we assume that ϕ is classification calibrated and L-Lipschitz for $L > 0$. On the other hand, for $\alpha > 0$, we assume a margin condition $\mathrm{M}_\alpha^{\mathrm{k}}$ on each $p_k, k = 1, \ldots, K$ (Tsybakov, 2004).

$\mathrm{M}_\alpha^{\mathrm{k}}$: $\mathbb{P}_X\left(0 < |p_k(X) - G^{-1}(\beta)| \leq t\right) \leq c_2 t^\alpha$, for some constant $c_2 > 0$ and for all $t > 0$.

It is important to note that since we assume that the distribution functions of $p_k(X)$ are continuous for each k, we have $\mathbb{P}_X\left(0 < |p_k(X) - G^{-1}(\beta)| \leq t\right) \to 0$ with $t \to 0$. Therefore, the margin condition only precise the rate of this decay to 0. Now, we provide the rate of convergence that we can expect for the empirical confidence sets defined by Equation (10).

Theorem 13 *Assume that $\|f\|_\infty \leq B$ for all $f \in \mathcal{F}$. Let $M_n = \mathcal{N}(1/n, L_\infty, \mathcal{F})$ be the cardinality of the set of closed balls with radius $1/n$ in L_∞-norm needed to cover $\mathcal{F}$. Under the assumptions of Theorem 10, with the modulos of convexity $\delta(\varepsilon) \geq c_1\epsilon^2$ with $c_1 > 0$, if ϕ is L-Lipschitz and if the margin assumptions $\mathrm{M}_\alpha^{\mathrm{k}}$ are satisfied with $\alpha > 0$, then*

$$\mathbf{E}\left[|\Delta\mathcal{R}(\hat{\Gamma}_\beta)|\right] \leq C(B, L, s, \alpha)K^{1-\alpha/(\alpha+s)}\left\{\inf_{f\in\mathcal{F}}\Delta R_\phi(f) + \frac{K\log(M_n)}{n}\right\}^{\alpha/(\alpha+s)} + C'\frac{K}{\sqrt{N}},$$

where $C' > 0$ is an absolute constant and $C(B, L, s, \alpha) > 0$ is a constant that depends only on L, B, s and α.

From this theorem we obtain the following rate of convergence: $K\left(\frac{\log(M_n)}{n}\right)^{\alpha/(\alpha+s)} + \frac{K}{\sqrt{N}}$. This is the first bound for confidence sets in multiclass setting. The regularity parameter α plays a crucial role and governs the rate; larger values of α lead to faster rates. Moreover,

this rate is linear in K, the number of classes, that seems to be the characteristic of multiclass classification as compared to binary classification. Note that the exponent $\alpha/(\alpha+s)$ is not classical and is due to the estimation of quantiles. The second part of the rates which is of order $K/\sqrt{N}$ relies on the estimation of the function $\tilde{G}$ given in (9). This part of the estimation is established under the mild Assumptions (A1) and (A2). For this term the proof of the linearity on K is tricky and is obtained thanks to a new technical probabilistic results on sums of cumulative distribution functions (see Lemma 1). Let us conclude this paragraph by mentioning that Theorem 13 applies for instance for $\mathcal{F}$ being the convex hull of a finite family of a score functions which is the scope of the next section.

4. Application to confidence sets aggregation

This Section is devoted to an aggregation procedure which relies on the superlearning principle. The specifics of our superlearning procedure is given in Section 4.1. In Section 4.2, we show the consistency of our procedure and finally we provide a numerical illustration in Section 4.3 of our algorithm.

4.1 Description of the procedure: superlearning

In this section, we describe the superlearning procedure for confidence sets. The superlearning procedure is based on the V-fold cross-validation procedure (see (van der Laan et al., 2007)). Initially, this aggregation procedure has been introduced in the context of regression and binary classification. Let $V \geq 2$ be an integer and let $(B_v)_{1\leq v\leq V}$ be a regular partition of $\{1, \ldots, n\}$, *i.e.*, a partition such that, for each $v = 1, \ldots, V$, $\mathrm{card}(B_v) \in \{\lfloor n/V\rfloor, \lfloor n/V\rfloor + 1\}$, where we write $\lfloor x\rfloor$ for the floor of any real number x (that is, $x - 1 \leq \lfloor x\rfloor \leq x$). For each $v \in \{1, \ldots, V\}$, we denote by $D_n^{(v)}$ (respectively $D_n^{(-v)}$) the dataset $\{(X_i, Y_i), i \in B_v\}$ (respectively $\{(X_i, Y_i), i \notin B_v\}$), and define the corresponding empirical measures

$$\begin{aligned} P_n^{(v)} &= \frac{1}{\mathrm{card}(B_v)}\sum_{i\in B_v}\mathrm{Dirac}(X_i, Y_i), \quad \text{and} \\ P_n^{(-v)} &= \frac{1}{n - \mathrm{card}(B_v)}\sum_{i\notin B_v}\mathrm{Dirac}(X_i, Y_i). \end{aligned}$$

For a score algorithm f, that is to say a function which maps the empirical distribution to a score function. We define the cross-validated risk of f as

$$\widehat{R}_\phi^n(f) = \frac{1}{V}\sum_{v=1}^{V} R_\phi^{(P_n^{(v)})}\left(f(P_n^{(-v)})\right), \qquad (12)$$

where for each $v \in \{1, \ldots, V\}$, $R_\phi^{(P_n^{(v)})}(f(P_n^{(-v)}))$ is the empirical estimator of $R_\phi(f(P_n^{(-v)})$, based on $D_n^{(v)}$ and conditionally on $D_n^{(-v)}$:

$$R_\phi^{(P_n^{(v)})}\left(f(P_n^{(-v)})\right) = \frac{1}{\mathrm{card}(B_v)}\sum_{i\in I_v}\sum_{k=1}^{K}\phi(Z_k^i f_k(P_n^{(-v)})(X_i)).$$

Next, we consider $\mathcal{F} = \left(f^1, \ldots, f^M\right)$ a family of M score algorithms. We define the cross-validated score by

$$\widehat{f} \in \underset{f \in \mathrm{conv}(\mathcal{F})}{\mathrm{argmin}} \ \widehat{R}^n_\phi(f). \tag{13}$$

Finally, we consider the resulting cross-validated confidence set defined by

$$\hat{\Gamma}^{\mathrm{CV}}_\beta(X) = \left\{k \in \{1, \ldots, K\} : \ \hat{G}(\hat{f}_k(X)) \leq \beta\right\},$$

that we analyse in the next section.

4.2 Consistency of the algorithm

In this Section, we show some results that illustrate the consistency of the superlearning procedure described above. To this end, we introduce the oracle counterpart of the cross-validated risk defined in (12). For a score f, we define

$$\tilde{R}^n_\phi(f) = \frac{1}{V}\sum_{v=1}^{V} R_\phi(f(P_n^{(-v)}),$$

that yields to the oracle counterpart of the cross-validated score defined in (13)

$$\tilde{f} \in \underset{f \in \mathrm{conv}(\mathcal{F})}{\mathrm{argmin}} \ \tilde{R}^n_\phi(f).$$

Here again, we assume that the loss function ϕ satisfies the properties described in Section 3.4 so that we can state the following result:

Proposition 14 *We assume that for each $f \in \mathrm{conv}(\mathcal{F})$ and $v \in \{1, \ldots, V\}$ we have $\|f(P_n^{(-v)})\|_\infty \leq B$. Then the following holds*

$$\mathbf{E}\left[\tilde{R}^n_\phi(\hat{f}) - \tilde{R}^n_\phi(\tilde{f})\right] \leq C(B, L)\frac{KM\log(n)}{\lfloor n/V \rfloor},$$

where $C(B, L) > 0$ is a constant that depends only on B and L.

As usual when one deals with cross-validated estimators, the theorem compares $\hat{f}$ to the oracle counterpart $\tilde{f}$ in terms of the oracle cross-validated ϕ-risk. The theorem teaches us that, for sufficiently large n, $\hat{f}$ perform as well as $\tilde{f}$. However our main goal remains confidence sets. Therefore the next step consists in showing that the confidence set associated to the cross-validated score $\hat{f}$ has good properties in terms of the classification risk. Let $\Gamma_{f,\delta}$ be the confidence set that results from a choice of $\beta \in (0, K)$ and a score function $f \in \mathrm{conv}(\mathcal{F})$. We introduce the following excess risks

$$\begin{aligned}
\Delta\tilde{\mathcal{R}}^n(\Gamma_{f,\delta}) &= \frac{1}{V}\sum_{v=1}^{V} \mathcal{R}\left(\Gamma_{f\left(P_n^{(-v)}\right),\delta}\right) - \mathcal{R}^*, \\
\Delta\tilde{R}^n_\phi(f) &= \frac{1}{V}\sum_{v=1}^{V} R_\phi(f(P_n^{(-v)})) - R_\phi(f^*).
\end{aligned}$$

These quantities can be view as cross-validated counterparts of the excess risks introduced in Section 3.2. Hereafter, we provide a result which can be view as the cross-validation counterpart of Theorem 13. It is interpreted in a same way.

`Forest` ($K = 4$)

		β-set				
β		`rforest`	`softmax reg`	`svm`	`kknn`	`CV`
2	**R**	0.02 (0.02)	0.06 (0.02)	0.02 (0.01)	0.05 (0.03)	0.02 (0.01)
	I	2.00 (0.09)	2.00 (0.08)	2.00 (0.09)	2.00 (0.08)	2.00 (0.08)

`Plant` ($K = 100$)

		β-set				
β		`rforest`	`softmax reg`	`svm`	`kknn`	`CV`
2	**R**	0.18 (0.03)	0.77 (0.02)	0.32 (0.04)	0.20 (0.03)	0.17 (0.03)
	I	2.00 (0.09)	2.02 (0.18)	1.99 (0.10)	2.00 (0.08)	2.00 (0.08)
10	**R**	0.02 (0.01)	0.42 (0.04)	0.03 (0.02)	0.08 (0.03)	0.02 (0.01)
	I	9.95 (0.38)	10.06 (0.58)	9.98 (0.22)	9.98 (0.23)	9.96 (0.37)

Table 1: For each of the $B = 100$ repetitions and for each dataset, we derive the estimated risks **R** and information **I** of the different β-sets w.r.t. β. We compute the means and standard deviations (between parentheses) over the $B = 100$ repetitions. For each β, the β-sets are based on, from left to right, `rforest`, `softmax reg` and `svm`, `kknn` and `CV` which are respectively the random forest, the softmax regression, support vector machines, k nearest neighbors and the superlearning procedure. Top: the dataset is the `Forest` – the dataset is the `Plant`.

Proposition 15 *We assume that for each $v \in \{1, \ldots, V\}$, $t \mapsto \mathbb{P}_X\left(\hat{f}\left(P_n^{(-v)}\right) \leq t | \mathcal{D}_n\right)$ is continuous. Under the assumptions of Proposition 14 and if the margin assumptions $\mathrm{M}^{\mathrm{k}}_\alpha$ are satisfied with $\alpha > 0$,*

$$\mathbf{E}\left[|\Delta\tilde{\mathcal{R}}^n(\hat{\Gamma}^{\mathrm{CV}}_\beta)|\right] \leq C(B, L, \alpha, s)K^{1-\alpha/(\alpha+s)}\left\{\mathbf{E}\left[\Delta\tilde{R}^n_\phi(\tilde{f})\right] + \frac{KM\log(n)}{\lfloor n/V \rfloor}\right\}^{\alpha/(\alpha+s)} + C'\frac{K}{\sqrt{N}}.$$

The proof of Proposition 15 relies on Proposition 14 and similar arguments as in Theorem 13.

4.3 Application to real datasets

In this section, we provide an application of our aggregation procedure described in Section 4.1. For the numerical experiment we focus on the boosting loss and consider the library of algorithms constituted by the random forest, the softmax regression, the support vector machines and the k nearest neighbors (with $k = 11$) procedures. To be more specifics, we respectively exploit the `R` packages `randomForest`, `polspline`, `e1071` and `kknn`. All the `R` functions are used with standard tuning parameters. Finally the parameter V of the aggregation procedure is fixed to 5.

We evaluate the performance of the procedure on two real datasets: the *Forest type mapping* dataset and the *one-hundred plant species leaves* dataset coming from the UCI database. In the sequel we refer to these two datasets as `Forest` and `Plant` respectively. The `Forest` dataset consists of $K = 4$ classes and 523 labeled observations (we gather the train and test sets) with 27 features. Here the classes are unbalanced. In the `Plant` dataset, there are $K = 100$ classes and 1600 labeled observations. This dataset is balanced

so that each class consists of 16 observations. The original dataset contains 3 covariates (each covariate consists of 64 features). In order to make the problem more challenging, we drop 2 covariates.

To get an indication of the statistical significance, it makes sense to compare our aggregated confidence set (referred as `CV`) to the confidence sets that result from each component of the library. Roughly speaking, we evaluate risks (and informations) of these confidence sets on each dataset. To do so, we use the *cross validation* principle. In particular, we run $B = 100$ times the procedure where we split the data each time in three: a sample of size n to build the scores $\hat{f}$; a sample of size N to estimate the function G and to get the confidence sets; and a sample of size M to evaluate the risk and the information. For both datasets, we make sure that in the sample of size n, there is the same number of observations in each class. As a benchmark, we notify that the misclassification risks of the best classifier from the library in the `Forest` dataset is evaluated at 0.15 , whereas in the `Plant` dataset, it is evaluated at 0.40. As planned, the performance of the classical methods are quite bad in this last dataset.

We set the sizes of the samples as $n = 200$, $N = 100$ and $M = 223$ for the `Forest` dataset, and $n = 1000$, $N = 200$ and $M = 400$ for the `Plant` one. The results are reported in Table 1, and confirm our expectations. In particular, our main observation is that the aggregated confidence set (`CV`) outperforms all components of the library in the sense that it is at least as good as the best component in all of the experiments. Second, let us state some remarks that hold for all of the confidence sets and in particular our aggregated confidence set. First, we note that the information $\mathbf{I}(\Gamma)$ has the good level β which is supported by our theory. Moreover, we see that the risk gets drastically better with moderate β as compared to the *best* misclassification risk. For instance, for the `Plant`, the error rate of the confidence set with $\beta = 2$ based on random forests is 0.18 whereas the misclassification error rate of the best component is 0.40.

5. Conclusion

In multiclass classification setting, the present paper propose a new procedure that assigns a set of labels instead of a single label to each instance. This set has a controlled expected size (or information) and its construction relies on cumulative distribution functions and on an empirical risk minimization procedure. Theoretical guarantees, especially rates of convergence are also provided and rely on the regularity of these cumulative distribution functions. The obtained rates of convergence highlight a linear dependence w.r.t the number of classes K. The procedure described in Section 3 is defined as a two steps algorithm whose second step consists in the estimation of the function G (which is a sum of tail distribution functions). Interestingly, this step does not require a set of labeled data and neither to explore the whole classes, that is suitable for semi-supervised learning. Moreover, we apply our methodology to derive an aggregation algorithm which is based on the V-fold cross-validation principle. Future works will focus on the optimality in the minimax sense with respect to the classification error. In particular, we will investigate whether the rates of convergence are optimal in terms of their dependence on K. We believe that this dependence (linearity on K) is the correct one. However we will instigate whether this dependence can be reduced under some sparsity assumption.

This section gathers the proofs of our results. Let us first add a notation that will be used throughout the Appendix: for any random variable (or vector) Z, we denote by $\mathbb{P}_Z$ the probability w.r.t. Z and by $\mathbb{E}_Z$, the corresponding expectation.

Appendix A. Technical Lemmas

We first lay out key lemmata, which are crucial to establish the main theory. We consider $K \geq 2$ be an integer, and $Z_1, \ldots, Z_K$, K random variables. Moreover we define function H by:

$$H(t) = \frac{1}{K} \sum_{k=1}^{K} F_k(t), \quad \forall t \in [0,1],$$

where for all $k = 1, \ldots, K$, F_k is the cumulative distribution function of Z_k. Finally, let us define the generalized inverse H^{-1} of H:

$$H^{-1}(p) = \inf\{t : H(t) \geq p\}, \quad \forall p \in (0,1).$$

Lemma 1 *Let ε distributed from a uniform distribution on $\{1, \ldots, K\}$ and independent of Z_k, $k = 1, \ldots, K$. Let U distributed from a uniform distribution on $[0,1]$. We consider*

$$Z = \sum_{k=1}^{K} Z_k \mathbf{1}_{\{\varepsilon = k\}}.$$

If H is continuous then

$$H(Z) \overset{\mathcal{L}}{=} U \quad \text{and} \quad H^{-1}(U) \overset{\mathcal{L}}{=} Z$$

Proof First we note that for every $t \in [0,1]$, $\mathbb{P}\left(H(Z) \leq t\right) = \mathbb{P}\left(Z \leq H^{-1}(t)\right)$. Moreover, we have

$$\begin{aligned}
\mathbb{P}\left(H(Z) \leq t\right) &= \sum_{k=1}^{K} \mathbb{P}(Z \leq H^{-1}(t), \varepsilon = k) \\
&= \frac{1}{K} \sum_{k=1}^{K} \mathbb{P}(Z_k \leq H^{-1}(t)) \text{ with } \varepsilon \text{ independent of } X \\
&= H(H^{-1}(t)) \\
&= t \text{ with } H \text{ continuous.}
\end{aligned}$$

To conclude the proof, we observe that

$$\begin{aligned}
\mathbb{P}\left(H^{-1}(U) \leq t\right) &= \mathbb{P}\left(U \leq H(t)\right) \\
&= \frac{1}{K} \sum_{k=1}^{K} F_k(t) \\
&= \sum_{k=1}^{K} \mathbb{P}(Z_k \leq t, \varepsilon = k) \\
&= \mathbb{P}(Z \leq t).
\end{aligned}$$

■

Lemma 2 *There exists an absolute constant $C' > 0$ such that*

$$\sum_{k=1}^{K} \mathbf{P}\left(|\hat{G}(\hat{f}_k(X) - \widetilde{G}(\hat{f}_k(X))| \geq |\widetilde{G}(\hat{f}_k(X)) - \beta|\right) \leq \frac{C'K}{\sqrt{N}}.$$

Proof We define, for $\gamma > 0$ and $k \in \{1, \dots K\}$

$$\begin{aligned} A_0^k &= \left\{\left|\widetilde{G}(\hat{f}_k(X)) - \beta\right| \leq \gamma\right\} \\ A_j^k &= \left\{2^{j-1}\gamma < |\widetilde{G}(\hat{f}_k(X)) - \beta| \leq 2^j\gamma\right\}, \ \ j \geq 1. \end{aligned}$$

Since, for every k, the events $(A_j^k)_{j\geq 0}$ are mutually exclusive, we deduce

$$\sum_{k=1}^{K} \mathbf{P}\left(|\hat{G}(\hat{f}_k(X) - \widetilde{G}(\hat{f}_k(X))| \geq |\widetilde{G}(\hat{f}_k(X)) - \beta|\right) = \sum_{k=1}^{K}\sum_{j\geq 0} \mathbf{P}\left(|\hat{G}(\hat{f}_k(X) - \widetilde{G}(\hat{f}_k(X))| \geq |\widetilde{G}(\hat{f}_k(X)) - \beta|, A_j^k\right). \quad (14)$$

Now, we consider a random variable ε uniformly distributed on $\{1, \dots, K\}$ and independent of $\mathcal{D}_n$ and X. Conditional on $\mathcal{D}_n$ and under Assumption $(A2)$, we apply Lemma 1 with $Z_k = \hat{f}_k(X)$, $Z = \sum_{k=1}^K Z_k \mathbf{1}_{\{\varepsilon=k\}}$ and then obtain that $\widetilde{G}(Z)$ is uniformly distributed on $[0, K]$. Therefore, for all $j \geq 0$ and $\gamma > 0$, we deduce

$$\begin{aligned} \frac{1}{K}\sum_{k=1}^{K} \mathbb{P}_X\left(|\widetilde{G}(\hat{f}_k(X)) - \beta| \leq 2^j\gamma|\mathcal{D}_n\right) &= \mathbb{P}_X\left(|\widetilde{G}(Z) - \beta| \leq 2^j\gamma|\mathcal{D}_n\right) \\ &\leq \frac{2^{j+1}\gamma}{K}. \end{aligned}$$

Hence, for all $j \geq 0$, we obtain

$$\sum_{k=1}^{K} \mathbf{P}(A_j^k) \leq 2^{j+1}\gamma. \quad (15)$$

Next, we observe that for all $j \geq 1$

$$\sum_{k=1}^{K} \mathbf{P}\left(|\hat{G}(\hat{f}_k(X) - \widetilde{G}(\hat{f}_k(X))| \geq |\widetilde{G}(\hat{f}_k(X)) - \beta|, A_j^k\right) \leq \sum_{k=1}^{K} \mathbb{E}_{(\mathcal{D}_n,X)}\left[\mathbb{P}_{\mathcal{D}_N}\left(|\hat{G}(\hat{f}_k(X)) - \widetilde{G}(\hat{f}_k(X))| \geq 2^{j-1}\gamma|\mathcal{D}_n, X\right)\mathbf{1}_{A_j^k}\right]. \quad (16)$$

Now, since conditional on $(\mathcal{D}_n, X)$, $\hat{G}(\hat{f}_k(X))$ is an empirical mean of i.i.d random variables of common mean $\widetilde{G}(\hat{f}_k(X)) \in [0, K]$, we deduce from Hoeffding's inequality that

$$\mathbb{P}_{\mathcal{D}_N}\left(|\hat{G}(\hat{f}_k(X)) - \widetilde{G}(\hat{f}_k(X))| \geq 2^{j-1}\gamma|\mathcal{D}_n, X\right) \leq 2\exp\left(-\frac{N\gamma^2 2^{2j-1}}{K^2}\right).$$

Therefore, from Inequalities (14), (15) and (16), we get

$$\begin{aligned} &\sum_{k=1}^{K} \mathbf{P}\left(|\hat{G}(\hat{f}_k(X) - \widetilde{G}(\hat{f}_k(X))| \geq |\widetilde{G}(\hat{f}_k(X)) - \beta|\right) \\ &\qquad \leq \sum_{k=1}^{K} \mathbf{P}\left(A_0^k\right) + \sum_{j\geq 1} 2\exp\left(-\frac{N\gamma^2 2^{2j-1}}{K^2}\right)\left(\sum_{k=1}^{K}\mathbf{P}\left(A_j^k\right)\right) \\ &\qquad \leq 2\gamma + \gamma\sum_{j\geq 1} 2^{j+2}\exp\left(-\frac{N\gamma^2 2^{2j-1}}{K^2}\right). \end{aligned}$$

Finally, choosing $\gamma = \dfrac{K}{\sqrt{N}}$ in the above inequality, we finish the proof of the lemma. ■

Appendix B. Proof of Proposition 4

Let $\beta > 0$ and Γ be a confidence set such that $\mathcal{I}(\Gamma) = \beta$. First, we note that the following decomposition holds

$$\begin{aligned} \mathcal{R}(\Gamma) - \mathcal{R}(\Gamma_\beta^*) &= \sum_{j=1}^{K}\sum_{k=1}^{K}\mathbb{E}\left[\sum_{l=1}^{K}(\mathbf{1}_{\{Y\notin\Gamma(X)\}} - \mathbf{1}_{\{Y\notin\Gamma_\beta^*(X)\}})\mathbf{1}_{\{Y=l\}}\mathbf{1}_{\{|\Gamma(X)|=k\}}\mathbf{1}_{\{|\Gamma_\beta^*(X)|=j\}}\right] \\ &= \sum_{j=1}^{K}\sum_{k=1}^{K}\mathbb{E}\left[\sum_{l=1}^{K}(\mathbf{1}_{\{l\notin\Gamma(X)\}} - \mathbf{1}_{\{l\notin\Gamma_\beta^*(X)\}})p_l(X)\mathbf{1}_{\{|\Gamma(X)|=k\}}\mathbf{1}_{\{|\Gamma_\beta^*(X)|=j\}}\right] \\ &= \sum_{j=1}^{K}\sum_{k=1}^{K}\mathbb{E}\left[\sum_{l=1}^{K}(\mathbf{1}_{\{l\in\Gamma_\beta^*(X)\setminus\Gamma(X)\}} - \mathbf{1}_{\{l\in\Gamma(X)\setminus\Gamma_\beta^*(X)\}})p_l(X)\mathbf{1}_{\{|\Gamma(X)|=k,|\Gamma_\beta^*(X)|=j\}}\right], \end{aligned}$$

where we conditioned by X to get the second equality. From the last decomposition and with

$$\begin{aligned} \mathbb{E}\left[|\Gamma(X)|\right] &= \mathbb{E}\left[|\Gamma_\beta^*(X)|\right] \\ &= \mathbb{E}\left[\sum_{j=1}^{K}\sum_{k=1}^{K} k\mathbf{1}_{\{|\Gamma(X)|=k\}}\mathbf{1}_{\{|\Gamma_\beta^*(X)|=j\}}\right] \\ &= \mathbb{E}\left[\sum_{j=1}^{K}\sum_{k=1}^{K} j\mathbf{1}_{\{|\Gamma(X)|=k\}}\mathbf{1}_{\{|\Gamma_\beta^*(X)|=j\}}\right], \end{aligned}$$

we can express the excess risk as the sum of two terms:

$$\mathcal{R}(\Gamma)-\mathcal{R}(\Gamma^*_\beta) = \sum_{j=1}^K\sum_{k=1}^K \mathbb{E}\left[\left(\sum_{l=1}^K \mathbf{1}_{\{l\in\Gamma^*_\beta(X)\setminus\Gamma(X)\}} p_l(X) - jG^{-1}(\beta)\right)\mathbf{1}_{\{|\Gamma(X)|=k\}}\mathbf{1}_{\{|\Gamma^*_\beta(X)|=j\}}\right] + \sum_{j=1}^K\sum_{k=1}^K \mathbb{E}\left[\left(kG^{-1}(\beta) - \sum_{l=1}^K \mathbf{1}_{\{l\in\Gamma(X)\setminus\Gamma^*_\beta(X)\}} p_l(X)\right)\mathbf{1}_{\{|\Gamma(X)|=k\}}\mathbf{1}_{\{|\Gamma^*_\beta(X)|=j\}}\right]. \quad (17)$$

Now, for $j,k\in\{1,\dots,K\}$ on the event $\{|\Gamma(X)|=k, |\Gamma^*_\beta(X)|=j\}$, we have

$$k = \sum_{l=1}^K \mathbf{1}_{\{l\in\Gamma(X)\setminus\Gamma^*_\beta(X)\}} + \sum_{l=1}^K \mathbf{1}_{\{l\in\Gamma(X)\cap\Gamma^*_\beta(X)\}},$$

and

$$j = \sum_{l=1}^K \mathbf{1}_{\{l\in\Gamma^*_\beta(X)\setminus\Gamma(X)\}} + \sum_{l=1}^K \mathbf{1}_{\{l\in\Gamma(X)\cap\Gamma^*_\beta(X)\}}.$$

Therefore, since

$$l\in\Gamma^*_\beta \Leftrightarrow p_l(X)\geq G^{-1}(\beta),$$

Equality (17) yields the result.

Appendix C. Proof of Theorem 10

First we recall that $\hat{f}_n = (\hat{f}_{n,1},\dots,\hat{f}_{n,K})$ is a sequence of score functions and $\delta_n\in\mathbb{R}$ is such that $G_{\hat{f}_n}(-\delta_n)=\beta$. We suppress the dependence on n to simplify notation and write $\hat{f}=(\hat{f}_1,\dots,\hat{f}_K)$ and δ for $\hat{f}$ and δ_n respectively. Moreover, since there is no doubt, we also suppress everywhere the dependence on X. We also define the events

$$B_k = \{\hat{f}_k\in(-\delta,-\delta^*) \text{ or } \hat{f}_k\in(-\delta^*,-\delta)\}, \quad (18)$$

for $k=1,\dots,K$. We aim at controlling the excess risk $\Delta\mathcal{R}(\Gamma_{\hat{f},\delta})$. Since the risk $\mathcal{R}$ is decomposable, it is convenient to introduce "marginal excess risks":

$$\Delta\mathcal{R}^k(\Gamma_{f,\delta}) = \mathbf{1}_{\{k\in(\Gamma_{\hat{f},\delta}\,\Delta\,\Gamma^*_\beta)\}}|p_k - G^{-1}(\beta)|.$$

Recall also that by convexity of the loss function ϕ, we have that for all $x,y\in\mathbb{R}$,

$$\phi(y)-\phi(x)\geq\phi'(x)(y-x). \quad (19)$$

Assume that $\hat{f}_k\leq-\delta$ and $\hat{f}_k\leq-\delta^*\leq f^*_k$, which translates as $p_k - G^{-1}(\beta)\geq 0$, we get thanks to (19)

$$p_k(\phi(\hat{f}_k)-\phi(-\delta)) - (1-p_k)(\phi(-\hat{f}_k)-\phi(\delta)) \geq (\phi'(\delta^*)-\phi'(-\delta^*))(p_k-G^{-1}(\beta))(\hat{f}_k+\delta^*)\geq 0.$$

Similarly, if $\hat{f}_k\geq-\delta$ and $\hat{f}_k\geq-\delta^*\geq f^*_k$, that is, $p_k - G^{-1}(\beta)\leq 0$ we have

$$p_k(\phi(\hat{f}_k)-\phi(-\delta)) - (1-p_k)(\phi(-\hat{f}_k)-\phi(\delta))\geq 0.$$

Note that in the following two cases

- $\hat{f}_k\leq -\delta$ and $f^*_k\leq-\delta^*$;
- $\hat{f}_k\geq -\delta$ and $f^*_k\geq-\delta^*$,

we have $\Delta\mathcal{R}^k(\Gamma_{\hat{f},\delta})=0$. Therefore, from the above inequalities, on B^c_k and by assumption (6) we get

$$\mathbf{1}_{B^c_k}\mathbf{1}_{\{k\in(\Gamma_{\hat{f},\delta}\,\Delta\,\Gamma^*_\beta)\}}|p_k - G^{-1}(\beta)|^s \leq C(\Delta R^k_\phi(\hat{f})). \quad (20)$$

Therefore, since $s\geq 1$, we have

$$\left(\mathbb{E}\left[\sum_{k=1}^K \mathbf{1}_{B^c_k\cap\{k\in(\Gamma_{\hat{f},\delta}\,\Delta\,\Gamma^*_\beta)\}}|p_k-G^{-1}(\beta)|\right]\right)^s \leq \frac{1}{K}\sum_{k=1}^K \mathbb{E}\left[\mathbf{1}_{B^c_k\cap\{k\in\Gamma_{\hat{f},\delta}\Delta\Gamma^*_\beta\}}K^s|p_k-G^{-1}(\beta)|^s\right] \leq CK^{s-1}\Delta R_\phi(\hat{f}).$$

Moreover

$$\mathbb{E}\left[\sum_{k=1}^K \mathbf{1}_{B_k}\mathbf{1}_{\{k\in(\Gamma_{\hat{f},\delta}\,\Delta\,\Gamma^*_\beta)\}}|p_k-G^{-1}(\beta)|\right] \leq \sum_{k=1}^K\mathbb{P}(B_k) \leq \mathbb{E}\left[|G_{\hat{f}}(-\delta)-G_{\hat{f}}(-\delta^*)|\right] \leq \mathbb{E}\left[|G_{f^*}(-\delta^*)-G_{\hat{f}}(-\delta^*)|\right].$$

Finally, we get the following bound

$$\Delta\mathcal{R}(\Gamma_{\hat{f},\delta}) \leq K^{\frac{s-1}{s}}\Delta R_\phi(\hat{f})^{1/s} + \mathbb{E}\left[|G_{f^*}(-\delta^*)-G_{\hat{f}}(-\delta^*)|\right].$$

Now we observe that

$$|\mathbf{1}_{\{\hat{f}_k\geq-\delta^*\}} - \mathbf{1}_{\{f^*_k\geq-\delta^*\}}| \leq \mathbf{1}_{\{|p_k-G^{-1}(\beta)|^s\leq C\Delta R^k_\phi(\hat{f})\}}. \quad (21)$$

Therefore, for each $\gamma>0$, we have

$$\mathbb{E}_X\left[|G_{f^*}(-\delta^*)-G_{\hat{f}}(-\delta^*)|\right] \leq \sum_{k=1}^K \mathbb{P}_X\left(|p_k(X)-G^{-1}(\beta)|^s\leq C\Delta R^k_\phi(\hat{f})\right) \leq \sum_{k=1}^K \mathbb{P}_X\left(|p_k(X)-G^{-1}(\beta)|\leq\gamma^{1/s}\right) + \mathbb{P}_X\left(\gamma\leq C\Delta R^k_\phi(\hat{f})\right).$$

Now using Markov Inequality, we have that

$$\sum_{k=1}^K \mathbb{P}_X\left(\gamma\leq C\Delta R^k_\phi(\hat{f})\right) \leq \frac{C}{\gamma}\sum_{k=1}^K\mathbb{E}_X\left[\Delta R^k_\phi(\hat{f})\right] \leq \frac{C\Delta R_\phi(\hat{f})}{\gamma}.$$

The above inequality yields

$$\mathbb{E}_X\left[|G_{f^*}(-\delta^*) - G_{\hat{f}}(-\delta^*)|\right] \leq \frac{C\Delta R_\phi(\hat{f})}{\gamma} + \sum_{k=1}^{K} \mathbb{P}_X\left(|p_k(X) - G^{-1}(\beta)| \leq \gamma^{1/s}\right). \quad (22)$$

Hence, with Equation (21), we get

$$\Delta\mathcal{R}(\Gamma_{\hat{f},\delta}) \leq K^{\frac{s-1}{s}} \Delta R_\phi(\hat{f})^{1/s} + \frac{C\Delta R_\phi(\hat{f})}{\gamma} + \sum_{k=1}^{K} \mathbb{P}_X\left(|p_k(X) - G^{-1}(\beta)| \leq \gamma^{1/s}\right).$$

The term $\sum_{k=1}^{K} \mathbb{P}_X\left(|p_k(X) - G^{-1}(\beta)| \leq \gamma^{1/s}\right) \to 0$ when $\gamma \to 0$, given that the distribution function of the $p_k's$ are continuous. Then using the convergence in distribution of $\Delta R_\phi(\hat{f})$ to zero, the last inequality ensures the desired result.

Appendix D. Proof of Proposition 12

For any $\beta \in (0, K)$, and conditional on $\mathcal{D}_n$ we define

$$\widetilde{G}^{-1}(\beta) = \inf\{t \in \mathbb{R} : \ \widetilde{G}(t) \leq \beta\}. \quad (23)$$

We note that Assumption $(A2)$ ensures that $t \mapsto \widetilde{G}(t)$ is continuous and then

$$\widetilde{G}(\widetilde{G}^{-1}(\beta)) = \beta.$$

Now, we have

$$\begin{aligned} |\widetilde{\Gamma}_\beta(X)| &= \sum_{k=1}^{K} \mathbf{1}_{\{\widetilde{G}(\hat{f}_k(X)) \leq \beta\}} \\ &= \sum_{k=1}^{K} \mathbf{1}_{\{\hat{f}_k(X) \geq \widetilde{G}^{-1}(\beta)\}}. \end{aligned}$$

Hence, the last equation implies that

$$\mathbb{E}_X\left[|\widetilde{\Gamma}_\beta(X)||\mathcal{D}_n\right] = \sum_{k=1}^{K} \mathbb{P}_X\left(\hat{f}_k(X) \geq \widetilde{G}^{-1}(\beta)|\mathcal{D}_n\right) = \widetilde{G}(\widetilde{G}^{-1}(\beta)) = \beta. \quad (24)$$

Therefore, we obtain $\mathbf{E}\left[|\widetilde{\Gamma}_\beta(X)|\right] = \beta$. Also, we can write

$$\begin{aligned} \left|\mathbf{E}\left[\left|\hat{\Gamma}_\beta(X)\right|\right] - \beta\right| &\leq \left|\mathbf{E}\left[|\hat{\Gamma}_\beta(X)| - |\widetilde{\Gamma}_\beta(X)|\right]\right| \\ &\leq \left|\mathbf{E}\left[\sum_{k=1}^{K}\left(\mathbf{1}_{\{\hat{G}(\hat{f}_k(X)) \leq \beta\}} - \mathbf{1}_{\{\widetilde{G}(\hat{f}_k(X)) \leq \beta\}}\right)\right]\right| \\ &\leq \mathbf{E}\left[|\hat{\Gamma}_\beta(X) \,\Delta\, \widetilde{\Gamma}_\beta(X)|\right] \\ &\leq \sum_{k=1}^{K} \mathbf{E}\left[|\mathbf{1}_{\{\hat{G}(\hat{f}_k(X)) \leq \beta\}} - \mathbf{1}_{\{\widetilde{G}(\hat{f}_k(X)) \leq \beta\}}|\right] \\ &\leq \sum_{k=1}^{K} \mathbf{P}\left(|\hat{G}(\hat{f}_k(X) - \widetilde{G}(\hat{f}_k(X))| \geq |\widetilde{G}(\hat{f}_k(X)) - \beta|\right). \end{aligned}$$

Hence, applying Lemma 2 in the above inequality, we obtain the desired result.

Appendix E. Proof of Theorem 13

When there is no doubt, we suppress the dependence on X. First, let us state a intermediate result that is also needed to prove the theorem.

Lemma 3 *Consider $\Gamma_{\hat{f},\delta}$ the confidence set based on the score function $\hat{f}$ with information β (that is, $G_{\hat{f}}(-\delta) = \beta$). Under assumptions M_α^k, the following holds*

$$\Delta\mathcal{R}(\Gamma_{\hat{f},\delta}) \leq C(\alpha, s)\left\{K^{1-1/(s+\lambda-\lambda s)}\Delta R_\phi(\hat{f})^{1/(s+\lambda-\lambda s)} + K^{1-\lambda/(s+\lambda-\lambda s)}\Delta R_\phi(\hat{f})^{\lambda/(s+\lambda-\lambda s)}\right\},$$

where $\lambda = \frac{\alpha}{\alpha+1}$ and $C(\alpha, s)$ is non negative constant which depends only on α and s.

Proof For each $k = 1, \ldots, K$, we define the following events $S_k = B_k^c \cap \{k \in (\Gamma_{\hat{f},\delta} \,\Delta\, \Gamma_\beta^*)\}$ and $T_k = B_k \cap \{k \in (\Gamma_{\hat{f},\delta} \,\Delta\, \Gamma_\beta^*)\}$, where the B_k's are the events given in Eq. (18). Now we observe that

$$\Delta\mathcal{R}(\Gamma_{\hat{f},\delta}) = \mathbb{E}\left[\sum_{k=1}^{K} \mathbf{1}_{S_k}|p_k - G^{-1}(\beta)|\right] + \mathbb{E}\left[\sum_{k=1}^{K} \mathbf{1}_{T_k}|p_k - G^{-1}(\beta)|\right] = \Delta\mathcal{R}_1(\Gamma_{\hat{f},\delta}) + \Delta\mathcal{R}_2(\Gamma_{\hat{f},\delta}),$$

where

$$\begin{aligned} \Delta\mathcal{R}_1(\Gamma_{\hat{f},\delta}) &= \mathbb{E}\left[\sum_{k=1}^{K} \mathbf{1}_{S_k}|p_k - G^{-1}(\beta)|\right], \\ \Delta\mathcal{R}_2(\Gamma_{\hat{f},\delta}) &= \mathbb{E}\left[\sum_{k=1}^{K} \mathbf{1}_{T_k}|p_k - G^{-1}(\beta)|\right]. \end{aligned}$$

The end of the proof consists in controlling each of these two terms. Let us first consider $\Delta\mathcal{R}_1(\Gamma_{\hat{f},\delta})$. For $\varepsilon > 0$, we have

$$\begin{aligned} \Delta\mathcal{R}_1(\Gamma_{\hat{f},\delta}) &= \mathbb{E}\left[\sum_{k=1}^{K} \mathbf{1}_{S_k}|p_k - G^{-1}(\beta)|\left(\mathbf{1}_{\{|p_k - G^{-1}(\beta)| \geq \varepsilon\}} + \mathbf{1}_{\{|p_k - G^{-1}(\beta)| \leq \varepsilon\}}\right)\right] \\ &\leq \mathbb{E}\left[\sum_{k=1}^{K} \mathbf{1}_{S_k}\varepsilon^{1-s}|p_k - G^{-1}(\beta)|^s\right] + \varepsilon\sum_{k=1}^{K}\mathbb{P}(S_k) \\ &\leq C\varepsilon^{1-s}\Delta R_\phi(\hat{f}) + \varepsilon\sum_{k=1}^{K}\mathbb{P}(S_k), \end{aligned} \quad (25)$$

where we used the assumption (6) and more precisely (20) to deduce the last inequality. To control $\sum_{k=1}^{K}\mathbb{P}(S_k)$, we require the following result that is a direct application of Lemma 5 in (Bartlett et al., 2006).

Lemma 4 *Under the assumptions M_α^k we have*

$$\sum_{k=1}^{K} \mathbb{P}(S_k) \leq C(\alpha) \left(K^{1/\alpha} \Delta\mathcal{R}_1(\Gamma_{\hat{f},\delta}) \right)^{\frac{\alpha}{\alpha+1}},$$

where $C(\alpha) > 0$ is a constant that depends only on α.

Proof The proof of this result relies on the following simple fact: for all $\varepsilon > 0$

$$\begin{aligned} \mathbb{E}\left[\mathbf{1}_{S_k} |p_k - G^{-1}(\beta)|\right] &\geq \varepsilon \mathbb{E}\left[\mathbf{1}_{S_k} \mathbf{1}_{\{|p_k - G^{-1}(\beta)| \geq \varepsilon\}}\right] \\ &\geq \varepsilon \left[\mathbb{P}(S_k) - c_2 \varepsilon^\alpha\right]. \end{aligned}$$

Choosing $\varepsilon = \left(\frac{1}{c_2 K(\alpha+1)} \sum_{k=1}^{K} \mathbb{P}(S_k) \right)^{1/\alpha}$ we get the lemma. ■

We go back to the proof of Lemma 3. Applying Lemma 4 to (25), we get

$$\Delta\mathcal{R}_1(\Gamma_{\hat{f},\delta}) \leq C\varepsilon^{1-s} \Delta R_\phi(\hat{f}) + \varepsilon C(\alpha) \left(K^{1/\alpha} \Delta\mathcal{R}_1(\Gamma_{\hat{f},\delta}) \right)^{\frac{\alpha}{\alpha+1}}.$$

Choosing $\varepsilon = \frac{s-1}{sC(\alpha)} K^{(\lambda-1)} \Delta\mathcal{R}_1(\Gamma_{\hat{f},\delta})^{(1-\lambda)}$, we obtain

$$\Delta\mathcal{R}_1(\Gamma_{\hat{f},\delta}) \leq C_1(\alpha, s) K^{1-1/(s-\lambda s+\lambda)} \Delta R_\phi(\hat{f})^{1/(s+\lambda-\lambda s)}, \tag{26}$$

for a non negative constant $C_1(\alpha, s)$ that depends only on α and s.

Let us now focus on the second term, $\Delta\mathcal{R}_2(\Gamma_{\hat{f},\delta})$. Since the assumptions of Theorem 10 are satisfied, we can use Equation (22) for any $\gamma > 0$. Combined with the Margin assumptions M_α^k, we obtain

$$\begin{aligned} \Delta\mathcal{R}_2(\Gamma_{\hat{f},\delta}) &\leq \mathbb{E}_X\left[|G_{f^*}(-\delta^*) - G_{\hat{f}}(-\delta^*)| \right] \\ &\leq \frac{C\Delta R_\phi(\hat{f})}{\gamma} + c_2 K \gamma^{\alpha/s}. \end{aligned}$$

Therefore, optimizing in γ, we have

$$\Delta\mathcal{R}_2(\Gamma_{\hat{f},\delta}) \leq C_2(\alpha, s) K^{1-\lambda/(s+\lambda-\lambda s)} \Delta R_\phi(\hat{f})^{\lambda/(s+\lambda-\lambda s)},$$

for a non negative constant $C_2(\alpha, s)$ that depends only on α and s. The result stated in the lemma is deduced by combining the last equation with Equation (26) and by setting $C(\alpha, s) = \max\{C_1(\alpha, s); C_2(\alpha, s)\}$. ■

We now state another important lemma that describes the behavior of the empirical minimizer of the ϕ-risk on the class $\mathcal{F}$:

Lemma 5 *Let $\bar{f} \in \mathcal{F}$ be the minimizer of $R_\phi(f)$ over $\mathcal{F}$. Under the assumptions of Theorem 13, we have that*

$$\mathbf{E}\left[R_\phi(\hat{f}_n) - R_\phi(\bar{f}) \right] \leq \frac{3KL}{n} + \frac{KC(B, L) \log(N_n)}{n},$$

where $C(B, L) > 0$ is a constant that only depends on the constant B given in the Proposition 14 and on the Lipschitz constant L.

Proof First, according to Eq. (11), for each $f \in \mathcal{F}$, we can write

$$\frac{R_\phi(f) + R_\phi(\bar{f})}{2} - R_\phi\left(\frac{f + \bar{f}}{2} \right) \geq \delta \left(\sqrt{\sum_{k=1}^{K} \mathbb{E}_X\left[(f_k - \bar{f}_k)^2(X) \right]} \right),$$

Hence, by assumption on the modulus of convexity, we deduce

$$\frac{R_\phi(f) + R_\phi(\bar{f})}{2} - R_\phi\left(\frac{f + \bar{f}}{2} \right) \geq c_1 \sum_{k=1}^{K} \mathbb{E}_X\left[(f_k - \bar{f}_k)^2(X) \right].$$

Since $R_\phi\left(\frac{f+\bar{f}}{2}\right) \geq R_\phi(\bar{f})$, we obtain

$$\sum_{k=1}^{K} \mathbb{E}_X\left[(f_k - \bar{f}_k)^2(X) \right] \leq \frac{1}{2c_1} R_\phi(f) - R_\phi(\bar{f}).$$

Now, denoting by $h(z, f(x)) = \sum_{k=1}^{K} \phi(z_k f_k(x)) - \phi(z_k \bar{f}_k(x))$, we get the following bound

$$\begin{aligned} \mathbb{E}_X\left[h^2(Zf(X)) \right] &\leq KL^2 \sum_{k=1}^{K} \mathbb{E}_X\left[(f_k - \bar{f}_k)^2(X) \right] \\ &\leq \frac{KL^2}{2c_1} \mathbb{E}_X\left[h(Zf(X)) \right], \end{aligned} \tag{27}$$

where L is the Lipschitz constant L for ϕ. On the other hand, we have the following decomposition

$$R_\phi(\hat{f}) - R_\phi(\bar{f}) = R_\phi(\hat{f}) + 2(\hat{R}_\phi(\hat{f}) - \hat{R}_\phi(\bar{f})) - 2(\hat{R}_\phi(\hat{f}) - \hat{R}_\phi(\bar{f})) - R_\phi(\bar{f}).$$

Also, since $\hat{R}_\phi(\hat{f}) - \hat{R}_\phi(\bar{f}) \leq 0$, we get

$$\begin{aligned} R_\phi(\hat{f}) - R_\phi(\bar{f}) &\leq (R_\phi(\hat{f}) - R_\phi(\bar{f})) - 2(\hat{R}_\phi(\hat{f}) - \hat{R}_\phi(\bar{f})) \\ &\leq \frac{3KL}{n} + \sup_{f \in \mathcal{F}_n} (R_\phi(f) - R_\phi(\bar{f})) - 2(\hat{R}_\phi(f) - \hat{R}_\phi(\bar{f})), \end{aligned}$$

where $\mathcal{F}_n$ is the ϵ-net of $\mathcal{F}$ w.r.t the L_∞-norm and with $\epsilon = 1/n$. Now, using Bernstein's Inequality, we have that for all $f \in \mathcal{F}_n$ and $t > 0$

$$\begin{aligned} &\mathbf{P}\left((R_\phi(f) - R_\phi(\bar{f})) - 2(\hat{R}_\phi(f) - \hat{R}_\phi(\bar{f})) \geq t \right) \leq \\ &\quad \mathbf{P}\left(2((R_\phi(f) - R_\phi(\bar{f})) - (\hat{R}_\phi(f) - \hat{R}_\phi(\bar{f})) \geq t + R_\phi(f) - R_\phi(\bar{f}) \right) \\ &\quad \leq \exp\left(- \frac{n(t + \mathbb{E}[h(Z, f(X))])^2/8}{\mathbb{E}[h^2(Z, f(X))]) + (2KLB/3)(t + \mathbb{E}[h(Z, f(X))]))} \right). \end{aligned}$$

Using Eq. (27), we get for all $f \in \mathcal{F}_n$

$$\mathbf{P}\left((R_\phi(f) - R_\phi(\bar{f})) - 2(\hat{R}_\phi(f) - \hat{R}_\phi(\bar{f})) \geq t \right) \leq \exp\left(- \frac{nt}{8(KL^2/(2c_1) + KLB/3)} \right),$$

Therefore, using a union bound argument, and then integrating we deduce that

$$\begin{aligned}\mathbf{E}\left[R_\phi(\hat{f}) - R_\phi(\bar{f})\right] &\leq \frac{3KL}{n} + \mathbf{E}\left[\sup_{f\in\mathcal{F}_n}(R_\phi(f) - R_\phi(\bar{f})) - 2(\hat{R}_\phi(f) - \hat{R}_\phi(\bar{f}))\right] \\ &\leq \frac{3KL}{n} + \frac{KC(B,L)\log(M_n)}{n}.\end{aligned}$$

■

We are now ready to conclude the proof of the theorem. We have the following inequality

$$|\mathcal{R}(\hat{\Gamma}_\beta) - \mathcal{R}^*_\beta| \leq \Delta\mathcal{R}(\widetilde{\Gamma}_\beta) + |\mathcal{R}(\hat{\Gamma}_\beta) - \mathcal{R}(\widetilde{\Gamma}_\beta)|. \tag{28}$$

We deal which each terms in the r.h.s separately. First, we have from Jensen's Inequality that

$$\left(\mathbf{E}\left[\Delta\mathcal{R}(\widetilde{\Gamma}_\beta)\right]\right)^{\frac{s+\lambda-\lambda s}{\lambda}} \leq \mathbf{E}\left[\Delta\mathcal{R}(\widetilde{\Gamma}_\beta)^{\frac{s+\lambda-\lambda s}{\lambda}}\right].$$

Hence, from Lemma 3, we deduce

$$\left(\mathbf{E}\left[\Delta\mathcal{R}(\widetilde{\Gamma}_\beta)\right]\right)^{\frac{s+\lambda-\lambda s}{\lambda}} \leq C(\alpha,s)^{\frac{s+\lambda-\lambda s}{\lambda}} K^{\frac{s+\lambda-\lambda s}{\lambda}-1}\mathbf{E}\left[\Delta R_\phi(\hat{f})\right].$$

Moreover, from Lemma 5, we have that

$$\mathbf{E}\left[\Delta R_\phi(\hat{f})\right] \leq \inf_{f\in\mathcal{F}}\Delta R_\phi(f) + \frac{3KL}{n} + \frac{KC(B,L)\log(M_n)}{n}.$$

Therefore, we can write

$$\mathbf{E}\left[\Delta\mathcal{R}(\widetilde{\Gamma}_\beta)\right] \leq C(\alpha,s)K^{1-\lambda/(s+\lambda-\lambda s)}\left\{\inf_{f\in\mathcal{F}}\Delta R_\phi(f) + \frac{3KL}{n} + \frac{KC(B,L)\log(N_n)}{n}\right\}^{\lambda/(s+\lambda-\lambda s)}. \tag{29}$$

For the second term $|\mathcal{R}(\hat{\Gamma}_\beta) - \mathcal{R}(\widetilde{\Gamma}_\beta)|$ in (28), we observe that

$$\mathbf{1}_{\{Y\notin\hat{\Gamma}_\beta(X)\}} - \mathbf{1}_{\{Y\notin\widetilde{\Gamma}_\beta(X)\}} = \sum_{k=1}^K \mathbf{1}_{\{Y=k\}}\mathbf{1}_{\left\{k\notin\hat{\Gamma}_\beta(X)\right\}} - \sum_{k=1}^K \mathbf{1}_{\{Y=k\}}\mathbf{1}_{\left\{k\notin\widetilde{\Gamma}_\beta(X)\right\}}.$$

Therefore, we can write

$$\begin{aligned}\mathbf{E}\left[\mathbf{1}_{Y\notin\hat{\Gamma}_\beta(X)\}} - \mathbf{1}_{\{Y\notin\widetilde{\Gamma}_\beta(X)\}}\right] &= \sum_{k=1}^K \mathbf{E}\left[p_k(X)\left(\mathbf{1}_{\left\{k\notin\hat{\Gamma}_\beta(X)\right\}} - \mathbf{1}_{\left\{k\notin\widetilde{\Gamma}_\beta(X)\right\}}\right)\right] \\ &= \sum_{k=1}^K \mathbf{E}\left[p_k(X)\left(\mathbf{1}_{\left\{\hat{G}(\hat{f}_k(X))>\beta\right\}} - \mathbf{1}_{\left\{\tilde{G}(\hat{f}_k(X))>\beta\right\}}\right)\right].\end{aligned}$$

Since $0 \leq p_k(X) \leq 1$ for all $k \in \{1,\ldots,K\}$, the last equality implies

$$\begin{aligned}\left|\mathbf{R}\left(\hat{\Gamma}_\beta\right) - \mathbf{R}\left(\tilde{\Gamma}_\beta\right)\right| &= \left|\mathbf{E}\left[\mathbf{1}_{Y\notin\hat{\Gamma}_\beta(X)\}} - \mathbf{1}_{\{Y\notin\widetilde{\Gamma}_\beta(X)\}}\right]\right| \\ &\leq \sum_{k=1}^K \mathbf{E}\left[\left|\mathbf{1}_{\left\{\hat{G}(\hat{f}_k(X))>\beta\right\}} - \mathbf{1}_{\left\{\tilde{G}(\hat{f}_k(X))>\beta\right\}}\right|\right] \\ &\leq \sum_{k=1}^K \mathbf{P}\left(|\hat{G}(\hat{f}_k(X) - \widetilde{G}(\hat{f}_k(X))| \geq |\widetilde{G}(\hat{f}_k(X)) - \beta|\right).\end{aligned}$$

Therefore, Lemma 2 implies

$$\left|\mathbf{R}\left(\hat{\Gamma}_\beta\right) - \mathbf{R}\left(\tilde{\Gamma}_\beta\right)\right| \leq \frac{C'K}{\sqrt{N}}. \tag{30}$$

Injecting Equations (29) and (30) to Equation (28) we conclude the proof of the theorem.

Appendix F. Proof of Proposition 14

We begin with the following decomposition

$$\tilde{R}^n_\phi(\hat{f}) - \tilde{R}^n_\phi(\tilde{f}) = \tilde{R}^n_\phi(\hat{f}) + 2(\hat{R}^n_\phi(\hat{f}) - \hat{R}^n_\phi(\tilde{f})) - 2(\hat{R}^n_\phi(\hat{f}) - \hat{R}^n_\phi(\tilde{f})) - \tilde{R}^n_\phi(\tilde{f}),$$

since $\hat{R}^n_\phi(\hat{f}) - \hat{R}^n_\phi(\tilde{f}) \leq 0$, we get

$$\tilde{R}^n_\phi(\hat{f}) - \tilde{R}^n_\phi(\tilde{f}) \leq (\tilde{R}^n_\phi(\hat{f}) - \tilde{R}^n_\phi(\tilde{f})) - 2(\hat{R}^n_\phi(\hat{f}) - \hat{R}^n_\phi(\tilde{f})). \tag{31}$$

Now, we denote by $\mathcal{C}_n = \{(i_1/n,\ldots,i_M/n),(i_1,\ldots,i_M)\in\{0,\ldots,n\}\}\cap\mathcal{C}_M$,
and $\mathcal{F}_n = \{f = \sum_{i=1}^M m_i f_i,\ \ m_1,\ldots m_M)\in\mathcal{C}_n\}$. For each $f\in\mathrm{conv}(\mathcal{F})$, there exists $f_n\in\mathcal{F}_n$ such that

$$\begin{aligned}|\tilde{R}^n_\phi(f) - \tilde{R}^n_\phi(f_n)| &\leq \frac{2KLBM}{n} \\ |\hat{R}^n_\phi(f) - \hat{R}^n_\phi(f_n)| &\leq \frac{2KLBM}{n}.\end{aligned}$$

Therefore, with Equation (31), we obtain

$$\tilde{R}^n_\phi(\hat{f}) - \tilde{R}^n_\phi(\tilde{f}) \leq \frac{6LKB}{n} + \sup_{f\in\mathcal{F}_n}(\tilde{R}^n_\phi(f) - \tilde{R}^n_\phi(\bar{f})) - 2(\hat{R}^n_\phi(f) - \hat{R}^n_\phi(\bar{f})).$$

Now, Similar arguments as in the paper by Dudoit and van der Laan (2005) and Lemma 5 yield the proposition.

References

P. Bartlett and S. Mendelson. Empirical minimization. *Probability Theory and Related Fields*, 135(3):311–334, 2006.

P. Bartlett, M. Jordan, and J. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

A. Choromanska, K. Choromanski, and M. Bojarski. On the boosting ability of top-down decision tree learning algorithm for multiclass classification. preprint, 2016.

C.K. Chow. On optimum error and reject trade-off. *IEEE Transactions on Information Theory*, 16:41–46, 1970.

J. del Coz, J. Díez, and A. Bahamonde. Learning nondeterministic classifiers. *Journal of Machine Learning Research*, 10:2273–2293, 2009.

C. Denis and M. Hebiri. Consistency of plug-in confidence sets for classification in semi-supervised learning. preprint, 2015.

S. Dudoit and M. van der Laan. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. *Statistical Methodology*, 2(2):131–154, 2005.

Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, 28(2):337–407, 2000.

R. Herbei and M. Wegkamp. Classification with reject option. *The Canadian Journal of Statistics*, 34(4):709–721, 2006.

J. Lei. Classification with confidence. *Biometrika*, 101(4):755–769, 2014.

J. Lei, J. Robins, and L. Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108(501):278–287, 2013.

A. Tewari and P. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.

A. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

M. van der Laan, E. Polley, and A. Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6, 2007.

A. van der Vaart. *Asymptotic statistics*, volume 3 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, Cambridge, 1998.

V. Vapnik. *Statistical learning theory*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons Inc., New York, 1998. A Wiley-Interscience Publication.

V. Vovk. Asymptotic optimality of transductive confidence machine. In *Algorithmic learning theory*, volume 2533 of *Lecture Notes in Computer Science*, pages 336–350. Springer, Berlin, 2002.

V. Vovk, A. Gammerman, and C. Saunders. Machine-learning applications of algorithmic randomness. In *International Conference on Machine Learning*, pages 444–453. 1999.

V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic learning in a random world*. Springer, New York, 2005.

M. Wegkamp and M. Yuan. Support vector machines with a reject option. *Bernoulli*, 17 (4):1368–1385, 2011.

T.-F. Wu, C.-J. Lin, and R. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.

M. Yuan and M. Wegkamp. Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, 11:111–130, 2010.

T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 02 2004.

Journal of Machine Learning Research 18 (2017) 1-50 Submitted 6/16; Revised 8/17; Published 10/17

Online Learning to Rank with Top-k Feedback

Sougata Chaudhuri [1] sougata@umich.edu
Ambuj Tewari [1,2] tewria@umich.edu
Department of Statistics [1]
Department of Electrical Engineering and Computer Science [2]
University of Michigan
Ann Arbor, MI 48109, USA

Editor: Alexander Rakhlin

Abstract

We consider two settings of online learning to rank where feedback is restricted to top ranked items. The problem is cast as an online game between a learner and sequence of users, over T rounds. In both settings, the learners objective is to present ranked list of items to the users. The learner's performance is judged on the entire ranked list and true relevances of the items. However, the learner receives highly restricted feedback at end of each round, in form of relevances of only the top k ranked items, where $k \ll m$. The first setting is *non-contextual*, where the list of items to be ranked is fixed. The second setting is *contextual*, where lists of items vary, in form of traditional query-document lists. No stochastic assumption is made on the generation process of relevances of items and contexts. We provide efficient ranking strategies for both the settings. The strategies achieve $O(T^{2/3})$ regret, where regret is based on popular ranking measures in first setting and ranking surrogates in second setting. We also provide impossibility results for certain ranking measures and a certain class of surrogates, when feedback is restricted to the top ranked item, i.e. $k = 1$. We empirically demonstrate the performance of our algorithms on simulated and real world data sets.

Keywords: Learning to Rank, Online Learning, Partial Monitoring, Online Bandits, Learning Theory

1. Introduction

Learning to rank (Liu, 2011) is a supervised machine learning problem, where the output space consists of *rankings* of objects. Most learning to rank methods are based on supervised *batch* learning, i.e., rankers are trained on batch data in an offline setting. The accuracy of a ranked list, in comparison to the actual relevance of the documents, is measured by various ranking measures, such as Discounted Cumulative Gain (DCG) (Järvelin and Kekäläinen, 2000), Average Precision (AP) (Baeza-Yates and Ribeiro-Neto, 1999) and others.

Collecting reliable training data can be expensive and time consuming. In certain applications, such as deploying a new web app or developing a custom search engine, collecting large amount of high quality labeled data might be infeasible (Sanderson, 2010). Moreover, a ranker trained from batch data might not be able to satisfy rapidly changing user needs and preferences. Thus, a promising direction of research is development of online ranking systems, where a ranker is updated on the fly. One type of online ranking models learns from implicit feedback inferred from user clicks on ranked lists (Hofmann et al., 2013). However, there are some potential drawbacks in learning from user clicks. It is possible that the system is designed for explicit ratings but not clicks. Moreover, a clicked item might not actually be relevant to the user and there is also the problem of bias towards top ranked items in inferring feedback from user clicks (Joachims, 2002).

We develop models for online learning of ranking systems, from explicit but *highly restricted* feedback. At a high level, we consider a ranking system which interacts with users over a time horizon, in a sequential manner. At each round, the system presents a ranked list of m items to the user, with the quality of the ranked list judged by the relevance of the items to the user. The relevance of the items, reflecting varying user preferences, is encoded as relevance vectors. The system's objective is to learn from the feedback it receives and update its ranker over time, to satisfy as many users as possible. However, the feedback that the system receives at end of each round is not the full relevance vector, but relevance of only the top k ranked items, where $k \ll m$ (typically, $k = 1$ or 2). We consider two problem settings under the general framework: *non-contextual* and *contextual*. In the first setting, we assume that the set of items to be ranked are fixed (i.e., there are no context on items), with the relevance vectors varying according to users' preferences. In the second setting, we assume that set of items vary, as traditional query-document lists. We highlight two motivating examples for such feedback model, encompassing *privacy* concerns and *economic and user-burden* constraints.

Privacy Concerns: Assume that a medical company wants to build an app to suggest activities (take a walk, meditate, watch relaxing videos, etc.) that can lead to reduction of stress in a certain highly stressed segment of the population. The activities do not have contextual representation and are fixed over time. Not all activities are likely to be equally suitable for everyone under all conditions since the effects of the activities vary depending on the user attributes like age & gender and on the context such as time of day & day of week. The user has liberty to browse through all the suggested activities, and the company would like the user to rate every activity (may be on an $1-5$ scale), reflecting the relevances, so that it can keep refining its ranking strategy. However, in practice, though the user may scan through all suggested activities and have a rough idea about how relevant each one is to her; she is unlikely to give feedback on the usefulness (relevance) of every activity due to privacy concerns and cognitive burden. Hence, in exchange of the user using the app, the company only asks for careful rating of the top 1 or 2 ranked activities. The apps performance would still be based on the full ranked list, compared to the implicit relevance vector that the user generates, but it gets feedback on the relevances of only top 1 or 2 ranked activities.

Economic Constraints: Assume that a small retail company wants to build an app that produces a ranked list of suggestions to a user query, for categories of different products. The app develops features representing the categories, and thus, the interaction with the users happens in a traditional query-documents lists framework (user query and retrieved activities are jointly represented through a feature matrix). Different categories are likely to have varying relevance to different users, depending on user characteristics such as age, gender, etc. Like in the first example, the user has liberty to browse through all the suggestions but she will likely feel too burdened to give carefully considered rating on

©2017 Sougata Chaudhuri and Ambuj Tewari.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-285.html.

each suggestion, unless the company provides some economic incentives to do so. Though the company needs high quality feedback on each suggestion to keep refining the ranking strategy, it cannot afford to give incentives due to budget constraints. Similar to the first example, the company only asks, and possibly pays, for rating on the top 1 or 2 ranked suggestions, in lieu of using the app, but its performance is judged on the full ranked list and implicit relevance vector.

We cast the online learning to rank problem as an online game between a learner and an adversary, played over time horizon T. That is, we do not make any *stochastic* assumption on the relevance vector generation process or the context (features) generation process (in the second problem setting). The adversary is considered to be *oblivious*, i.e., an adversary who generates moves without knowledge of the learner's algorithm. We separately discuss the two problem settings, and our contributions in each, in greater details.

1.1 Non-contextual setting

Existing work loosely related to ranking of a fixed set of items to satisfy diverse user preferences (Radlinski et al., 2008, 2009; Agrawal et al., 2009; Wen et al., 2014) has focused on learning an optimal ranking of a subset of items, to be presented to an user, with performance judged by a simple 0-1 loss. The loss in a round is 0 if among the top k (out of m) items presented to a user, the user finds at least one relevant item. All of the work falls under the framework of *online bandit* learning. In contrast, our model focuses on optimal ranking of the entire list of items, where the performance of the system is judged by practical ranking measures like DCG and AP. The challenge is to decide when and how efficient learning is possible with the highly restricted feedback model. Theoretically, the top k feedback model is neither full-feedback nor bandit-feedback since not even the loss (quantified by some ranking measure) at each round is revealed to the learner. The appropriate framework to study the problem is that of *partial monitoring* (Cesa-Bianchi et al., 2006). A very recent paper shows another practical application of partial monitoring in the stochastic setting (Lin et al., 2014). Recent advances in the classification of partial monitoring games tell us that the minimax regret, in an adversarial setting, is governed by a (two-class) property of the loss and feedback functions, called *global observability* and *local observability* (Bartok et al., 2014; Foster and Rakhlin, 2012).

Our contributions: We instantiate these general observability notions for our problem with top 1 ($k = 1$) feedback. We prove that, for some ranking measures, namely PairwiseLoss (Duchi et al., 2010), DCG and Precision@n (Liu et al., 2007), global observability holds. This immediately shows that the upper bound on regret scales as $O(T^{2/3})$. Specifically for PairwiseLoss and DCG, we further prove that local observability fails, when restricted to the top 1 feedback case, illustrating that their *minimax* regret scales as $\Theta(T^{2/3})$. However, the generic algorithm that enjoys $O(T^{2/3})$ regret for globally observable games necessarily maintains explicit weights on each action in learner's action set. It is impractical in our case to do so, since the learner's action set is the exponentially large set of $m!$ rankings over m objects. We propose a generic algorithm for learning with top k feedback, which uses blocking and a black-box full information algorithm. Specifically, we instantiate the black box algorithm with Follow The Perturbed Leader (FTPL) strategy, which leads to an efficient algorithm achieving $O(T^{2/3})$ regret bound for PairwiseLoss, DCG and

Precision@n, with $O(m \log m)$ time spent per step. Moreover, the regret of our efficient algorithm has a logarithmic dependence on number of learner's actions (i.e., polynomial dependence on m), whereas the generic algorithm has a linear dependence on number of actions (i.e., exponential dependence on m).

For several measures, their *normalized* versions are also considered. For example, the normalized versions of PairwiseLoss, DCG and Precision@n are called AUC (Cortes and Mohri, 2004), NDCG (Järvelin and Kekäläinen, 2002) and AP respectively. We show an unexpected result for the normalized versions: *they do not* admit sub-linear regret algorithms with top 1 feedback. This is despite the fact that the opposite is true for their unnormalized counterparts. Intuitively, the normalization makes it hard to construct an unbiased estimator of the (unobserved) relevance vectors. We are able to translate this intuitive hurdle into a provable impossibility.

We also present some preliminary experiments on simulated data sets to explore the performance of our efficient algorithm and compare its regret to its full information counterpart.

1.2 Contextual Setting

The requirement of having a fixed set of items to rank, in the first part of our work, somewhat limits practical applicability. In fact, in the classic multi-armed bandit problem, while non-contextual bandits have received a lot of attention, the authors Langford and Zhang (2008) mention that "settings with no context information are rare in practice". The second part of our work introduces context, by combining query-level ranking with the explicit but restricted feedback model. At each round, the adversary generates a document list of length m, pertaining to a query. The learner sees the list and produces a real valued score vector to rank the documents. We assume that the ranking is generated by sorting the score vector in descending order of its entries. The adversary then generates a relevance vector but, like in the non-contextual setting, the learner gets to see the relevance of only the top k items of the ranked list. The learner's loss in each round, based on the learner's score vector and the *full* relevance vector, is measured by some continuous ranking surrogates. We focus on continuous surrogates, e.g., the cross entropy surrogate in ListNet (Cao et al., 2007) and hinge surrogate in RankSVM (Joachims, 2002), instead of discontinuous ranking measures like DCG, or AP, because the latter lead to intractable optimization problems in the query-documents setting. Just like in the non-contextual setting, we note that the top k feedback model is neither full feedback nor bandit feedback models. The problem is an instance of partial monitoring, *extended to a setting with side information* (documents list) and an *infinite set of learner's moves* (all real valued score vectors). For such an extension of partial monitoring there exists no generic theoretical or algorithmic framework to the best of our knowledge.

Our contributions: In this setting, first, we propose a general, efficient algorithm for online learning to rank with top k feedback and show that it works in conjunction with a number of ranking surrogates. We characterize the minimum feedback required, i.e., the value of k, for the algorithm to work with a particular surrogate by formally relating the feedback mechanism with the structure of the surrogates. We then apply our general techniques to three convex ranking surrogates and one non-convex surrogate. The

convex surrogates considered are from three major learning to ranking methods: squared loss from a *pointwise* method (Cossock and Zhang, 2008), hinge loss used in the *pairwise* RankSVM (Joachims, 2002) method, and (modified) cross-entropy surrogate used in the *listwise* ListNet (Cao et al., 2007) method. The non-convex surrogate considered is the SmoothDCG surrogate (Chapelle and Wu, 2010). For the three convex surrogates, we establish an $O(T^{2/3})$ regret bound.

The convex surrogates we mentioned above are widely used but are known to fail to be calibrated with respect to NDCG (Ravikumar et al., 2011). Our second contribution is to show that for the entire class of NDCG calibrated surrogates, no online algorithm can have sub-linear (in T) regret with top 1 feedback, i.e., the minimax regret of an online game for any NDCG calibrated surrogate is $\Omega(T)$. The proof for this result relies on exploiting a connection between the construction of optimal adversary strategies for hopeless *finite action* partial monitoring games (Piccolboni and Schindelhauer, 2001) and the structure of NDCG calibrated surrogates. We only focus on NDCG calibrated surrogates for the *impossibility* results since no (convex) surrogate can be calibrated for AP and ERR (Calauzenes et al., 2012). This impossibility result is the first of its kind for a natural partial monitoring problem with side information when the learner's action space is infinite. Note, however, that there does exist work on partial monitoring problems with continuous learner actions, but without side information (Kleinberg and Leighton, 2003; Cesa-Bianchi et al., 2006), and vice versa (Bartók and Szepesvári, 2012; Gentile and Orabona, 2014).

We apply our algorithms on benchmark ranking data sets, demonstrating the ability to efficiently learn a ranking function in an online fashion, from highly restricted feedback.

The rest of the paper is divided into the following sections. Section 2 and its subsections detail the notations, definitions and technicalities associated with online ranking with restricted feedback in the non-contextual setting. Section 3 and its subsections detail the notations, definitions and technicalities associated with online ranking with restricted feedback in the contextual setting. Section 4 demonstrates the performance of our algorithms on simulated and commercial data sets. Section 5 discusses open questions and future directions of research.

2. Online Ranking with Restricted Feedback- Non Contextual Setting

All proofs not in main text are in Appendix A.

2.1 Notation and Preliminaries

The fixed m items to be ranked are numbered $\{1, 2, \ldots, m\}$. A permutation σ gives a mapping from ranks to items and its inverse σ^{-1} gives a mapping from items to ranks. Thus, $\sigma^{-1}(i) = j$ means item i is placed at position j while $\sigma(i) = j$ means item j is placed at position i. The supervision is in form of a relevance vector $R = \{0, 1, \ldots, n\}^m$, representing relevance of each document to the query. If $n = 1$, the relevance vector is binary graded. For $n > 1$, relevance vector is multi-graded. $R(i)$ denotes ith component of R, i.e., relevance of item i. The subscript t is exclusively used to denote time t. We denote $\{1, \ldots, n\}$ by $[n]$. The learner can choose from $m!$ actions (permutations) whereas nature/adversary can choose from $(n+1)^m$ outcomes (when there are n relevance levels, $n \geq 1$). We sometimes refer to the learner's ith action (in some fixed ordering of $m!$ available

actions) as σ_i (resp. adversary's ith action as R_i). Note that σ_i^{-1} simply means that we are viewing permutation σ_i as mapping from items to ranks. Also, a vector can be row or column vector depending on context.

At round t, the learner outputs a permutation (ranking) σ_t of the objects (possibly using some internal randomization, based on feedback history so far), and simultaneously, adversary generates relevance vector R_t. The quality of σ_t is judged against R_t by some ranking measure RL. *Crucially, only the relevance of the top k ranked objects are revealed to the learner at end of round t.* Thus, the learner gets to know neither R_t (full information problem) nor $RL(\sigma_t, R_t)$ (bandit problem). The objective of the learner is to minimize the expected regret with respect to best permutation in hindsight:

$$\mathbb{E}_{\sigma_1,\ldots,\sigma_T}\left[\sum_{t=1}^{T} RL(\sigma_t, R_t)\right] - \min_{\sigma} \sum_{t=1}^{T} RL(\sigma, R_t). \tag{1}$$

When RL is a gain, not loss, we need to negate the quantity above. The worst-case regret of a learner strategy is its maximal regret over all possible choices of $R_1, \ldots, R_T$. The **minimax regret** is the minimal worst-case regret over all learner strategies.

2.2 Ranking Measures

We consider ranking measures which can be expressed in the form $f(\sigma) \cdot R$, where the function $f : \mathbb{R}^m \to \mathbb{R}^m$ is composed of m copies of a univariate, monotonic, scalar valued function. Thus, $f(\sigma) = [f^s(\sigma^{-1}(1)), f^s(\sigma^{-1}(2)), \ldots, f^s(\sigma^{-1}(m))]$, where $f^s : \mathbb{R} \to \mathbb{R}$. Monotonic (increasing) means $f^s(\sigma^{-1}(i)) \geq f^s(\sigma^{-1}(j))$, whenever $\sigma^{-1}(i) > \sigma^{-1}(j)$. Monotonic (decreasing) is defined similarly. The following popular ranking measures can be expressed in the form $f(\sigma) \cdot R$.

PairwiseLoss & SumLoss: PairwiseLoss is restricted to binary relevance vectors and defined as:

$$PL(\sigma, R) = \sum_{i=1}^{m}\sum_{j=1}^{m} I(\sigma^{-1}(i) < \sigma^{-1}(j))I(R(i) < R(j))$$

PairwiseLoss cannot be directly expressed in the form of $f(\sigma) \cdot R$. Instead, we consider **SumLoss**, defined as:

$$SumLoss(\sigma, R) = \sum_{i=1}^{m} \sigma^{-1}(i)\ R(i)$$

SumLoss has the form $f(\sigma) \cdot R$, where $f(\sigma) = \sigma^{-1}$. It has been shown by Ailon (2014) that regret under the two measures are equal:

$$\sum_{t=1}^{T} PL(\sigma_t, R_t) - \sum_{t=1}^{T} PL(\sigma, R_t) = \sum_{t=1}^{T} SumLoss(\sigma_t, R_t) - \sum_{t=1}^{T} SumLoss(\sigma, R_t). \tag{2}$$

Discounted Cumulative Gain: DCG is a gain function which admits non-binary relevance vectors and is defined as:

$$DCG(\sigma, R) = \sum_{i=1}^{m} \frac{2^{R(i)} - 1}{\log_2(1 + \sigma^{-1}(i))}$$

and becomes $\sum_{i=1}^{m} \frac{R(i)}{\log_2(1+\sigma^{-1}(i))}$ for $R(i) \in \{0,1\}$. Thus, for binary relevance, $DCG(\sigma, R)$ has the form $f(\sigma) \cdot R$, where $f(\sigma) = [\frac{1}{\log_2(1+\sigma^{-1}(1))}, \frac{1}{\log_2(1+\sigma^{-1}(2))}, \ldots, \frac{1}{\log_2(1+\sigma^{-1}(m))}]$.

Precision@n Gain: Precision@n is a gain function restricted to binary relevance and is defined as

$$Precision@n(\sigma, R) = \sum_{i=1}^{m} I(\sigma^{-1}(i) \leq n)\ R(i)$$

Precision@n can be written as $f(\sigma) \cdot R$ where $f(\sigma) = [I(\sigma^{-1}(1) < n), \ldots, I(\sigma^{-1}(m) < n)]$. It should be noted that for $n = k$ (i.e., when feedback is on top n items), feedback is actually the same as full information feedback, for which efficient algorithms already exist (Kalai and Vempala, 2005).

Normalized measures are not of the form $\mathbf{f(\sigma) \cdot R}$: PairwiseLoss, DCG and Precision@n are unnormalized versions of popular ranking measures, namely, Area Under Curve (AUC), Normalized Discounted Cumulative Gain (NDCG) and Average Precision (AP) respectively. None of these can be expressed in the form $f(\sigma) \cdot R$.

NDCG: NDCG is a gain function, admits non-binary relevance and is defined as:

$$NDCG(\sigma, R) = \frac{1}{Z(R)} \sum_{i=1}^{m} \frac{2^{R(i)} - 1}{\log_2(1 + \sigma^{-1}(i))}$$

and becomes $\frac{1}{Z(R)} \sum_{i=1}^{m} \frac{R(i)}{\log_2(1+\sigma^{-1}(i))}$ for $R(i) \in \{0,1\}$. Here $Z(R) = \max_{\sigma} \sum_{i=1}^{m} \frac{2^{R(i)}-1}{\log_2(1+\sigma^{-1}(i))}$ is the normalizing factor ($Z(R) = \max_{\sigma} \sum_{i=1}^{m} \frac{R(i)}{\log_2(1+\sigma^{-1}(i))}$ for binary relevance). It can be clearly seen that $NDCG(\sigma, R) = f(\sigma) \cdot g(R)$, where $f(\sigma)$ is same as in DCG but $g(R) = \frac{R}{Z(R)}$ is non-linear in R.

AP: Average Precision is a gain function, restricted to binary relevance and is defined as:

$$AP(\sigma, R) = \frac{1}{\|R\|_1} \sum_{i=1}^{m} \frac{\sum_{j \leq i} I(R(\sigma(j)) = 1)}{i} I(R(\sigma(i) = 1))$$

It can be clearly seen that AP cannot be expressed in the form $f(\sigma) \cdot R$.

AUC: AUC is a loss function, restricted to binary relevance and is defined as:

$$AUC(\sigma, R) = \frac{1}{N(R)} \sum_{i=1}^{m} \sum_{j=1}^{m} I(\sigma^{-1}(i) < \sigma^{-1}(j)) I(R(i) < R(j))$$

where $N(R) = (\sum_{i=1}^{m} I(R(i) = 1)) \cdot (m - \sum_{i=1}^{m} I(R(i) = 1))$. It can be clearly seen that AUC cannot be expressed in the form $f(\sigma) \cdot R$.

Note: We will develop our subsequent theory and algorithms for binary valued relevance vectors, and show how they can be extended to multi-graded vectors when ranking measure is DCG/NDCG.

2.3 Relevant Definitions from Partial Monitoring

We develop all results in context of SumLoss, with binary relevance vector. We then extend the results to other ranking measures. Our main results on regret bounds build on some

Table 1: Loss matrix L for $m = 3$

Objects	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
123	000	001	010	011	100	101	110	111
$\sigma_1 = 123$	0	3	2	5	1	4	3	6
$\sigma_2 = 132$	0	2	3	5	1	3	4	6
$\sigma_3 = 213$	0	3	1	4	2	5	3	6
$\sigma_4 = 231$	0	1	3	4	2	3	5	6
$\sigma_5 = 312$	0	2	1	3	3	5	4	6
$\sigma_6 = 321$	0	1	2	3	3	4	5	6

Table 2: Feedback matrix H for $m = 3$

Objects	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
123	000	001	010	011	100	101	110	111
$\sigma_1 = 123$	0	0	0	0	1	1	1	1
$\sigma_2 = 132$	0	0	0	0	1	1	1	1
$\sigma_3 = 213$	0	0	1	1	0	0	1	1
$\sigma_4 = 231$	0	1	0	1	0	1	0	1
$\sigma_5 = 312$	0	0	1	1	0	0	1	1
$\sigma_6 = 321$	0	1	0	1	0	1	0	1

of the theory for abstract partial monitoring games developed by Bartok et al. (2014) and Foster and Rakhlin (2012). For ease of understanding, we reproduce the relevant notations and definitions in context of SumLoss. *We will specifically mention when we derive results for top k feedback, with general k, and when we restrict to top 1 feedback.*

Loss and Feedback Matrices: The online learning game with the SumLoss measure and top 1 feedback can be expressed in form of a pair of *loss matrix* and *feedback matrix*. The *loss matrix* L is an $m! \times 2^m$ dimensional matrix, with rows indicating the learner's actions (permutations) and columns representing adversary's actions (relevance vectors). The entry in cell (i, j) of L indicates loss suffered when learner plays action i (i.e., σ_i) and adversary plays action j (i.e., R_j), that is, $L_{i,j} = \sigma_i^{-1} \cdot R_j = \sum_{k=1}^{m} \sigma_i^{-1}(k) R_j(k)$. The *feedback matrix* H has same dimension as *loss matrix*, with (i, j) entry being the relevance of top ranked object, i.e., $H_{i,j} = R_j(\sigma_i(1))$. When the learner plays action σ_i and adversary plays action R_j, the true loss is $L_{i,j}$, while the feedback received is $H_{i,j}$.

Table 1 and 2 illustrate the matrices, with number of objects $m = 3$. In both the tables, the permutations indicate rank of each object and relevance vector indicates relevance of each object. For example, $\sigma_5 = 312$ means object 1 is ranked 3, object 2 is ranked 1 and object 3 is ranked 2. $R_5 = 100$ means object 1 has relevance level 1 and other two objects have relevance level 0. Also, $L_{3,4} = \sigma_3 \cdot R_4 = \sum_{i=1}^{3} \sigma_3^{-1}(i) R_4(i) = 2 \cdot 0 + 1 \cdot 1 + 3 \cdot 1 = 4$; $H_{3,4} = R_4(\sigma_3(1)) = R_4(2) = 1$. Other entries are computed similarly.

Let $\ell_i \in \mathbb{R}^{2^m}$ denote row i of L. Let Δ be the probability simplex in $\mathbb{R}^{2^m}$, i.e., $\Delta = \{p \in \mathbb{R}^{2^m} : \forall\ 1 \le i \le 2^m,\ p_i \ge 0,\ \sum p_i = 1\}$. The following definitions, given for abstract problems by Bartok et al. (2014), has been refined to fit our problem context.

Definition 1: Learner action i is called optimal under distribution $p \in \Delta$, if $\ell_i \cdot p \le \ell_j \cdot p$, for all other learner actions $1 \le j \le m!,\ j \ne i$. For every action $i \in [m!]$, probability cell of i is defined as $C_i = \{p \in \Delta : \text{action } i \text{ is optimal under } p\}$. If a non-empty cell C_i is $2^m - 1$ dimensional (i.e, elements in C_i are defined by only 1 equality constraint), then associated action i is called *Pareto-optimal*.

Note that since entries in H are relevance levels of objects, there can be maximum of 2 distinct elements in each row of H, i.e., 0 or 1 (assuming binary relevance).

Definition 2: The *signal matrix* S_i, associated with learner's action σ_i, is a matrix with 2 rows and 2^m columns, with each entry 0 or 1, i.e., $S_i \in \{0,1\}^{2 \times 2^m}$. The entries of ℓth column of S_i are respectively: $(S_i)_{1,\ell} = I(H_{i,\ell} = 0)$ and $(S_i)_{2,\ell} = I(H_{i,\ell} = 1)$.

Note that by definitions of signal and feedback matrices, the 2nd row of S_i (2nd column of $S_i^\top$)) is precisely the ith row of H. The 1st row of S_i (1st column of $S_i^\top$)) is the (boolean) complement of ith row of H.

2.4 Minimax Regret for SumLoss

The minimax regret for SumLoss, restricted to top 1 feedback, will be established by showing that: a) SumLoss satisfies *global observability*, and b) it does not satisfy *local observability*.

2.4.1 Global Observability

Definition 3: The condition of *global observability* holds, w.r.t. loss matrix L and feedback matrix H, if for every pair of learner's actions $\{\sigma_i, \sigma_j\}$, it is true that $\ell_i - \ell_j \in \oplus_{k \in [m!]} Col(S_k^\top)$, where Col refers to column space (i.e., the vector difference belongs in the column span).

The global observability condition states that the (vector) loss difference between any pair of learner's actions has to belong to the vector space spanned by columns of (transposed) signal matrices corresponding to all possible learner's actions. We derive the following theorem on global observability for *SumLoss*.

Theorem 1. *The global observability condition, as per Definition 3, holds w.r.t. loss matrix L and feedback matrix H defined for SumLoss, for any $m \ge 1$.*

Proof. For any σ_a (learner's action) and R_b (adversary's action), we have (please see the end of the proof for explanation of notations and equalities):

$$L_{a,b} = \sigma_a^{-1} \cdot R_b = \sum_{i=1}^{m} \sigma_a^{-1}(i) R_b(i) \overset{1}{=} \sum_{j=1}^{m} j\ R_b(\sigma_a(j)) \overset{2}{=} \sum_{j=1}^{m} j\ R_b(\tilde{\sigma}_{j(a)}(1)) \overset{3}{=} \sum_{j=1}^{m} j\ (S^\top_{\tilde{\sigma}_{j(a)}})_{R_b,2}.$$

Thus, we have

$$\ell_a = [L_{a,1}, L_{a,2}, \ldots, L_{a,2^m}] =$$
$$[\sum_{j=1}^{m} j\ (S^\top_{\tilde{\sigma}_{j(a)}})_{R_1,2}, \sum_{j=1}^{m} j\ (S^\top_{\tilde{\sigma}_{j(a)}})_{R_2,2}, .., \sum_{j=1}^{m} j\ (S^\top_{\tilde{\sigma}_{j(a)}})_{R_{2^m},2}] \overset{4}{=} \sum_{j=1}^{m} j\ (S^\top_{\tilde{\sigma}_{j(a)}})_{:,2}\ (:,2 \text{ indicates 2nd column}).$$

Equality 4 shows that ℓ_a is in the column span of m of the $m!$ possible (transposed) signal matrices, specifically in the span of the 2nd columns of those (transposed) m matrices. Hence, for all actions σ_a, it holds that $\ell_a \in \oplus_{k \in [m!]} Col(S_k^\top)$. This implies that $\ell_a - \ell_b \in \oplus_{k \in [m!]} Col(S_k^\top),\ \forall\ \sigma_a, \sigma_b$.

1. Equality 1 holds because $\sigma_a^{-1}(i) = j \Rightarrow i = \sigma_a(j)$.
2. Equality 2 holds because of the following reason. For any permutation σ_a and for every $j \in [m]$, $\exists$ a permutation $\tilde{\sigma}_{j(a)}$, s.t. the object which is assigned rank j by σ_a is the same object assigned rank 1 by $\tilde{\sigma}_{j(a)}$, i.e., $\sigma_a(j) = \tilde{\sigma}_{j(a)}(1)$.
3. In Equality 3, $(S^\top_{\tilde{\sigma}^{-1}_{j(a)}})_{R_b,2}$ indicates the R_bth row and 2nd column of (transposed) signal matrix $S_{\tilde{\sigma}_{j(a)}}$, corresponding to learner action $\tilde{\sigma}_{j(a)}$. Equality 3 holds because $R_b(\tilde{\sigma}_{j(a)}(1))$ is the entry in the row corresponding to action $\tilde{\sigma}_{j(a)}$ and column corresponding to action R_b of H (see Definition 2).
4. Equality 4 holds from the observation that for a particular j, $[(S^\top_{\tilde{\sigma}_{j(a)}})_{R_1,2}, (S^\top_{\tilde{\sigma}_{j(a)}})_{R_2,2}, \ldots, (S^\top_{\tilde{\sigma}_{j(a)}})_{R_{2^m},2}]$ forms the 2nd column of $(S^\top_{\tilde{\sigma}_{j(a)}})$, i.e., $(S^\top_{\tilde{\sigma}_{j(a)}})_{:,2}$. □

2.4.2 Local Observability

Definition 4: Two Pareto-optimal (learner's) actions i and j are called *neighboring actions* if $C_i \cap C_j$ is a $(2^m - 2)$ dimensional polytope (where C_i is probability cell of action σ_i). The *neighborhood action set* of two neighboring (learner's) actions i and j is defined as $N^+_{i,j} = \{k \in [m!] : C_i \cap C_j \subseteq C_k\}$.

Definition 5: A pair of neighboring (learner's) actions i and j is said to be locally observable if $\ell_i - \ell_j \in \oplus_{k \in N^+_{i,j}} Col(S_k^\top)$. The condition of *local observability* holds if every pair of neighboring (learner's) actions is locally observable.

We now show that local observability condition fails for L, H under SumLoss. First, we present the following two lemmas characterizing Pareto-optimal actions and neighboring actions for SumLoss.

Lemma 2. *For SumLoss, each of learner's action σ_i is Pareto-optimal, where Pareto-optimality has been defined in Definition 1.*

Proof. For any $p \in \Delta$, we have $\ell_i \cdot p = \sum_{j=1}^{2^m} p_j\ (\sigma_i^{-1} \cdot R_j) = \sigma_i^{-1} \cdot (\sum_{j=1}^{2^m} p_j R_j) = \sigma_i^{-1} \cdot \mathbb{E}[R]$, where the expectation is taken w.r.t. p. $l_i \cdot p$ is minimized when ranking of objects according to σ_i and expected relevance of objects are in opposite order. That is, the object with highest expected relevance is ranked 1 and so on. Formally, $l_i \cdot p$ is minimized when $\mathbb{E}[R(\sigma_i(1))] \ge \mathbb{E}[R(\sigma_i(2))] \ge \ldots \ge \mathbb{E}[R(\sigma_i(m))]$.

Thus, for action σ_i, probability cell is defined as $C_i = \{p \in \Delta : \sum_{j=1}^{2^m} p_j = 1,\ \mathbb{E}[R(\sigma_i(1))] \ge \mathbb{E}[R(\sigma_i(2))] \ge \ldots \ge \mathbb{E}[R(\sigma_i(m))]\}$. Note that, $p \in C_i$ iff action i is optimal w.r.t. p. Since C_i is obviously non-empty and it has only 1 equality constraint (hence $2^m - 1$ dimensional), action i is Pareto optimal.

The above holds true for all learner's actions σ_i. □

Lemma 3. *A pair of learner's actions $\{\sigma_i,\ \sigma_j\}$ is a neighboring actions pair, if there is exactly one pair of objects, numbered $\{a, b\}$, whose positions differ in σ_i and σ_j. Moreover, a needs to be placed just before b in σ_i and b needs to placed just before a in σ_j.*

Proof. From Lemma 2, we know that every one of learner's actions is Pareto-optimal and C_i, associated with action σ_i, has structure $C_i = \{p \in \Delta : \sum_{j=1}^{2^m} p_j = 1,\ \mathbb{E}[R(\sigma_i(1))] \geq \mathbb{E}[R(\sigma_i(2))] \geq \ldots \geq \mathbb{E}[R(\sigma_i(m))]\}$.

Let $\sigma_i(k) = a,\ \sigma_i(k+1) = b$. Let it also be true that $\sigma_j(k) = b,\ \sigma_j(k+1) = a$ and $\sigma_i(n) = \sigma_j(n),\ \forall n \neq \{k,\ k+1\}$. Thus, objects in $\{\sigma_i, \sigma_j\}$ are same in all places except in a pair of consecutive places where the objects are interchanged.

Then, $C_i \cap C_j = \{p \in \Delta : \sum_{j=1}^{2^m} p_j = 1,\ \mathbb{E}[R(\sigma_i(1)] \geq \ldots \geq \mathbb{E}[R(\sigma_i(k)] = \mathbb{E}[R(\sigma_i(k+1)] \geq \ldots \geq \mathbb{E}[R(\sigma_i(m)]\}$. Hence, there are two equalities in the non-empty set $C_i \cap C_j$ and it is an $(2^m - 2)$ dimensional polytope. Hence condition of Definition 4 holds true and $\{\sigma_i, \sigma_j\}$ are neighboring actions pair.

□

Lemma 2 and 3 are used to establish the following result:

Theorem 4. *The local observability condition, as per Definition 5, fails w.r.t. loss matrix L and feedback matrix H defined for SumLoss, already at $m = 3$.*

Proof. We will explicitly show that local observability condition fails by considering the case when number of objects is $m = 3$. Specifically, action pair $\{\sigma_1,\ \sigma_2\}$, in Table 1 are neighboring actions, using Lemma 3 . Now every other action $\{\sigma_3, \sigma_4, \sigma_5, \sigma_6\}$ either places object 2 at top or object 3 at top. It is obvious that the set of probabilities for which $E[R(1)] \geq E[R(2)] = E[R(3)]$ cannot be a subset of any C_3, C_4, C_5, C_6. From Def. 4, the neighborhood action set of actions $\{\sigma_1, \sigma_2\}$ is precisely σ_1 and σ_2 and contains no other actions. By definition of signal matrices $S_{\sigma_1},\ S_{\sigma_2}$ and entries $\ell_1,\ \ell_2$ in Table 1 and 2, we have,

$$S_{\sigma_1} = S_{\sigma_2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$
$$\ell_1 - \ell_2 = \begin{bmatrix} 0 & 1 & -1 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}. \tag{3}$$

It is clear that $\ell_1 - \ell_2 \notin Col(S_{\sigma_1}^\top)$. Hence, Definition 5 fails to hold. □

2.5 Minimax Regret Bound

We establish the minimax regret for SumLoss by combining results on global and local observability. First, we get a lower bound by combining our Theorem 4 with Theorem 4 of Bartok et al. (2014).

Corollary 5. *Consider the online game for SumLoss with top-1 feedback and $m = 3$. Then, for every learner's algorithm, there is an adversary strategy generating relevance vectors, such that the expected regret of the learner is $\Omega(T^{2/3})$.*

The fact that the game is globally observable (Theorem 1), combined with Theorem 3.1 in Cesa-Bianchi et al. (2006), gives an algorithm (inspired by the algorithm originally given in Piccolboni and Schindelhauer (2001)) obtaining $O(T^{2/3})$ regret.

Corollary 6. *The algorithm in Figure 1 of Cesa-Bianchi et al. (2006) achieves $O(T^{2/3})$ regret bound for SumLoss.*

However, the algorithm in Cesa-Bianchi et al. (2006) is intractable in our setting since the algorithm necessarily enumerates all the actions of the learner in each round, which is exponential in m in our case ($m!$ to be exact). Moreover, the regret bound of the algorithm also has a linear dependence on the number of actions, which renders the bound useless.

Discussion: The results above establish that the minimax regret for SumLoss, *restricted to top-1 feedback*, is $\Theta(T^{2/3})$. Theorem 4 of Bartok et al. (2014) says the following: A partial monitoring game which is both globally and locally observable has minimax regret $\Theta(T^{1/2})$, while a game which is globally observable but *not* locally observable has minimax regret $\Theta(T^{2/3})$. In Theorem 1, we proved global observability, when feedback is restricted to relevance of top ranked item. The global observability result automatically extends to feedback on top k items, for $k > 1$. This is because for top k feedback, with $k > 1$, the learner receives strictly greater information at end of each round than top 1 feedback (for example, the learner can just throw away relevance feedback on items ranked 2nd onwards). So, with top k feedback, for general k, the game will remain at least globally observable. In fact, our algorithm in the next section will achieve $O(T^{2/3})$ regret bound for SumLoss with top k feedback, $k \geq 1$. However, the same is not true for failure of local observability. Feedback on more than top ranked item can make the game strictly easier for the learner and may make local observability condition hold, for some $k > 1$. In fact, for $k = m$ (full feedback), the game will be a simple full information game (disregarding computational complexity), and hence locally observable.

2.6 Algorithm for Obtaining Minimax Regret under SumLoss with Top k Feedback

We first provide a general algorithmic framework for getting an $O(T^{2/3})$ regret bound for SumLoss, with feedback on top k ranked items per round, for $k \geq 1$. We then instantiate a specific algorithm, which spends $O(m \log m)$ time per round (thus, highly efficient) and obtains a regret of rate $O(poly(m)\ T^{2/3})$.

2.6.1 General Algorithmic Framework

Our algorithm combines *blocking* with a randomized *full information* algorithm. We first divide time horizon T into blocks (referred to as blocking). Within each block, we allot a small number of rounds for pure *exploration*, which allows us to estimate the average of the full relevance vectors generated by the adversary in that block. The estimated average vector is cumulated over blocks and then fed to a full information algorithm for the next block. The randomized full information algorithm *exploits* the information received at the beginning of the block to maintain distribution over permutations (learner's actions). In each round in the new block, actions are chosen according to the distribution and presented to the user.

The *key property* of the randomized full information algorithm is this: the algorithm should have an expected regret rate of $O(C\sqrt{T})$, where the regret is the difference between cumulative loss of the algorithm and cumulative loss of best action in hindsight, over T rounds, and C is a parameter specific to the full information algorithm.

Our algorithm is motivated by the reduction from bandit-feedback to full feedback scheme given in Blum and Mansour (2007). However, the reduction *cannot be directly applied to our problem*, because we are not in the bandit setting and hence do not know loss of any action. Further, the algorithm of Blum and Mansour (2007) necessarily spends N rounds per block to try out *each* of the N available actions — this is impractical in our setting since $N = m!$.

Algorithm 1 describes our approach. A key aspect is the formation of estimate of average relevance vector of a block (line 16), for which we have the following lemma:

Lemma 7. *Let the average of (full) relevance vectors over the time period $\{1, 2, \ldots, t\}$ be denoted as $R^{avg}_{1:t}$, that is, $R^{avg}_{1:t} = \sum_{n=1}^{t} \frac{R_n}{t} \in \mathbb{R}^m$. Let $\{i_1, i_2, \ldots, i_{\lceil m/k \rceil}\}$ be $\lceil m/k \rceil$ arbitrary time points, chosen uniformly at random, without replacement, from $\{1, \ldots, t\}$. At time point i_j, only k distinct components of relevance vector R_{i_j}, i.e., $\{R_{i_j}(k \cdot (j-1)+1), R_{i_j}(k \cdot (j-1)+2), \ldots, R_{i_j}(k \cdot j)\}$, become known, $\forall j \in \{1, \ldots, \lceil m/k \rceil\}$ (for $j = \lceil m/k \rceil$, there might be less than k components available). Then the vector formed from the m revealed components, i.e. $\hat{R}_t = [R_{i_j}(k \cdot (j-1)+1), R_{i_j}(k \cdot (j-1)+2), \ldots, R_{i_j}(k \cdot j)]_{\{j=1,2,\ldots,\lceil m/k \rceil\}}$ is an unbiased estimator of $R^{avg}_{1:t}$.*

Proof. We can write $\hat{R}_t = \sum_{j=1}^{\lceil m/k \rceil} \sum_{\ell=1}^{k} R_{i_j}(k \cdot (j-1)+\ell) e_{k \cdot (j-1)+\ell}$, where e_i is the m dimensional standard basis vector along coordinate j. Then, taking expectation over the randomly chosen time points, we have: $E_{i_1,\ldots,i_{\lceil m/k \rceil}}(\hat{R}_t) = \sum_{j=1}^{\lceil m/k \rceil} E_{i_j}[\sum_{\ell=1}^{k} R_{i_j}(k \cdot (j-1)+\ell) e_{k \cdot (j-1)+\ell}] = \sum_{j=1}^{m} \sum_{\ell=1}^{k} \sum_{n=1}^{t} \frac{R_n(k \cdot (j-1)+\ell) e_{k \cdot (j-1)+\ell}}{t} = R^{avg}_{1:t}$. □

Suppose we have a full information algorithm whose regret in T rounds is upper bounded by $C\sqrt{T}$ for some constant C and let C^I be the maximum loss that the learner can suffer in a round. Note that C^I depends on the loss used and on the range of the relevance scores. We have the following regret bound, obtained from application of Algorithm 1 on SumLoss with top k feedback.

Theorem 8. *Let C, C^I be the constants defined above. The expected regret under SumLoss, obtained by applying Algorithm 1, with relevance feedback on top k ranked items per round ($k \geq 1$), and the expectation being taken over randomized learner's actions σ_t, is*

$$\mathbb{E}\left[\sum_{t=1}^{T} SumLoss(\sigma_t, R_t)\right] - \min_{\sigma} \sum_{t=1}^{T} SumLoss(\sigma_t, R_t) \leq C^I \lceil m/k \rceil K + C\frac{T}{\sqrt{K}}. \quad (4)$$

Optimizing over block size K, the final regret bound is:

$$\mathbb{E}\left[\sum_{t=1}^{T} SumLoss(\sigma_t, R_t)\right] - \min_{\sigma} \sum_{t=1}^{T} SumLoss(\sigma_t, R_t) \leq 2(C^I)^{1/3} C^{2/3} \lceil m/k \rceil^{1/3} T^{2/3}. \quad (5)$$

2.6.2 Computationally Efficient Algorithm with FTPL

We instantiate our general algorithm with *Follow The Perturbed Leader* (FTPL) full information algorithm (Kalai and Vempala, 2005). The following choices need to be made in Algorithm 1 to implement FTPL as the full information algorithm:

Algorithm 1 RankingwithTop-kFeedback(RTop-kF)- Non Contextual

1: T = Time horizon, K = No. of (equal sized) blocks, FI= randomized full information algorithm.
2: Time horizon divided into equal sized blocks $\{B_1, \ldots, B_K\}$, where $B_i = \{(i-1)(T/K)+1, \ldots, i(T/K)\}$.
3: Initialize $\hat{s}_0 = \mathbf{0} \in \mathbb{R}^m$. Initialize any other parameter specific to FI.
4: **For** $i = 1, \ldots, K$
5: Select $\lceil m/k \rceil$ time points $\{i_1, \ldots, i_{\lceil m/k \rceil}\}$ from block B_i, uniformly at random, without replacement.
6: Divide the m items into $\lceil m/k \rceil$ cells, with k distinct items in each cell.[1]
7: **For** $t \in B_i$
8: **If** $t = i_j \in \{i_1, \ldots, i_{\lceil m/k \rceil}\}$
9: **Exploration round**:
10: Output any permutation σ_t which places items of jth cell in top k positions (in any order).
11: Receive feedback as relevance of top k items of σ_t (i.e., items of jth cell).
12: **Else**
13: **Exploitation round**:
14: Feed $\hat{s}_{i-1}$ to the randomized full information algorithm FI and output σ_t according to FI.
15: **end for**
16: Set $\hat{R}_i \in \mathbb{R}^m$ as vector of relevances of the m items collected during exploration rounds.
17: Update $\hat{s}_i = \hat{s}_{i-1} + \hat{R}_i$.
18: **end for**

Initialization of parameters: In line 3 of the algorithm, the parameter specific to FTPL is randomization parameter $\epsilon \in \mathbb{R}$.

Exploitation round: σ_t, during exploitation, is sampled by FTPL as follows: sample $p_t \in [0, 1/\epsilon]^m$ from the product of uniform distribution in each dimension. Output permutation $\sigma_t = M(\hat{s}_{i-1} + p_t)$ where $M(y) = \operatorname{argmin}_{\sigma} \sigma^{-1} \cdot y$.

Discussion: The key reason for using FTPL as the full information algorithm is that the structure of our problem allows the permutation σ_t to be chosen during exploitation round via a simple sorting operation on m objects. This leads to an easily implementable algorithm which spends only $O(m \log m)$ time per round (sorting is in fact the most expensive step in the algorithm). The reason that the simple sorting operation does the trick is the following: FTPL only *implicitly* maintains a distribution over $m!$ actions (permutations) at beginning of each round. Instead of having an explicit probability distribution over each action and sampling from it, FTPL mimics sampling from a distribution over actions by randomly perturbing the information vector received so far (say $\hat{s}_{i-1}$ in block B_i) and then sorting the items by perturbed score. The random perturbation puts an implicit weight on each of the $m!$ actions and sorting is basically sampling according to the weights. This is an advantage over general full information algorithms based on exponential weights, which maintain explicit weight on actions and samples from it. We also note that we are using FTPL with uniform distribution over a hypercube as our perturbation distribution. Other choices, such as the Gaussian distribution, can lead to slightly better dependence on the dimension m (see the discussion following the proof of Corollary 9 in the appendix). Our bounds, therefore, do not necessarily yield the optimal dependence on m.

[1] For example, assume $m = 7$ and $k = 2$. Then place items $(1, 2)$ in cell 1, items $(3, 4)$ in cell 2, items $(5, 6)$ in cell 3 and item 7 in cell 4.

We have the following corollary:

Corollary 9. *The expected regret of SumLoss, obtained by applying Algorithm 1, with FTPL full information algorithm and feedback on top k ranked items at end of each round ($k \geq 1$), and* $K = O\left(\frac{m^{1/3}T^{2/3}}{\lceil m/k \rceil^{2/3}}\right)$, $\epsilon = O(\frac{1}{\sqrt{mK}})$, *is:*

$$\mathbb{E}\left[\sum_{t=1}^{T} SumLoss(\sigma_t, R_t)\right] - \min_{\sigma} \sum_{t=1}^{T} SumLoss(\sigma, R_t) \leq O(m^{7/3}\lceil m/k \rceil^{1/3} T^{2/3}). \quad (6)$$

where $O(\cdot)$ hides some numeric constants.

Assuming that $\lceil m/k \rceil \sim m/k$, the regret rate in Corollary 9 is $O\left(\frac{m^{8/3}T^{2/3}}{k^{1/3}}\right)$

2.7 Regret Bounds for PairwiseLoss, DCG and Precision@n

PairwiseLoss: As we saw in Eq. 2, the regret of SumLoss is same as regret of PairwiseLoss. Thus, SumLoss in Corollary 9 can be replaced by PairwiseLoss to get exactly same result.

DCG: All the results of SumLoss can be extended to DCG (see Appendix A). Moreover, the results can be extended even for multi-graded relevance vectors. Thus, the minimax regret under DCG, *restricted to feedback on top ranked item*, even when the adversary can play multi-graded relevance vectors, is $\Theta(T^{2/3})$.

The main differences between SumLoss and DCG are the following. The former is a loss function; the latter is a gain function. Also, for DCG, $f(\sigma) \neq \sigma^{-1}$ (see definition in Sec.2.2) and when relevance is multi-graded, DCG cannot be expressed as $f(\sigma) \cdot R$, as clear from definition. Nevertheless, DCG can be expressed as $f(\sigma) \cdot g(R)$, , where $g(R) = [g^s(R(1)), g^s(R(2)), \ldots, g^s(R(m))]$, $g^s(i) = 2^i - 1$ is constructed from univariate, monotonic, scalar valued functions ($g(R) = R$ for binary graded relevance vectors). Thus, Algorithm 1 can be applied (with slight variation), with FTPL full information algorithm and top k feedback, to achieve regret of $O(T^{2/3})$ [2]. The slight variation is that during *exploration* rounds, when relevance feedback is collected to form the estimator at end of the block, the relevances should be transformed by function $g^s(\cdot)$.. The estimate is then constructed in the transformed space and fed to the full information algorithm. In the *exploitation* round, the selection of σ_t remains exactly same as in SumLoss, i.e., $\sigma_t = M(\hat{s}_{i-1} + p_t)$ where $M(y) = \operatorname{argmin}_{\sigma} \sigma^{-1} \cdot y$. This is because $\operatorname{argmax}_{\sigma} f(\sigma) \cdot y = \operatorname{argmin}_{\sigma} \sigma^{-1} \cdot y$, by definition of $f(\sigma)$ in DCG.

Let relevance vectors chosen by adversary have $n+1$ grades, i.e., $R \in \{0, 1, \ldots, n\}^m$. In practice, n is almost always less than 5. We have the following corollary:

Corollary 10. *The expected regret of DCG, obtained by applying Algorithm 1, with FTPL full information algorithm and feedback on top k ranked items at end of each round ($k \geq 1$), and* $K = O\left(\frac{m^{1/3}T^{2/3}}{\lceil m/k \rceil^{2/3}}\right)$, $\epsilon = O(\frac{1}{(2^n-1)^2\sqrt{mK}})$, *is:*

$$\max_{\sigma} \sum_{t=1}^{T} DCG(\sigma_t, R_t) - \mathbb{E}\left[\sum_{t=1}^{T} DCG(\sigma_t, R_t)\right] \leq O((2^n - 1)m^{4/3}\lceil m/k \rceil^{1/3} T^{2/3}). \quad (7)$$

Assuming that $\lceil m/k \rceil \sim m/k$, the regret rate in Corollary 10 is $O\left(\frac{(2^n-1)m^{5/3}T^{2/3}}{k^{1/3}}\right)$.

Precision@n: Since Precision@$n = f(\sigma) \cdot R$, the global observability property of SumLoss can be easily extended to it and Algorithm 1 can be applied, with FTPL full information algorithm and top k feedback, to achieve regret of $O(T^{2/3})$. In the *exploitation* round, the selection of σ_t remains exactly same as in SumLoss, i.e., $\sigma_t = M(\hat{s}_{i-1} + p_t)$ where $M(y) = \operatorname{argmin}_{\sigma} \sigma^{-1} \cdot y$.

However, the local observability property of SumLoss does not extend to Precision@n. The reason is that while $f(\cdot)$ of SumLoss is strictly monotonic, $f(\cdot)$ of Precision@n is monotonic but not strict. Precision@n depends only on the objects in the top n positions of the ranked list, *irrespective of the order*. A careful review shows that Lemma 3 fails to extend to the case of Precision@n, due to lack of strict monotonicity. Thus, we cannot define the neighboring action set of the Pareto optimal action pairs, and hence cannot prove or disprove local observability.

We have the following corollary:

Corollary 11. *The expected regret of Precision@n, obtained by applying Algorithm 1, with FTPL full information algorithm and feedback on top k ranked items at end of each round ($k \geq 1$), and* $K = O\left(\frac{m^{1/3}T^{2/3}}{\lceil m/k \rceil^{2/3}}\right)$, $\epsilon = O(\frac{1}{\sqrt{mK}})$, *is:*

$$\max_{\sigma} \sum_{t=1}^{T} Precision@n(\sigma_t, R_t) - \mathbb{E}\left[\sum_{t=1}^{T} Precision@n(\sigma_t, R_t)\right] \leq O(n\ m^{1/3}\lceil m/k \rceil^{1/3} T^{2/3}). \quad (8)$$

Assuming that $\lceil m/k \rceil \sim m/k$, the regret rate in Corollary 11 is $O\left(\frac{n\ m^{2/3}T^{2/3}}{k^{1/3}}\right)$.

2.8 Non-Existence of Sublinear Regret Bounds for NDCG, AP and AUC

As stated in Sec. 2.2, NDCG, AP and AUC are normalized versions of measures DCG, Precision@n and PairwiseLoss. We have the following lemma for all these normalized ranking measures.

Lemma 12. *The global observability condition, as per Definition 1, fails for NDCG, AP and AUC, when feedback is restricted to top ranked item.*

Combining the above lemma with Theorem 2 of Bartok et al. (2014), we conclude that there *cannot exist any algorithm which has sub-linear regret for any of the following measures: NDCG, AP or AUC, when restricted to top* 1 *feedback.*

[2] Note that there is no problem in applying Lemma 7 to DCG. This is because we are trying to estimate $\sum_{n=1}^{t} \frac{g(R_n)}{t}$, which can be estimated by considering the relevance feedbacks themselves transformed by $g(\cdot)$

Theorem 13. *There exists an online game, for NDCG with top-1 feedback, such that for every learner's algorithm, there is an adversary strategy generating relevance vectors, such that the expected regret of the learner is $\Omega(T)$. Furthermore, the same lower bound holds if NDCG is replaced by AP or AUC.*

3. Online Ranking with Restricted Feedback- Contextual Setting

All proofs not in the main text are in Appendix B.

3.1 Problem Setting and Learning to Rank Algorithm

First, we introduce some additional notations to Section 2.1. In the contextual setting, each query and associated items (documents) are represented jointly as a feature matrix. Each feature matrix, $X \in \mathbb{R}^{m\times d}$, consists of a list of m documents, each represented as a feature vector in $\mathbb{R}^d$. The feature matrices are considered side-information (context) and represents varying items, as opposed to the fixed set of items in the first part of our work. $X_{i:}$ denotes ith row of X. We assume feature vectors representing documents are bounded by R_D in ℓ_2 norm. The relevance vectors are same as before.

As per traditional learning to rank setting with query-document matrices, documents are ranked by a ranking function. The prevalent technique is to represent a ranking function as a scoring function and get ranking by sorting scores in descending order. A linear scoring function produces score vector as $f_w(X) = Xw = s^w \in \mathbb{R}^m$, with $w \in \mathbb{R}^d$. Here, $s^w(i)$ represents score of ith document (s^w points to score s being generated by using parameter w). We assume that ranking parameter space is bounded in ℓ_2 norm, i.e, $\|w\|_2 \le U$, $\forall\ w$. $\pi_s = \text{argsort}(s)$ is the permutation induced by sorting score vector s in descending order. As a reminder, a permutation π gives a mapping from ranks to documents and π^{-1} gives a mapping from documents to ranks.

Performance of ranking functions are judged, based on the rankings obtained from score vectors, by ranking measures like DCG, AP and others. However, the measures themselves are discontinuous in the score vector produced by the ranking function, leading to intractable optimization problems. Thus, most learning to rank methods are based on minimizing *surrogate* losses, which can be optimized efficiently. A surrogate ϕ takes in a score vector s and relevance vector R and produces a real number, i.e., $\phi : \mathbb{R}^m \times \{0, 1, \ldots, n\}^m \mapsto \mathbb{R}$. $\phi(\cdot, \cdot)$ is said to be convex if it is convex in its first argument, for any value of the second argument. Ranking surrogates are designed in such a way that the ranking function learnt by optimizing the surrogates has good performance with respect to ranking measures.

Formal problem setting: We formalize the problem as a game being played between a learner and an adversary over T rounds. The learner's action set is the uncountably infinite set of score vectors in $\mathbb{R}^m$ and the adversary's action set is all possible relevance vectors, i.e., $(n+1)^m$ possible vectors. At round t, the adversary generates a list of documents, represented by a matrix $X_t \in \mathbb{R}^{m\times d}$, pertaining to a query (the document list is considered as side information). The learner receives X_t, produces a score vector $\tilde{s}_t \in \mathbb{R}^m$ and ranks the documents by sorting according to score vector. The adversary then generates a relevance vector R_t but only reveals the relevances of top k ranked documents to the learner. The learner uses the feedback to choose its action for the next round (updates an internal scoring function). The learner suffers a loss as measured in terms of a surrogate ϕ, i.e, $\phi(\tilde{s}_t, R_t)$.

As is standard in online learning setting, the learner's performance is measured in terms of its expected regret:

$$\mathbb{E}\left[\sum_{t=1}^{T}\phi(\tilde{s}_t, R_t)\right] - \min_{\|w\|_2\le U}\sum_{t=1}^{T}\phi(X_t w, R_t),$$

where the expectation is taken w.r.t. to randomization of learner's strategy and $X_t w = s_t^w$ is the score produced by the linear function parameterized by w.

Algorithm 2 Ranking with Top-k Feedback (RTop-kF)- Contextual

1: Exploration parameter $\gamma \in (0, \frac{1}{2})$, learning parameter $\eta > 0$, ranking parameter $w_1 = \mathbf{0} \in \mathbb{R}^d$
2: **For** $t = 1$ to T
3: Receive X_t (document list pertaining to query q_t)
4: Construct score vector $s_t^{w_t} = X_t w_t$ and get permutation $\sigma_t = \text{argsort}(s_t^{w_t})$
5: $\mathbb{Q}_t(s) = (1-\gamma)\delta(s - s_t^{w_t}) + \gamma\text{Uniform}([0,1]^m)$ (δ is the Dirac Delta function).
6: Sample $\tilde{s}_t \sim \mathbb{Q}_t$ and output the ranked list $\tilde{\sigma}_t = \text{argsort}(\tilde{s}_t)$
(Effectively[3], it means $\tilde{\sigma}_t$ is drawn from $\mathbb{P}_t(\sigma) = (1-\gamma)I(\sigma = \sigma_t) + \frac{\gamma}{m!}$)
7: Receive relevance feedback on top-k items, i.e., $(R_t(\tilde{\sigma}_t(1)), \ldots, R_t(\tilde{\sigma}_t(k)))$
8: Suffer loss $\phi(\tilde{s}_t, R_t)$ (Neither loss nor R_t revealed to learner)
9: Construct $\tilde{z}_t$, an unbiased estimator of gradient $\nabla_{w=w_t}\phi(X_t w, R_t)$, from top-$k$ feedback.
10: Update $w = w_t - \eta\tilde{z}_t$
11: $w_{t+1} = \min\{1, \frac{U}{\|w\|_2}\}w$ (Projection onto Euclidean ball of radius U).
12: **End For**

Relation between feedback and structure of surrogates: Algorithm 2 is our general algorithm for learning a ranking function, online, from partial feedback. The key step in Algorithm 2 is the construction of the unbiased estimator $\tilde{z}_t$ of the surrogate gradient $\nabla_{w=w_t}\phi(X_t w, R_t)$. The information present for the construction process, at end of round t, is the random score vector $\tilde{s}_t$ (and associated permutation $\tilde{\sigma}_t$) and relevance of top-k items of $\tilde{\sigma}_t$, i.e., $\{R_t(\tilde{\sigma}_t(1)), \ldots, R_t(\tilde{\sigma}_t(k)\}$. Let $\mathbb{E}_t[\cdot]$ be the expectation operator w.r.t. to randomization at round t, conditioned on $(w_1, \ldots, w_t)$. Then $\tilde{z}_t$ being an unbiased estimator of gradient of surrogate, w.r.t w_t, means the following: $\mathbb{E}_t[\tilde{z}_t] = \nabla_{w=w_t}\phi(X_t w, R_t)$. We note that conditioned on the past, the score vector $s_t^{w_t} = X_t w_t$ is deterministic. We start with a general result relating feedback to the construction of unbiased estimator of a vector valued function. Let S_m be the set of $m!$ permutations of $[m]$. Let $\mathbb{P}$ denote a probability distribution on S_m, i.e, $\sum_{\sigma\in S_m}\mathbb{P}(\sigma) = 1$. For a distinct set of indices $(j_1, j_2, \ldots, j_k) \subseteq [m]$, we denote $p(j_i, j_2, \ldots, j_k)$ as the the sum of probability of permutations whose first k objects match objects $(j_1, \ldots, j_k)$, in order. Formally,

$$p(j_1, \ldots, j_k) = \sum_{\pi\in S_m}\mathbb{P}(\pi)I(\pi(1) = j_1, \ldots, \pi(k) = j_k). \tag{9}$$

We have the following lemma relating feedback and structure of surrogates:

[3] $\tilde{s}_t \sim s_t^{w_t} \implies \tilde{\sigma}_t \sim \sigma_t$; otherwise $\tilde{\sigma}_t$ is any other permutation.

Lemma 14. *Let $F : \mathbb{R}^m \mapsto \mathbb{R}^a$ be a vector valued function, where $m \geq 1$, $a \geq 1$. For a fixed $x \in \mathbb{R}^m$, let k entries of x be observed at random. That is, for a fixed probability distribution $\mathbb{P}$ and some random $\sigma \sim \mathbb{P}(S_m)$, observed tuple is $\{\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}\}$. A necessary condition for existence of an unbiased estimator of $F(x)$, that can be constructed from $\{\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}\}$, is that it should be possible to decompose $F(x)$ over k (or less) coordinates of x at a time. That is, $F(x)$ should have the structure:*

$$F(x) = \sum_{(i_1, i_2, \ldots, i_\ell) \in \ {}^mP_\ell} h_{i_1, i_2, \ldots, i_\ell}(x_{i_1}, x_{i_2}, \ldots, x_{i_\ell}) \tag{10}$$

where $\ell \leq k$, ${}^mP_\ell$ is ℓ permutations of m and $h : \mathbb{R}^\ell \mapsto \mathbb{R}^a$ (the subscripts in h are used to denote possibly different functions in the decomposition structure). Moreover, when $F(x)$ can be written in form of Eq 10 , with $\ell = k$, an unbiased estimator of $F(x)$, based on $\{\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}\}$, is,

$$g(\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}) = \frac{\sum_{(j_1, j_2, \ldots, j_k) \in S_k} h_{\sigma(j_1), \ldots, \sigma(j_k)}(x_{\sigma(j_1)}, \ldots, x_{\sigma(j_k)})}{\sum_{(j_1, \ldots, j_k) \in S_k} p(\sigma(j_1), \ldots, \sigma(j_k))} \tag{11}$$

where S_k is the set of $k!$ permutations of $[k]$ and $p(\sigma(1), \ldots, \sigma(k))$ is as in Eq 9 .

Illustrative Examples: We provide simple examples to concretely illustrate the abstract functions in Lemma 14. Let $F(\cdot)$ be the identity function, and $x \in \mathbb{R}^m$. Thus, $F(x) = x$ and the function decomposes over $k = 1$ coordinate of x as follows: $F(x) = \sum_{i=1}^m x_i e_i$, where $e_i \in \mathbb{R}^m$ is the standard basis vector along coordinate i. Hence, $h_i(x_i) = x_i e_i$. Based on top-1 feedback, following is an unbiased estimator of $F(x)$: $g(\sigma, x_{\sigma(1)}) = \frac{x_{\sigma(1)} e_{\sigma(1)}}{p(\sigma(1))}$, where $p(\sigma(1)) = \sum_{\pi \in S_m} \mathbb{P}(\pi) I(\pi(1) = \sigma(1))$. In another example, let $F : \mathbb{R}^3 \mapsto \mathbb{R}^2$ and $x \in \mathbb{R}^3$. Let $F(x) = [x_1 + x_2; x_2 + x_3]^\top$. Then the function decomposes over $k = 1$ coordinate of x as $F(x) = x_1 e_1 + x_2(e_1 + e_2) + x_3 e_2$, where $e_i \in \mathbb{R}^2$. Hence, $h_1(x_1) = x_1 e_1$, $h_2(x_2) = x_2(e_1 + e_2)$ and $h_3(x_3) = x_3 e_2$. An unbiased estimator based on top-1 feedback is: $g(\sigma, x_{\sigma(1)}) = \frac{h_{\sigma(1)}(x_{\sigma(1)})}{p(\sigma(1))}$.

3.2 Unbiased Estimators of Gradients of Surrogates

Algorithm 2 can be implemented for any ranking surrogate as long as an unbiased estimator of the gradient can be constructed from the random feedback. We will use techniques from *online convex optimization* to obtain formal regret guarantees. We will thus construct the unbiased estimator of four major ranking surrogates. Three of them are popular *convex* surrogates, one each from the three major learning to rank methods, i.e., *pointwise*, *pairwise* and *listwise* methods. The fourth one is a popular *non-convex* surrogate.

Shorthand notations: We note that by chain rule, $\nabla_{w=w_t} \phi(X_t w, R_t) = X_t^\top \nabla_{s_t^{w_t}} \phi(s_t^{w_t}, R_t)$, where $s_t^{w_t} = X_t w_t$. Since X_t is deterministic in our setting, we focus on unbiased estimators of $\nabla_{s_t^{w_t}} \phi(s_t^{w_t}, R_t)$ and take a matrix-vector product with X_t. To reduce notational clutter

in our derivations, we drop w from s^w and the subscript t throughout. Thus, in our derivations, $\tilde{z} = \tilde{z}_t$, $X = X_t$, $s = s_t^{w_t}$ (and not $\tilde{s}_t$), $\sigma = \tilde{\sigma}_t$ (and not σ_t), $R = R_t$, e_i is standard basis vector in $\mathbb{R}^m$ along coordinate i and $p(\cdot)$ as in Eq. 9 with $\mathbb{P} = \mathbb{P}_t$ where $\mathbb{P}_t$ is the distribution in round t in Algorithm 2.

3.2.1 Convex Surrogates

Pointwise Method: We will construct the unbiased estimator of the gradient of squared loss (Cossock and Zhang, 2006): $\phi_{sq}(s, R) = \|s - R\|_2^2$. The gradient $\nabla_s \phi_{sq}(s, R)$ is $2(s - R) \in \mathbb{R}^m$. As we have already demonstrated in the example following Lemma 14, *we can construct unbiased estimator of R from top-1 feedback* ($\{\sigma, R(\sigma(1))\}$). Concretely, the unbiased estimator is:

$$\tilde{z} = X^\top \left(2 \left(s - \frac{R(\sigma(1)) e_{\sigma(1)}}{p(\sigma(1))} \right) \right).$$

Pairwise Method: We will construct the unbiased estimator of the gradient of hinge-like surrogate in RankSVM (Joachims, 2002): $\phi_{svm}(s, R) = \sum_{i \neq j = 1} I(R(i) > R(j)) \max(0, 1 + s(j) - s(i))$. The gradient is given by:

$$\nabla_s \phi_{svm}(s, R) = \sum_{i=1}^m \sum_{j=1, j \neq i}^m I(R(i) > R(j)) I(1 + s(j) > s(i))(e_j - e_i) \in \mathbb{R}^m.$$

Since s is a known quantity, from Lemma 14, we can construct $F(R)$ as follows:

$$F(R) = F_s(R) = \sum_{i=1}^m \sum_{j=1, j \neq i}^m h_{s,i,j}(R(i), R(j)), \ \ h_{s,i,j}(R(i), R(j)) = I(R(i) > R(j)) I(1 + s(j) > s(i))(e_j - e_i).$$

Since $F_s(R)$ is decomposable over 2 coordinates of R at a time, *we can construct an unbiased estimator from top-2 feedback* ($\{\sigma, R(\sigma(1)), R(\sigma(2))\}$). The unbiased estimator is:

$$\tilde{z} = X^\top \left(\frac{h_{s,\sigma(1),\sigma(2)}(R(\sigma(1)), R(\sigma(2))) + h_{s,\sigma(2),\sigma(1)}(R(\sigma(2)), R(\sigma(1)))}{p(\sigma(1), \sigma(2)) + p(\sigma(2), \sigma(1))} \right).$$

We note that the unbiased estimator was constructed from top-2 feedback. The following lemma, in conjunction with the necessary condition of Lemma 14 shows that it is the minimum information required to construct the unbiased estimator.

Lemma 15. *The gradient of RankSVM surrogate, i.e., $\phi_{svm}(s, R)$ cannot be decomposed over 1 coordinate of R at a time.*

Listwise Method: Convex surrogates developed for listwise methods of learning to rank are defined over the entire score vector and relevance vector. Gradients of such surrogates cannot usually be decomposed over coordinates of the relevance vector. We will focus on the cross-entropy surrogate used in the highly cited ListNet (Cao et al., 2007) ranking algorithm and show how a very natural modification to the surrogate makes its gradient estimable in our partial feedback setting.

The authors of the ListNet method use a cross-entropy surrogate on two probability distributions on permutations, induced by score and relevance vector respectively. More

formally, the surrogate is defined as follows[4]. Define m maps from $\mathbb{R}^m$ to $\mathbb{R}$ as: $P_j(v) = \exp(v(j))/\sum_{j=1}^m \exp(v(j))$ for $j \in [m]$. Then, for score vector s and relevance vector R, $\phi_{\text{LN}}(s,R) = -\sum_{i=1}^m P_i(R)\log P_i(s)$ and $\nabla_s \phi_{\text{LN}}(s,R) = \sum_{i=1}^m \left(-\frac{\exp(R(i))}{\sum_{j=1}^m \exp(R(j))} + \frac{\exp(s(i))}{\sum_{j=1}^m \exp(s(j))}\right) e_i$. We have the following lemma about the gradient of ϕ_{LN}.

Lemma 16. *The gradient of ListNet surrogate $\phi_{LN}(s,R)$ cannot be decomposed over k, for $k = 1, 2$, coordinates of R at a time.*

In fact, an examination of the proof of the above lemma reveals that decomposability at any $k < m$ does not hold for the gradient of LisNet surrogate, though we only prove it for $k = 1, 2$ (since feedback for top k items with $k > 2$ does not seem practical). Due to Lemma 14, this means that if we want to run Alg. 2 under top-k feedback, a modification of ListNet is needed. We now make such a modification.

We first note that the cross-entropy surrogate of ListNet can be easily obtained from a standard divergence, viz. Kullback-Liebler divergence. Let $p, q \in \mathbb{R}^m$ be 2 probability distributions ($\sum_{i=1}^m p_i = \sum_{i=1}^m q_i = 1$). Then $KL(p,q) = \sum_{i=1}^m p_i \log(p_i) - \sum_{i=1}^m p_i \log(q_i) - \sum_{i=1}^m p_i + \sum_{i=1}^m q_i$. Taking $p_i = P_i(R)$ and $q_i = P_i(s)$, $\forall\, i \in [m]$ (where $P_i(v)$ is as defined in ϕ_{LN}) and noting that $\phi_{\text{LN}}(s,R)$ needs to be minimized w.r.t. s (thus we can ignore the $\sum_{i=1}^m p_i \log(p_i)$ term in $KL(p,q)$), we get the cross entropy surrogate from KL.

Our natural modification now easily follows by considering KL divergence for *un-normalized* vectors (it should be noted that KL divergence is an instance of a Bregman divergence). Define m maps from $\mathbb{R}^m$ to $\mathbb{R}$ as: $P'_j(v) = \exp(v(j))$ for $j \in [m]$. Now define $p_i = P'_i(R)$ and $q_i = P'_i(s)$. Then, the modified surrogate $\phi_{KL}(s,R)$ is:

$$\sum_{i=1}^m e^{R(i)} \log(e^{R(i)}) - \sum_{i=1}^m e^{R(i)} \log(e^{s(i)}) - \sum_{i=1}^m e^{R(i)} + \sum_{i=1}^m e^{s(i)},$$

and $\sum_{i=1}^m (\exp(s(i)) - \exp(R(i)))\, e_i$ is its gradient w.r.t. s. Note that $\phi_{KL}(s,R)$ is non-negative and convex in s. Equating gradient to $\mathbf{0} \in \mathbb{R}^m$, at the minimum point, $s(i) = R(i)$, $\forall\, i \in [m]$. Thus, the sorted order of optimal score vector agrees with sorted order of relevance vector and it is a valid ranking surrogate.

Now, from Lemma 14, we can construct $F(R)$ as follows: $F(R) = F_s(R) = \sum_{i=1}^m h_{s,i}(R(i))$, where $h_{s,i}(R(i)) = (\exp(s(i)) - \exp(R(i)))\, e_i$. Since $F_s(R)$ is decomposable over 1 coordinate of R at a time, *we can construct an unbiased estimator from top-1 feedback* ($\{\sigma, R(\sigma(1))\}$). The unbiased estimator is:

$$\tilde{\mathbf{z}} = X^\top \left(\frac{(\exp(s(\sigma(1))) - \exp(R(\sigma(1))))e_{\sigma(1)}}{p(\sigma(1))}\right)$$

Other Listwise Methods: As we mentioned before, most listwise convex surrogates will not be suitable for Algorithm 2 with top-k feedback. For example, the class of popular listwise surrogates that are developed from structured prediction perspective (Chapelle et al., 2007; Yue et al., 2007) cannot have unbiased estimator of gradients from top-k feedback since they are based on maps from full relevance vectors to full rankings and thus cannot be decomposed over $k = 1$ or 2 coordinates of R. It does not appear they have any natural modification to make them amenable to our approach.

[4]The ListNet paper actually defines a family of losses based on probability models for top r documents, with $r \leq m$. We use $r = 1$ in our definition since that is the version implemented in their experimental results.

3.2.2 Non-convex Surrogate

We provide an example of a non-convex surrogate for which Alg. 2 is applicable (however it will not have any regret guarantees due to non-convexity). We choose the SmoothDCG surrogate given in (Chapelle and Wu, 2010), which has been shown to have very competitive empirical performance. SmoothDCG, like ListNet, defines a family of surrogates, based on the cut-off point of DCG (see original paper (Chapelle and Wu, 2010) for details). We consider SmoothDCG@1, which is the smooth version of DCG@1 (i.e., DCG which focuses just on the top-ranked document). The surrogate is defined as: $\phi_{SD}(s,R) = \frac{1}{\sum_{j=1}^m \exp(s(j)/\epsilon)} \sum_{i=1}^m G(R(i)) \exp(s(i)/\epsilon)$, where ϵ is a (known) smoothing parameter and $G(a) = 2^a - 1$. The gradient of the surrogate is:

$$[\nabla_s \phi_{\text{SD}}(s,R)] = \sum_{i=1}^m h_{s,i}(R(i)),$$

$$h_{s,i}(R(i)) = G(R_i)\left(\sum_{j=1}^m \frac{1}{\epsilon}\left[\frac{\exp(s(i)/\epsilon)}{\sum_{j'} \exp(s(j')/\epsilon)} I_{(i=j)} - \frac{\exp((s(i)+s(j))/\epsilon)}{(\sum_{j'} \exp(s(j')/\epsilon))^2}\right] e_j\right)$$

Using Lemma 14, we can write $F(R) = F_s(R) = \sum_{i=1}^m h_{s,i}(R(i))$ where $h_{s,i}(R(i))$ is defined above. Since $F_s(R)$ is decomposable over 1 coordinate of R at a time, *we can construct an unbiased estimator from top-1 feedback* ($\{\sigma, R(\sigma(1))\}$), with unbiased estimator being:

$$s(\sigma(1))\tilde{z} = X^\top \left(\frac{G(R(\sigma(1)))}{p(\sigma(1))} \sum_{j=1}^m \frac{1}{\epsilon}\left[\frac{\exp(s(\sigma(1))/\epsilon)}{\sum_{j'} \exp(s(j')/\epsilon)} I_{(\sigma(1)=j)} - \frac{\exp((s(\sigma(1)) + s(j))/\epsilon)}{(\sum_{j'} \exp(s(j')/\epsilon))^2}\right] e_j\right)$$

3.3 Computational Complexity of Algorithm 2

Three of the four key steps governing the complexity of Algorithm 2, i.e., construction of $\tilde{s}_t$, $\tilde{\sigma}_t$ and sorting can all be done in $O(m \log(m))$ time.The only bottleneck could have been calculations of $p(\tilde{\sigma}_t(1))$ in squared loss, (modified) ListNet loss and SmoothDCG loss, and $p(\tilde{\sigma}_t(1), \tilde{\sigma}_t(2))$ in RankSVM loss, since they involve sum over permutations. However, they have a compact representation, i.e., $p(\tilde{\sigma}_t(1)) = (1 - \gamma + \frac{\gamma}{m}) I(\tilde{\sigma}_t(1) = \sigma_t(1)) + \frac{\gamma}{m} I(\tilde{\sigma}_t(1) \neq \sigma_t(1))$ and $p(\tilde{\sigma}_t(1), \tilde{\sigma}_t(2)) = (1 - \gamma + \frac{\gamma}{m(m-1)}) I(\tilde{\sigma}_t(1) = \sigma_t(1), \tilde{\sigma}_t(2) = \sigma_t(2)) + \frac{\gamma}{m(m-1)}[\sim I(\tilde{\sigma}_t(1) = \sigma_t(1), \tilde{\sigma}_t(2) = \sigma_t(2))]$. The calculations follow easily due to the nature of $\mathbb{P}_t$ (step-6 in algorithm) which put equal weights on all permutations other than σ_t.

3.4 Regret Bounds

The underlying deterministic part of our algorithm is online gradient descent (OGD) (Zinkevich, 2003). The regret of OGD, run with unbiased estimator of gradient of a **convex**

function, as given in Theorem 3.1 of (Flaxman et al., 2005), in our problem setting is:

$$\mathbb{E}\left[\sum_{t=1}^{T}\phi(X_t w_t, R_t)\right] \leq \min_{w:\|w\|_2\leq U}\sum_{t=1}^{T}\phi(X_t w, R_t) + \frac{U^2}{2\eta} + \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T}\|\tilde{z}_t\|_2^2\right] \tag{12}$$

where $\tilde{z}_t$ is unbiased estimator of $\nabla_{w=w_t}\phi(X_t w, R_t)$, conditioned on past events, η is the learning rate and the expectation is taken over all randomness in the algorithm.

However, from the perspective of the loss $\phi(\tilde{s}_t, R_t)$ incurred by Algorithm 2, at each round t, the RHS above is not a valid upper bound. The algorithms plays the score vector suggested by OGD ($\tilde{s}_t = X_t w_t$) with probability $1-\gamma$ (exploitation) and plays a randomly selected score vector (i.e., a draw from the uniform distribution on $[0,1]^m$), with probability γ (exploration). Thus, the expected number of rounds in which the algorithm does not follow the score suggested by OGD is γT, leading to an extra regret of order γT. Thus, we have [5]

$$\mathbb{E}\left[\sum_{t=1}^{T}\phi(\tilde{s}_t, R_t)\right] \leq \mathbb{E}\left[\sum_{t=1}^{T}\phi(X_t w_t, R_t)\right] + O\left(\gamma T\right) \tag{13}$$

We first control $\mathbb{E}_t\|\tilde{z}_t\|_2^2$, for all convex surrogates considered in our problem (we remind that $\tilde{z}_t$ is the estimator of a gradient of a surrogate, calculated at time t. In Sec 3.2.1 , we omitted showing w in s^w and index t). To get bound on $\mathbb{E}_t\|\tilde{z}_t\|_2^2$, we used the following norm relation that holds for any matrix X (Bhaskara and Vijayaraghavan, 2011): $\|X\|_{p\to q} = \sup_{v\neq 0}\frac{\|Xv\|_q}{\|v\|_p}$, where q is the dual exponent of p (i.e., $\frac{1}{q}+\frac{1}{p}=1$), and the following lemma derived from it:

Lemma 17. *For any $1 \leq p \leq \infty$, $\|X^\top\|_{1\to p} = \|X\|_{q\to\infty} = \max_{j=1}^{m}\|X_{j:}\|_p$, where $X_{j:}$ denotes jth row of X and m is the number of rows of matrix.*

We have the following result:

Lemma 18. *For parameter γ in Algorithm 2 , R_D being the bound on ℓ_2 norm of the feature vectors (rows of document matrix X), m being the upper bound on number of documents per query, U being the radius of the Euclidean ball denoting the space of ranking parameters and $R_{\max}$ being the maximum possible relevance value (in practice always ≤ 5), let $C^\phi \in \{C^{sq}, C^{svm}, C^{KL}\}$ be polynomial functions of R_D, m, U, R_{max}, where the degrees of the polynomials depend on the surrogate ($\phi_{sq}, \phi_{svm}, \phi_{KL}$), with no degree ever greater than four. Then we have,*

$$\mathbb{E}_t\left[\|\tilde{z}_t\|_2^2\right] \leq \frac{C^\phi}{\gamma} \tag{14}$$

Plugging Eq. 14 and Eq. 13 in Eq. 12, and optimizing over η and γ, (which gives $\eta = O(T^{-2/3})$ and $\gamma = O(T^{-1/3})$), we get the final regret bound:

[5] The instantaneous loss suffered at each of the exploration round can be maximum of $O(1)$, as long as $\phi(s,R)$ is bounded, $\forall\ s$ and $\forall\ R$. This is true because the score space is ℓ_2 norm bounded, maximum relevance grade is finite in practice and we consider Lipschitz, convex surrogates.

Theorem 19. *For any sequence of instances and labels $(X_t, R_t)_{\{t\in[T]\}}$, applying Algorithm 2 with top-1 feedback for ϕ_{sq} and ϕ_{KL} and top-2 feedback for ϕ_{svm}, will produce the following bound:*

$$\mathbb{E}\left[\sum_{t=1}^{T}\phi(\tilde{s}_t, R_t)\right] - \min_{w:\|w\|_2\leq U}\sum_{t=1}^{T}\phi(X_t w, R_t) \leq C^\phi O\left(T^{2/3}\right) \tag{15}$$

where C^ϕ is a surrogate dependent function, as described in Lemma 18 , and expectation is taken over underlying randomness of the algorithm, over T rounds.

3.5 Impossibility of Sublinear Regret for NDCG Calibrated Surrogates

Learning to rank methods optimize surrogates to learn a ranking function, even though performance is measured by target measures like NDCG. This is done because direct optimization of the measures lead to NP-hard optimization problems. One of the most desirable properties of any surrogate is *calibration*, i.e., the surrogate should be calibrated w.r.t the target (Bartlett et al., 2006). Intuitively, it means that a function with small expected surrogate loss on unseen data should have small expected target loss on unseen data. We focus on NDCG calibrated surrogates (both convex and non-convex) that have been characterized by Ravikumar et al. (2011). We first state the necessary and sufficient condition for a surrogate to be calibrated w.r.t NDCG. For any score vector s and distribution η on relevance space $\mathcal{Y}$, let $\bar{\phi}(s,\eta) = \mathbb{E}_{R\sim\eta}\phi(s,R)$. Moreover, we define $G(\mathbf{R}) = (G(R_1),\ldots,G(R_m))^\top$. $Z(R)$ is defined in Sec 2.2.

Theorem 20. *(Ravikumar et al., 2011, Thm. 6) A surrogate ϕ is NDCG calibrated iff for any distribution η on relevance space $\mathcal{Y}$, there exists an invertible, order preserving map $g:\mathbb{R}^m \mapsto \mathbb{R}^m$ s.t. the unique minimizer $s^*_\phi(\eta)$ can be written as*

$$s^*_\phi(\eta) = g\left(\mathbb{E}_{R\sim\eta}\left[\frac{G(\mathbf{R})}{Z(R)}\right]\right). \tag{16}$$

Informally, Eq. 16 states that $\text{argsort}(s^*_\phi(\eta)) \subseteq \text{argsort}(\mathbb{E}_{R\sim\eta}\left[\frac{G(\mathbf{R})}{Z(R)}\right])$. Ravikumar et al. (2011) give concrete examples of NDCG calibrated surrogates, including how some of the popular surrogates can be converted into NDCG calibrated ones: e.g., the NDCG calibrated version of squared loss is $\|s - \frac{G(\mathbf{R})}{Z(R)}\|_2^2$.

We now state the *impossibility* result for the class of NDCG calibrated surrogates when feedback is restricted to top ranked item.

Theorem 21. *Fix the online learning to rank game with top 1 feedback and any NDCG calibrated surrogate. Then, for every learner's algorithm, there exists an adversary strategy such that the learner's expected regret is $\Omega(T)$.*

We note that the proof of Theorem. 3 of Piccolboni and Schindelhauer (2001) cannot be directly extended to prove the impossibility result because it relies on constructing a connected graph on vertices defined by neighboring actions of learner. In our case, due to the continuous nature of learner's actions, the graph will be an empty graph and proof will break down.

4. Experiments

We conducted experiments on simulated and commercial data sets to demonstrate the performance of our algorithms.

4.1 Non Contextual Setting

Objectives: We had the following objectives while conducting experiments in the non-contextual, online ranking with partial feedback setting:

- Investigate how performance of Algorithm 1 is affected by size of blocks during blocking.
- Investigate how performance of the algorithm is affected by amount of feedback received (i.e., generalizing k in top k feedback).
- Demonstrate the difference between regret rate of our algorithm, which operates in partial feedback setting, with regret rate of a full information algorithm which receives full relevance vector feedback at end of each round.

We applied Algorithm 1 in conjunction with Follow-The-Perturbed-Leader (FTPL) full information algorithm, as described in Sec. 2.6. Note that the online learning to rank problem is not as well studied as the batch problem. This is especially true for the partial feedback setting we have. Therefore benchmark data sets with published baseline performance measures, are not available for comparison, to the best of our knowledge. The generic partial monitoring algorithms that do exist cannot be applied due to computational inefficiency (Sec. 2.5).

Experimental Setting: All our experiments were conducted with respect to the DCG measure, which is quite popular in practice, and binary graded relevance vectors. Our experiments were conducted on the following simulated data set. We fixed number of items to 20 ($m = 20$). We then fixed a "true" relevance vector which had 5 items with relevance level 1 and 15 items with relevance level 0. We then created a total of T=10000 relevance vectors by corrupting the true relevance vector. The corrupted copies were created by independently flipping each relevance level (0 to 1 and vice-versa) with a small probability. The reason for creating adversarial relevance vectors in such a way was to reflect diversity of preferences in practice. In reality, it is likely that most users will have certain similarity in preferences, with small deviations on certain items and certain users. That is, some items are likely to be relevant in general, with most items not relevant to majority of users, with slight deviation from user to user. Moreover, the comparator term in our regret bound (i.e., cumulative loss/gain of the true best ranking in hindsight) makes practical sense if there is a ranking which satisfies most users.

Results: We plotted average regret over time under DCG. Average regret over time means cumulative regret up to time t, divided by t, for $1 \leq t \leq T$. Figure 1 demonstrates the effect of block size on the regret rate under DCG. We fixed feedback to relevance of top ranked item ($k = 1$). As can be seen in Corollary 10, the optimal regret rate is achieved by optimizing over number of blocks (hence block size), which requires prior knowledge of

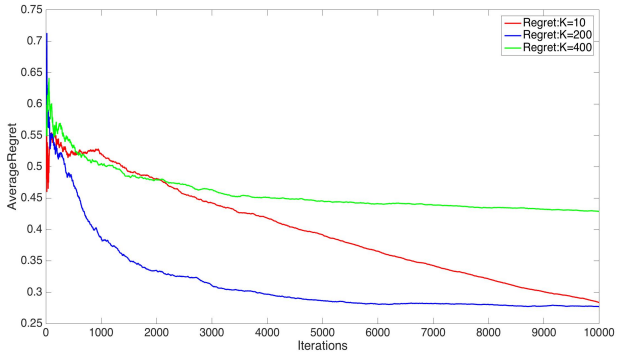


Figure 1: Average regret under DCG, with feedback on top ranked object, for varying block size, where block size is $\lceil T/K \rceil$. *Best viewed in color.*

time horizon T. We wanted to demonstrate how the regret rate (and hence the performance of the algorithm) differs with different block sizes. The optimal number of blocks in our setting is $K \sim 200$, with corresponding block size being $\lceil T/K \rceil = 50$. As can be clearly seen, with optimal block size, the regret drops fastest and becomes steady after a point. $K = 10$ means that block size is 1000. This means over the time horizon, number of exploitation rounds greatly dominates number of exploration rounds, leading to regret dropping at a slower rate initially than than the case with optimal block size. However, the regret drops of pretty sharply later on. This is because the relevance vectors are slightly corrupted copies of a "true" relevance vector and the algorithm gets a good estimate of the true relevance vector quickly and then more exploitation helps. When $K = 400$ (i.e., block size is 25), most of the time, the algorithm is exploring, leading to a substantially worse regret and poor performance.

Figure 2 demonstrates the effect of amount of feedback on the regret rate under DCG. We fixed $K = 200$, and varied feedback as relevance of top k ranked items per round, where $k = 1, 5, 10$. Validating our regret bound, we see that as k increases, the regret decreases.

Figure 3 compares regret of our algorithm, working with top 1 feedback, and FTPL full information algorithm, working with full relevance vector feedback at end of each round. We fixed $K = 200$ and the comparison was done from 1000 iterations onwards, i.e., roughly after the initial learning phase. FTPL full information algorithm has regret rate of $O(T^{1/2})$ (ignoring other parameters). So, as expected, FTPL with full information feedback outperforms our algorithm with highly restricted feedback; yet, we have demonstrated, both theoretically and empirically, that it is possible to have a good ranking strategy with highly restricted feedback.

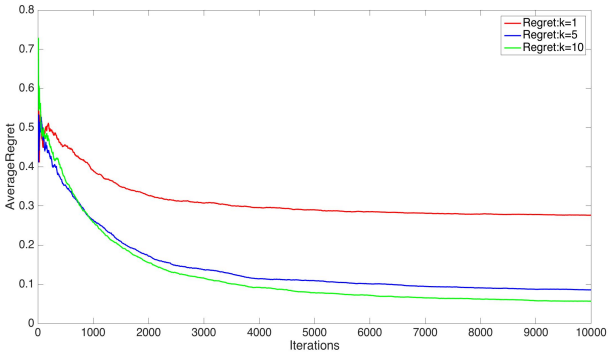


Figure 2: Average regret under DCG, where $K = 200$, for varying amount of feedback. *Best viewed in color.*

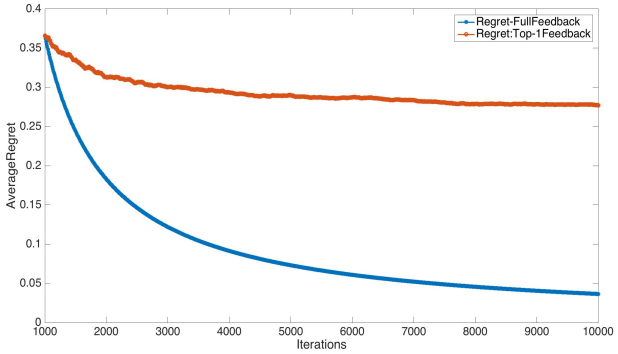


Figure 3: Comparison of average regret over time, for DCG, between top-1 feedback and full relevance vector feedback. *Best viewed in color.*

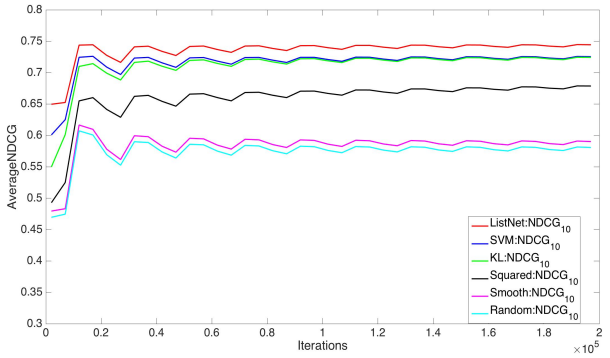


Figure 4: Average NDCG_{10} values of different algorithms, for Yahoo data set. ListNet:NDCG_{10} (in cyan) operates on full feedback and Random:NDCG_{10} (in red) does not receive any feedback. *Best viewed in color* .

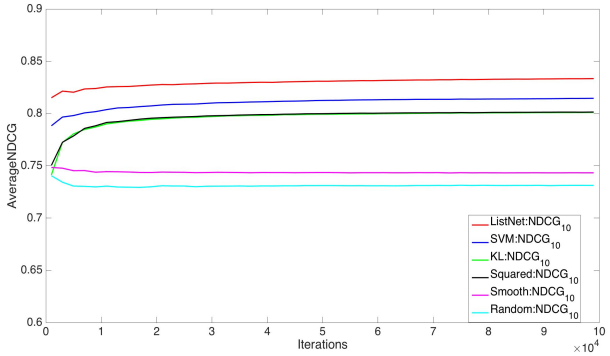


Figure 5: Average NDCG_{10} values of different algorithms, for Yandex data set. ListNet:NDCG_{10} (in cyan) operates on full feedback and Random:NDCG_{10} (in red) does not receive any feedback. *Best viewed in color.*

4.2 Contextual Setting

Objective: Since our contextual, online learning to rank with restricted feedback setting involves query-document matrices, we could conduct experiments on commercial, publicly available ranking data sets. Our objective was to demonstrate that it is possible to learn a good ranking function, even with highly restricted feedback, when standard online learning to rank algorithms would require full feedback at end of each round. As stated before, though our algorithm is designed to minimize surrogate based regret, the surrogate loss is only a proxy for ranking measures like NDCG. The users care about the ranking presented to them, and indeed the algorithm interacts with users by presenting ranked lists and getting feedback on top ranked item(s). We tested the quality of the ranked lists, and hence the performance of the evolving ranking functions, against the full relevance vectors, via ranking measure NDCG, cutoff at the 10th item. NDCG, cutoff at a point n, is defined as follows:

$$NDCG_n(\sigma, R) = \frac{1}{Z_n(R)} \sum_{i=1}^{n} \frac{2^{R(\sigma(i))} - 1}{\log_2(1+i)}$$

where $Z_n(R) = \max_{\sigma \in S_m} \sum_{i=1}^{n} \frac{2^{R(\sigma(i))}-1}{\log_2(1+i)}$. We want to emphasize that the algorithm was in no way affected by the fact that we were measuring its performance with respect to NDCG cutoff at 10. In fact, the cutoff point can be varied, but usually, researchers report performance under NDCG cutoff at 5 or 10.

Baselines: We applied Algorithm 2, with top 1 feedback, on Squared, KL (un-normalized ListNet) and SmoothDCG surrogates, and with top 2 feedback, on the RankSVM surrogate. Based on the objective of our work, we selected two different ranking algorithms as baselines. The first one is the online version of ListNet ranking algorithm (which is essentially OGD on cross-entropy function), with full relevance vector revealed at end of every round. ListNet is not only one of the most cited ranking algorithms (over 700 citations according to Google Scholar), but also one of the most validated algorithms (Tax et al., 2015). We emphasize that some of the ranking algorithms in literature, which have shown better empirical performance than ListNet, are based on non-convex surrogates with complex, non-linear ranking functions. These algorithms cannot usually be converted into online algorithms which learn from streaming data. Our second algorithm is a simple, fully random algorithm , which outputs completely random ranking of documents at each round. This algorithm, in effect, receives no feedback at end of each round. *Thus, we are comparing Algorithm 2, which learns from highly restricted feedback, with an algorithm which learns from full feedback and an algorithm which receives no feedback and learns nothing.*

Data sets: We compared the various ranking functions on two large scale commercial data sets. They were Yahoo's Learning to Rank Challenge data set (Chapelle and Chang, 2011) and a data set published by Russian search engine Yandex (IM-2009). The Yahoo data set had 19944 unique queries with 5 distinct relevance levels, while Yandex had 9126 unique queries with 5 distinct relevance levels.

Experimental Setting: We selected time horizon $T = 200,000$ (Yahoo) and $T = 100,000$ (Yandex) iterations for our experiments (thus, each algorithm went over each data set multiple times). The reason for choosing different time horizons is that there were roughly double the number of queries in Yahoo data set as compared to Yandex data set. All the online algorithms, other than the fully random one, involve learning rate η and exploration parameter γ (full information ListNet does not involve γ and SmoothDCG has an additional smoothing parameter ϵ). While obtaining our regret guarantees, we had established that $\eta = O(T^{-2/3})$ and $\gamma = O(T^{-1/3})$. In our experiments, for each instance of Algorithm 2, we selected a time varying $\eta = \frac{0.01}{t^{2/3}}$ and $\gamma = \frac{0.1}{t^{1/3}}$, for round t . We fixed $\epsilon = 0.01$. For ListNet, we selected $\eta = \frac{0.01}{t^{1/2}}$, since regret guarantee in OGD is established with $\eta = O(T^{-1/2})$. It should be noted that, while we varied the numerators in the parameters between $\{0.001, 0.01, 0.1, 1, 10\}$, and selected the parameters which produced the best (avg.) NDCG value over the entire time horizon, more emphasis could have been given on the issue of parameter tuning. However, better tuning of parameters could have only improved the performance of our algorithm. We plotted average NDCG_{10} against time, where average NDCG_{10} at time t is the cumulative NDCG_{10} up to time t, divided by t. We made an important observation while comparing the performance plots of the algorithms. As we have shown, construction of the unbiased estimators involve division by a probability value (Eq 11). The particular probability value can be $\frac{\gamma}{m}$, which is very small since γ goes to 0, when the top ranked item of the randomly drawn permutation does not match the top ranked item of the permutation given by the deterministic score vector (Sec 3.3). The mismatch happens with very low probability (since the random permutation is actually the deterministic permutation with high probability). While theoretically useful, in practice, dividing by such small value negatively affected the gradient estimation and hence the performance of our algorithm. So, when the mismatch happened, we made the denominator γ, instead of $\frac{\gamma}{m}$, and saw signficantly better performance.

Results: Figure 4 and Figure 5 show that ListNet, with full information feedback at end of each round, has highest average NDCG value throughout, as expected. However, Algorithm 2, with the convex surrogates, produce competitive performance. In fact, in the Yahoo data set, our algorithm, with RankSVM and KL, are very close to the performance of ListNet. RanSVM based algorithm does better than the others, since the estimator of RankSVM gradient is constructed from top 2 feedback, leading to lower variance of the estimator. KL based algorithm does much better than Squared loss based algorithm on Yahoo and equally as well on Yandex data set. *Crucially, our algorithm, based on all three convex surrogates, perform significantly better than the purely random algorithm, and are much closer to ListNet in performance, despite being much closer to the purely random algorithm in terms of feedback.* Our algorithm, with SmoothDCG, on the other hand, produce poor performance. We believe the reason is the non-convexity of the surrogate, which leads to the optimization procedure possibly getting stuck at a local minima. In batch setting, such problem is avoided by an annealing technique that successively reduces ϵ. We are not aware of an analogue in an online setting. Possible algorithms optimizing non-convex surrogates in an online manner, which require gradient of the surrogate, may be adapted to this partial feedback setting. The main purpose for including SmoothDCG in our work was to show that unbiased estimation of gradient, from restricted feedback, is possible even for non-convex surrogates.

5. Conclusion and Future Directions

We studied the problem of online learning to rank with a novel, restricted feedback model. The work is divided into two parts: in the first part, the set of items to be ranked is fixed, with varying user preferences, and in the second part, the items vary, as traditional query-documents matrices. The parts are tied by the feedback model; where the user gives feedback only on top k ranked items at end of each round, though the performance of the learner's ranking strategy is judged against full, *implicit* relevance vectors. In the first part, we gave comprehensive results on learnability with respect to a number of practically important ranking measures. We also gave a generic algorithm, along with an efficient instantiation, which achieves sub-linear regret rate for certain ranking measures. In the second part, we gave an efficient algorithm, which works on a number of popular ranking surrogates, to achieve sub-linear regret rate. We also gave an impossibility result for an entire class of ranking surrogates. Finally, we conducted experiments on simulated and commercial ranking data sets to demonstrate the performance of our algorithms.

We highlight some of the open questions of interest:

- What are the minimax regret rates with top k feedback model, for $k > 1$, for the ranking measures DCG, PairwiseLoss, Precision@n and their normalized versions NDCG, AUC and AP? Specifically, NDCG and AP are very popular in the learning to rank community. We showed that with top 1 feedback model, no algorithm can achieve sub-linear regret for NDCG and AP. Is it possible to get sub-linear regret with $1 < k < m$?

- We used FTPL as the sub-routine in Algorithm 1 to get an efficient algorithm. It might be possible to use other full information algorithms as sub-routine, retaining the efficiency, but getting tighter rates in terms of parameters (other than T) and better empirical performance.

- We applied Algorithm 2 on three convex surrogates and one non-convex surrogates. It would be interesting to investigate what other surrogates the algorithm can be applied on, guided by Lemma 14, and test its empirical performance. Since the algorithm learns a ranking function in the traditional query-documents setting, the question is more of practical interest.

- We saw that Algorithm 2, when applied to SmoothDCG, does not produce competitive empirical performance. It has been shown that a ranking function, learnt by optimizing SmoothDCG in the batch setting, has extremely competitive empirical performance (Qin and Liu, 2006). In the batch setting, simulated annealing is used to prevent the optimization procedure getting stuck in local minima. Any algorithm that optimizes non-convex surrogates in an online manner, by accessing its gradient, can replace the online gradient descent part in our algorithm and tested on SmoothDCG for empirical performance.

- We proved an impossibility result for NDCG calibrated surrogates with top 1 feedback. What is the minimax regret for NDCG calibrated surrogates, with top k feedback, for $k > 1$?

Acknowledgments

The authors acknowledge the support of NSF under grants IIS-1319810 and CAREER IIS-1452099.

Appendix A. Proofs for Non Contextual Setting

We provide technical details of results of Online Ranking with Restricted Feedback- Non Contextual Setting.

Proof of Theorem 8:

Proof. **Full information feedback**: Instead of top k feedback, assume that at end of each round, after learner reveals its action, the full relevance vector R is revealed to the learner. Since the knowledge of full relevance vector allows the learner to calculate the loss for every action (SumLoss(σ, R), $\forall\ \sigma$), the game is in full information setting, and the learner, using the full information algorithm, will have an $O(C\sqrt{T})$ expected regret for $SumLoss$ (ignoring computational complexity). Here, C denotes parameter specific to the full information algorithm used.

Blocking with full information feedback: We consider a blocked variant of the full information algorithm. We still assume that full relevance vector is revealed at end of each round. Let the time horizon T be divided into K blocks, i.e., $\{B_1, \ldots, B_K\}$, of equal size. Here, $B_i = \{(i-1)(T/K)+1, (i-1)(T/K)+2, (i-1)T/K+3, \ldots, i(T/K)\}$. While operating in a block, the relevance vectors revealed at end of each round are accumulated, but not used to generate learner's actions like in the "without blocking" variant. Assume at the start of block B_i, there was some vector $s_{i-1} \in \mathbb{R}^m$. Then, at each round in the block, the randomized full information algorithm exploits s_{i-1} and outputs a permutation (basically maintains a distribution over actions, using s_{i-1}, and samples from the distribution). At the end of a block, the average of the accumulated relevance vectors (R_i^{avg}) for the block is used to update, as $s_{i-1} + R_i^{avg}$, to get s_i for the next block. The process is repeated for each block.

Formally, the full information algorithm creates distribution ρ_i over the actions, at beginning of block B_i, exploiting information s_{i-1}. Thus, $\rho_i \in \Delta$, where Δ is the probability simplex over $m!$ actions. Note that ρ_i is a deterministic function of $\{R_1^{avg}, \ldots, R_{i-1}^{avg}\}$.

Since action σ_t, for $t \in B_i$, is generated according to distribution ρ_i (we will denote this as $\sigma_t \sim \rho_i$), and in block i, distribution ρ_i is fixed, we have

$$\mathbb{E}_{\sigma_t \sim \rho_i}[\sum_{t \in [B_i]} SumLoss(\sigma_t, R_t)] = \sum_{t \in B_i} \rho_i \cdot [SumLoss(\sigma_1, R_t), \ldots, SumLoss(\sigma_{m!}, R_t)].$$

Thus, the total expected loss of this variant of the full information problem is:

$$\begin{aligned}
\mathbb{E}\sum_{t=1}^{T}[SumLoss(\sigma_t, R_t)] &= \sum_{i=1}^{K} \mathbb{E}_{\sigma_t \sim \rho_i}[\sum_{t \in B_i} SumLoss(\sigma_t, R_t)] \\
&= \sum_{i=1}^{K} \sum_{t \in B_i} \rho_i \cdot [SumLoss(\sigma_1, R_t), \ldots, SumLoss(\sigma_{m!}, R_t)] \\
&= \sum_{i=1}^{K} \sum_{t \in B_i} \rho_i \cdot [\sigma_1^{-1} \cdot R_t, \ldots, \sigma_{m!}^{-1} \cdot R_t)] \quad \text{By defn. of SumLoss} \\
&= \frac{T}{K} \sum_{i=1}^{K} \rho_i \cdot [\sigma_1^{-1} \cdot R_i^{avg}, \ldots, \sigma_{m!}^{-1} \cdot R_i^{avg}]
\end{aligned}$$

$$\begin{aligned}
&= \frac{T}{K} \sum_{i=1}^{K} \mathbb{E}_{\sigma_i \sim \rho_i}[SumLoss(\sigma_i, R_i^{avg})] \\
&= \frac{T}{K} \mathbb{E}_{\sigma_1 \sim \rho_1, \ldots, \sigma_K \sim \rho_K} \sum_{i=1}^{K} SumLoss(\sigma_i, R_i^{avg}) \qquad (17)
\end{aligned}$$

where $R_i^{avg} = \sum_{t \in B_i} \frac{R_t}{T/K}$. Note that, at end of every block $i \in [K]$, ρ_i is updated to ρ_{i+1}. By the regret bound of the full information algorithm, for K rounds of full information problem, we have:

$$\begin{aligned}
\mathbb{E}_{\sigma_1 \sim \rho_1, \ldots, \sigma_K \sim \rho_K} \sum_{i=1}^{K} SumLoss(\sigma_i, R_i^{avg}) &\leq \min_{\sigma} \sum_{i=1}^{K} SumLoss(\sigma, R_i^{avg}) + C\sqrt{K} \\
&= \min_{\sigma} \sum_{i=1}^{K} \sigma^{-1} \cdot R_i^{avg} + C\sqrt{K} \qquad (18) \\
&= \min_{\sigma} \sum_{t=1}^{T} \sigma^{-1} \cdot \frac{R_t}{T/K} + C\sqrt{K}
\end{aligned}$$

Now, since

$$\min_{\sigma} \sum_{t=1}^{T} \sigma^{-1} \cdot \frac{R_t}{T/K} = \min_{\sigma} \frac{1}{T/K} \sum_{t=1}^{T} SumLoss(\sigma, R_t),$$

combining Eq. 17 and Eq. 18, we get:

$$\sum_{t=1}^{T} \mathbb{E}_{\sigma_t \in \rho_i}[SumLoss(\sigma_t, R_t)] \leq \min_{\sigma} \sum_{t=1}^{T} SumLoss(\sigma, R_t) + C\frac{T}{\sqrt{K}}. \qquad (19)$$

Blocking with top k feedback: However, in our top k feedback model, the learner does not get to see the full relevance vector at each end of round. Thus, we form the unbiased estimator $\hat{R}_i$ of R_i^{avg}, using Lemma 7. That is, at start of each block, we choose $\lceil m/k \rceil$ time points uniformly at random, and at those time points, we output a random permutation which places k distinct objects on top (refer to Algorithm 1). At the end of the block, we form the vector $\hat{R}_i$ which is the unbiased estimator of R_i^{avg}. Note that using random vector $\hat{R}_i$ instead of true R_i^{avg} introduces randomness in the distribution ρ_i itself . But significantly, ρ_i is dependent only on information received up to the beginning of block i and is independent of the information collected in the block. We show the exclusive dependence as $\rho_i(\hat{R}_1, \hat{R}_2, .., \hat{R}_{i-1})$. Thus, for block i, we have:

$$\begin{aligned}
&\mathbb{E}_{\sigma_t \sim \rho_i(\hat{R}_1, \hat{R}_2, .., \hat{R}_{i-1})} \sum_{t \in [B_i]} SumLoss(\sigma_t, R_t) \\
&= \frac{T}{K} \mathbb{E}_{\sigma_i \sim \rho_i(\hat{R}_1, \hat{R}_2, .., \hat{R}_{i-1})} SumLoss(\sigma_i, R_i^{avg}) \\
&\quad \text{(From Eq. 17)}
\end{aligned}$$

$$= \frac{T}{K}\mathbb{E}_{\sigma_i\sim\rho_i(\hat{R}_1,\hat{R}_2,..,\hat{R}_{i-1})}\mathbb{E}_{\hat{R}_i}SumLoss(\sigma_i,\hat{R}_i)$$
$$(\because \text{SumLoss is linear in both arguments and } \hat{R}_i \text{ is unbiased})$$
$$= \frac{T}{K}\mathbb{E}_{\hat{R}_i}\mathbb{E}_{\sigma_i\sim\rho_i(\hat{R}_1,\hat{R}_2,..,\hat{R}_{i-1})}SumLoss(\sigma_i,\hat{R}_i).$$

In the last step above, we crucially used the fact that, since random distribution ρ_i is independent of $\hat{R}_i$, the order of expectations is interchangeable. Taking expectation w.r.t. $\hat{R}_1, \hat{R}_2, .., \hat{R}_{i-1}$, we get,

$$\mathbb{E}_{\hat{R}_1,..,\hat{R}_{i-1}}\mathbb{E}_{\sigma_t\sim\rho_i(\hat{R}_1,\hat{R}_2,..,\hat{R}_{i-1})}\sum_{t\in[B_i]}SumLoss(\sigma_t,R_t) = \frac{T}{K}\mathbb{E}_{\hat{R}_1,..,\hat{R}_{i-1},\hat{R}_i}E_{\sigma_i\sim\rho_i(\hat{R}_1,\hat{R}_2,..,\hat{R}_{i-1})}SumLoss(\sigma_i,\hat{R}_i). \tag{20}$$

Thus,

$$\begin{aligned}
\mathbb{E}\sum_{t=1}^{T}SumLoss(\sigma_t,R_t) &= \mathbb{E}\sum_{i=1}^{K}\sum_{t\in[B_i]}SumLoss(\sigma_t,R_t)\\
&= \sum_{i=1}^{K}E_{\hat{R}_1,..,\hat{R}_{i-1}}E_{\sigma_t\sim\rho_i(\hat{R}_1,\hat{R}_2,..,\hat{R}_{i-1})}\sum_{t\in[B_i]}SumLoss(\sigma_t,R_t)\\
&= \frac{T}{K}\sum_{i=1}^{K}\mathbb{E}_{\hat{R}_1,..,\hat{R}_{i-1},\hat{R}_i}\mathbb{E}_{\sigma_i\sim\rho_i(\hat{R}_1,\hat{R}_2,..,\hat{R}_{i-1})}SumLoss(\sigma_i,\hat{R}_i)\\
&\quad\text{(From Eq. 20)}\\
&= \frac{T}{K}\mathbb{E}_{\hat{R}_1,..,\hat{R}_K}\mathbb{E}_{\sigma_i\sim\rho_i(\hat{R}_1,\hat{R}_2,..,\hat{R}_{i-1})}\sum_{i=1}^{K}SumLoss(\sigma_i,\hat{R}_i)
\end{aligned}$$

Now using Eq. 18, we can upper bound the last term above as

$$\begin{aligned}
&\le \frac{T}{K}\{\mathbb{E}_{\hat{R}_1,..,\hat{R}_K}[\min_{\sigma}\sum_{i=1}^{K}\sigma^{-1}\cdot\hat{R}_i] + C\sqrt{K}\}\\
&\le \frac{T}{K}\{\min_{\sigma}\sum_{i=1}^{K}\sigma^{-1}\cdot R_i^{avg} + C\sqrt{K}\}\\
&\quad\text{(Jensen's Inequality)}\\
&\le \min_{\sigma}\sum_{t=1}^{T}\sigma^{-1}\cdot R_t + C\frac{T}{\sqrt{K}}\\
&= \min_{\sigma}\sum_{t=1}^{T}SumLoss(\sigma,R_t) + C\frac{T}{\sqrt{K}}.
\end{aligned}$$

Effect of exploration: Since in each block B_i, $\lceil m/k\rceil$ rounds are reserved for exploration, where we do not draw σ_t from distribution ρ_i, we need to account for it in our

regret bound. Exploration leads to an extra regret of $C^I\lceil m/k\rceil K$, where C^I is a constant depending on the loss under consideration and specific full information algorithm used. The extra regret is because loss in each of the exploration rounds is at most C^I and there are a total of $\lceil m/k\rceil K$ exploration rounds over all K blocks. Thus, overall regret :

$$\mathbb{E}\left[\sum_{t=1}^{T}SumLoss(\sigma_t,R_t)\right] - \min_{\sigma}\sum_{t=1}^{T}SumLoss(\sigma,R_t) \le C^I\lceil m/k\rceil K + C\frac{T}{\sqrt{K}}. \tag{21}$$

Now we optimize over K, to get:

$$\mathbb{E}\left[\sum_{t=1}^{T}SumLoss(\sigma_t,R_t)\right] \le \min_{\sigma}\sum_{t=1}^{T}SumLoss(\sigma,R_t) + 2(C^I)^{1/3}C^{2/3}\lceil m/k\rceil^{1/3}T^{2/3} \tag{22}$$

□

Proof of Corollary 9:

Proof. We only need to instantiate the constants C and C^I from Theorem 8, with respect to SumLoss and FTPL. FTPL has the following parameters in its regret bound, for any online full information linear optimization problem: D is the ℓ_1 diameter of learner's action set, R is upper bound on difference between losses of 2 actions on same information vector and A is the ℓ_1 diameter of the set of information vectors (adversary's action set).

For SumLoss, it can be easily calculated that $R = \sum_{i=1}^{m}\sigma^{-1}(i)R(i) = O(m^2)$, $D = \sum_{i=1}^{m}\sigma^{-1}(i) = O(m^2)$, and $A = \sum_{i=1}^{m}R(i) = O(m)$.

FTPL gets $O(C\sqrt{T})$ regret over T rounds when $\epsilon = \sqrt{\frac{D}{RAT}}$. Here, $C = 2\sqrt{DRA} = O(m^{5/2})$ and $C^I = R$. Substituting the values of D, R, A, we conclude. □

In the proof above, we used the standard FTPL bound from the work of Kalai and Vempala (2005). In particular, we used uniform noise distribution drawn from a hypercube. We could have used other distributions, e.g. Gaussian. For Gaussian perturbations, Theorem 1.7 of Abernethy et al. (2016) gives an FTPL regret bound of $B_1B_2m^{1/4}\sqrt{2T}$ where B_1, B_2 are upper bounds on the maximum ℓ_1 norm of vectors in the learner's and adversary's sets respectively. For us, B_1 is $O(m^{3/2})$ and B_2 is $O(m^{1/2})$ giving a bound of $O(m^{2.25}\sqrt{T})$ which is slightly better than the $O(m^{2.5}\sqrt{T})$ bound used in the proof above.

Extension of results from SumLoss to DCG and Precision@n:

DCG: Due to structural differences, there are minor differences in definitions and proofs of theorems for SumLoss and DCG. We prove that local observability condition fails to hold for DCG, when restricted to top 1 feedback. We can skip the explicit proof of global observability, since the application of Algorithm 1 already establishes that $O(T^{2/3})$ regret can be achieved.

With slight abuse of notations, the loss matrix L implicitly means gain matrix, where entry in cell $\{i,j\}$ of L is $f(\sigma_i)\cdot g(R_j)$. The columns of feedback matrix H are expanded to account for greater number of moves available to adversary (due to multi-graded relevance vectors). In Definition 1, learner action i is optimal if $\ell_i\cdot p \ge \ell_j\cdot p,\ \forall j\ne i$.

In Definition 2, the maximum number of distinct elements that can be in a row of H is $n+1$. The signal matrix now becomes $S_i\in\{0,1\}^{(n+1)\times 2^m}$, where $(S_i)_{j,\ell} = I(H_{i,\ell} = j-1)$.

Local Observability Fails: Since we are trying to establish a lower bound, it is sufficient to show it for binary relevance vectors, since the adversary can only be more powerful otherwise.

In Lemma 2, proved for SumLoss, $\ell_i \cdot p$ equates to $f(\sigma) \cdot \mathbb{E}[R]$. From definition of DCG, and from the structure and properties of $f(\cdot)$, it is clear that $\ell_i \cdot p$ is *maximized* under the same condition, i.e, $\mathbb{E}[R(\sigma_i(1)] \geq \mathbb{E}[R(\sigma_i(2)] \geq \ldots \geq \mathbb{E}[R(\sigma_i(m)]$. Thus, all actions are Pareto-optimal.

Careful observation of Lemma 3 shows that it is directly applicable to DCG, in light of extension of Lemma 2 to DCG.

Finally, just like in SumLoss, simple calculations with $m = 3$ and $n = 1$, in light of Lemma 2 and 3, show that local observability condition fails to hold.

We show the calculations:

$$S_{\sigma_1} = S_{\sigma_2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\begin{aligned} \ell_{\sigma_1} =& [0, 1/2, 1/\log_2 3, 1/2 + 1/\log_2 3, 1, 3/2, \\ & 1 + 1/\log_2 3, 3/2 + 1/\log_2 3] \\ \ell_{\sigma_2} =& [0, 1/\log_2 3, 1/2, 1/2 + 1/\log_2 3, 1, 1 + 1/\log_2 3, \\ & 3/2, 3/2 + 1/\log_2 3] \end{aligned}$$

It is clear that $\ell_1 - \ell_2 \notin Col(S_{\sigma_1}^\top)$. Hence, Definition 5 fails to hold.

Proof of Corrolary 10

For DCG, the parameters of FTPL are: $R = \sum_{i=1}^m f^s(\sigma^{-1}(i)) g^s(R(i)) = O(m(2^n - 1)), D = \sum_{i=1}^m f^s(\sigma^{-1}(i)) = O(m), A = \sum_{i=1}^m g^s(R(i)) = O(m(2^n - 1))$. Again, $C = 2\sqrt{DRA}$ and $C^I = R$.

Precision@n:

Proof of Corrolary 11

For Precision@n, the parameters of FTPL are: $D = \sum_{i=1}^m f^s(\sigma^{-1}(i)) = O(n), R = \sum_{i=1}^m f^s(\sigma^{-1}(i)) g^s(R(i)) = O(n), A = \sum_{i=1}^m g^s(R(i)) = O(m)$. Again, $C = 2\sqrt{DRA}$ and $C^I = R$.

Non-existence of Sublinear Regret Bounds for NDCG, AP and AUC

We show via simple calculations that for the case $m = 3$, global observability condition fails to hold for NDCG, when feedback is restricted to top ranked item, and relevance vectors are restricted to take binary values. It should be noted that allowing for multi-graded relevance vectors only makes the adversary more powerful; hence proving for binary relevance vectors is enough.

The intuition behind failure to satisfy global observability condition is that the $NDCG(\sigma, R) = f(\sigma) \cdot g(R)$, where where $g(r) = R/Z(R)$ (see Sec.2.2). Thus, $g(\cdot)$ cannot be represented by univariate, scalar valued functions. This makes it impossible to write the difference between two rows of the loss matrix as linear combination of columns of (transposed) signal matrices.

Similar intuitions hold for AP and AUC.

Proof of Lemma 12

Proof. We will first consider NDCG and then, AP and AUC.

NDCG:

The first and last row of Table 1, when calculated for NDCG, are:

$$\begin{aligned} \ell_{\sigma_1} &= [1, 1/2, 1/\log_2 3, (1 + \log_2 3/2))/(1 + \log_2 3), 1, 3/(2(1 + 1/\log_2 3)), 1, 1] \\ \ell_{\sigma_6} &= [1, 1, \log_2 2/\log_2 3, 1, 1/2, 3/(2(1 + 1/\log_2 3)), (1 + (\log_2 3)/2))/(1 + \log_2 3), 1] \end{aligned}$$

We remind once again that NDCG is a gain function, as opposed to SumLoss.

The difference between the two vectors is:

$$\ell_{\sigma_1} - \ell_{\sigma_6} = [0, -1/2, 0, -\log_2 3/(2(1 + \log_2 3)), 1/2, 0, \log_2 3/(2(1 + \log_2 3)), 0].$$

The signal matrices are same as SumLoss:

$$S_{\sigma_1} = S_{\sigma_2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$S_{\sigma_3} = S_{\sigma_5} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$S_{\sigma_4} = S_{\sigma_6} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

It can now be easily checked that $\ell_{\sigma_1} - \ell_{\sigma_6}$ does not lie in the (combined) column span of the (transposed) signal matrices.

We show similar calculations for AP and AUC.

AP:

We once again take $m = 3$. The first and last row of Table 1, when calculated for AP, is:

$$\begin{aligned} \ell_{\sigma_1} &= [1, 1/3, 1/2, 7/12, 1, 5/6, 1, 1] \\ \ell_{\sigma_6} &= [1, 1, 1/2, 1, 1/3, 5/6, 7/12, 1] \end{aligned}$$

Like NDCG, AP is also a gain function.

The difference between the two vectors is:

$$\ell_{\sigma_1} - \ell_{\sigma_6} = [0, -2/3, 0, -5/12, 2/3, 0, 5/12, 0].$$

The signal matrices are same as in the SumLoss case:

$$S_{\sigma_1} = S_{\sigma_2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$S_{\sigma_3} = S_{\sigma_5} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$S_{\sigma_4} = S_{\sigma_6} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

It can now be easily checked that $\ell_{\sigma_1} - \ell_{\sigma_6}$ does not lie in the (combined) column span of the (transposed) signal matrices. □

AUC:

For AUC, we will show the calculations for $m = 4$. This is because global observability does hold with $m = 3$, as the normalizing factors for all relevance vectors with non-trivial mixture of 0 and 1 are same (i.e, when relevance vector has 1 irrelevant and 2 relevant objects, and 1 relevant and 2 irrelevant objects, the normalizing factors are same). The normalizing factor changes from $m = 4$ onwards; hence global observability fails.

Table 1 will be extended since $m = 4$. Instead of illustrating the full table, we point out the important facts about the loss matrix table with $m = 4$ for AUC.

The 2^4 relevance vectors heading the columns are:

$R_1 = 0000,\ R_2 = 0001,\ R_3 = 0010,\ R_4 = 0100,\ R_5 = 1000,\ R_6 = 0011,\ R_7 = 0101,\ R_8 = 1001,\ R_9 = 0110,\ R_{10} = 1010,\ R_{11} = 1100,\ R_{12} = 0111,\ R_{13} = 1011,\ R_{14} = 1101,\ R_{15} = 1110,\ R_{16} = 1111.$

We will calculate the losses of 1st and last (24th) action, where $\sigma_1 = 1234$ and $\sigma_{24} = 4321$.

$$\ell_{\sigma_1} = [0, 1, 2/3, 1/3, 0, 1, 3/4, 1/2, 1/2, 1/4, 0, 1, 2/3, 1/3, 0, 0]$$
$$\ell_{\sigma_{24}} = [0, 0, 1/3, 2/3, 1, 0, 1/4, 1/2, 1/2, 3/4, 1, 0, 1/3, 2/3, 1, 0]$$

AUC, like SumLoss, is a loss function.

The difference between the two vectors is:

$$\ell_{\sigma_1} - \ell_{\sigma_{24}} = [0, 1, 1/3, -1/3, -1, 1, 1/2, 0, 0, -1/2, -1, 1, 1/3, -1/3, -1, 0].$$

The signal matrices for AUC with $m = 4$ will be slightly different. This is because there are 24 signal matrices, corresponding to 24 actions. However, groups of 6 actions will share the same signal matrix. For example, all 6 permutations that place object 1 first will have same signal matrix, all 6 permutations that place object 2 first will have same signal matrix, and so on. For simplicity, we denote the signal matrices as S_1, S_2, S_3, S_4, where S_i corresponds to signal matrix where object i is placed at top. We have:

$$S_1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}$$

$$S_4 = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

It can now be easily checked that $\ell_{\sigma_1} - \ell_{\sigma_{24}}$ does not lie in the (combined) column span of transposes of S_1, S_2, S_3, S_4.

Appendix B. Proofs for Contextual Setting

We provide technical details of results of Online Ranking with Restricted Feedback- Contextual Setting.

Proof of Lemma 14: We restate the lemma before giving the proof, for ease of reading:

Lemma 14: Let $F : \mathbb{R}^m \mapsto \mathbb{R}^a$ be a vector valued function, where $m \geq 1$, $a \geq 1$. For a fixed $x \in \mathbb{R}^m$, let k entries of x be observed at random. That is, for a fixed probability distribution $\mathbb{P}$ and some random $\sigma \sim \mathbb{P}(S_m)$, observed tuple is $\{\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}\}$. The necessary condition for existence of an unbiased estimator of $F(x)$, that can be constructed from $\{\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}\}$, is that it should be possible to decompose $F(x)$ over k (or less) coordinates of x at a time. That is, $F(x)$ should have the following structure:

$$F(x) = \sum_{(i_1, i_2, \ldots, i_\ell) \in\ {}^mP_\ell} h_{i_1, i_2, \ldots, i_\ell}(x_{i_1}, x_{i_2}, \ldots, x_{i_\ell})$$

where $\ell \leq k$, ${}^mP_\ell$ is ℓ permutations of m and $h : \mathbb{R}^\ell \mapsto \mathbb{R}^a$. Moreover, when $F(x)$ can be written in form of Eq 10 , with $\ell = k$, an unbiased estimator of $F(x)$, based on $\{\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}\}$, is,

$$g(\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}) = \frac{\sum_{(j_1, j_2, \ldots, j_k) \in S_k} h_{\sigma(j_1), \ldots, \sigma(j_k)}(x_{\sigma(j_1)}, \ldots, x_{\sigma(j_k)})}{\sum_{(j_1, \ldots, j_k) \in S_k} p(\sigma(j_1), \ldots, \sigma(j_k))}$$

where S_k is the set of $k!$ permutations of [k] and $p(\sigma(1), \ldots, \sigma(k))$ is as in Eq 9 .

Proof. For a fixed $x \in \mathbb{R}^m$ and probability distribution $\mathbb{P}$, let the random permutation be $\sigma \sim \mathbb{P}(S_m)$ and the observed tuple be $\{\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}\}$. Let $\hat{G} = G(\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)})$ be an unbiased estimator of $F(x)$ based on the random observed tuple. Taking expectation, we get:

$$\begin{aligned} F(x) =& \mathbb{E}_{\sigma \sim \mathbb{P}}\left[\hat{G}\right] = \sum_{\pi \in S_m} \mathbb{P}(\pi) G(\pi, x_{\pi(1)}, \ldots, x_{\pi(k)}) \\ =& \sum_{(i_1, i_2, \ldots, i_k) \in\ {}^mP_k} \sum_{\pi \in S_m} \mathbb{P}(\pi) I(\pi(1) = i_1, \pi(2) = i_2, \ldots, \pi(k) = i_k) G(\pi, x_{i_1}, x_{i_2}, \ldots, x_{i_k}) \end{aligned}$$

We note that $\mathbb{P}(\pi) \in [0, 1]$ is independent of x for all $\pi \in S_m$. Then we can use the following construction of function $h(\cdot)$:

$$h_{i_1, i_2, \ldots, i_k}(x_{i_1}, \ldots, x_{i_k}) = \sum_{\pi \in S_m} \mathbb{P}(\pi) I(\pi(1) = i_1, \pi(2) = i_2, \ldots, \pi(k) = i_k) G(\pi, x_{i_1}, x_{i_2}, \ldots, x_{i_k})$$

and thus,

$$F(x) = \sum_{(i_1, i_2, \ldots, i_k) \in\ {}^mP_k} h_{i_1, i_2, \ldots, i_k}(x_{i_1}, x_{i_2}, \ldots, x_i)$$

Hence, we conclude that for existence of an unbiased estimator based on the random observed tuple, it should be possible to decompose $F(x)$ over k (or less) coordinates of x at a time. The "less than k" coordinates arguement follows simply by noting that if

$F(x)$ can be decomposed over ℓ coordinates at a time ($\ell < k$) and observation tuple is $\{\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)})\}$, then any $k-\ell$ observations can be thrown away and the rest used for construction of the unbiased estimator.

The construction of the unbiased estimator proceeds as follows:

Let $F(x) = \sum_{i=1}^{m} h_i(x_i)$ and feedback is for top-1 item ($k = 1$). The unbiased estimator according to Lemma. 14 is:

$$g(\sigma, x_{\sigma(1)}) = \frac{h_{\sigma(1)}(x_{\sigma(1)})}{p(\sigma(1))} = \frac{h_{\sigma(1)}(x_{\sigma(1)})}{\sum_{\pi} \mathbb{P}(\pi) I(\pi(1) = \sigma(1))}$$

Taking expectation w.r.t. σ, we get:

$$\mathbb{E}_{\sigma}[g(\sigma, x_{\sigma(1)})] = \sum_{i=1}^{m} \frac{h_i(x_i)(\sum_{\pi} \mathbb{P}(\pi) I(\pi(1) = i))}{\sum_{\pi} \mathbb{P}(\pi) I(\pi(1) = i)} = \sum_{i=1}^{m} h_i(x_i) = F(x)$$

Now, let $F(x) = \sum\limits_{i \neq j = 1}^{m} h_{i,j}(x_i, x_j)$ and the feedback is for top-2 item ($k = 2$). The unbiased estimator according to Lemma. 14 is:

$$g(\sigma, x_{\sigma(1)}, x_{\sigma(2)}) = \frac{h_{\sigma(1),\sigma(2)}(x_{\sigma(1)}, x_{\sigma(2)}) + h_{\sigma(2),\sigma(1)}(x_{\sigma(2)}, x_{\sigma(1)})}{p(\sigma(1), \sigma(2)) + p(\sigma(2), \sigma(1))}$$

We will use the fact that for any 2 permutations σ_1, σ_2, which places the same 2 objects in top-2 positions but in opposite order, estimators based on σ_1 (i.e, $g(\sigma_1, x_{\sigma_1(1)}, x_{\sigma_1(2)})$) and σ_2 (i.e, $g(\sigma_2, x_{\sigma_2(1)}, x_{\sigma_2(2)})$) have same numerator and denominator. For eg., let $\sigma_1(1) = i, \sigma_1(2) = j$. Numerator and denominator for $g(\sigma_1, x_{\sigma_1(1)}, x_{\sigma_1(2)})$ are $h_{i,j}(x_i, x_j) + h_{j,i}(x_j, x_i)$ and $p(i,j) + p(j,i)$ respectively. Now let $\sigma_2(1) = j, \sigma_2(2) = i$. Then numerator and denominator for $g(\sigma_2, x_{\sigma_2(1)}, x_{\sigma_2(2)})$ are $h_{j,i}(x_j, x_i) + h_{i,j}(x_i, x_j)$ and $p(j,i) + p(i,j)$ respectively.

Then, taking expectation w.r.t. σ, we get:

$$\begin{aligned}
\mathbb{E}_{\sigma} g(\sigma, x_{\sigma(1)}, x_{\sigma(2)}) &= \sum_{i \neq j = 1}^{m} \frac{(h_{i,j}(x_i, x_j) + h_{j,i}(x_j, x_i)) p(i,j)}{p(i,j) + p(j,i)} \\
&= \sum_{i > j = 1}^{m} \frac{(h_{i,j}(x_i, x_j) + h_{j,i}(x_j, x_i))(p(i,j) + p(j,i))}{p(i,j) + p(j,i)} \\
&= \sum_{i > j = 1}^{m} (h_{i,j}(x_i, x_j) + h_{j,i}(x_j, x_i)) = \sum_{i \neq j = 1}^{m} h_{i,j}(x_i, x_j) = F(x)
\end{aligned}$$

This chain of logic can be extended for any $k \geq 3$. Explicitly, for general $k \leq m$, let $\mathbb{S}(i_1, i_2, \ldots, i_k)$ denote all permutations of the set $\{i_1, \ldots, i_k\}$. Then, taking expectation of the unbiased estimator will give:

$$\begin{aligned}
&\mathbb{E}_{\sigma} g(\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(k)}) \\
&= \sum_{(i_1, i_2, \ldots, i_k) \in {}^{m}P_k} \frac{\left(\sum\limits_{(j_1, \ldots, j_k) \in \mathbb{S}(i_1, \ldots, i_k)} h_{j_1, \ldots, j_k}(x_{j_1}, \ldots, x_{j_k}) \right) p(i_1, \ldots, i_k)}{\sum\limits_{(j_1, \ldots, j_k) \in \mathbb{S}(i_1, \ldots, i_k)} p(j_1, \ldots, j_k)} \\
&= \sum_{i_1 > i_2 > \ldots > i_k = 1}^{m} \frac{\left(\sum\limits_{(j_1, \ldots, j_k) \in \mathbb{S}(i_1, \ldots, i_k)} h_{j_1, \ldots, j_k}(x_{j_1}, \ldots, x_{j_k}) \right) \left(\sum\limits_{(j_1, \ldots, j_k) \in \mathbb{S}(i_1, \ldots, i_k)} p(j_1, \ldots, j_k) \right)}{\sum\limits_{(j_1, \ldots, j_k) \in \mathbb{S}(i_1, \ldots, i_k)} p(j_1, \ldots, j_k)} \\
&= \sum_{i_1 > i_2 > \ldots > i_k = 1}^{m} \left(\sum_{(j_1, \ldots, j_k) \in \mathbb{S}(i_1, \ldots, i_k)} h_{j_1, \ldots, j_k}(x_{j_1}, \ldots, x_{j_k}) \right) \\
&= \sum_{(i_1, i_2, \ldots, i_k) \in {}^{m}P_k} h_{i_1, i_2, \ldots, i_k}(x_{i_1}, x_{i_2}, \ldots, x_{i_k}) = F(x)
\end{aligned}$$

Note: For $k = m$, i.e., when the full feedback is received, the unbiased estimator is:

$$\begin{aligned}
g(\sigma, x_{\sigma(1)}, \ldots, x_{\sigma(m)}) &= \frac{\sum\limits_{(j_1, j_2, \ldots, j_m) \in S_m} h_{\sigma(j_1), \ldots, \sigma(j_m)}(x_{\sigma(j_1)}, \ldots, x_{\sigma(j_m)})}{\sum\limits_{(j_1, \ldots, j_m) \in S_m} p(\sigma(j_1), \ldots, \sigma(j_m))} \\
&= \frac{\sum\limits_{(i_1, i_2, \ldots, i_m) \in {}^{m}P_m} h_{i_1, \ldots, i_m}(x_{i_1}, \ldots, x_{i_m})}{1} = F(x)
\end{aligned}$$

Hence, with full information, the unbiased estimator of $F(x)$ is actually $F(x)$ itself, which is consistent with the theory of unbiased estimator.

□

Proof of Lemma 18 :

Proof. All our unbiased estimators are of the form $X^{\top} f(s, R, \sigma)$. We will actually get a bound on $f(s, R, \sigma)$ by using Lemma 17 and $p \to q$ norm relation, to equate out X:

$$\|\tilde{z}\|_2 = \|X^{\top} f(s, R, \sigma)\|_2 \leq \|X^{\top}\|_{1 \to 2} \|f(s, R, \sigma)\|_1 \leq R_D \|f(s, R, \sigma)\|_1$$

since $R_D \geq \max_{j=1}^{m} \|X_{j:}\|_2$.

Squared Loss: The unbiased estimator of gradient of squared loss, as given in the main text, is:

$$\tilde{z} = X^{\top}(2(s - \frac{R(\sigma(1)) e_{\sigma(1)}}{p(\sigma(1))}))$$

where $p(\sigma(1)) = \sum_{\pi \in S_m} \mathbb{P}(\pi) I(\pi(1) = \sigma(1))$ ($\mathbb{P} = \mathbb{P}_t$ is the distribution at round t as in Algorithm 2)

Now we have:

$$\|s - \frac{R(\sigma(1))e_{\sigma(1)}}{p(\sigma(1))}\|_1 \leq mR_DU + \frac{R_{max}}{p(\sigma(1))} \leq \frac{mR_DUR_{max}}{p(\sigma(1)}$$

Thus, taking expectation w.r.t σ, we get:

$$\mathbb{E}_\sigma\|\tilde{z}\|_2^2 \leq m^2R_D^4U^2R_{max}^2\mathbb{E}_\sigma\frac{1}{p(\sigma(1))^2} = m^2R_D^4U^2R_{max}^2\sum_{i=1}^{m}\frac{p(i)}{p^2(i)}$$

Now, since $p(i) \geq \frac{\gamma}{m}$, $\forall\ i$, we get: $\mathbb{E}_\sigma\|\tilde{z}\|_2^2 \leq \frac{C^{sq}}{\gamma}$, where $C^{sq} = m^4R_D^4U^2R_{max}^2$.

RankSVM Surrogate: The unbiased estimator of gradient of the RankSVM surrogate, as given in the main text, is:

$$\tilde{z} = X^\top\left(\frac{h_{s,\sigma(1),\sigma(2)}(R(\sigma(1)),R(\sigma(2))) + h_{s,\sigma(2),\sigma(1)}(R(\sigma(2)),R(\sigma(1)))}{p(\sigma(1),\sigma(2)) + p(\sigma(2),\sigma(1))}\right)$$

where $h_{s,i,j}(R(i),R(j)) = I(R(i) > R(j))I(1 + s(j) > s(i))(e_j - e_i)$ and $p(\sigma(1),\sigma(2)) = \sum_{\pi\in S_m}\mathbb{P}(\pi)I(\pi(1) = \sigma(1),\pi(2) = \sigma(2))$ ($\mathbb{P} = \mathbb{P}_t$ as in Algorithm 2)).

Now we have:

$$\|\frac{h_{s,\sigma(1),\sigma(2)}(R_{\sigma(1)},R_{\sigma(2)}) + h_{s,\sigma(2),\sigma(1)}(R_{\sigma(2)},R_{\sigma(1)})}{p(\sigma(1),\sigma(2)) + p(\sigma(2),\sigma(1))}\|_1 \leq \frac{2}{p(\sigma(1),\sigma(2)) + p(\sigma(2),\sigma(1))}$$

Thus, taking expectation w.r.t σ, we get:

$$\mathbb{E}_\sigma\|\tilde{z}\|_2^2 \leq 4R_D^2\mathbb{E}_\sigma\frac{1}{(p(\sigma(1),\sigma(2)) + p(\sigma(2),\sigma(1)))^2} \leq 4R_D^2\sum_{i>j}^{m}\frac{p(i,j) + p(j,i)}{(p(i,j) + p(j,i))^2}$$

Now, since $p(i,j) \geq \frac{\gamma}{m^2}$, $\forall\ i,j$, we get: $\mathbb{E}_\sigma\|\tilde{z}\|_2^2 \leq \frac{C^{svm}}{\gamma}$, where $C^{svm} = O(m^4R_D^2)$.

KL based Surrogate: The unbiased estimator of gradient of the KL based surrogate, as given in the main text, is:

$$\tilde{z} = X^\top\left(\frac{(\exp(s(\sigma(1))) - \exp(R(\sigma(1))))e_{\sigma(1)}}{p(\sigma(1))}\right)$$

where $p(\sigma(1)) = \sum_{\pi\in S_m}\mathbb{P}(\pi)I(\pi(1) = \sigma(1))$ ($\mathbb{P} = \mathbb{P}_t$ as in Alg. 2)).

Now we have:

$$\|\frac{(\exp(s(\sigma(1))) - \exp(R(\sigma(1))))e_{\sigma(1)}}{p(\sigma(1))}\|_1 \leq \frac{\exp(R_DU)}{p(\sigma(1))}$$

Thus, taking expectation w.r.t σ, we get:

$$\mathbb{E}_\sigma\|\tilde{z}\|_2^2 \leq R_D^2\exp(2R_DU)\mathbb{E}_\sigma\frac{1}{p(\sigma(1))^2}$$

Following the same arguement as in squared loss, we get: $\mathbb{E}_\sigma\|\tilde{z}\|_2^2 \leq \frac{C^{KL}}{\gamma}$, where $C^{KL} = m^2R_D^2\exp(2R_DU)$.

□

Proof of Lemma 15 :

Proof. Let $m = 3$. Collection of all terms which are functions of 1st coordinate of R, i.e, $R(1)$, in the gradient of RankSVM is:

$$I(R(1) > R(2))I(1 + s(2) > s(1))(e_2 - e_1) + I(R(2) > R(1))I(1 + s(1) > s(2))(e_1 - e_2)$$
$$+ I(R(1) > R(3))I(1 + s(3) > s(1))(e_3 - e_1) + I(R(3) > R(1))I(1 + s(1) > s(3))(e_1 - e_3).$$

Now let $s(1) = 1, s(2) = 0, s(3) = 0$. Then the collection becomes:

$$I(R(2) > R(1))(e_1 - e_2) + I(R(3) > R(1))(e_1 - e_3) =$$
$$(I(R(2) > R(1)) + I(R(3) > R(1)))e_1 - I(R(2) > R(1))e_2 - I(R(3) > R(1))e_3$$

Now, if the gradient can be decomposed over each coordinate of R, then the collection of terms associated with $R(1)$ should *only and only be a function* of $R(1)$. Specifically, $(I(R(2) > R(1)) + I(R(3) > R(1)))$ (the non-zero coefficient of e_1) should be a function of only $R(1)$ (similarly for e_2 and e_3).

Now assume that the $(I(R(2) > R(1)) + I(R(3) > R(1)))$ can be expressed as a function of $R(1)$ only. Then the difference between the coefficient's values, for the following two cases: $R(1) = 0, R(2) = 0, R(3) = 0$ and $R(1) = 1, R(2) = 0, R(3) = 0$, would be same as the difference between the coefficient's values, for the following two cases: $R(1) = 0, R(2) = 1, R(3) = 1$ and $R(1) = 1, R(2) = 1, R(3) = 1$ (Since the difference would be affected only by change in $R(1)$ value). It can be clearly seen that the change in value between the first two cases is: $0 - 0 = 0$, while the change in value between the second two cases is: $2 - 0 = 2$. Thus, we reach a contradiction. □

Proof of Lemma 16 :

Proof. The term associated with the 1st coordinate of R, i.e, $R(1)$, in the gradient of ListNet is $= \sum_{i=1}^{m}\left(\frac{-\exp(R(i))}{\sum_{j=1}^{m}\exp(R(j))} + \frac{\exp(s(i))}{\sum_{j=1}^{m}\exp(s(j))}\right)e_i$ (in fact, the same term is associated with every coordinate of R).

Specifically, $f(R) = \left(\frac{-\exp(R(1))}{\sum_{j=1}^{m}\exp(R(j))} + \frac{\exp(s_1)}{\sum_{j=1}^{m}\exp(s(j))}\right)$ is the non-zero coefficient of e_1, associated with $R(1)$. Now, if $f(R)$ would have only been a function of $R(1)$, then $\frac{\partial^2 f(R)}{\partial R(1)\partial R(j)}$, $\forall\ j \neq 1$ would have been zero. It can be clearly seen this is not the case.

Now, the term associated jointly with $R(1)$ and $R(2)$, in the gradient of ListNet is same as before, i.e, $\sum_{i=1}^{m}\left(\frac{-\exp(R(i))}{\sum_{j=1}^{m}\exp(R(j))} + \frac{\exp(s(i))}{\sum_{j=1}^{m}\exp(s(j))}\right)e_i$ (since $R(1)$ and $R(2)$ are present in all the summation terms of the gradient).

Specifically, $f(R) = \left(\frac{-\exp(R(i))}{\sum_{j=1}^{m}\exp(R(j))} + \frac{\exp(s(i))}{\sum_{j=1}^{m}\exp(s(j))}\right)$ is the non-zero coefficient of e_1. Now, if $f(R)$ would have only been a function of $R(1)$ and $R(2)$, then $\frac{\partial^3 f(R)}{\partial R(1)\partial R(2)\partial R(j)}$, $\forall j \neq 1, j \neq 2$ would have been zero. It can be clearly seen this is not the case.

The same argument can be extended for any $k < m$.

□

Proof of Theorem. 21:

Proof. (*Sketch*) The proof builds on the proof of hopeless finite action partial monitoring games given by Piccolboni and Schindelhauer (2001). An examination of their proof of Theorem. 3 indicates that for hopeless games, there have to exist two probability distributions (over adversary's actions), which are indistinguishable in terms of feedback but the optimal learner's actions for the distributions are different. We first provide a mathematical explanation as to why such existence lead to hopeless games. Then, we provide a characterization of indistinguishable probability distributions in our problem setting, and then exploit the characterization of optimal actions for NDCG calibrated surrogates (Theorem 20) to explicitly construct two such probability distributions. This proves the result. Full proof is given below. □

Proof. We will first fix the setting of the online game. We consider $m = 3$ and fixed the document matrix $X \in \mathbb{R}^{3\times 3}$ to be the identity. At each round of the game, the adversary generates the fixed X and the learner chooses a score vector $s \in \mathbb{R}^3$. Making the matrix X identity makes the distinction between weight vectors w and scores s irrelevant since $s = Xw = w$. We note that allowing the adversary to vary X over the rounds only makes him more powerful, which can only increase the regret. We also restrict the adversary to choose binary relevance vectors. Once again, allowing adversary to choose multi-graded relevance vectors only makes it more powerful. Thus, in this setting, the adversary can now choose among $2^3 = 8$ possible relevance vectors. The learner's action set is infinite, i.e., the learner can choose any score vector $s = Xw = \mathbb{R}^m$. The loss function $\phi(s, R)$ is any NDCG calibrated surrogate and feedback is the relevance of top-ranked item at each round, where ranking is induced by sorted order (descending) of score vector. We will use p to denote randomized adversary one-short strategies, i.e. distributions over the 8 possible relevance score vectors. Let $s^*_p = \text{argmin}_s \mathbb{E}_{R\sim p}\phi(s, R)$. We note that in the definition of NDCG calibrated surrogates, Ravikumar et al. (2011) assume that the optimal score vector for each distribution over relevance vectors is unique and we subscribe to that assumption. The assumption was taken to avoid some boundary conditions.

It remains to specify the choice of U, a bound on the Euclidean norm of the weight vectors (same as score vectors for us right now) that is used to define the best loss in hindsight. It never makes sense for the learner to play anything outside the set $\cup_p s^*_p$ so that we can set $U = \max\{\|s\|_2 : s \in \cup_p s^*_p\}$.

The paragraph following Lemma 6 of Thm. 3 in Piccolboni and Schindelhauer (2001) gives the main intuition behind the argument the authors developed to prove hopelessness of finite action partial monitoring games. To make our proof self contained, we will explain the intuition in a rigorous way.

Key insight: Two adversary strategies $p, \tilde{p}$ are said to be indistinguishable from the learner's feedback perspective, if for every action of the learner, the probability distribution over the feedbacks received by learner is the same for p and $\tilde{p}$. Now assume that adversary always selects actions according to one of the two such indistinguishable strategies. Thus, the learner will always play one of s^*_p and $s^*_{\tilde{p}}$. By uniqueness, $s^*_p \neq s^*_{\tilde{p}}$. Then, the learner

incurs a constant (non-zero) regret on any round where adversary plays according to p and learner plays $s^*_{\tilde{p}}$, or if the adversary plays according to $\tilde{p}$ and learner plays s^*_p. We show that in such a setting, adversary can simply play according to $(p + \tilde{p})/2$ and the learner suffers an expected regret of $\Omega(T)$.

Assume that the adversary selects $\{R_1, \ldots, R_T\}$ from product distribution $\otimes p$. Let the number of times the learner plays s^*_p and $s^*_{\tilde{p}}$ be denoted by random variables N^p_1 and N^p_2 respectively, where N^p shows the exclusive dependence on p. It is always true that $N^p_1 + N^p_2 = T$. Moreover, let the expected per round regret be ϵ_p when learner plays $s^*_{\tilde{p}}$, where the expectation is taken over the randomization of adversary. Now, assume that adversary selects $\{R_1, \ldots, R_T\}$ from product distribution $\otimes\tilde{p}$. The corresponding notations become $N^{\tilde{p}}_1$ and $N^{\tilde{p}}_2$ and $\epsilon_{\tilde{p}}$. Then,

$$\mathbb{E}_{(R_1,\ldots,R_T)\sim\otimes p}\mathbb{E}_{(s_1,\ldots,s_T)}[\text{Regret}((s_1,\ldots,s_T),(R_1,\ldots,R_T))] = 0\cdot\mathbb{E}[N^p_1] + \epsilon_p\cdot\mathbb{E}[N^p_2]$$

and

$$\mathbb{E}_{(R_1,\ldots,R_T)\sim\otimes\tilde{p}}\mathbb{E}_{(s_1,\ldots,s_T)}[\text{Regret}((s_1,\ldots,s_T),(R_1,\ldots,R_T))] = \epsilon_{\tilde{p}}\cdot\mathbb{E}[N^{\tilde{p}}_1] + 0\cdot\mathbb{E}[N^{\tilde{p}}_2]$$

Since p and $\tilde{p}$ are indistinguishable from perspective of learner, $\mathbb{E}[N^p_1] = \mathbb{E}[N^{\tilde{p}}_1] = \mathbb{E}[N_1]$ and $\mathbb{E}[N^p_2] = \mathbb{E}[N^{\tilde{p}}_2] = \mathbb{E}[N_2]$. That is, the random variable denoting number of times s^*_p is played by learner does not depend on adversary distribution (same for $s^*_{\tilde{p}}$.). Using this fact and averaging the two expectations, we get:

$$\begin{aligned}\mathbb{E}_{(R_1,\ldots,R_T)\sim\frac{\otimes p+\otimes\tilde{p}}{2}}\mathbb{E}_{(s_1,\ldots,s_T)}[\text{Regret}((s_1,\ldots,s_T),(R_1,\ldots,R_T))] &= \frac{\epsilon_{\tilde{p}}}{2}\cdot\mathbb{E}[N_1] + \frac{\epsilon_p}{2}\cdot\mathbb{E}[N_2]\\ &\geq \min(\frac{\epsilon_p}{2},\frac{\epsilon_{\tilde{p}}}{2})\cdot\mathbb{E}[N_1+N_2] = \epsilon\cdot T\end{aligned}$$

Since

$$\sup_{R_1,\ldots,R_T}\mathbb{E}[\text{Regret}((s_1,\ldots,s_T),(R_1,\ldots,R_T))] \geq \mathbb{E}_{(R_1,\ldots,R_T)\sim\frac{\otimes p+\otimes\tilde{p}}{2}}\mathbb{E}_{(s_1,\ldots,s_T)}[\text{Regret}((s_1,\ldots,s_T),(R_1,\ldots,R_T))]$$

we conclude that for every learner algorithm, adversary has a strategy, s.t. learner suffers an expected regret of $\Omega(T)$.

Now, the thing left to be shown is the existence of two indistinguishable distributions p and $\tilde{p}$, s.t. $s^*_p \neq s^*_{\tilde{p}}$.

Characterization of indistinguishable strategies in our problem setting: Two adversary's strategies p and $\tilde{p}$ will be indistinguishable, in our problem setting, if for every score vector s, the relevances of the top-ranked item, according to s, are same for relevance vector drawn from p and $\tilde{p}$. Since relevance vectors are restricted to be binary, mathematically, it means that $\forall s$, $\mathbb{P}_{R\sim p}(R(\pi_s(1)) = 1) = \mathbb{P}_{R\sim\tilde{p}}(R(\pi_s(1)) = 1)$ (actually, we also need $\forall s$, $\mathbb{P}_{R\sim p}(R(\pi_s(1)) = 0) = \mathbb{P}_{R\sim\tilde{p}}(R(\pi_s(1)) = 0)$, but due to the binary nature, $\mathbb{P}_{R\sim p}(R(\pi_s(1)) = 1) = \mathbb{P}_{R\sim\tilde{p}}(R(\pi_s(1)) = 1) \implies \mathbb{P}_{R\sim p}(R(\pi_s(1)) = 0) = \mathbb{P}_{R\sim\tilde{p}}(R(\pi_s(1)) = 0)$). Since the equality has to hold $\forall s$, this implies $\forall j \in [m]$, $\mathbb{P}_{R\sim p}(R(j) = 1) = \mathbb{P}_{R\sim\tilde{p}}(R(j) = 1)$ (as every item will be ranked at top by some score vector). Hence, $\forall j \in [m]$, $\mathbb{E}_{R\sim p}[R(j)] = \mathbb{E}_{R\sim\tilde{p}}[R(j)] \implies \mathbb{E}_{R\sim p}[R] = \mathbb{E}_{R\sim\tilde{p}}[R]$. It can be

Table 3: Relevance and probability vectors.

p	0.0	0.1	0.15	0.05	0.2	0.3	0.2	0.0
$\tilde{p}$	0.0	0.3	0.0	0.0	0.15	0.15	0.4	0.0
Rel.	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
	0	1	1	0	1	0	0	1
	0	1	0	1	0	1	0	1
	0	0	1	1	0	0	1	1

seen clearly that the chain of implications can be reversed. Hence, $\forall s$, $\mathbb{P}_{R\sim p}(R(\pi_s(1)) = 1) = \mathbb{P}_{R\sim\tilde{p}}(R(\pi_s(1)) = 1) \iff \mathbb{E}_{R\sim p}[R] = \mathbb{E}_{R\sim\tilde{p}}[R]$.

Explicit adversary strategies: Following from the discussion so far and Theorem 20, if we can show existence of two strategies p and $\tilde{p}$ s.t. $\mathbb{E}_{R\sim p}[R] = \mathbb{E}_{R\sim\tilde{p}}[R]$, but argsort $\left(\mathbb{E}_{R\sim p}\left[\frac{G(\mathbf{R})}{Z(R)}\right]\right) \neq$ argsort $\left(\mathbb{E}_{R\sim\tilde{p}}\left[\frac{G(\mathbf{R})}{Z(R)}\right]\right)$, we are done.

The 8 possible relevance vectors (adversary's actions) are $(R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8) = (000, 110, 101, 011, 100, 010, 001, 111)$. Let the two probability vectors be:

$$p = (0.0, 0.1, 0.15, 0.05, 0.2, 0.3, 0.2, 0.0)$$
$$\tilde{p} = (0.0, 0.3, 0.0, 0.0, 0.15, 0.15, 0.4, 0.0).$$

The data is provided in table format in Table. 3.

Under the two distributions, it can be checked that $\mathbb{E}_{R\sim p}[R] = \mathbb{E}_{R\sim\tilde{p}}[R] = (0.45, 0.45, 0.4)^\top$.

However, $\mathbb{E}_{R\sim p}\left[\frac{G(\mathbf{R})}{Z(R)}\right] = (0.3533, 0.3920, 0.3226)^\top$, but $\mathbb{E}_{R\sim\tilde{p}}\left[\frac{G(\mathbf{R})}{Z(R)}\right] = (0.3339, 0.3339, 0.4000)^\top$.

Hence, argsort $\left(\mathbb{E}_{R\sim p}\left[\frac{G(\mathbf{R})}{Z(R)}\right]\right) = [2, 1, 3]^\top$ but argsort $\left(\mathbb{E}_{R\sim\tilde{p}}\left[\frac{G(\mathbf{R})}{Z(R)}\right]\right) \in \{[3, 1, 2]^\top, [3, 2, 1]^\top\}$. □

References

Jacob Abernethy, Chansoo Lee, and Ambuj Tewari. Perturbation techniques in online learning and optimization. In Tamir Hazan, George Papandreou, and Daniel Tarlow, editors, *Perturbations, Optimization, and Statistics*, Neural Information Processing Series, chapter 8. MIT Press, 2016.

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

Nir Ailon. Improved bounds for online learning over the permutahedron and other ranking polytopes. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 29–37, 2014.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval*, volume 463. ACM Press, 1999.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Gábor Bartók and Csaba Szepesvári. Partial monitoring with side information. In *Algorithmic Learning Theory*, pages 305–319, 2012.

Gabor Bartok, Dean P. Foster, David Pal, Alexander Rakhlin, and Csaba Szepesvari. Partial monitoringclassification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997, 2014.

Aditya Bhaskara and Aravindan Vijayaraghavan. Approximating matrix p-norms. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 497–511. SIAM, 2011.

A. Blum and Y. Mansour. Learning, regret minimization, and equilibria. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic game theory*. Cambridge University Press, 2007.

Clément Calauzenes, Nicolas Usunier, and Patrick Gallinari. On the (non-) existence of convex, calibrated surrogate losses for ranking. In *Advances in Neural Information Processing Systems*, 2012.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International conference on Machine learning*, pages 129–136. ACM, 2007.

Nicolo Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Regret minimization under partial monitoring. *Mathematics of Operations Research*, 31(3):562–580, 2006.

Olivier Chapelle and Yi Chang. Yahoo! learning to rank challenge overview. *Journal of Machine Learning Research-Proceedings Track*, pages 1–24, 2011.

Olivier Chapelle and Mingrui Wu. Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval*, 13(3):216–235, 2010.

Olivier Chapelle, Quoc Le, and Alex Smola. Large margin optimization of ranking measures. In *NIPS Workshop: Machine Learning for Web Search*, 2007.

Corinna Cortes and Mehryar Mohri. Auc optimization vs. error rate minimization. *Advances in neural information processing systems*, 16(16):313–320, 2004.

David Cossock and Tong Zhang. Subset ranking using regression. In *Conference on Learning theory*, pages 605–619, 2006.

David Cossock and Tong Zhang. Statistical analysis of bayes optimal subset ranking. *Information Theory, IEEE Transactions on*, 54(11):5140–5154, 2008.

John C. Duchi, Lester W. Mackey, and Michael I. Jordan. On the consistency of ranking algorithms. In *Proceedings of the 27th International Conference on Machine Learning*, pages 327–334, 2010.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394, 2005.

Dean P. Foster and Alexander Rakhlin. No internal regret via neighborhood watch. In *International Conference on Artificial Intelligence and Statistics*, pages 382–390, 2012.

Claudio Gentile and Francesco Orabona. On multilabel classification and ranking with bandit feedback. *The Journal of Machine Learning Research*, 15(1):2451–2487, 2014.

Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Balancing exploration and exploitation in listwise and pairwise online learning to rank. *Information Retrieval*, 16 (1):63–90, 2013.

IM-2009. `http://imat2009.yandex.ru/en/`, 2009.

Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48. ACM, 2000.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, pages 422–446, 2002.

Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD*, pages 133–142. ACM, 2002.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, pages 291–307, 2005.

Robert Kleinberg and Tom Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Foundations of Computer Science, 2003*, pages 594–605, 2003.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.

Tian Lin, Bruno Abrahao, Robert Kleinberg, and John Lui. Combinatorial partial monitoring game with linear feedback and its applications. In *Proceedings of the 31th International Conference on Machine Learning*, pages 901–909. ACM, 2014.

Tie-Yan Liu. *Learning to rank for information retrieval*. Springer Science & Business Media, 2011.

Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval*, pages 3–10, 2007.

Antonio Piccolboni and Christian Schindelhauer. Discrete prediction games with arbitrary feedback and loss. In *Computational Learning Theory*, pages 208–223. Springer, 2001.

Tao Qin and Tie-Yan Liu. Microsoft letor website, 2006. `http://research.microsoft.com/en-us/um/beijing/projects/letor/`.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International conference on Machine learning*, pages 784–791. ACM, 2008.

Filip Radlinski, Paul N Bennett, Ben Carterette, and Thorsten Joachims. Redundancy, diversity and interdependent document relevance. In *ACM SIGIR*, pages 46–52. ACM, 2009.

Pradeep Ravikumar, Ambuj Tewari, and Eunho Yang. On NDCG consistency of listwise ranking methods. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 618–626, 2011.

Mark Sanderson. *Test collection based evaluation of information retrieval systems*, volume 13. Now Publishers Inc, 2010.

Niek Tax, Sander Bockting, and Djoerd Hiemstra. A cross-benchmark comparison of 87 learning to rank methods. *Information Processing and Management*, pages 757–772, 2015.

Zheng Wen, Branislav Kveton, and Azin Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In *Proceedings of the 32nd International Conference on Machine Learning*, 2014.

Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *Proceedings of ACM SIGIR*, pages 271–278, 2007.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.

Journal of Machine Learning Research 18 (2017) 1-72 Submitted 11/16; Revised 10/17; Published 10/17

A Unifying Framework for Gaussian Process Pseudo-Point Approximations using Power Expectation Propagation

Thang D. Bui TDB40@CAM.AC.UK
Josiah Yan JOSIAH.YAN@GMAIL.COM
Richard E. Turner RET26@CAM.AC.UK
Computational and Biological Learning Lab, Department of Engineering
University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, UK

Editor: Neil Lawrence

Abstract

Gaussian processes (GPs) are flexible distributions over functions that enable high-level assumptions about unknown functions to be encoded in a parsimonious, flexible and general way. Although elegant, the application of GPs is limited by computational and analytical intractabilities that arise when data are sufficiently numerous or when employing non-Gaussian models. Consequently, a wealth of GP approximation schemes have been developed over the last 15 years to address these key limitations. Many of these schemes employ a small set of pseudo data points to summarise the actual data. In this paper we develop a new pseudo-point approximation framework using Power Expectation Propagation (Power EP) that unifies a large number of these pseudo-point approximations. Unlike much of the previous venerable work in this area, the new framework is built on standard methods for approximate inference (variational free-energy, EP and Power EP methods) rather than employing approximations to the probabilistic generative model itself. In this way all of the approximation is performed at 'inference time' rather than at 'modelling time', resolving awkward philosophical and empirical questions that trouble previous approaches. Crucially, we demonstrate that the new framework includes new pseudo-point approximation methods that outperform current approaches on regression and classification tasks.

Keywords: Gaussian process, expectation propagation, variational inference, sparse approximation

1. Introduction

Gaussian Processes (GPs) are powerful nonparametric distributions over continuous functions that are routinely deployed in probabilistic modelling for applications including regression and classification (Rasmussen and Williams, 2005), representation learning (Lawrence, 2005), state space modelling (Wang et al., 2005), active learning (Houlsby et al., 2011), reinforcement learning (Deisenroth, 2010), black-box optimisation (Snoek et al., 2012), and numerical methods (Mahsereci and Hennig, 2015). GPs have many elegant theoretical properties, but their use in probabilistic modelling is greatly hindered by analytic and computational intractabilities. A large research effort has been directed at this fundamental problem resulting in the development of a plethora of sparse approximation methods that can sidestep these intractabilities (Csató, 2002; Csató and Opper, 2002; Schwaighofer and Tresp, 2002; Seeger et al., 2003; Quiñonero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006; Snelson, 2007; Naish-Guzman and Holden, 2007; Titsias, 2009; Figueiras-Vidal and Lázaro-Gredilla, 2009; Álvarez et al., 2010; Qi et al., 2010; Bui and Turner, 2014; Frigola et al., 2014; McHutchon, 2014; Hensman et al., 2015; Hernández-Lobato and Hernández-Lobato, 2016; Matthews et al., 2016)

This paper develops a general sparse approximate inference framework based upon Power Expectation Propagation (PEP) (Minka, 2004) that unifies many of these approximations, extends them significantly, and provides improvements in practical settings. In this way, the paper provides a complementary perspective to the seminal review of Quiñonero-Candela and Rasmussen (2005) viewing sparse approximations through the lens of approximate *inference*, rather than approximate *generative models*.

The paper begins by reviewing several frameworks for sparse approximation focussing on the GP regression and classification setting (Section 2). It then lays out the new unifying framework and the relationship to existing techniques (Section 3). Readers whose focus is to understand the new framework might want to move directly to this section. Finally, a thorough experimental evaluation is presented in Section 4.

2. Pseudo-point Approximations for GP Regression and Classification

This section provides a concise introduction to GP regression and classification and then reviews several pseudo-point based sparse approximation schemes for these models. For simplicity, we first consider a supervised learning setting in which the training set comprises N D-dimensional input and scalar output pairs $\{\mathbf{x}_n, y_n\}_{n=1}^N$ and the goal is to produce probabilistic predictions for the outputs corresponding to novel inputs. A non-linear function, $f(\mathbf{x})$, can be used to parameterise the probabilistic mapping between inputs and outputs, $p(y_n|f, \mathbf{x}_n, \theta)$ which may also depend on hyperparameters θ. Typical choices for the probabilistic mapping are Gaussian $p(y_n|f, \mathbf{x}_n, \theta) = \mathcal{N}(y_n; f(\mathbf{x}_n), \sigma_y^2)$ for the regression setting ($y_n \in \mathbb{R}$) and Bernoulli $p(y_n|f, \mathbf{x}_n, \theta) = \mathcal{B}(y_n; \Phi(f(\mathbf{x}_n)))$ with a sigmoidal link function $\Phi(f)$ for the binary classification setting ($y_n \in \{0, 1\}$). Whilst it is possible to specify the non-linear function f via an explicit parametric form, a more flexible and elegant approach employs a GP prior over the functions directly, $p(f|\theta) = \mathcal{GP}(f; 0, k_\theta(\cdot, \cdot))$, here assumed without loss of generality to have a zero mean-function and a covariance function $k_\theta(\mathbf{x}, \mathbf{x}')$. This class of probabilistic models has a joint distribution

$$p(f, \mathbf{y}|\theta) = p(f|\theta) \prod_{n=1}^{N} p(y_n|f(\mathbf{x}_n), \theta)$$

where we have collected the observations into the vector $\mathbf{y}$ and suppressed the inputs on the left hand side to lighten the notation.

This model class contains two potential sources of intractability. First, the possibly non-linear likelihood function can introduce analytic intractabilities that require approximation. Second, the GP prior entails an $\mathcal{O}(N^3)$ complexity that is computationally intractable for many practical problems. These two types of intractability can be handled by combining standard approximate inference methods with pseudo-point approximations that summarise the full Gaussian process via M pseudo data points leading to an $\mathcal{O}(NM^2)$ cost. The main

©2017 Thang D. Bui, Josiah Yan, Richard E. Turner.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-603.html.

approaches of this sort can be characterised in terms of two parallel frameworks that are described in the following sections.

2.1 Sparse GP Approximation via Approximate Generative Models

The first framework begins by constructing a new generative model that is similar to the original, so that inference in the new model might be expected to produce similar results, but which has a special structure that supports efficient computation. Typically this approach involves approximating the Gaussian process prior as it is the origin of the cubic cost. If there are analytic intractabilities in the approximate model, as will be the case in e.g. classification or state-space models, then these will require approximate inference to be performed in the approximate model.

The seminal review by Quiñonero-Candela and Rasmussen (Quiñonero-Candela and Rasmussen, 2005) reinterprets a family of approximations in terms of this unifying framework. The GP prior is approximated by identifying a small set of $M \leq N$ pseudo-points $\mathbf{u}$, here assumed to be disjoint from the training function values $\mathbf{f}$ so that $f = \{\mathbf{u}, \mathbf{f}, f_{\neq \mathbf{u},\mathbf{f}}\}$. Here $f_{\neq \mathbf{u},\mathbf{f}}$ denotes the function values which are not at the training inputs or pseudo-inputs. The GP prior is then decomposed using the product rule

$$p(f|\theta) = p(\mathbf{u}|\theta)p(\mathbf{f}|\mathbf{u},\theta)p(f_{\neq \mathbf{u},\mathbf{f}}|\mathbf{f},\mathbf{u},\theta).$$

Of central interest is the relationship between the pseudo-points and the training function values $p(\mathbf{f}|\mathbf{u},\theta) = \mathcal{N}(\mathbf{f}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \mathbf{D_{ff}})$ where $\mathbf{D_{ff}} = \mathbf{K_{ff}} - \mathbf{Q_{ff}}$ and $\mathbf{Q_{ff}} = \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}$. Here we have introduced matrices corresponding to the covariance function's evaluation at the pseudo-input locations $\{\mathbf{z}_m\}_{m=1}^M$, so that $[\mathbf{K_{uu}}]_{mm'} = k_\theta(\mathbf{z}_m, \mathbf{z}_{m'})$ and similarly for the covariance between the pseudo-input and data locations $[\mathbf{K_{uf}}]_{mn} = k_\theta(\mathbf{z}_m, \mathbf{x}_n)$. Importantly, this term saddles inference and learning with a complexity cost that is cubic in the number of data points. Computationally efficient approximations can be constructed by simplifying these dependencies between the pseudo-points and the data function values $q(\mathbf{f}|\mathbf{u},\theta) \approx p(\mathbf{f}|\mathbf{u},\theta)$. In order to benefit from these efficiencies at prediction time as well, a second approximation is made whereby the pseudo-points form a bottleneck between the data function values and test function values $p(f_{\neq \mathbf{u},\mathbf{f}}|\mathbf{u},\theta) \approx p(f_{\neq \mathbf{u},\mathbf{f}}|\mathbf{f},\mathbf{u},\theta)$. Together, the two approximations result in an approximate prior process,

$$q(f|\theta) = p(\mathbf{u}|\theta)q(\mathbf{f}|\mathbf{u},\theta)p(f_{\neq \mathbf{u},\mathbf{f}}|\mathbf{u},\theta).$$

We can now compactly summarise a number of previous approaches to GP approximation as special cases of the choice

$$q(\mathbf{f}|\mathbf{u},\theta) = \prod_{b=1}^{B} \mathcal{N}(\mathbf{f}_b; \mathbf{K}_{\mathbf{f}_b,\mathbf{u}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \alpha\mathbf{D}_{\mathbf{f}_b,\mathbf{f}_b})$$

where b indexes B disjoint blocks of data-function values. The Deterministic Training Conditional (DTC) approximation uses $\alpha \to 0$; the Fully Independent Training Conditional (FITC) approximation uses $\alpha = 1$ and $B = N$; the Partially Independent Training Conditional (PITC) approximation uses $\alpha = 1$ (Quiñonero-Candela and Rasmussen, 2005; Schwaighofer and Tresp, 2002).

Before considering inference in the modified models, note that it is possible to construct more flexible modified prior processes using the inter-domain approach that places the pseudo-points in a different domain from the data, defined by a linear integral transform $g(z) = \int w(z,z')f(z')\mathrm{d}z'$. Here the window $w(z,z')$ might be a Gaussian blur or a wavelet transform. The pseudo-points are now placed in the new domain $g = \{\mathbf{u}, \mathbf{g}_{\neq \mathbf{u}}\}$ where they induce a potentially more flexible Gaussian process in the old domain f through the linear transform (see Figueiras-Vidal and Lázaro-Gredilla, 2009, for FITC). The expressions in this section still hold, but the covariance matrices involving pseudo-points are modified to take account of the transform,

$$[\mathbf{K_{uu}}]_{mm'} = \int w(\mathbf{z}_m,\mathbf{z})k_\theta(\mathbf{z},\mathbf{z}')w(\mathbf{z}',\mathbf{z}_{m'})\mathbf{dzdz}', \quad [\mathbf{K_{uf}}]_{mn} = \int w(\mathbf{z}_m,\mathbf{z})k_\theta(\mathbf{z},\mathbf{x}_n)\mathbf{dz}.$$

Having specified modified prior processes, these can be combined with the original likelihood function to produce a new generative model. In the case of point-wise likelihoods we have

$$q(\mathbf{y}, f|\theta) = q(f|\theta)\prod_{n=1}^{N} p(y_n|f(\mathbf{x}_n),\theta).$$

Inference and learning can now be performed using the modified model using standard techniques. Due to the form of the new prior process, the computational complexity is $\mathcal{O}(NM^2)$ (for testing, N becomes the number of test data points, assuming dependencies between the test-points are not computed).[1] For example, in the case of regression, the posterior distribution over function values f (necessary for inference and prediction) has a simple analytic form

$$q(f|\mathbf{y},\theta) = \mathcal{GP}(f; \mu_{f|\mathbf{y}}, \Sigma_{f|\mathbf{y}}), \quad \mu_{f|\mathbf{y}} = \mathbf{Q}_{f\mathbf{f}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}, \quad \Sigma_{f|\mathbf{y}} = \mathbf{K}_{ff} - \mathbf{Q}_{f\mathbf{f}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{Q}_{\mathbf{f}f}, \qquad (1)$$

where $\overline{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q_{ff}} + \mathrm{blkdiag}(\{\alpha_b\mathbf{D}_{\mathbf{f}_b\mathbf{f}_b}\}_{b=1}^B) + \sigma_y^2\mathrm{I}$ and blkdiag builds a block-diagonal matrix from its inputs. One way of understanding the origin of the computational gains is that the new generative model corresponds to a form of factor analysis in which the M pseudo-points determine the N function values at the observed data (as well as at potential test locations) via a linear Gaussian relationship. This results in low rank (sparse) structure in $\overline{\mathbf{K}}_{\mathbf{ff}}$ that can be exploited through the matrix inversion and determinant lemmas. In the case of regression, the new model's marginal likelihood also has an analytic form that allows the hyperparameters, θ, to be learned via optimisation

$$\log q(\mathbf{y}|\theta) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\overline{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^\intercal\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}. \qquad (2)$$

The approximate generative model framework has attractive properties. The cost of inference, learning, and prediction has been reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(NM^2)$ and in many cases accuracy can be maintained with a relatively small number of pseudo-points. The pseudo-point input locations can be optimised by maximising the new model's marginal

1. It is assumed that the maximum size of the blocks is not greater than the number of pseudo-points $\dim(\mathbf{f}_b) \leq M$.

likelihood (Snelson and Ghahramani, 2006). When $M = N$ and the pseudo-points and observed data inputs coincide, then FITC and PITC are exact which appears reassuring. However, the framework is philosophically challenging as the elegant separation of model and (approximate) inference has been lost. Are we allowed in an online inference setting, for example, to add new pseudo-points as more data are acquired and the complexity of the underlying function is revealed? This seems sensible, but effectively changes the modelling assumptions as more data are seen. Devout Bayesians might then demand that we perform model averaging for coherence. Similarly, if the pseudo-input locations are optimised, the principled non-parametric model has suddenly acquired MD parameters and with them all of the concomitant issues of parametric models including overfitting and optimisation difficulties (Bauer et al., 2016). As the pseudo-inputs are considered part of the model, the Bayesians might then suggest that we place priors over the pseudo-inputs and perform full-blown probabilistic inference over them.

These awkward questions arise because the generative modelling interpretation of pseudo-data entangles the assumptions made about the data with the approximations required to perform inference. Instead, the modelling assumptions (which encapsulate prior understanding of the data) should remain decoupled from inferential assumptions (which leverage structure in the posterior for tractability). In this way pseudo-data should be introduced when we seek to perform computationally efficient approximate inference, leaving the modelling assumptions unchanged as we refine and improve approximate inference. Indeed, even under the generative modelling perspective, for analytically intractable likelihood functions an additional approximate inference step is required, begging the question: why not handle computational and analytic intractabilities together at inference time?

2.2 Sparse GP Approximation via Approximate Inference: VFE

The approximate generative model framework for constructing sparse approximations is philosophically troubling. In addition, learning pseudo-point input locations via optimisation of the model likelihood can perform poorly e.g. for DTC it is prone to overfitting even for $M \ll N$ (Titsias, 2009). This motivates a more direct approach that commits to the true generative model and performs all of the necessary approximation at inference time.

Perhaps the most well known approach in this vein is Titsias's beautiful sparse variational free energy (VFE) method (Titsias, 2009). The original presentation of this work employs finite variable sets and an augmentation trick that arguably obscures its full elegance. Here instead we follow the presentation in Matthews et al. (2016) and lower bound the marginal likelihood using a distribution $q(f)$ over the entire infinite-dimensional function,

$$\log p(\mathbf{y}|\theta) = \log \int p(\mathbf{y}, f|\theta) \mathrm{d}f \geq \int q(f) \log \frac{p(\mathbf{y}, f|\theta)}{q(f)} \mathrm{d}f = \mathbb{E}_{q(f)}\left[\log \frac{p(\mathbf{y}, f|\theta)}{q(f)}\right] = \mathcal{F}(q, \theta).$$

The VFE bound can be written as the difference between the model log-marginal likelihood and the KL divergence between the variational distribution and the true posterior $\mathcal{F}(q, \theta) = \log p(\mathbf{y}|\theta) - \mathrm{KL}[q(f)||p(f|\mathbf{y}, \theta)]$. The bound is therefore saturated when $q(f) = p(f|\mathbf{y}, \theta)$, but this is intractable. Instead, pseudo-points are made explicit, $f = \{\mathbf{u}, f_{\neq \mathbf{u}}\}$, and an approximate posterior distribution used of the following form $q(f) = q(\mathbf{u}, f_{\neq \mathbf{u}}|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$. Under this approximation, the set of variables $f_{\neq \mathbf{u}}$ do not experience the data directly, but

rather only through the pseudo-points, as can be seen by comparison to the true posterior $p(f|\mathbf{y}, \theta) = p(f_{\neq \mathbf{u}}|\mathbf{y}, \mathbf{u}, \theta)p(\mathbf{u}|\mathbf{y}, \theta)$. Importantly, the form of the approximate posterior causes a cancellation of the prior conditional term, which gives rise to a bound with $\mathcal{O}(NM^2)$ complexity,

$$\begin{aligned}\mathcal{F}(q, \theta) &= \mathbb{E}_{q(f|\theta)}\left[\log \frac{p(\mathbf{y}|f, \theta)\cancel{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)}p(\mathbf{u}|\theta)}{\cancel{p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)}q(\mathbf{u})}\right] \\ &= \sum_n \mathbb{E}_{q(f|\theta)}\left[\log p(y_n|\mathrm{f}_n, \theta)\right] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u}|\theta)].\end{aligned}$$

For regression with Gaussian observation noise, the calculus of variations can be used to find the optimal approximate posterior Gaussian process over pseudo-data $q^{\mathrm{opt}}(f|\theta) = p(f_{\neq \mathbf{u}}|\mathbf{u}, \theta)q^{\mathrm{opt}}(\mathbf{u})$ which has the form

$$q^{\mathrm{opt}}(f|\theta) = \mathcal{GP}(f; \mu_{f|\mathbf{y}}, \Sigma_{f|\mathbf{y}}), \;\; \mu_{f|\mathbf{y}} = \mathbf{Q}_{f\mathbf{f}}\tilde{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}, \;\; \Sigma_{f|\mathbf{y}} = \mathbf{K}_{ff} - \mathbf{Q}_{f\mathbf{f}}\tilde{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{Q}_{\mathbf{f}f}, \qquad (3)$$

where $\tilde{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \sigma_y^2\mathrm{I}$. This process is identical to that recovered when performing exact inference under the DTC approximate regression generative model (Titsias, 2009) (see Equation (1) as $\alpha \to 0$). In fact, DTC was originally derived using a related KL argument (Csató, 2002; Seeger et al., 2003). The optimised free-energy is

$$\mathcal{F}(q^{\mathrm{opt}}, \theta) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\tilde{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^\intercal\tilde{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y} - \frac{1}{2\sigma_y^2}\mathrm{trace}(\mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}). \qquad (4)$$

Notice that the free-energy has an additional trace term as compared to the marginal likelihood obtained from the DTC generative model approach (see Equation (2) as $\alpha \to 0$). The trace term is proportional to the sum of the variances of the training function values given the pseudo-points, $p(\mathbf{f}|\mathbf{u})$, it thereby encourages pseudo-input locations that explain the observed data well. This term acts as a regulariser that prevents overfitting which plagues the generative model formulation of DTC.

The VFE approach can be extended to non-linear models including classification (Hensman et al., 2015), latent variable models (Titsias and Lawrence, 2010) and state space models (Frigola et al., 2014; McHutchon, 2014) by restricting $q(\mathbf{u})$ to be Gaussian and optimising its parameters. Indeed, this uncollapsed form of the bound can be beneficial in the context of regression too as it is amenable to stochastic optimisation (Hensman et al., 2013). Additional approximation is sometimes required to compute any remaining intractable non-linear integrals, but these are often low-dimensional. For example, when the likelihood depends on only one latent function value, as is typically the case for regression and classification, the bound requires only 1D integrals $\mathbb{E}_{q(f_n)}\left[\log p(y_n|\mathrm{f}_n, \theta)\right]$ which may be evaluated using quadrature (Hensman et al., 2015).

The VFE approach can also be extended to employ inter-domain variables (Álvarez et al., 2010; Tobar et al., 2015; Matthews et al., 2016). The approach considers the augmented generative model $p(f, g|\theta)$ where to remind the reader the auxiliary process is defined by a linear integral transformation, $g(z) = \int w(z, z')f(z')\mathrm{d}z'$. Variational inference is now performed over both latent processes $q(f, g) = q(f, \mathbf{u}, g_{\neq \mathbf{u}}|\theta) = p(f, g_{\neq \mathbf{u}}|\mathbf{u}, \theta)q(\mathbf{u})$. Here the pseudo-data have been placed into the auxiliary process with the idea being that they can

induce richer dependencies in the original domain that model the true posterior more accurately. In fact, if the linear integral transformation is parameterised then the transformation can be learned so that it approximates the posterior more accurately.

A key concept underpinning the VFE framework is that the pseudo-input locations (and the parameters of the inter-domain transformation, if employed) are purely parameters of the approximate posterior, hence the name 'variational parameters'. This distinction is important as it means, for example, that we are free to add pseudo-data as more structure of the underlying function is revealed, without altering the modelling assumptions (e.g. see Bui et al. (2017) for an example in online inference). Moreover, since the pseudo-input locations are variational parameters, placing priors over them is unnecessary in this framework. Unlike the model parameters, optimisation of variational parameters is automatically protected from overfitting as the optimisation is minimising the KL divergence between the approximate posterior and the true posterior. Indeed, although the DTC posterior is recovered in the regression setting, as we have seen the free-energy is *not* equal to the log-marginal likelihood of the DTC generative model, containing an additional term that substantially improves the quality of the optimised pseudo-point input locations.

The facts that the form of the DTC approximation can be recovered from a direct approximate inference approach and that this new perspective leads to superior pseudo-input optimisation, raises the question: can this also be done for FITC and PITC?

2.3 Sparse GP Approximation via Approximate Inference: EP

Expectation Propagation (EP) is a deterministic inference method (Minka, 2001) that is known to outperform VFE methods in GP classification when non-sparse fully-factored approximations $q(\mathbf{f}) = \prod_n q_n(\mathrm{f}_n)$ are used (Nickisch and Rasmussen, 2008). Motivated by this observation, EP has been combined with the approximate generative modelling approach to handle non-linear likelihoods (Naish-Guzman and Holden, 2007; Hernández-Lobato and Hernández-Lobato, 2016). This begs the question: can the sparsification and the non-linear approximation be handled in a single EP inference stage, as for VFE? Astonishingly Csató and Opper not only developed such a method in 2002 (Csató and Opper, 2002), predating much of the work mentioned above, they showed that it is equivalent to applying the FITC approximation and running EP if further approximation is required. In our view, this is a central result, but it appears to have been largely overlooked by the field. Snelson was made aware of it when writing his thesis (Snelson, 2007), briefly acknowledging Csató and Opper's contribution. Qi et al. (2010) extended Csató and Opper's work to utilise inter-domain pseudo-points and they additionally recognised that the EP energy function at convergence is equal to the FITC log-marginal likelihood approximation. Interestingly, no additional term arises as it does when the VFE approach generalised the DTC generative model approach. We are unaware of other work in this vein.

It is hard to know for certain why these important results are not widely known, but a contributing factor is that the exposition in these papers is largely at Marr's algorithmic level (Dawson, 1998), and does not focus on the computational level, making them challenging to understand. Moreover, Csató and Opper's paper was written before EP was formulated in a general way and the presentation, therefore, does not follow what has become the standard

approach. In fact, as the focus was online inference, Assumed Density Filtering (Kushner and Budhiraja, 2000; Ito and Xiong, 2000) was employed rather than full-blown EP.

2.4 Contributions

One of the primary contributions of this paper is to provide a clear computational exposition of Csató and Opper's EP procedure including an explicit form of the approximating distribution and full details about each step. In addition to bringing clarity we make the following novel contributions:

- We show that a generalisation of EP called Power EP can subsume the EP and VFE approaches (and therefore FITC and DTC) into a single unified framework. More precisely, the fixed points of Power EP yield the FITC and VFE posterior distribution under different limits and the Power EP marginal likelihood estimate (the negative 'Power EP energy') recovers the FITC marginal likelihood and the VFE too. Critically, the connection to the VFE method leans on the new interpretation of Titsias's approach (Matthews et al., 2016) outlined in the previous section that directly employs the approximate posterior over function values (rather than augmenting the model with pseudo-points). The connection therefore also requires a formulation of Power EP that involves KL divergence minimisation between stochastic processes.

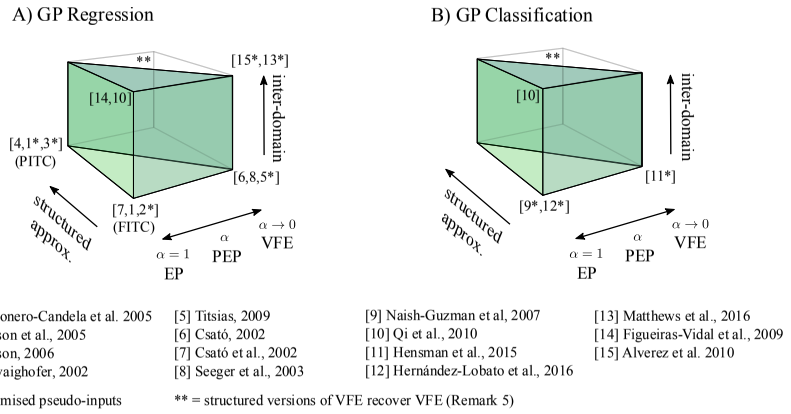
- We show how versions of PEP that are intermediate between the existing VFE and EP approaches can be derived, as well as mixed approaches that treat some data variationally and others using EP. We also show how PITC emerges from the same framework and how to incorporate inter-domain transforms. For regression with Gaussian observation noise, we obtain analytical expressions for the fixed points of Power EP in a general case that includes all of these extensions as well as the form of the Power EP marginal likelihood estimate at convergence that is useful for hyperparameter and pseudo-input optimisation.

- We consider (Gaussian) regression and probit classification as canonical models on which to test the new framework and demonstrate through exhaustive testing that versions of PEP intermediate between VFE and EP perform substantially better on average. The experiments also shed light on situations where VFE is to be preferred to EP and vice versa, an important open area of research.

Many of the new theoretical contributions described above are summarised in Figure 1 along with their relationship to previous work.

3. A New Unifying View using Power Expectation Propagation

In this section, we provide a new unifying view of sparse approximation using Power Expectation Propagation (PEP or Power EP) (Minka, 2004). We review Power EP, describe how to apply it for sparse GP regression and classification, and then discuss its relationship to existing methods.



[1] Quiñonero-Candela et al. 2005
[2] Snelson et al., 2005
[3] Snelson, 2006
[4] Schwaighofer, 2002
[5] Titsias, 2009
[6] Csató, 2002
[7] Csató et al., 2002
[8] Seeger et al., 2003
[9] Naish-Guzman et al, 2007
[10] Qi et al., 2010
[11] Hensman et al., 2015
[12] Hernández-Lobato et al., 2016
[13] Matthews et al., 2016
[14] Figueiras-Vidal et al., 2009
[15] Alverez et al. 2010

* = optimised pseudo-inputs
** = structured versions of VFE recover VFE (Remark 5)

Figure 1: A unified view of pseudo-point GP approximations applied to A) regression, and B) classification. Every point in the algorithm polygons corresponds to a form of GP approximation. Previous algorithms correspond to labelled vertices. The new Power EP framework encompasses the three polygons, including their interior.

3.1 The Joint-Distribution View of Approximate Inference and Learning

One way of understanding the goal of distributional inference approximations, including the VFE method, EP and Power EP, is that they return an approximation of a tractable form to the model *joint-distribution* evaluated on the observed data. In the case of GP regression and classification, this means $q^*(f) \approx p(f, \mathbf{y}|\theta)$ where $*$ is used to denote an unnormalised process. Why is the model joint-distribution a sensible object of approximation? The joint distribution can be decomposed into the product of the posterior distribution and the marginal likelihood, $p(f, \mathbf{y}|\theta) = p^*(f|\mathbf{y}, \theta) = p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta)$, the two inferential objects of interest. A tractable approximation to the joint can therefore be similarly decomposed $q^*(f) = Zq(f)$ into a normalised component that approximates the posterior $q(f) \approx p(f|\mathbf{y}, \theta)$ and the normalisation constant which approximates the marginal likelihood $Z \approx p(\mathbf{y}|\theta)$. In other words, the approximation of the joint simultaneously returns approximations to the posterior and marginal likelihood. In the current context, tractability of the approximating family means that it is analytically integrable and that this integration can be performed with an appropriate computational complexity. We consider the approximating family comprising unnormalised GPs, $q^*(f) = Z\ \mathcal{GP}(f; m_{\mathrm{f}}, V_{\mathrm{ff}'})$.

The VFE approach can be reformulated in the new context using the unnormalised KL divergence (Zhu and Rohwer, 1997) to measure the similarity between the approximation

and the joint distribution

$$\overline{\mathrm{KL}}[q^*(f)||p(f, \mathbf{y}|\theta)] = \int q^*(f) \log \frac{q^*(f)}{p(f, \mathbf{y}|\theta)} \mathrm{d}f + \int \left(p(f, \mathbf{y}|\theta) - q^*(f)\right) \mathrm{d}f. \tag{5}$$

The unnormalised KL divergence generalises the KL divergence to accommodate unnormalised densities. It is always non-negative and collapses back to the standard form when its arguments are normalised. Minimising the unnormalised KL with respect to $q^*(f) = Z_{\mathrm{VFE}}\, q(f)$ encourages the approximation to match both the posterior and marginal-likelihood, and it yields analytic solutions

$$q^{\mathrm{opt}}(f) = \operatorname*{argmin}_{q(f)\in\mathcal{Q}} \mathrm{KL}[q(f)||p(f|\mathbf{y}, \theta)], \quad \text{and} \quad Z^{\mathrm{opt}}_{\mathrm{VFE}} = \exp(\mathcal{F}(q^{\mathrm{opt}}(f), \theta)).$$

That is, the standard variational free-energy approximation to the posterior and marginal likelihood is recovered. One of the pedagogical advantages of framing VFE in this way is that approximation of the posterior and marginal likelihood are committed to upfront, in contrast to the traditional derivation which begins by targeting approximation of the marginal likelihood, but shows that approximation of the posterior emerges as an essential part of this scheme (see Section 2.2). A disadvantage is that optimisation of hyperparameters must logically proceed by optimising the marginal likelihood approximation, $Z^{\mathrm{opt}}_{\mathrm{VFE}}$, and at first sight therefore appears to necessitate different objective functions for $q^*(f|\theta)$ and θ (unlike the standard view which uses a single objective from the beginning). However, it is easy to show that maximising the single objective $p(\mathbf{y}|\theta) - \overline{\mathrm{KL}}[q^*(f|\theta)||p(f, \mathbf{y}|\theta)]$ directly for both $q^*(f|\theta)$ and θ is equivalent and that this also recovers the standard VFE method (see Appendix A).

3.2 The Approximating Distribution Employed by Power EP

Power EP also approximates the joint-distribution, employing an approximating family whose form mirrors that of the target,

$$p^*(f|\mathbf{y}, \theta) = p(f|\mathbf{y}, \theta)p(\mathbf{y}|\theta) = p(f|\theta) \prod_n p(y_n|f, \theta) \approx p(f|\theta) \prod_n t_n(\mathbf{u}) = q^*(f|\theta). \tag{6}$$

Here, the approximation retains the exact prior, but each likelihood term in the exact posterior, $p(y_n|\mathrm{f}_n, \theta)$, is approximated by a simple factor $t_n(\mathbf{u})$ that is assumed Gaussian. These simple factors will be iteratively refined by the PEP algorithm such that they will capture the effect that each true likelihood has on the posterior. As the approximation retains the exact prior it explicitly depends on the hyperparameters θ. However, we will suppress this dependence to lighten the notation.

Before describing the details of the PEP algorithm, it is illuminating to consider an alternative interpretation of the approximation. Together, the approximate likelihood functions specify an unnormalised Gaussian over the pseudo-points that can be written $\prod_n t_n(\mathbf{u}) = \mathcal{N}(\tilde{\mathbf{y}}; \tilde{\mathbf{W}}\mathbf{u}, \tilde{\Sigma})$ (assuming that the product of these factors is normalisable which may not be the case for heavy tailed likelihoods, for example). The approximate posterior above can therefore be thought of as the (exact) GP posterior resulting from a surrogate regression problem with surrogate observations $\tilde{\mathbf{y}}$ that are generated from linear combinations of the pseudo-points and additive surrogate noise $\tilde{\mathbf{y}} = \tilde{\mathbf{W}}\mathbf{u} + \tilde{\Sigma}^{1/2}\epsilon$. We note that the

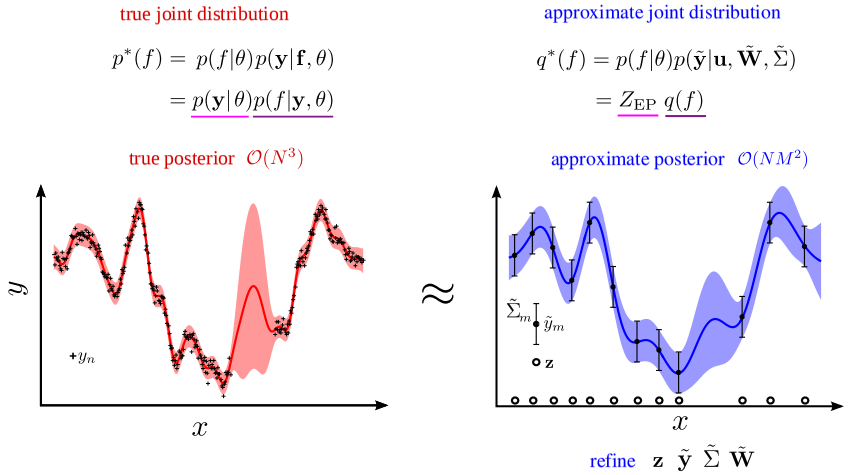


Figure 2: Perspectives on the approximating family. The true joint distribution over the unknown function f and the N data points $\mathbf{y}$ (top left) comprises the GP prior and an intractable likelihood function. This is approximated by a surrogate regression model with a joint distribution over the function f and M surrogate data points $\tilde{\mathbf{y}}$ (top right). The surrogate regression model employs the same GP prior, but uses a Gaussian likelihood function $p(\tilde{\mathbf{y}}|\mathbf{u},\tilde{\mathbf{W}},\tilde{\Sigma}) = \mathcal{N}(\tilde{\mathbf{y}};\tilde{\mathbf{W}}\mathbf{u},\tilde{\Sigma})$. The intractable true posterior (bottom left) is approximated by refining the surrogate data $\tilde{\mathbf{y}}$, their input locations $\mathbf{z}$ and the parameters of the surrogate model $\tilde{\mathbf{W}}$ and $\tilde{\Sigma}$.

pseudo-points $\mathbf{u}$ live on the latent function (or an inter-domain transformation thereof) and the surrogate observations $\tilde{\mathbf{y}}$ will not generally lie on the latent function. The surrogate observations and the pseudo-points are therefore analogous to the data $\mathbf{y}$ and the function values $\mathbf{f}$ in a normal Gaussian Process regression problem, respectively. To make the paper more specific on this point, we have defined parameters for the surrogate regression problem explicitly in Appendix H. The PEP algorithm will implicitly iteratively refine $\{\tilde{\mathbf{y}},\tilde{\mathbf{W}},\tilde{\Sigma}\}$ such that exact inference in the simple surrogate regression model returns a posterior and marginal likelihood estimate that is 'close' to that returned by performing exact inference in the intractable complex model (see Figure 2).

3.3 The EP Algorithm

One method for updating the approximate likelihood factors $t_n(\mathbf{u})$ would be to minimise the unnormalised KL divergence between the joint distribution and each of the distributions formed by replacing one of the likelihoods by the corresponding approximating factor (Li et al., 2015),

$$\underset{t_n(\mathbf{u})}{\operatorname{argmin}}\ \overline{\mathrm{KL}}\left[p(f,\mathbf{y}|\theta)\middle|\middle|\frac{p(f,\mathbf{y}|\theta)t_n(\mathbf{u})}{p(y_n|\mathrm{f}_n,\theta)}\right] = \underset{t_n(\mathbf{u})}{\operatorname{argmin}}\ \overline{\mathrm{KL}}[p^*_{\backslash n}(f)p(y_n|\mathrm{f}_n,\theta)||p^*_{\backslash n}(f)t_n(\mathbf{u})].$$

Here we have introduced the leave-one-out joint $p^*_{\backslash n}(f) = p(f,\mathbf{y}|\theta)/p(y_n|\mathrm{f}_n,\theta)$ which makes clear that the minimisation will cause the approximate factors to approximate the likelihoods in the context of the leave-one-out joint. Unfortunately, such an update is still intractable. Instead, EP approximates this idealised procedure by replacing the exact leave-one-out joint on both sides of the KL by the approximate leave-one-out joint (called the cavity) $p^*_{\backslash n}(f) \approx q^*_{\backslash n}(f) = q^*(f)/t_n(\mathbf{u})$. Not only does this improve tractability, but it also means that the new procedure effectively refines the approximating distribution directly at each stage, rather than setting the component parts in isolation,

$$\overline{\mathrm{KL}}[q^*_{\backslash n}(f)p(y_n|\mathrm{f}_n,\theta)||q^*_{\backslash n}(f)t_n(\mathbf{u})] = \overline{\mathrm{KL}}[q^*_{\backslash n}(f)p(y_n|\mathrm{f}_n,\theta)||q^*(f)].$$

However, the updates for the approximating factors are now coupled and so the updates must now be iterated, unlike in the idealised procedure. In this way, EP iteratively refines the approximate factors or surrogate likelihoods so that the GP posterior of the surrogate regression task 'best' approximates the posterior of the original regression/classification problem.

3.4 The Power EP Algorithm

Power EP is, algorithmically, a mild generalisation of the EP algorithm that instead removes (or includes) a fraction α of the approximate (or true) likelihood functions in the following steps:

1. **Deletion**: compute the cavity distribution by removing a fraction of one approximate factor, $q^*_{\backslash n}(f) \propto q^*(f)/t^\alpha_n(\mathbf{u})$.
2. **Projection**: first, compute the tilted distribution by incorporating a corresponding fraction of the true likelihood into the cavity, $\tilde{p}(f) = q^*_{\backslash n}(f)p^\alpha(y_n|\mathrm{f}_n,\theta)$. Second, project the tilted distribution onto the approximate posterior using the KL divergence for unnormalised densities,

$$q^*(f) \leftarrow \underset{q^*(f)\in\mathcal{Q}}{\operatorname{argmin}}\ \overline{\mathrm{KL}}[\tilde{p}(f)||q^*(f)].$$

   Here $\mathcal{Q}$ is the set of allowed $q^*(f)$ defined by Equation (6).
3. **Update**: compute a new fraction of the approximate factor by dividing the new approximate posterior by the cavity, $t^\alpha_{n,\text{new}}(\mathbf{u}) = q^*(f)/q^*_{\backslash n}(f)$, and incorporate this fraction back in to obtain the updated factor, $t_n(\mathbf{u}) = t^{1-\alpha}_{n,\text{old}}(\mathbf{u})t^\alpha_{n,\text{new}}(\mathbf{u})$.

The above steps are iteratively repeated for each factor that needs to be approximated. Notice that the procedure only involves one likelihood factor to be handled at a time. In the case of analytically intractable likelihood functions, this often requires only low dimensional integrals to be computed. In other words, PEP has transformed a high dimensional intractable integral that is hard to approximate into a set of low dimensional intractable

integrals that are simpler to approximate. The procedure is not guaranteed to converge in general, but we did not observe any convergence issues in our experiments. Furthermore, it can be shown to be numerically stable when the factors are log-concave (as in GP regression and classification without pseudo-data) (Seeger, 2008).

If Power EP converges, the fractional updates are equivalent to running the original EP procedure, but replacing the KL minimisation with an alpha-divergence minimisation (Zhu and Rohwer, 1995; Minka, 2005),

$$\overline{\mathrm{D}}_\alpha[p^*(f)||q^*(f)] = \frac{1}{\alpha(1-\alpha)}\int \left[\alpha p^*(f) + (1-\alpha)q^*(f) - p^*(f)^\alpha q^*(f)^{1-\alpha}\right] \mathrm{d}f.$$

When $\alpha = 1$, the alpha-divergence is the inclusive KL divergence $\overline{\mathrm{D}}_1[p^*(f)||q^*(f)] = \overline{\mathrm{KL}}[p^*(f)||q^*(f)]$ recovering EP as expected from the PEP algorithm. As $\alpha \to 0$ the exclusive KL divergence is recovered, $\overline{\mathrm{D}}_{\to 0}[p^*(f)||q^*(f)] = \overline{\mathrm{KL}}[q^*(f)||p^*(f)]$, and since minimising a set of local exclusive KL divergences is equivalent to minimising a single global exclusive KL divergence (Minka, 2005), the Power EP solution is the minimum of a variational free-energy (see Appendix B for more details). In the current case, we will now show explicitly that these cases of Power EP recover FITC and Titsias's VFE solution respectively.

3.5 General Results for Gaussian Process Power EP

This section describes the Power EP steps in finer detail showing the complexity is $\mathcal{O}(NM^2)$ and laying the ground work for the equivalence relationships. The Appendix F includes a full derivation.

We start by defining the approximate factors to be in natural parameter form, making it simple to combine and delete them, $t_n(\mathbf{u}) = \tilde{\mathcal{N}}(\mathbf{u}; z_n, \mathbf{T}_{1,n}, \mathbf{T}_{2,n}) = z_n \exp(\mathbf{u}^\intercal \mathbf{T}_{1,n} - \frac{1}{2}\mathbf{u}^\intercal \mathbf{T}_{2,n}\mathbf{u})$. We consider full rank $\mathbf{T}_{2,n}$, but will show that the optimal form is rank 1. The parameterisation means the approximate posterior over the pseudo-points has natural parameters $\mathbf{T}_{1,\mathbf{u}} = \sum_n \mathbf{T}_{1,n}$ and $\mathbf{T}_{2,\mathbf{u}} = \mathbf{K}_{\mathbf{uu}}^{-1} + \sum_n \mathbf{T}_{2,n}$ inducing an approximate posterior, $q^*(f|\theta) = \mathcal{Z}_{\mathrm{PEP}}\mathcal{GP}(f; m_{\mathrm{f}}, V_{\mathrm{ff}'})$. The mean and covariance functions of the approximate posterior are

$$m_{\mathrm{f}} = \mathbf{K}_{\mathrm{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{T}_{1,\mathbf{u}}; \quad V_{\mathrm{ff}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}'}.$$

Deletion: The cavity for data point n, $q^*_{\backslash n}(f) \propto q^*(f)/t_n^\alpha(\mathbf{u})$, has a similar form to the posterior, but the natural parameters are modified by the deletion step, $\mathbf{T}_{1,\mathbf{u}}^{\backslash n} = \mathbf{T}_{1,\mathbf{u}} - \alpha\mathbf{T}_{1,n}$ and $\mathbf{T}_{2,\mathbf{u}}^{\backslash n} = \mathbf{T}_{2,\mathbf{u}} - \alpha\mathbf{T}_{2,n}$, yielding the following mean and covariance functions

$$m_{\mathrm{f}}^{\backslash n} = \mathbf{K}_{\mathrm{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{\backslash n,-1}\mathbf{T}_{1,\mathbf{u}}^{\backslash n}; \quad V_{\mathrm{ff}'}^{\backslash n} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{\backslash n,-1}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}'}.$$

Projection: The central step in Power EP is the projection. Obtaining the new approximate unnormalised posterior $q^*(f)$ by minimising $\overline{\mathrm{KL}}[\tilde{p}(f)||q^*(f)]$ would naïvely appear intractable. Fortunately,

Remark 1 *Due to the structure of the approximate posterior, $q^*(f) = p(f_{\neq\mathbf{u}}|\mathbf{u})q^*(\mathbf{u})$, the objective, $\overline{\mathrm{KL}}[\tilde{p}(f)||q^*(f)]$ is minimised when $\mathbb{E}_{\tilde{p}(f)}[\phi(\mathbf{u})] = \mathbb{E}_{q^*(\mathbf{u})}[\phi(\mathbf{u})]$, where $\phi(\mathbf{u}) = \{\mathbf{u}, \mathbf{u}\mathbf{u}^\intercal\}$ are the sufficient statistics, that is when the moments at the pseudo-inputs are matched.*

This is the central result from which computational savings are derived. Furthermore, this moment matching condition would appear to necessitate computation of a set of integrals to find the zeroth, first and second moments. However, the technique known as 'differentiation under the integral sign'[2] provides a useful shortcut that only requires one integral to compute the log-normaliser of the tilted distribution, $\log \tilde{Z}_n = \log \mathbb{E}_{q^*_{\backslash n}(f)}[p^\alpha(y_n|\mathrm{f}_n)]$, before differentiating w.r.t. the cavity mean to give

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\frac{\mathrm{d}\log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^{\backslash n}}; \qquad \mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\frac{\mathrm{d}^2\log \tilde{Z}_n}{\mathrm{d}(m_{\mathrm{f}_n}^{\backslash n})^2}\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}. \tag{7}$$

Update: Having computed the new approximate posterior, the approximate factor $t_{n,\mathrm{new}}(\mathbf{u}) = q^*(f)/q^*_{\backslash n}(f)$ can be straightforwardly obtained, resulting in,

$$\mathbf{T}_{1,n,\mathrm{new}} = \mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} - (\mathbf{V}_{\mathbf{u}}^{\backslash n})^{-1}\mathbf{m}_{\mathbf{u}}^{\backslash n}, \ \mathbf{T}_{2,n,\mathrm{new}} = \mathbf{V}_{\mathbf{u}}^{-1} - (\mathbf{V}_{\mathbf{u}}^{\backslash n})^{-1}, \ z_n^\alpha = \tilde{Z}_n \mathrm{e}^{\mathcal{G}(q_*^{\backslash n}(\mathbf{u})) - \mathcal{G}(q^*(\mathbf{u}))},$$

where we have defined the log-normaliser as the functional $\mathcal{G}(\tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2)) = \log \int \tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2)\mathrm{d}\mathbf{u}$. Remarkably, these results and Equation (7) reveals that $\mathbf{T}_{2,n,new}$ is a rank-1 matrix. As such, the minimal and simplest way to parameterise the approximate factor is $t_n(\mathbf{u}) = z_n\mathcal{N}(\mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; g_n, v_n)$, where g_n and v_n are scalars, resulting in a significant memory saving and $\mathcal{O}(NM^2)$ cost.

In addition to providing the approximate posterior after convergence, Power EP also provides an approximate log-marginal likelihood for model selection and hyperparameter optimisation,

$$\log \mathcal{Z}_{\mathrm{PEP}} = \log \int p(f)\prod_n t_n(\mathbf{u})\mathrm{d}f = \mathcal{G}(q^*(\mathbf{u})) - \mathcal{G}(p^*(\mathbf{u})) + \sum_n \log z_n. \tag{8}$$

Armed with these general results, we now consider the implications for Gaussian Process regression.

3.6 Gaussian Regression case

When the model contains Gaussian likelihood functions, closed-form expressions for the Power EP approximate factors at convergence can be obtained and hence the approximate posterior:

$$t_n(\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; y_n, \alpha D_{\mathrm{f}_n\mathrm{f}_n} + \sigma_y^2), \ \ q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{K}_{\mathbf{fu}})$$

where $\overline{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \alpha\mathrm{diag}(\mathbf{D}_{\mathbf{ff}}) + \sigma_y^2\mathrm{I}$ and $\mathbf{D}_{\mathbf{ff}} = \mathbf{K}_{\mathbf{ff}} - \mathbf{Q}_{\mathbf{ff}}$ as defined in Section 2. These analytic expressions can be rigorously proven to be the stable fixed point of the Power EP procedure using Remark 1. Briefly, assuming the factors take the form above, the natural parameters of the cavity $q^*_{\backslash n}(\mathbf{u})$ become,

$$\mathbf{T}_{1,\mathbf{u}}^{\backslash n} = \mathbf{T}_{1,\mathbf{u}} - \alpha\gamma_n y_n \mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}, \quad \mathbf{T}_{2,\mathbf{u}}^{\backslash n} = \mathbf{T}_{2,\mathbf{u}} - \alpha\gamma_n \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}_n}\mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1},$$

2. In this case, the dominated convergence theorem can be used to justify the interchange of integration and differentiation (see e.g. Brown, 1986).

where $\gamma_n^{-1} = \alpha D_{\mathrm{f}_n\mathrm{f}_n} + \sigma_y^2$. The subtracted quantities in the equations above are exactly the contribution the likelihood factor makes to the cavity distribution (see Remark 1) so $\int q^*_{\backslash n}(f) p^\alpha(y_n|\mathrm{f}_n)\mathrm{d}f_{\neq \mathbf{u}} = q^*_{\backslash n}(\mathbf{u}) \int p(\mathrm{f}_n|\mathbf{u}) p^\alpha(y_n|\mathrm{f}_n)\mathrm{d}\mathrm{f}_n \propto q^*(\mathbf{u})$. Therefore, the posterior approximation remains unchanged after an update and the form for the factors above is the fixed point. Moreover, the approximate log-marginal likelihood is also analytically tractable,

$$\log \mathcal{Z}_{\mathrm{PEP}} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\overline{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^\intercal \overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y} - \frac{1-\alpha}{2\alpha}\sum_n \log\left(1 + \alpha D_{\mathrm{f}_n\mathrm{f}_n}/\sigma_y^2\right).$$

We now look at special cases and the correspondence to the methods discussed in Section 2.

Remark 2 *When $\alpha = 1$ [EP], the Power EP posterior becomes the FITC posterior in Equation* (1) *and the Power EP approximate marginal likelihood becomes the FITC marginal likelihood in Equation* (2)*. In other words, the FITC approximation for GP regression is, surprisingly, equivalent to running an EP algorithm for sparse GP posterior approximation to convergence.*

Remark 3 *As $\alpha \to 0$ the approximate posterior and approximate marginal likelihood are identical to that of the VFE approach in Equations* (3) *and* (4) *(Titsias, 2009). This result uses the limit: $\lim_{x\to 0} x^{-1}\log(1+x) = 1$. So FITC and Titsias's VFE approach employ the same form of pseudo-point approximation, but refine it in different ways.*

Remark 4 *For fixed hyperparameters, a single pass of Power EP is sufficient for convergence in the regression case.*

3.7 Extensions: Structured, Inter-domain and Multi-power Power EP Approximations

The framework can now be generalised in three orthogonal directions:

1. enable structured approximations to be handled that retain more dependencies in the spirit of PITC (see Section 2.1)
2. incorporate inter-domain pseudo-points thereby adding further flexibility to the form of the approximate posterior
3. employ different powers α for each factor (thereby enabling e.g. VFE updates to be used for some data points and EP for others).

Given the groundwork above, these three extensions are straightforward. In order to handle structured approximations, we take inspiration from PITC and partition the data into B disjoint blocks $\mathbf{y}_b = \{y_n\}_{n\in\mathcal{B}_b}$ (see Section 2.1). Each PEP factor update will then approximate an entire block which will contain a set of data points, rather than just a single one. This is related to a form of EP approximation that has recently been used to distribute Monte Carlo algorithms across many machines (Gelman et al., 2014; Xu et al., 2014).

In order to handle inter-domain variables, we define a new domain via a linear transform $g(\mathbf{x}) = \int \mathrm{d}\mathbf{x}' W(\mathbf{x},\mathbf{x}') f(\mathbf{x}')$ which now contains the pseudo-points $g = \{g_{\neq\mathbf{u}}, \mathbf{u}\}$. Choices for

$W(\mathbf{x},\mathbf{x}')$ include Gaussians or wavelets. These two extensions mean that the approximation becomes,

$$p(f,g)\prod_b p(\mathbf{y}_b|f) \approx p(f,g)\prod_b t_b(\mathbf{u}) = q^*(f).$$

Power EP is then performed using private powers α_b for each data block, which is the third generalisation mentioned above. Analytic solutions are again available (covariance matrices now incorporate the inter-domain transform)

$$t_b(\mathbf{u}) = \mathcal{N}(\mathbf{K}_{\mathbf{f}_b\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; \mathbf{y}_b, \alpha_b\mathbf{D}_{\mathbf{f}_b\mathbf{f}_b} + \sigma_y^2\mathrm{I}), \quad q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uf}}\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{K}_{\mathbf{fu}}),$$

where $\overline{\mathbf{K}}_{\mathbf{ff}} = \mathbf{Q}_{\mathbf{ff}} + \mathrm{blkdiag}(\{\alpha_b\mathbf{D}_{\mathbf{f}_b\mathbf{f}_b}\}_{b=1}^B) + \sigma_y^2\mathrm{I}$ and blkdiag builds a block-diagonal matrix from its inputs. The approximate log-marginal likelihood can also be obtained in closed-form,

$$\log \mathcal{Z}_{\mathrm{PEP}} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\overline{\mathbf{K}}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^\intercal\overline{\mathbf{K}}_{\mathbf{ff}}^{-1}\mathbf{y} + \sum_b \frac{1-\alpha_b}{2\alpha_b}\log\left(\mathrm{I} + \alpha_b\mathbf{D}_{\mathbf{f}_b\mathbf{f}_b}/\sigma_y^2\right).$$

Remark 5 *When $\alpha_b = 1$ and $W(\mathbf{x},\mathbf{x}') = \delta(\mathbf{x}-\mathbf{x}')$ the structured Power EP posterior becomes the PITC posterior and the Power EP approximate marginal likelihood becomes the PITC marginal likelihood. Additionally, when $B = N$ we recover FITC as discussed in Section 3.6.*

Remark 6 *When $\alpha_b \to 0$ and $W(\mathbf{x},\mathbf{x}') = \delta(\mathbf{x}-\mathbf{x}')$ the structured Power EP posterior and approximate marginal likelihood becomes identical to the VFE approach (Titsias, 2009). This is a result of the equivalence of local and global exclusive KL divergence minimisation. See Appendix B for more details and Figure 1 for more relationships.*

3.8 Classification

For classification, the non-Gaussian likelihood prevents an analytic solution. As such, the iterative Power EP procedure is required to obtain the approximate posterior. The projection step requires computation of the log-normaliser of the tilted distribution, $\log \tilde{Z}_n = \log \mathbb{E}_{q^*_{\backslash n}(f)}[p^\alpha(y_n|f)] = \log \mathbb{E}_{q^*_{\backslash n}(\mathrm{f}_n)}[\Phi^\alpha(y_n\mathrm{f}_n)]$. For general α, this quantity is not available in closed form[3]. However, it involves a one-dimensional expectation of a non-linear function of a normally-distributed random variable and, therefore, can be approximated using numerical methods, e.g. Gauss-Hermite quadrature. This procedure gives an approximation to the expectation, resulting in an approximate update for the posterior mean and covariance. The approximate log-marginal likelihood can also be obtained and used for hyperparameter optimisation. As $\alpha \to 0$, it becomes the variational free-energy used in Hensman et al. (2015) which employs quadrature for the same purpose. These relationships are shown in Figure 1 which also shows that inter-domain transformations and structured approximations have

3. except for special cases, e.g. when $\alpha = 1$ and $\Phi(x)$ is the probit inverse link function, $\Phi(x) = \int_{-\infty}^x \mathcal{N}(a;0,1)\mathrm{d}a$.

not yet been fully explored in the classification setting. In our view, the inter-domain generalisation would be a sensible one to pursue and it is mathematically and algorithmically straightforward. The structured approximation variant is more complicated as it requires multiple non-linear likelihoods to be handled at each step of EP. This will require further approximation such as using Monte Carlo methods (Gelman et al., 2014; Xu et al., 2014). In addition, when $\alpha = 1$, $M = N$ and the pseudo-points are at the training inputs, the standard EP algorithm for GP classification is recovered (Rasmussen and Williams, 2005, sec. 3.6).

Since the proposed Power EP approach is general, an extension to other likelihood functions is as simple as for VFE methods (Dezfouli and Bonilla, 2015). For example, the multinomial probit likelihood can be handled in the same way as the binary case, where the log-normaliser of the tilted distribution can be computed using a C-dimensional Gaussian quadrature [C is the number of classes] (Seeger and Jordan, 2004) or nested EP (Riihimäki et al., 2013).

3.9 Complexity

The computational complexity of all the regression and classification methods described in this section is $\mathcal{O}(NM^2)$ for training, and $\mathcal{O}(M^2)$ per test point for prediction. The training cost can be further reduced to $\mathcal{O}(M^3)$, in a similar vein to the uncollapsed VFE approach (Hensman et al., 2013, 2015), by employing stochastic updates of the posterior and stochastic optimisation of the hyperparameters using minibatches of data points (Hernández-Lobato and Hernández-Lobato, 2016). In particular, the Power EP update steps in Section 3.2 are repeated for only a small subset of training points and for only a small number of iterations. The approximate log-marginal likelihood in Equation (8) is then computed using this minibatch and optimised as if the Power EP procedure has converged. This approach results in a computationally efficient training scheme, at the cost of returning noisy hyperparameter gradients. In practice, we find that the noise can be handled using stochastic optimisers such as Adam (Kingma and Ba, 2015). In summary, given these advances the general PEP framework is as scalable as variational inference.

4. Experiments

The general framework described above lays out a large space of potential inference algorithms suggesting many exciting directions for innovation. The experiments considered in the paper will investigate only one aspect of this space; how do algorithms that are intermediate between VFE ($\alpha = 0$) and EP/FITC ($\alpha = 1$) perform? Specifically, we will investigate how the performance of the inference scheme varies as a function of α and whether this depends on: the type of problem (classification or regression); the data set (synthetic data sets, 8 real world regression data sets and 6 classification data sets); the performance metric (we compare metrics that require point-estimates to those that are uncertainty sensitive). An important by-product of the experiments is that they provide a comprehensive comparison between the VFE and EP approaches which has been an important area of debate in its own right.

The results presented below are compact summaries of a large number of experiments full details of which are included in Appendix I (along with additional experiments). Python

and Matlab implementations are available at `http://github.com/thangbui/sparseGP_powerEP`.

4.1 Regression on Synthetic Data Sets

In the first experiment, we investigate the performance of the proposed Power EP method on toy regression data sets where ground truth is known. We vary α (from 0 VFE to 1 EP/FITC) and the number of pseudo-points (from 5 to 500). We use thirty data sets, each comprising 1000 data points with five input dimensions and one output dimension, that were drawn from a GP with an Automatic Relevance Determination squared exponential kernel. A 50:50 train/test split was used. The hyperparameters and pseudo-inputs were found by optimising the PEP energy using L-BFGS with a maximum of 2000 function evaluations. The performances are compared using two metrics: standardised mean squared error (SMSE) and standardised mean log loss (SMLL) as described in Rasmussen and Williams (2005). The approximate negative log-marginal likelihood (NLML) for each experiment is also computed. The mean performance using Power EP with different α values and full GP regression is shown in Figure 3. The results demonstrate that as M increases, the SMLL and SMSE of the sparse methods approach that of full GP. Power EP with $\alpha = 0.8$ or $\alpha = 1$ (EP) overestimates the log-marginal likelihood when intermediate numbers of pseudo-points are used, but the overestimation is markedly less when $M = N = 500$. Importantly, however, an intermediate value of α in the range 0.5-0.8 seems to be best for prediction on average, outperforming both EP and VFE.

4.2 Regression on Real-world Data Sets

The experiment above was replicated on 8 UCI regression data sets, each with 20 train/test splits. We varied α between 0 and 1, and M was varied between 5 and 200. Full details of the experiments along with extensive additional analysis is presented in the appendices. Here we concentrate on several key aspects. First we consider pairwise comparisons between VFE ($\alpha \to 0$), Power EP with $\alpha = 0.5$ and EP/FITC ($\alpha = 1$) on both the SMSE and SMLL evaluation metrics. Power EP with $\alpha = 0.5$ was chosen because it is the mid-point between VFE and EP and because settings around this value empirically performed the best on average across all data sets, splits, numbers of inducing points, and evaluation metrics.

In Figure 4A we plot (for each data set, each split and each setting of M) the evaluation scores obtained using one inference algorithm (e.g. PEP $\alpha = 0.5$) against the score obtained using another (e.g. VFE $\alpha = 0$). In this way, points falling below the identity line indicate experiments where the method on the y-axis outperformed the method on the x-axis. These results have been collapsed by forming histograms of the difference in the performance of the two algorithms, such that mass to the right of zero indicates the method on the y-axis outperformed that on the x-axis. The proportion of mass on each side of the histogram, also indicated on the plots, shows in what fraction of experiments one method returns a more accurate result than the other. This is a useful summary statistic, linearly related to the average rank, that we will use to unpack the results. The average rank is insensitive to the magnitude of the performance differences and readers might worry that this might give an overly favourable view of a method that performs the best frequently, but only by a tiny margin, and when it fails it does so catastrophically. However, the histograms indicate that

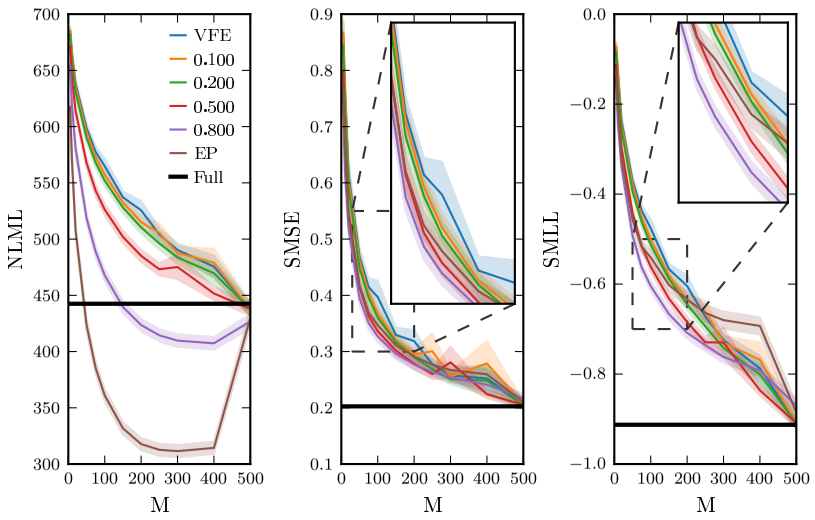


Figure 3: The performance of various α values averaged over 30 trials. See text for more details

the methods that win most frequently tend also to 'win big' and 'lose small', although EP is a possible exception to this trend (see the outliers below the identity line on the bottom right-hand plot).

A clear pattern emerges from these plots. First, PEP $\alpha = 0.5$ is the best performing approach on the SMSE metric, outperforming VFE 67% of the time and EP 78% of the time. VFE is better than EP on the SMSE metric 64% of the time. Second, EP performs the best on the SMLL metric, outperforming VFE 93% of the time and PEP $\alpha = 0.5$ 71% of the time. PEP $\alpha = 0.5$ outperforms VFE in terms of the SMLL metric 93% of the time.

These pairwise rank comparisons have been extended to other values of α in Figure 5A. Here, each row of the figure compares one approximation with all others. Horizontal bars indicate that the methods have equal average rank. Upward sloping bars indicate the method shown on that row has lower average rank (better performance), and downward sloping bars indicate higher average rank (worse performance). The plots show that PEP $\alpha = 0.5$ outperforms all other methods on the SMSE metric, except for PEP $\alpha = 0.6$ which is marginally better. EP is outperformed by all other methods, and VFE only outperforms EP on this metric. On the other hand, EP is the clear winner on the SMLL metric, with performance monotonically decreasing with α so that VFE is the worst.

The same pattern of results is seen when we simultaneously compare all of the methods, rather than considering sets of pairwise comparisons. The average rank plots shown in Figure 4B were produced by sorting the performances of the nine different approximating

methods for each data set, split, and number of pseudo-points M and assigning a rank. These ranks are then averaged over all data sets and their splits, and settings of M. PEP $\alpha = 0.5$ is the best for the SMSE metric, and the two worst methods are EP and VFE. PEP $\alpha = 0.8$ is the best for the SMLL metric, with EP and PEP $\alpha = 0.6$ not far behind (when EP performs poorly it can do so with a large magnitude, explaining the discrepancy with the pairwise ranks).

There is some variability between individual data sets, but the same general trends are clear: For MSE $\alpha = 0.5$ is better than VFE on 6/8 data sets and EP on 8/8 data sets, whilst VFE is better than EP on 3 data sets (the difference on the others being small). For SMLL EP is better than $\alpha = 0.5$ on 5/8 data sets and VFE on 7/8 data sets, whilst $\alpha = 0.5$ is better than VFE on 8/8 data sets. Performance tends to increase for all methods as a function of the number of pseudo-points M. The interaction between the choice of M and the best performing inference method is often complex and variable across data sets making it hard to give precise advice about selecting α in an M dependent way.

In summary, we make the following recommendations based on these results for GP regression problems. For a MSE loss, we recommend using $\alpha = 0.5$. For SMLL we recommend using EP. It is possible that more fine grained recommendations are possible based upon details of the data set and the computational resources available for processing, but further work will be needed to establish this.

4.3 Binary Classification

We also evaluated the Power EP method on 6 UCI classification data sets, each has 20 train/test splits. The details of the data sets are included in Appendix I.3. The data sets are all roughly balanced, and the most imbalanced is `pima` with 500 positive and 267 negative data points. Again α was varied between 0 and 1, and M was varied between 10 and 100. We adopt the experimental protocol discussed in Section 3.9, including: (i) not waiting for Power EP to converge before making hyperparameter updates, (ii) using minibatches of data points for each Power EP sweep, (iii) parallel factor updates. The Adam optimiser was used with default hyperparameters to handle the noisy gradients produced by these approximations (Kingma and Ba, 2015). We also implemented the VFE approach of Hensman et al. (2015) and include this in the comparison to the PEP methods. The VFE approach should be theoretically identical to PEP with small α, however, we note that the results can be slightly different due to differences in the implementation—optimisation for VFE vs. the iterative PEP procedure, and we also note that each step of PEP only gets to see a tiny fraction of each data point when α is small which can slow the learning speed. Similar to the regression experiment, we compare the methods using the pairwise ranking plots on the test error and negative log-likelihood (NLL) evaluation metrics.

In Figure 6, we plot (for each data set, each split and each setting of M) the evaluation scores using one inference algorithm against the score obtained using another [see Section 4.2 for a detailed explanation of the plots]. In contrast to the regression results in Section 4.2, there are no clear-cut winners among the methods. The test error results show that PEP $\alpha = 0.5$ is marginally better than VFE and EP, while VFE edges EP out in this metric. Similarly, all methods perform comparably on the NLL scale, except with PEP $\alpha = 0.5$ outperforming EP by a narrow margin (65% of the time vs. 35%)

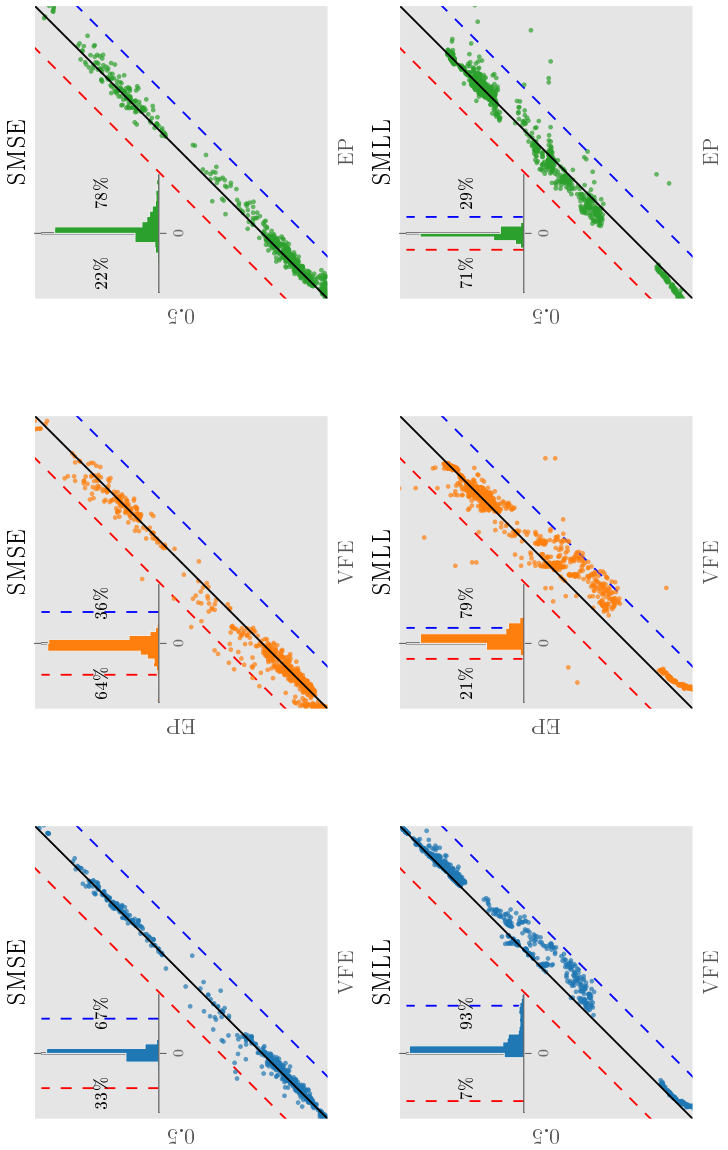


Figure 4: Pair-wise comparisons between Power EP with $\alpha = 0.5$, EP ($\alpha = 1$) and VFE ($\alpha \to 0$), evaluated on several regression data sets and various settings of M. Each coloured point is the result for one split. Points that are below the diagonal line illustrate the method on the y-axis is better than the method on the x-axis. The inset diagrams show the histograms of the difference between methods (x-value $-$ y-value), and the counts of negative and positive differences. Note that this indicates pairwise ranking of the two methods. Positive differences mean the y-axis method is better than the x-axis method and vice versa. For example, the middle, bottom plot shows EP is on average better than VFE.

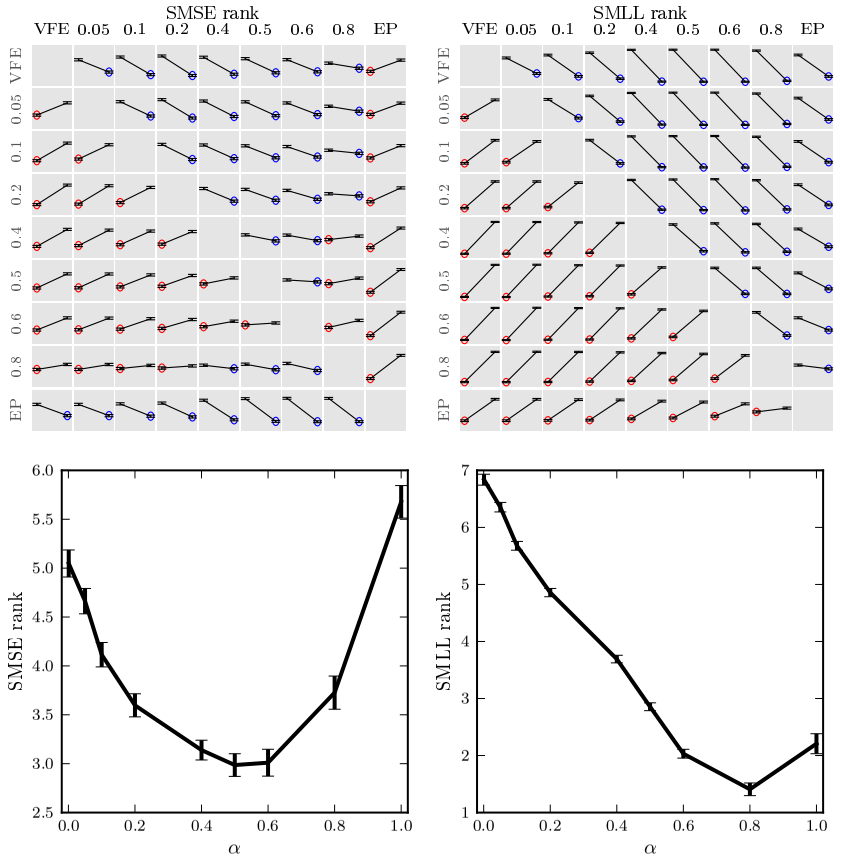


Figure 5: Average ranking of various α values in the regression experiment, lower is better. Top plots show the pairwise comparisons. Red circles denote rows being better than the corresponding columns, and blue circles mean vice versa. Bottom plots show the ranks of all methods when being compared together. Intermediate α values (not EP or VFE) are best on average.

We repeat the pairwise comparison above to all methods and show the results in Figure 7. The plots show that there is no conclusive winner on the test error metric with VFE and PEP $\alpha \leq 0.5$ performing similarly. On the NLL metric VFE, PEP $\alpha = 0.4$ and PEP $\alpha = 0.5$ have a slight edge over other α values. Notably, methods corresponding to bigger α values, such as PEP $\alpha = 0.8$ and EP, are outperformed by all other methods. Similar to the regression experiment, we observe the same pattern of results when all methods are simultaneously compared, as shown in Figure 7. However, the large errorbars suggest the difference between the methods is small in both metrics.

There is some variability between individual data sets, but the general trends are clear and consistent with the pattern noted above. For test error, PEP $\alpha = 0.5$ is better than VFE on 1/6 data set and is better than EP on 3/6 data sets (the differences on the other data sets are small). VFE outperforms EP on 2/6 data sets, while EP beats VFE on only 1/6 data sets. For NLL, PEP $\alpha = 0.5$ only clearly outperforms VFE on 1/6 data set, but is worse compared to VFE on 1 data set (the other 4 data sets have no clear winner). PEP $\alpha = 0.5$ is better than EP on 5/6 data sets and EP is better on the remaining data set). EP is only better than VFE on 2/6 data sets, and is outperformed by VFE on the other 4/6 data sets. The finding that PEP and VFE are slightly better than EP on the NLL metric is surprising as we expected EP perform the best on the uncertainty sensitive metric (just as was discovered in the regression case). The full results are included in the appendices (see figs 25, 26 and 27). Similar to the regression case, we observe that as M increases, the performance tends to be better for all methods and the differences between the methods tend to become smaller, but we have not found evidence for systematic sensitivity to the nature of the approximation.

In summary, we make the following recommendations based on these results for GP classification problems. For a raw test error loss and for NLL, we recommend using $\alpha = 0.5$ (or $\alpha = 0.4$). It is possible that more fine grained recommendations are possible based upon details of the data set and the computational resources available for processing, but further work will be needed to establish this.

5. Discussion

It is difficult to identify precisely where the best approximation methods derive their advantages, but here we will speculate. Since the negative variational free-energy is a lower-bound on the log-marginal likelihood it has the enviable theoretical guarantee that pseudo-input optimisation is always guaranteed to improve the estimate of the log marginal likelihood and the posterior (as measured by the inclusive KL). The negative EP energy, in contrast, is not generally a lower bound which can mean that pseudo-input optimisation drives the solution to the point where the EP energy over-estimates the log marginal likelihood the most, rather than to the point where the marginal likelihood and/or posterior estimate is best. For this reason, we believe that variational methods are likely to be better than EP if the goal is to derive accurate marginal likelihood estimates, or accurate predictive distributions, for fixed hyperparameter settings. For hyperparameter optimisation, things are less clear-cut since variational methods are biased away from the maximal marginal likelihood, towards hyperparameter settings for which the posterior approximation is accurate. Often this bias is severe and also creates local-optima (Turner and Sahani, 2011). So,

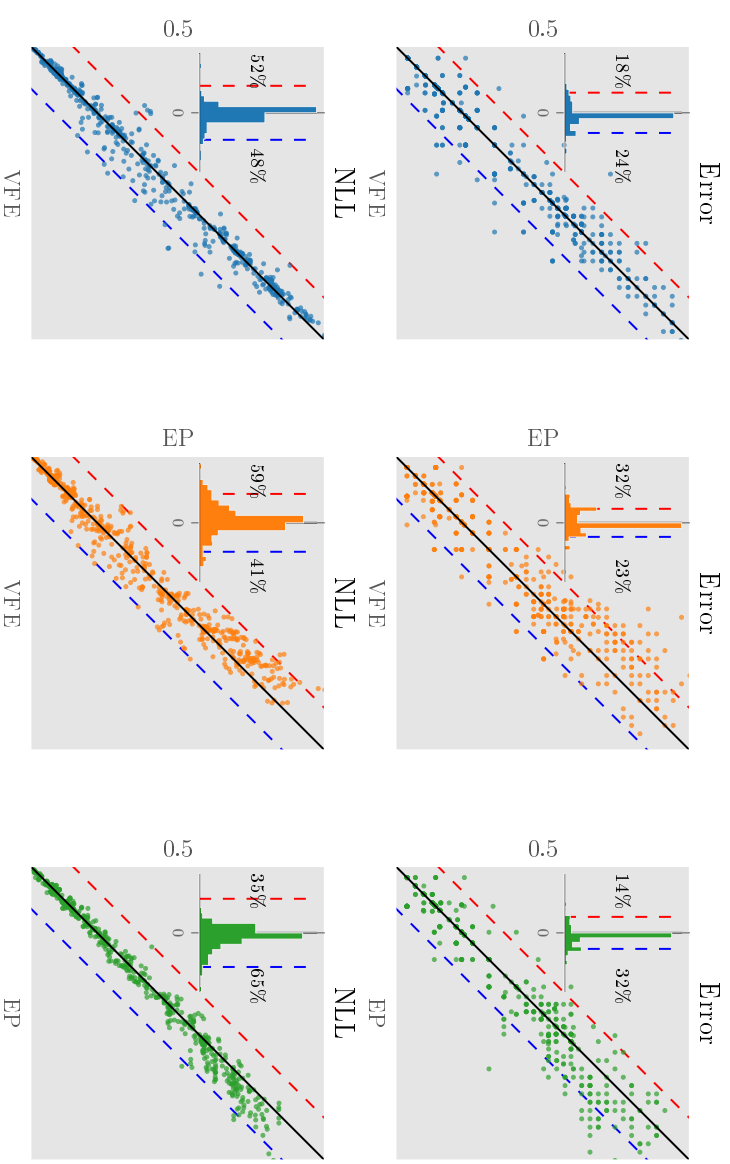


Figure 6: Pair-wise comparisons between Power EP with $\alpha = 0.5$, EP ($\alpha = 1$) and VFE ($\alpha \to 0$), evaluated on several classification data sets and various settings of M. Each coloured point is the result for one split. Points that are below the diagonal line illustrate the method on the y-axis is better than the method on the x-axis. The inset diagrams show the histograms of the difference between methods (x-value $-$ y-value), and the counts of negative and positive differences. Note that this indicates pairwise ranking of the two methods. Positive differences means the y-axis method is better than the x-axis method and vice versa.

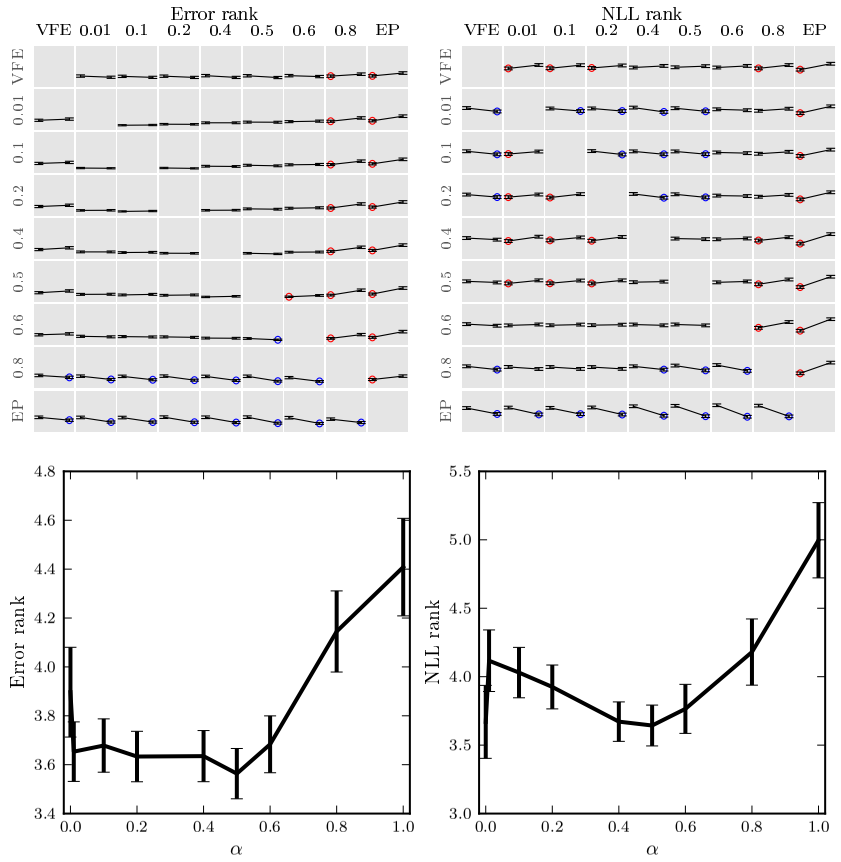


Figure 7: Average ranking of various α values in the classification experiment, lower is better. Top plots show the pairwise comparisons. Red circles denote rows being better than the corresponding columns, and blue circles mean vice versa. Bottom plots show the ranks of all methods when being compared together. Intermediate α values (not EP or VFE) are best on average.

although EP will generally also be biased away from the maximal marginal likelihood and potentially towards areas of over-estimation, it can still outperform variational methods. Superposed onto these factors, is a general trend for variational methods to minimise MSE or classification error-rate and EP methods to minimise negative log-likelihood, due to the form of their respective energies (the variational free-energy includes the average training MSE in the regression case, for example). Intermediate methods will blend the strengths and weaknesses of the two extremes. It is interesting that values of α around a half are arguably the best performing on average. Similar empirical conclusions have been made elsewhere (see e.g. Minka, 2005; Hernández-Lobato et al., 2016; Depeweg et al., 2016). In this case, the alpha-divergence interpretation of Power EP shows that it is minimising the Hellinger distance whose square root is a valid distance metric. Further experimental and theoretical work is required to clarify these issues.

The results presented above employed (approximate) type-II maximum likelihood fitting of the hyperparameters. This estimation method is known in some circumstances to overfit the data. It is therefore conceivable therefore that pseudo-point approximations, which have a tendency to encourage under-fitting due to their limited representational capacity, could be beneficial due to them mitigating overfitting. We do not believe that this is a strong effect in the experiments above. For example, in the synthetic data experiments the NLML, SMSE and SMLL obtained from fitting the unapproximated GP were similar to those obtained using the GP from which the data were generated, indicating that overfitting is not a strong effect (see fig. 9 in the appendix). It is true that EP and $\alpha = 0.8$ over-estimates the marginal likelihood in the synthetic data experiments, but this is a distinct effect from over-fitting which would, for example, result in overconfident predictions on the test data set. The SMSE and SMLL on the training and test sets, for example, are similar which is indicative of a well-fit model. It would be interesting to explore distributional hyperparameter estimates (see e.g. Piironen and Vehtari, 2017) that employ these pseudo-point approximations.

One of the features of the approximate generative models introduced in Section 2.1 for regression, is that they contain input-dependent noise, unlike the original model. Many data sets contain noise of this sort and so approximate models like FITC and PITC, or models in which the observation noise is explicitly modelled are arguably more appropriate than the original unapproximated regression model (Snelson, 2007; Saul et al., 2016). Motivated by this train of reasoning, Titsias (2009) applied the variational free-energy approximation to the FITC generative model, an approach that was later generalised by Hoang et al. (2016) to encompass a more general class of input dependent noise, including Markov structure (Low et al., 2015). Here the insight is that the resulting variational lower bound separates over data points (Hensman et al., 2013) and is, therefore, amenable to stochastic optimisation using minibatches, unlike the marginal likelihood. In a sense, these approaches unify the approximate generative modelling approach, including the FITC and PITC variants, with the variational free-energy methods. Indeed, one approach is to posit the desired form of the optimal variational posterior, and to work backwards from this to construct the generative model implied (Hoang et al., 2016). However, these approaches are quite different from the one described in this paper where FITC and PITC are shown to emerge in the context of approximating the original unapproximated GP regression model using Power EP. Indeed, if the goal really is to model input dependent noise, it is not at all clear that generative models

like FITC are the most sensible. For example, FITC uses a single set of hyperparameters to describe the variation of the underlying function and the input dependent noise.

6. Conclusion

This paper provided a new unifying framework for GP pseudo-point approximations based on Power EP that subsumes many previous approaches including FITC, PITC, DTC, Titsias's VFE method, Qi et al's EP method, and inter-domain variants. It provided a clean computational perspective on the seminal work of Csató and Opper that related FITC to EP, before extending their analysis significantly to include a closed form Power EP marginal likelihood approximation for regression, connections to PITC, and further results on classification and GPSSMs. The new framework was used to devise new algorithms for GP regression and GP classification. Extensive experiments indicate that intermediate values of Power EP with the power parameter set to $\alpha = 0.5$ often outperform the state-of-the-art EP and VFE approaches. The new framework suggests many interesting directions for future work in this area that we have not explored, for example, extensions to online inference, combinations with special structured matrices (e.g. circulant and Kronecker structure), Bayesian hyperparameter learning, and applications to richer models. The current work has only scratched the surface, but we believe that the new framework will form a useful theoretical foundation for the next generation of GP approximation schemes.

Acknowledgments

The authors would like to thank Prof. Carl Edward Rasmussen, Nilesh Tripuraneni, Matthias Bauer, James Hensman, and Hugh Salimbeni for insightful comments and discussion. TDB thanks Google for funding his European Doctoral Fellowship. RET thanks EPSRC grants EP/G050821/1, EP/L000776/1 and EP/M026957/1.

Appendix A. A Unified Objective for Unnormalised KL Variational Free-energy Methods

Here we show that performing variational inference by optimising the unnormalised KL naturally leads to a single objective for both the approximation to the joint distribution, $q^*(f|\theta)$ and the hyperparameters θ.

The unnormalised KL is given by

$$\overline{\mathrm{KL}}[q^*(f|\theta)||p(f,\mathbf{y}|\theta)] = \int q^*(f|\theta)\log\frac{q^*(f|\theta)}{p(f,\mathbf{y}|\theta)}\mathrm{d}f + \int \left(p(f,\mathbf{y}|\theta) - q^*(f|\theta)\right)\mathrm{d}f.$$

This is intractable as it includes the marginal likelihood $p(\mathbf{y}|\theta) = \int p(f,\mathbf{y}|\theta)\mathrm{d}f$. However, since we are interested in minimising this objective with respect to $q^*(f|\theta)$ we can ignore the intractable term,

$$\begin{aligned}\operatorname*{argmin}_{q^*(f|\theta)} \overline{\mathrm{KL}}[q^*(f|\theta)||p(f,\mathbf{y}|\theta)] &= \operatorname*{argmax}_{q^*(f|\theta)} \left(p(\mathbf{y}|\theta) - \overline{\mathrm{KL}}[q^*(f|\theta)||p(f,\mathbf{y}|\theta)]\right) \\ &= \operatorname*{argmax}_{q^*(f|\theta)} \left(\int q^*(f|\theta)\log\frac{p(f,\mathbf{y}|\theta)}{q^*(f|\theta)}\mathrm{d}f + \int q^*(f|\theta)\mathrm{d}f\right).\end{aligned}$$

In other words, we have turned the unnormalised KL into a tractable lower-bound of the marginal likelihood $\mathcal{G}(q^*(f|\theta),\theta) = p(\mathbf{y}|\theta) - \overline{\mathrm{KL}}[q^*(f|\theta)||p(f,\mathbf{y}|\theta)]$. The structure of this new lower-bound can be understood by decomposing the approximation to the joint distribution into a normalised posterior approximation $q(f|\theta)$ and an approximation to the marginal likelihood, Z_{VFE}, that is $q^*(f|\theta) = Z_{\mathrm{VFE}}\, q(f|\theta)$:

$$\mathcal{G}(Z_{\mathrm{VFE}}q(f|\theta),\theta) = Z_{\mathrm{VFE}}\left(1 - \log Z_{\mathrm{VFE}} + \int q(f|\theta)\log\frac{p(f,\mathbf{y}|\theta)}{q(f|\theta)}\mathrm{d}f\right).$$

We can see that optimising the lower-bound with respect to θ is equivalent to optimising the standard variational free-energy $\mathcal{F}(q(f|\theta),\theta) = \int q(f|\theta)\log\frac{p(f,\mathbf{y}|\theta)}{q(f|\theta)}\mathrm{d}f$. Moreover, optimising for Z_{VFE} recovers $Z^{\mathrm{opt}}_{\mathrm{VFE}} = \exp(\mathcal{F}(q(f|\theta),\theta))$. Substituting this back into the bound

$$\mathcal{G}(Z^{\mathrm{opt}}_{\mathrm{VFE}}q(f|\theta),\theta) = Z^{\mathrm{opt}}_{\mathrm{VFE}} = \exp(\mathcal{F}(q(f|\theta),\theta)).$$

In other words, the new collapsed bound is just the exponential of the original variational free-energy and optimising the collapsed bound for θ is equivalent to optimising the approximation to the marginal likelihood.

Appendix B. Global and Local Inclusive KL Minimisations

In this section, we will show that optimising a single global inclusive KL divergence, $\mathrm{KL}[q||p]$, is equivalent to optimising a sum of a set of local inclusive KL divergence, $\mathrm{KL}[q||\tilde{p}]$, where p, q and $\tilde{p}$ are the exact posterior, the approximate posterior and the tilted distribution accordingly. Without loss of generality, we assume that $p(\theta) = \prod_n f_n(\theta) \approx \prod_n t_n(\theta) = q(\theta)$, that is the exact posterior is a product of factors, $\{f_n(\theta)\}_n$, each of which is approximated by an approximate factor $t_n(\theta)$. Substituting these distributions into the global KL divergence

gives,

$$\begin{aligned}
\mathrm{KL}[q(\theta)||p(\theta)] &= \int \mathrm{d}\theta q(\theta) \log \frac{q(\theta)}{p(\theta)} \\
&= \int \mathrm{d}\theta q(\theta) \log \frac{\prod_n t_n(\theta)}{\prod_n f_n(\theta)} \\
&= \int \mathrm{d}\theta q(\theta) \log \left[\frac{\prod_n t_n(\theta)}{\prod_n f_n(\theta)} \frac{\prod_n \prod_{i\neq n} t_i(\theta)}{\prod_n \prod_{i\neq n} t_i(\theta)}\right] \\
&= \int \mathrm{d}\theta q(\theta) \log \frac{\prod_n [\prod_i t_i(\theta)]}{\prod_n [f_n(\theta) \prod_{i\neq n} t_i(\theta)]} \\
&= \sum_n \int \mathrm{d}\theta q(\theta) \log \frac{\prod_i t_i(\theta)}{f_n(\theta) \prod_{i\neq n} t_i(\theta)} \\
&= \sum_n \mathrm{KL}[q(\theta)||\tilde{p}_n(\theta)],
\end{aligned}$$

which means running the EP procedure, where we use $\mathrm{KL}[q(\theta)||\tilde{p}_n(\theta)]$ in place of $\mathrm{KL}[\tilde{p}_n(\theta)||q(\theta)]$, is *equivalent* to the VFE approach which optimises a single global KL divergence, $\mathrm{KL}[q(\theta)||p(\theta)]$.

Appendix C. Some Relevant Linear Algebra and Function Expansion Identities

The Woodbury matrix identity or Woodbury formula is:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}. \tag{9}$$

In general, C need not be invertible, we can use the Binomial inverse theorem,

$$(A + UCV)^{-1} = A^{-1} - A^{-1}UC(C + CVA^{-1}UC)^{-1}CVA^{-1}. \tag{10}$$

When C is an identity matrix and U and V are vectors, the Woodbury identity can be shortened and become the Sherman-Morrison formula,

$$(A + uv^\intercal)^{-1} = A^{-1} - \frac{A^{-1}uv^\intercal A^{-1}}{1 + v^\intercal A^{-1}u}. \tag{11}$$

Another useful identity is the matrix determinant lemma,

$$\det(A + uv^\intercal) = (1 + v^\intercal A^{-1}u)\det(A). \tag{12}$$

The above theorem can be extend for matrices U and V,

$$\det(A + UV^\intercal) = \det(\mathrm{I} + V^\intercal A^{-1}U)\det(A). \tag{13}$$

We also make use of the following Maclaurin series,

$$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots, \tag{14}$$

$$\text{and} \quad \log(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \cdots. \tag{15}$$

Appendix D. KL Minimisation between Gaussian Processes and Moment Matching

The difficult step of Power-EP is the projection step, that is how to find the posterior approximation $q(f)$ that minimises the KL divergence, $\mathrm{KL}[\tilde{p}(f)||q(f)]$, where $\tilde{p}(f)$ is the tilted distribution. We have chosen the form of the approximate posterior

$$q(f) = p(f_{\neq \mathbf{u}}|\mathbf{u})q(\mathbf{u}) = p(f_{\neq \mathbf{u}}|\mathbf{u})\frac{\exp(\theta_\mathbf{u}^\intercal \phi(\mathbf{u}))}{\mathcal{Z}(\theta_\mathbf{u})},$$

where $\mathcal{Z}(\theta_\mathbf{u}) = \int \exp(\theta_\mathbf{u}^\intercal \phi(\mathbf{u}))\mathrm{d}\mathbf{u}$ to ensure normalisation. We can then write the KL minimisation objective as follows,

$$\begin{aligned}
\mathcal{F}_{\mathrm{KL}} &= \mathrm{KL}[\tilde{p}(f)||q(f)] \\
&= \int \tilde{p}(f) \log \frac{\tilde{p}(f)}{q(f)} \mathrm{d}f \\
&= \langle \log \tilde{p}(f) \rangle_{\tilde{p}(f)} - \langle \log p(f_{\neq \mathbf{u}}|\mathbf{u}) \rangle_{\tilde{p}(f)} - \theta_\mathbf{u}^\intercal \langle \phi(\mathbf{u}) \rangle_{\tilde{p}(f)} + \log \mathcal{Z}(\theta_\mathbf{u}).
\end{aligned}$$

Since $p(f_{\neq \mathbf{u}}|\mathbf{u})$ is the prior conditional distribution, the only free parameter that controls our posterior approximation is $\theta_\mathbf{u}$. As such, to find $\theta_\mathbf{u}$ that minimises F_{KL}, we find the gradient of F_{KL} w.r.t $\theta_\mathbf{u}$ and set it to zero,

$$\begin{aligned}
0 = \frac{\mathrm{d}\mathcal{F}_{\mathrm{KL}}}{\mathrm{d}\theta_\mathbf{u}} &= -\langle \phi(\mathbf{u}) \rangle_{\tilde{p}(f)} + \frac{\mathrm{d} \log \mathcal{Z}(\theta_\mathbf{u})}{\mathrm{d}\theta_\mathbf{u}} \\
&= -\langle \phi(\mathbf{u}) \rangle_{\tilde{p}(f)} + \langle \phi(\mathbf{u}) \rangle_{q(\mathbf{u})},
\end{aligned}$$

therefore, $\langle \phi(\mathbf{u}) \rangle_{\tilde{p}(f)} = \langle \phi(\mathbf{u}) \rangle_{q(\mathbf{u})}$. That is, though we are trying to perform the KL minimisation between two Gaussian processes, due to the special form of the posterior approximation, *it is sufficient to only match the moments at the inducing points* $\mathbf{u}$.[4]

Appendix E. Shortcuts to the Moment Matching Equations

The most crucial step in Power-EP is the moment matching step as discussed above. This step can be done analytically for the Gaussian case, as the mean and covariance of the approximate posterior can be linked to the cavity distribution as follows,

$$\mathbf{m}_\mathbf{u} = \mathbf{m}_\mathbf{u}^{\backslash n} + \mathbf{V}_{\mathbf{u}f}^{\backslash n} \frac{\mathrm{d} \log \mathcal{Z}_{\mathrm{tilted},n}}{\mathrm{d}\mathbf{m}_f^{\backslash n}},$$

$$\mathbf{V}_\mathbf{u} = \mathbf{V}_\mathbf{u}^{\backslash n} + \mathbf{V}_{\mathbf{u}f}^{\backslash n} \frac{\mathrm{d}^2 \log \mathcal{Z}_{\mathrm{tilted},n}}{\mathrm{d}\mathbf{m}_f^{\backslash n,2}} \mathbf{V}_{f\mathbf{u}}^{\backslash n},$$

4. We can show that this condition gives the minimum of $\mathcal{F}_{\mathrm{KL}}$ by computing the second derivative.

where $\mathcal{Z}_{\text{tilted},n}$ is the normaliser of the tilted distribution,

$$\begin{aligned}\mathcal{Z}_{\text{tilted},n} &= \int q^{\backslash n}(f)p(y_n|f)\mathrm{d}f \\ &= \int q^{\backslash n}(f)p(y_n|f_n)\mathrm{d}f \\ &= \int q^{\backslash n}(f_n)p(y_n|f_n)\mathrm{d}f_n.\end{aligned}$$

In words, $\mathcal{Z}_{\text{tilted},n}$ only depends on the marginal distribution of the cavity process, $q^{\backslash n}(f_n)$, simplifying the moment matching equations above,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f_n}^{\backslash n}\frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}m_{f_n}^{\backslash n}}, \tag{16}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}f_n}^{\backslash n}\frac{\mathrm{d}^2\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}m_{f_n}^{\backslash n,2}}\mathbf{V}_{f_n\mathbf{u}}^{\backslash n}. \tag{17}$$

We can rewrite the cross-covariance $\mathbf{V}_{\mathbf{u}f_n}^{\backslash n} = \mathbf{V}_{\mathbf{u}}^{\backslash n}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n}$. We also note that, $m_{f_n}^{\backslash n} = \mathbf{K}_{f_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}_{\mathbf{u}}^{\backslash n}$, resulting in,

$$\begin{aligned}\frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}\mathbf{m}_{\mathbf{u}}^{\backslash n}} &= \frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}m_{f_n}^{\backslash n}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n}, \\ \frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}\mathbf{V}_{\mathbf{u}}^{\backslash n}} &= \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n}\frac{\mathrm{d}^2\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}m_{f_n}^{\backslash n,2}}\mathbf{K}_{f_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}.\end{aligned}$$

Substituting these results back in Equations (16) and (17), we obtain

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n}\frac{\mathrm{d}\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}\mathbf{m}_{\mathbf{u}}^{\backslash n}}, \tag{18}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n}\frac{\mathrm{d}^2\log\mathcal{Z}_{\text{tilted},n}}{\mathrm{d}\mathbf{m}_{\mathbf{u}}^{\backslash n,2}}\mathbf{V}_{\mathbf{u}}^{\backslash n}. \tag{19}$$

Therefore, using Equations (16) and (17), or Equations (18) and (19) are equivalent in our approximation settings.

Appendix F. Full Derivation of the Power-EP Procedure

We provide the full derivation of the Power-EP procedure in this section. We follow the derivation in (Qi et al., 2010) closely, but provide a clearer exposition and details how to get to each step used in the implementation, and how to handle powered/fractional deletion and update in Power-EP.

F.1 Optimal Factor Parameterisation

We start by defining the approximate factors to be in natural parameter form as this makes it simple to combine and delete them, $t_n(\mathbf{u}) = \tilde{\mathcal{N}}(\mathbf{u}; z_n, \mathbf{T}_{1,n}, \mathbf{T}_{2,n}) = z_n\exp(\mathbf{u}^\intercal\mathbf{T}_{1,n} - \frac{1}{2}\mathbf{u}^\intercal\mathbf{T}_{2,n}\mathbf{u})$. We initially consider full rank $\mathbf{T}_{2,n}$, but will show that the optimal form is rank 1.

The next goal is to relate these parameters to the approximate GP posterior. The approximate posterior over the pseudo-outputs has natural parameters $\mathbf{T}_{1,\mathbf{u}} = \sum_n \mathbf{T}_{1,n}$ and $\mathbf{T}_{2,\mathbf{u}} = \mathbf{K}_{\mathbf{uu}}^{-1} + \sum_n \mathbf{T}_{2,n}$. This induces an approximate GP posterior with mean and covariance function,

$$\begin{aligned}m_{\mathrm{f}} &= \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{T}_{1,\mathbf{u}} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\gamma, \\ V_{\mathrm{ff}'} &= \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{K}_{\mathrm{f}\mathbf{u}}\beta\mathbf{K}_{\mathbf{u}\mathrm{f}'},\end{aligned}$$

where γ and β are likelihood-dependent terms we wish to store and update using PEP; γ and β fully specify the approximate posterior.

Deletion step: The cavity for data point n, $q^{\backslash n}(f) \propto q^*(f)/t_n^\alpha(\mathbf{u})$, has a similar form to the posterior, but the natural parameters are modified by the deletion, $\mathbf{T}_{1,\mathbf{u}}^{\backslash n} = \mathbf{T}_{1,\mathbf{u}} - \alpha\mathbf{T}_{1,n}$ and $\mathbf{T}_{2,\mathbf{u}}^{\backslash n} = \mathbf{T}_{2,\mathbf{u}} - \alpha\mathbf{T}_{2,n}$, yielding a new mean and covariance function

$$\begin{aligned}m_{\mathrm{f}}^{\backslash n} &= \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}_{2,\mathbf{u}}^{\backslash n,-1}\mathbf{T}_{1,\mathbf{u}}^{\backslash n} = \mathbf{K}_{\mathrm{f}\mathbf{u}}\gamma^{\backslash n}, \\ V_{\mathrm{ff}'}^{\backslash n} &= \mathbf{K}_{\mathrm{ff}'} - \mathbf{Q}_{\mathrm{ff}'} + \mathbf{K}_{\mathrm{f}\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{T}\backslash n, -1_{2,\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}'} = \mathbf{K}_{\mathrm{ff}'} - \mathbf{K}_{\mathrm{f}\mathbf{u}}\beta^{\backslash n}\mathbf{K}_{\mathbf{u}\mathrm{f}'}.\end{aligned}$$

Projection step: The central step in Power EP is the projection step. Obtaining the new approximate unnormalised posterior $q^*(f)$ such that $\mathrm{KL}[\tilde{p}(f)||q^*(f)]$ is minimised would naïvely appear intractable. Fortunately, as shown in the previous section, because of the structure of the approximate posterior, $q(f) = p(f_{\neq\mathbf{u}}|\mathbf{u})q(\mathbf{u})$, the objective, $\mathrm{KL}[\tilde{p}(f)||q^*(f)]$ is minimised when $\mathbb{E}_{\tilde{p}(f)}[\phi(\mathbf{u})] = \mathbb{E}_{q(u)}[\phi(\mathbf{u})]$, where $\phi(\mathbf{u})$ are the sufficient statistics, that is when the moments at the pseudo-inputs are matched. This is the central result from which computational savings are derived. Furthermore, this moment matching condition would appear to necessitate computation of a set of integrals to find the zeroth, first and second moments. Using results from the previous section simplifies and provides the following shortcuts,

$$\mathbf{m}_{\mathbf{u}} = \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\frac{\mathrm{d}\log\tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^{\backslash n}}, \tag{20}$$

$$\mathbf{V}_{\mathbf{u}} = \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\frac{\mathrm{d}^2\log\tilde{Z}_n}{\mathrm{d}(m_{\mathrm{f}_n}^{\backslash n})^2}\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}, \tag{21}$$

where $\log\tilde{Z}_n = \log\mathbb{E}_{q^{\backslash n}(f)}[p^\alpha(y_n|\mathrm{f}_n)]$ is the log-normaliser of the tilted distribution.

Update step: Having computed the new approximate posterior, the fractional approximate factor $t_{n,\text{new}}(\mathbf{u}) = q^*(f)/q^{\backslash n}(f)$ can be straightforwardly obtained, resulting in,

$$\mathbf{T}_{1,n,\text{new}} = \mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{m}_{\mathbf{u}}^{\backslash n}, \tag{22}$$

$$\mathbf{T}_{2,n,\text{new}} = \mathbf{V}_{\mathbf{u}}^{-1} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}, \tag{23}$$

$$z_n^\alpha = \tilde{Z}_n\exp(\mathcal{G}_{q_*^{\backslash n}(\mathbf{u})} - \mathcal{G}_{q^*(\mathbf{u})}),$$

where $\mathcal{G}_{\tilde{\mathcal{N}}(\mathbf{u};z,\mathbf{T}_1,\mathbf{T}_2)} = \int \tilde{\mathcal{N}}(\mathbf{u}; z, \mathbf{T}_1, \mathbf{T}_2)\mathrm{d}\mathbf{u}$. Let $d_1 = \frac{\mathrm{d}\log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^{\backslash n}}$ and $d_2 = \frac{\mathrm{d}^2\log \tilde{Z}_n}{\mathrm{d}(m_{\mathrm{f}_n}^{\backslash n})^2}$. Using Equation (9) and Equation (21), we have,

$$\mathbf{V}_{\mathbf{u}}^{-1} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1} = -\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\left[d_2^{-1} + \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}\right]^{-1}\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}. \tag{24}$$

Let $v_n = \alpha(-d_2^{-1} - \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n})$, and $\mathbf{w}_n = \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}$. Combining Equation (24) and Equation (23) gives

$$\mathbf{T}_{2,n,\mathrm{new}} = \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal. \tag{25}$$

At convergence, we have $t_n(\mathbf{u})^\alpha = t_{n,\mathrm{new}}(\mathbf{u})$, hence $\mathbf{T}_{2,n} = \mathbf{w}_n v_n^{-1}\mathbf{w}_n^\intercal$. In words, $\mathbf{T}_{2,n}$ is optimally a rank-1 matrix. Note that,

$$\begin{aligned}
\mathbf{w}_n &= \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} \\
&= (\mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n}\mathbf{K}_{\mathbf{uu}})^{-1}(\mathbf{K}_{\mathbf{u}\mathrm{f}_n} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n}\mathbf{K}_{\mathbf{u}\mathrm{f}_n}) \\
&= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathrm{I} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n})^{-1}(\mathrm{I} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n})\mathbf{K}_{\mathbf{u}\mathrm{f}_n} \\
&= \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}\mathrm{f}_n}.
\end{aligned}$$

Using Equation (20) an Equation (25) gives,

$$\begin{aligned}
\mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} &= (\mathbf{V}_{\mathbf{u}}^{\backslash n,-1} + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal)(\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1) \\
&= \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1.
\end{aligned}$$

Substituting this result into Equation (22),

$$\begin{aligned}
\mathbf{T}_{1,n,\mathrm{new}} &= \mathbf{V}_{\mathbf{u}}^{-1}\mathbf{m}_{\mathbf{u}} - \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{m}_{\mathbf{u}}^{\backslash n} \\
&= \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 + \mathbf{w}_n\alpha v_n^{-1}\mathbf{w}_n^\intercal\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1 \\
&= \mathbf{w}_n\alpha v_n^{-1}\left(\mathbf{w}_n^\intercal\mathbf{m}_{\mathbf{u}}^{\backslash n} + d_1 v_n/\alpha + \mathbf{w}_n^\intercal\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1\right).
\end{aligned}$$

Let $\mathbf{T}_{1,n,\mathrm{new}} = \mathbf{w}_n\alpha v_n^{-1}g_n$, we obtain,

$$g_n = -\frac{d_1}{d_2} + \mathbf{K}_{\mathrm{f}_n\mathbf{u}}\gamma^{\backslash n}.$$

At convergence, $\mathbf{T}_{1,n} = \mathbf{w}_n v_n^{-1}g_n$. Re-writing the form of the approximate factor using $\mathbf{T}_{1,n}$ and $\mathbf{T}_{2,n}$ at convergence,

$$\begin{aligned}
t_n(\mathbf{u}) &= \tilde{\mathcal{N}}(\mathbf{u}; z_n, \mathbf{T}_{1,n}, \mathbf{T}_{2,n}) \\
&= z_n \exp(\mathbf{u}^\intercal\mathbf{T}_{1,n} - \frac{1}{2}\mathbf{u}^\intercal\mathbf{T}_{2,n}\mathbf{u}) \\
&= z_n \exp(\mathbf{u}^\intercal\mathbf{w}_n v_n^{-1}g_n - \frac{1}{2}\mathbf{u}^\intercal\mathbf{w}_n v_n^{-1}\mathbf{w}_n^\intercal\mathbf{u}).
\end{aligned}$$

As a result, the minimal and simplest way to parameterise the approximate factor is $t_n(\mathbf{u}) = \tilde{z}_n\mathcal{N}(\mathbf{w}_n^\intercal\mathbf{u}; g_n, v_n) = \tilde{z}_n\mathcal{N}(\mathbf{K}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{u}; g_n, v_n)$, where g_n and v_n are scalars, resulting in a significant memory saving compared to the parameterisation using $\mathbf{T}_{1,n}$ and $\mathbf{T}_{2,n}$.

F.2 Projection

We now recall the update equations in the projection step (Equations (20) and (21)):

$$\begin{aligned}
\mathbf{m}_{\mathbf{u}} &= \mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1, \\
\mathbf{V}_{\mathbf{u}} &= \mathbf{V}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_2\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}.
\end{aligned}$$

Note that:

$$\begin{aligned}
\mathbf{m}_{\mathbf{u}} &= \mathbf{K}_{\mathbf{uu}}\gamma, \\
\mathbf{V}_{\mathbf{u}} &= \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}\beta\mathbf{K}_{\mathbf{uu}},
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{m}_{\mathbf{u}}^{\backslash n} &= \mathbf{K}_{\mathbf{uu}}\gamma^{\backslash n}, \\
\mathbf{V}_{\mathbf{u}}^{\backslash n} &= \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}\beta^{\backslash n}\mathbf{K}_{\mathbf{uu}}.
\end{aligned}$$

Using these results, we can convert the update for the mean and covariance, $\mathbf{m}_{\mathbf{u}}$ and $\mathbf{V}_{\mathbf{u}}$, into an update for γ and β,

$$\begin{aligned}
\gamma &= \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}_{\mathbf{u}} \\
&= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{m}_{\mathbf{u}}^{\backslash n} + \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1) \\
&= \gamma^{\backslash n} + \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_1, \text{ and} \qquad (26) \\
\beta &= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}})\mathbf{K}_{\mathbf{uu}}^{-1} \\
&= \mathbf{K}_{\mathbf{uu}}^{-1}(\mathbf{K}_{\mathbf{uu}} - \mathbf{V}_{\mathbf{u}}^{\backslash n} - \mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_2\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n})\mathbf{K}_{\mathbf{uu}}^{-1} \\
&= \beta^{\backslash n} - \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}d_2\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{K}_{\mathbf{uu}}^{-1}. \qquad (27)
\end{aligned}$$

F.3 Deletion Step

Finally, we present how deletion might be accomplished. One direct approach to this step is to divide out the cavity from the cavity, that is,

$$q^{\backslash n}(f) \propto \frac{q(f)}{t_n^\alpha(\mathbf{u})} = \frac{p(f_{\neq\mathbf{u}}|\mathbf{u})q(\mathbf{u})}{t_n^\alpha(\mathbf{u})} = p(f_{\neq\mathbf{u}}|\mathbf{u})q^{\backslash n}(\mathbf{u}).$$

Instead, we use an alternative using the KL minimisation as used in (Qi et al., 2010), by realising that doing this will result in an identical outcome as the direct approach since the factor and distributions are Gaussian. Furthermore, we can re-use results from the projection and inclusion steps, by simply swapping the quantities and negating the site approximation variance. In particular, we present projection and deletion side-by-side, to facilitate the comparison,

$$\begin{aligned}
&\text{Projection:} \quad q(f) \approx q^{\backslash n}(f)p(y_n|\mathrm{f}_n), \\
&\text{Deletion:} \quad q^{\backslash n}(f) \propto q(f)\frac{1}{t_n^\alpha(\mathbf{u})}.
\end{aligned}$$

The projection step minimises the KL between the LHS and RHS while moment matching, to get $q(f)$. We would like to do the same for the deletion step to find $q^{\backslash n}(f)$, and thus reuse the same moment matching results for γ and β with some modifications.

Our task will be to reuse Equations (26) and (27), the moment matching equations in γ and β. We have two differences to account for. Firstly, we need to change any uses of the parameters of the cavity distribution to the parameters of the approximate posterior, $\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n}$ to $\mathbf{V}_{\mathbf{u}\mathrm{f}_n}$, $\gamma^{\backslash n}$ to γ and $\beta^{\backslash n}$ to β. This is the equivalent of re-deriving the entire projection operation while swapping the symbols (and quantities) for the cavity and the full distribution. Secondly, the derivatives d_1 and d_2 are different here, as

$$\log \tilde{Z}_n = \log \int q(f) \frac{1}{t_n^{\alpha}(\mathbf{u})} \mathrm{d}f.$$

Now, we note

$$\begin{aligned}
\frac{1}{t_n(\mathbf{u})} &\propto \frac{1}{\mathcal{N}^{\alpha}(\mathbf{w}_n^{\intercal}\mathbf{u}; g_n, v_n)} \\
&\propto \frac{1}{\exp\left(-\frac{\alpha}{2} v_n^{-1} \left(\mathbf{w}_n^{\intercal}\mathbf{u} - g_n\right)^2\right)} \\
&= \exp\left(\frac{1}{2}\alpha v_n^{-1} \left(\mathbf{w}_n^{\intercal}\mathbf{u} - g_n\right)^2\right) \\
&\propto \mathcal{N}(\mathbf{w}_n^{\intercal}\mathbf{u}; g_n, -v_n/\alpha).
\end{aligned}$$

Then we obtain the derivatives of $\log \tilde{Z}_n$

$$\begin{aligned}
\tilde{d}_2 &= \frac{\mathrm{d}^2 \log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}^2} = -\left[\mathbf{K}_{\mathrm{f}_n,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathrm{f}_n} - \mathbf{K}_{\mathrm{f}_n,\mathbf{u}}\beta\mathbf{K}_{\mathbf{u},\mathrm{f}_n} - v_n/\alpha\right]^{-1}, \\
\tilde{d}_1 &= \frac{\mathrm{d}\log \tilde{Z}_n}{\mathrm{d}m_{\mathrm{f}_n}} = (\mathbf{K}_{\mathrm{f}_n,\mathbf{u}}\gamma - g_n)\tilde{d}_2.
\end{aligned}$$

Putting the above results together, we obtain,

$$\begin{aligned}
\gamma^{\backslash n} &= \gamma + \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}\tilde{d}_1, \text{ and} \\
\beta^{\backslash n} &= \beta - \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}\tilde{d}_2\mathbf{V}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}.
\end{aligned}$$

F.4 Summary of the PEP Procedure

We summarise here the key steps and equations that we have obtained, that are used in the implementation:

1. Initialise the parameters: $\{g_n = 0\}_{n=1}^N$, $\{v_n = \infty\}_{n=1}^N$, $\gamma = \mathbf{0}_{M\times 1}$ and $\beta = \mathbf{0}_{M\times M}$
2. Loop through all data points until convergence:

(a) Deletion step: find $\gamma^{\backslash n}$ and $\beta^{\backslash n}$

$$\begin{aligned}
\gamma^{\backslash n} &= \gamma + \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}\tilde{d}_1, \text{ and} \\
\beta^{\backslash n} &= \beta - \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}\tilde{d}_2\mathbf{V}_{\mathrm{f}_n\mathbf{u}}\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}.
\end{aligned}$$

(b) Projection step: find γ and β

$$\begin{aligned}
\gamma &= \gamma^{\backslash n} + \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_1, \\
\beta &= \beta^{\backslash n} - \mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n} d_2\mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1}.
\end{aligned}$$

(c) Update step: find $g_{n,\text{new}}$ and $v_{n,\text{new}}$

$$\begin{aligned}
g_{n,\text{new}} &= -\frac{d_1}{d_2} + \mathbf{K}_{\mathrm{f}_n\mathbf{u}}\gamma^{\backslash n}, \\
v_{n,\text{new}} &= -d_2^{-1} - \mathbf{V}_{\mathrm{f}_n\mathbf{u}}^{\backslash n}\mathbf{V}_{\mathbf{u}}^{\backslash n,-1}\mathbf{V}_{\mathbf{u}\mathrm{f}_n}^{\backslash n},
\end{aligned}$$

and parameters for the full factor,

$$\begin{aligned}
v_n &\leftarrow (v_{n,\text{new}}^{-1} + (1-\alpha)v_n^{-1})^{-1}, \\
g_n &\leftarrow v_n(g_{n,\text{new}}v_{n,\text{new}}^{-1} + (1-\alpha)g_n v_n^{-1}).
\end{aligned}$$

Appendix G. Power-EP Energy for Sparse GP Regression and Classification

The Power-EP procedure gives an approximate marginal likelihood, which is the negative Power-EP energy, as follows,

$$\mathcal{F} = \mathcal{G}(q_*(\mathbf{u})) - \mathcal{G}(p_*(\mathbf{u})) + \frac{1}{\alpha}\sum_n \left[\log \mathcal{Z}_{\text{tilted},n} + \mathcal{G}(q_*^{\backslash n}(\mathbf{u})) - \mathcal{G}(q_*(\mathbf{u}))\right],$$

where $\mathcal{G}(q_*(\mathbf{u}))$ is the log-normaliser of the approximate posterior, that is,

$$\begin{aligned}
\mathcal{G}(q_*(\mathbf{u})) &= \log \int p(f_{\neq\mathbf{u}}|\mathbf{u}) \exp(\theta_{\mathbf{u}}^{\intercal}\phi(\mathbf{u})) \mathrm{d}f_{\neq\mathbf{u}}\mathrm{d}\mathbf{u} \\
&= \log \int \exp(\theta_{\mathbf{u}}^{\intercal}\phi(\mathbf{u}))\mathrm{d}\mathbf{u} \\
&= \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^{\intercal}\mathbf{V}^{-1}\mathbf{m},
\end{aligned} \tag{28}$$

where $\mathbf{m}$ and $\mathbf{V}$ are the mean and covariance of the posterior distribution over $\mathbf{u}$, respectively. Similarly,

$$\begin{aligned}
\mathcal{G}(q_*^{\backslash n}(\mathbf{u})) &= \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{V}_{\text{cav},n}| + \frac{1}{2}\mathbf{m}_{\text{cav},n}^{\intercal}\mathbf{V}_{\text{cav},n}^{-1}\mathbf{m}_{\text{cav},n}, \\
\text{and } \mathcal{G}(p_*(\mathbf{u})) &= \frac{M}{2}\log(2\pi) + \frac{1}{2}\log|\mathbf{K}_{\mathbf{u}\mathbf{u}}|.
\end{aligned} \tag{29}$$

Finally, $\log \mathcal{Z}_{\text{tilted},n}$ is the log-normalising constant of the tilted distribution,

$$\begin{aligned}\log \mathcal{Z}_{\text{tilted}} &= \log \int q_{\text{cav}}(f) p^{\alpha}(y_n|f) \mathrm{d}f \\ &= \log \int p(f_{\neq \mathbf{u}}|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) p^{\alpha}(y_n|f) \mathrm{d}f_{\neq \mathbf{u}} \mathrm{d}\mathbf{u} \\ &= \log \int p(f_n|\mathbf{u}) q_{\text{cav}}(\mathbf{u}) p^{\alpha}(y_n|f_n) \mathrm{d}f_n \mathrm{d}\mathbf{u}. \end{aligned} \tag{30}$$

Next, we can write down the form of the natural parameters of the approximate posterior and the cavity distribution, based on the approximate factor's parameters, as follows,

$$\mathbf{V}^{-1} = \mathbf{K}_{\mathbf{uu}}^{-1} + \sum_i \mathbf{w}_i \tau_i \mathbf{w}_i^{\intercal}, \tag{31}$$

$$\mathbf{V}^{-1}\mathbf{m} = \sum_i \mathbf{w}_i \tau_i \tilde{y}_i, \tag{32}$$

$$\mathbf{V}_{\text{cav},n}^{-1} = \mathbf{V}^{-1} - \alpha \mathbf{w}_n \tau_n \mathbf{w}_n^{\intercal}, \tag{33}$$

$$\mathbf{V}_{\text{cav},\mathbf{n}}{}^{-1}\mathbf{m}_{\text{cav},n} = \mathbf{V}^{-1}\mathbf{m} - \alpha \mathbf{w}_n \tau_n g_n. \tag{34}$$

Note that $\tau_i \coloneqq v_i^{-1}$. Using Equation (11) and Equation (33) gives,

$$\mathbf{V}_{\text{cav},n} = \mathbf{V} + \frac{\mathbf{V}\mathbf{w}_n \alpha \tau_n \mathbf{w}_n^{\intercal}\mathbf{V}}{1 - \mathbf{w}_n^{\intercal}\alpha\tau_n \mathbf{V}\mathbf{w}_n}. \tag{35}$$

Using Equation (12) and Equation (33) gives,

$$\log\det(\mathbf{V}_{\text{cav},n}) = \log\det(\mathbf{V}) - \log(1 - \mathbf{w}_n^{\intercal}\alpha\tau_n\mathbf{V}\mathbf{w}_n). \tag{36}$$

Substituting Equation (35) and Equation (36) back to Equation (29) results in,

$$\begin{aligned}\mathcal{G}(q_*^{\setminus n}(\mathbf{u})) = &\frac{M}{2}\log(2\pi) + \frac{1}{2}\log\det(\mathbf{V}) + \frac{1}{2}\mathbf{m}^{\intercal}\mathbf{V}^{-1}\mathbf{m} \\ &- \frac{1}{2}\log(1 - \mathbf{w}_n^{\intercal}\alpha\tau_n\mathbf{V}\mathbf{w}_n) + \frac{1}{2}\frac{\mathbf{m}^{\intercal}\mathbf{w}_n\alpha\tau_n\mathbf{w}_n^{\intercal}\mathbf{m}}{1 - \mathbf{w}_n^{\intercal}\alpha\tau_n\mathbf{V}\mathbf{w}_n} \\ &+ \frac{1}{2}g_n\alpha\tau_n\mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{w}_n\alpha\tau_n g_n - g_n\alpha\tau_n\mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m}. \end{aligned} \tag{37}$$

We now plug the above result back into the approximate marginal likelihood, yielding,

$$\begin{aligned}\mathcal{F} = &\frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^{\intercal}\mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| + \frac{1}{\alpha}\sum_n \log \mathcal{Z}_{\text{tilted},n} \\ &+ \sum_n \left[-\frac{1}{2\alpha}\log(1 - \mathbf{w}_n^{\intercal}\alpha\tau_n\mathbf{V}\mathbf{w}_n) + \frac{1}{2}\frac{\mathbf{m}^{\intercal}\mathbf{w}_n\tau_n\mathbf{w}_n^{\intercal}\mathbf{m}}{1 - \mathbf{w}_n^{\intercal}\alpha\tau_n\mathbf{V}\mathbf{w}_n} \right. \\ &\left. + \frac{1}{2}g_n\tau_n\mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{w}_n\alpha\tau_n g_n - g_n\tau_n\mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m} \right]. \end{aligned} \tag{38}$$

G.1 Regression

We have shown in the previous section that the fixed point solution of the Power-EP iterations can be obtained analytically for the regression case, $g_n = y_n$ and $\tau_n^{-1} = d_n = \alpha(K_{f_n f_n} - \mathbf{K}_{f_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n}) + \sigma_y^2$. Crucially, we can obtain a closed form expression for $\log \mathcal{Z}_{\text{tilted},n}$,

$$\log \mathcal{Z}_{\text{tilted},n} = -\frac{\alpha}{2}\log(2\pi\sigma_y^2) + \frac{1}{2}\log(\sigma_y^2) - \frac{1}{2}\log(\alpha v_n + \sigma_y^2) - \frac{1}{2}\frac{(y_n - \mu_n)^2}{v_n + \sigma_y^2/\alpha},$$

where $\mu_n = \mathbf{w}_n^{\intercal}\mathbf{m}_{\text{cav}} = \mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav}}(\mathbf{V}^{-1}\mathbf{m} - \mathbf{w}_n\alpha\tau_n y_n)$ and $v_n = \frac{d_n - \sigma_y^2}{\alpha} + \mathbf{w}_n^{\intercal}\mathbf{V}_{\text{cav}}\mathbf{w}_n$. We can therefore simplify the approximate marginal likelihood F further,

$$\begin{aligned}\mathcal{F} &= \frac{1}{2}\log|\mathbf{V}| + \frac{1}{2}\mathbf{m}^{\intercal}\mathbf{V}^{-1}\mathbf{m} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| \\ &\quad + \sum_n \left[-\frac{1}{2}\log(2\pi\sigma_y^2) + \frac{1}{2\alpha}\log\sigma_y^2 - \frac{1}{2\alpha}\log d_n - \frac{y_n^2}{2d_n} \right] \\ &= -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{D} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^T(\mathbf{D} + \mathbf{Q}_{\mathbf{ff}})^{-1}\mathbf{y} - \frac{1-\alpha}{2\alpha}\sum_n \log(\frac{d_n}{\sigma_y^2}), \end{aligned}$$

where $\mathbf{Q}_{\mathbf{ff}} = \mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}}$ and $\mathbf{D}$ is a diagonal matrix, $\mathbf{D}_{nn} = d_n$.

When $\alpha = 1$, the approximate marginal likelihood takes the same form as the FITC marginal likelihood,

$$\mathcal{F} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{D} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^T(\mathbf{D} + \mathbf{Q}_{\mathbf{ff}})^{-1}\mathbf{y},$$

where $\mathbf{D}_{nn} = d_n = K_{f_n f_n} - \mathbf{K}_{f_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n} + \sigma_y^2$.

When α tends to 0, we have,

$$\lim_{\alpha\to 0}\frac{1-\alpha}{2\alpha}\sum_n \log(\frac{d_n}{\sigma_y^2}) = \frac{1}{2}\sum_n \lim_{\alpha\to 0}\frac{\log(1 + \alpha\frac{g_n}{\sigma_y^2})}{\alpha} = \frac{\sum_n h_n}{2\sigma_y^2},$$

where $h_n = K_{f_n f_n} - \mathbf{K}_{f_n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{u}f_n}$. Therefore,

$$\mathcal{F} = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\sigma_y^2\mathbf{I} + \mathbf{Q}_{\mathbf{ff}}| - \frac{1}{2}\mathbf{y}^T(\sigma_y^2\mathbf{I} + \mathbf{Q}_{\mathbf{ff}})^{-1}\mathbf{y} - \frac{\sum_n h_n}{2\sigma_y^2},$$

which is the variational lower bound of Titsias (Titsias, 2009).

G.2 Classification

In contrast to the regression case, the approximate marginal likelihood for classification cannot be simplified due to the non-Gaussian likelihood. Specifically, $\log \mathcal{Z}_{\text{tilted},n}$ is not analytically tractable, except when $\alpha = 1$ and the classification link function is the Gaussian CDF. However, this quantity can be evaluated numerically, using sampling or Gauss-Hermite quadrature, since it only involves a one-dimensional integral.

We now consider the case when α tends to 0 and verify that in such case the approximate marginal likelihood becomes the variational lower bound. We first find the limits of individual terms in Equation (38):

$$\lim_{\alpha\to 0} -\frac{1}{2\alpha}\log(1-\mathbf{w}_n^\intercal\alpha\tau_n\mathbf{V}\mathbf{w}_n) = \frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n, \tag{39}$$

$$\frac{1}{2}\frac{\mathbf{m}^\intercal\mathbf{w}_n\tau_n\mathbf{w}_n^\intercal\mathbf{m}}{1-\mathbf{w}_n^\intercal\alpha\tau_n\mathbf{V}\mathbf{w}_n}\bigg|_{\alpha=0} = \frac{1}{2}\mathbf{m}^\intercal\mathbf{w}_n\tau_n\mathbf{w}_n^\intercal\mathbf{m}, \tag{40}$$

$$\frac{1}{2}g_n\tau_n\mathbf{w}_n^\intercal\mathbf{V}_{\text{cav},n}\mathbf{w}_n\alpha\tau_n g_n\bigg|_{\alpha=0} = 0, \tag{41}$$

$$-g_n\tau_n\mathbf{w}_n^\intercal\mathbf{V}_{\text{cav},n}\mathbf{V}^{-1}\mathbf{m}\bigg|_{\alpha=0} = -g_n\tau_n\mathbf{w}_n^\intercal\mathbf{m}. \tag{42}$$

We turn our attention to $\log\mathcal{Z}_{\text{tilted},n}$. First, we expand $p^\alpha(y_n|f_n)$ using Equation (14):

$$\begin{aligned} p^\alpha(y_n|f_n) &= \exp(\alpha\log p(y_n|f_n)) \\ &= 1+\alpha\log p(y_n|f_n)+\xi(\alpha^2). \end{aligned}$$

Substituting this result back into $\log\mathcal{Z}\text{tilted}/\alpha$ gives,

$$\begin{aligned} \frac{1}{\alpha}\log\mathcal{Z}_{\text{tilted}} &= \frac{1}{\alpha}\log\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})p^\alpha(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \\ &= \frac{1}{\alpha}\log\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})[1+\alpha\log p(y_n|f_n)+\xi(\alpha^2)]\mathrm{d}f_n\mathrm{d}\mathbf{u} \\ &= \frac{1}{\alpha}\log\left[1+\alpha\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}+\alpha^2\xi(1)\right] \\ &= \frac{1}{\alpha}\left[\alpha\int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}+\alpha^2\xi(1)\right] \\ &= \int p(f_n|\mathbf{u})q_{\text{cav}}(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}+\alpha\xi(1). \end{aligned}$$

Therefore,

$$\lim_{\alpha\to 0}\frac{1}{\alpha}\log\mathcal{Z}_{\text{tilted}} = \int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}. \tag{43}$$

Putting these results into Equation (38), we obtain,

$$\begin{aligned} \mathcal{F} &= \frac{1}{2}\log|\mathbf{V}|+\frac{1}{2}\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m}-\frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}| \\ &\quad+\sum_n\frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n+\frac{1}{2}\mathbf{m}^\intercal\mathbf{w}_n\tau_n\mathbf{w}_n^\intercal\mathbf{m}-g_n\tau_n\mathbf{w}_n^\intercal\mathbf{m}+\int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \\ &= \frac{1}{2}\log|\mathbf{V}|+\frac{1}{2}\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m}-\frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}|+\frac{1}{2}\mathbf{m}^\intercal(\mathbf{V}^{-1}-\mathbf{K}_{\mathbf{uu}}^{-1})\mathbf{m}-\mathbf{m}^\intercal\mathbf{V}^{-1}\mathbf{m} \\ &\quad+\sum_n\frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n+\int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u} \\ &= \frac{1}{2}\log|\mathbf{V}|-\frac{1}{2}\mathbf{m}^\intercal\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}-\frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}|+\sum_n\frac{1}{2}\mathbf{w}_n^\intercal\tau_n\mathbf{V}\mathbf{w}_n \\ &\quad+\sum_n\int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}. \end{aligned} \tag{44}$$

We now write down the evidence lower bound of the global variational approach of Titsias (Titsias, 2009), as applied to the classification case (Hensman et al., 2015),

$$\mathcal{F}_{\text{VFE}} = -\text{KL}[q(\mathbf{u})||p(\mathbf{u})]+\sum_n\int p(f_n|\mathbf{u})q(\mathbf{u})\log p(y_n|f_n)\mathrm{d}f_n\mathrm{d}\mathbf{u}, \tag{45}$$

where

$$\begin{aligned} -\text{KL}[q(\mathbf{u})||p(\mathbf{u})] &= -\frac{1}{2}\text{trace}(\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{V})-\frac{1}{2}\mathbf{m}^\intercal\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}+\frac{M}{2}-\frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}|+\frac{1}{2}\log|\mathbf{V}| \\ &= -\frac{1}{2}\text{trace}([\mathbf{V}^{-1}-\sum_n\mathbf{w}_n\tau_n\mathbf{w}_n]\mathbf{V})-\frac{1}{2}\mathbf{m}^\intercal\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m} \\ &\quad+\frac{M}{2}-\frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}|+\frac{1}{2}\log|\mathbf{V}| \\ &= \frac{1}{2}\text{trace}(\sum_n\mathbf{w}_n\tau_n\mathbf{w}_n\mathbf{V})-\frac{1}{2}\mathbf{m}^\intercal\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}-\frac{1}{2}\log|\mathbf{K}_{\mathbf{uu}}|+\frac{1}{2}\log|\mathbf{V}|. \end{aligned} \tag{46}$$

Therefore, $\mathcal{F}_{\text{VFE}}$ is identical to the limit of the approximate marginal likelihood provided by power-EP as shown in Equation (44).

Appendix H. The Surrogate Regression Viewpoint

It was written in the main text that it is instructive to view the approximation using pseudo-points as forming a surrogate exact Gaussian process regression problem such that the posterior and the marginal likelihood of this surrogate problem are close to that of the original intractable regression/classification problem. This approximation view is useful and could potentially be used for other intractable probabilistic model, despite that we have not used this view in the practical implementation of the algorithms/PEP procedure discussed in this paper. In this section, we detail the surrogate model and how the parameters of this model can be tuned to match the approximate posterior and approximate marginal likelihood.

We consider the exact GP regression problem with M surrogate observations $\tilde{\mathbf{y}}$ that are formed by linear combinations the pseudo-outputs and additive surrogate Gaussian noise, $\tilde{\mathbf{y}} = \tilde{\mathbf{W}}\mathbf{u} + \tilde{\Sigma}^{1/2}\epsilon$. The exact posterior and log marginal likelihood can be obtained for this model as follows,

$$\begin{aligned}\tilde{p}(\mathbf{u}|\mathbf{y}) &= \mathcal{N}^{-1}(\mathbf{u}; \tilde{\mathbf{W}}\tilde{\Sigma}^{-1}\tilde{\mathbf{y}}, \mathbf{K}_{\mathbf{uu}}^{-1} + \tilde{\mathbf{W}}^\intercal\tilde{\Sigma}^{-1}\tilde{\mathbf{W}}),\\ \log p(\tilde{\mathbf{y}}) &= -\frac{M}{2}\log(2\pi) - \frac{1}{2}(\log|\mathbf{K}_{\mathbf{uu}}^{-1} + \tilde{\mathbf{W}}^\intercal\tilde{\Sigma}^{-1}\tilde{\mathbf{W}}| + \log|\mathbf{K}_{\mathbf{uu}}^{-1}| + \log|\tilde{\Sigma}|)\\ &\quad - \frac{1}{2}\tilde{\mathbf{y}}^\intercal\tilde{\Sigma}^{-1}\tilde{\mathbf{y}} - \frac{1}{2}\tilde{\mathbf{y}}^\intercal\tilde{\Sigma}^{-1}\tilde{\mathbf{W}}(\mathbf{K}_{\mathbf{uu}}^{-1} + \tilde{\mathbf{W}}^\intercal\tilde{\Sigma}^{-1}\tilde{\mathbf{W}})^{-1}\tilde{\mathbf{W}}^\intercal\tilde{\Sigma}^{-1}\tilde{\mathbf{y}},\end{aligned}$$

where we have used the matrix inversion lemma and the matrix determinant lemma in the equations above, and that $\mathcal{N}^{-1}$ denotes the Gaussian distribution with natural parameters.

The aim is to show that we can use the above quantities is to match a given approximate posterior $q(\mathbf{u}) = \mathcal{N}^{-1}(\mathbf{u}; \mathbf{S}^{-1}\mathbf{m}, \mathbf{S}^{-1})$ and an approximate marginal likelihood $\mathcal{F}$, that is, $\tilde{p}(\mathbf{u}|\mathbf{y}) = q(\mathbf{u})$ and $\log p(\tilde{\mathbf{y}}) = \mathcal{F}$. Substituting the above results into the constraints leading to the following simplified constraints:

$$\begin{aligned}\tilde{\mathbf{W}}\tilde{\Sigma}^{-1}\tilde{\mathbf{y}} &= \mathbf{m},\\ \tilde{\mathbf{W}}^\intercal\tilde{\Sigma}^{-1}\tilde{\mathbf{W}} &= \mathbf{R} = \mathbf{K}_{\mathbf{uu}}^{-1} - \mathbf{S}^{-1},\\ \tilde{\mathbf{y}}^\intercal\tilde{\Sigma}^{-1}\tilde{\mathbf{y}} + \log|\tilde{\Sigma}| &= c,\end{aligned}$$

where c is a constant. Assume that $\mathbf{R}$ is invertible, we can simplified the above results further,

$$\begin{aligned}\tilde{\Sigma}^{-1/2}\tilde{\mathbf{y}} &= \mathbf{R}^{-1/2}\mathbf{m},\\ \tilde{\Sigma}^{-1/2}\tilde{\mathbf{W}} &= \mathbf{R}^{\intercal/2},\\ \log|\tilde{\Sigma}| &= d,\end{aligned}$$

where d is a constant. We can choose $\tilde{\Sigma}$, e.g. a diagonal matrix, that satisfies the third equality above. Given $\tilde{\Sigma}$, obtaining $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{W}}$ from the first two equalities is trivial.

Appendix I. Extra Experimental Results

I.1 Comparison Between Various α Values on a Toy Regression Problem

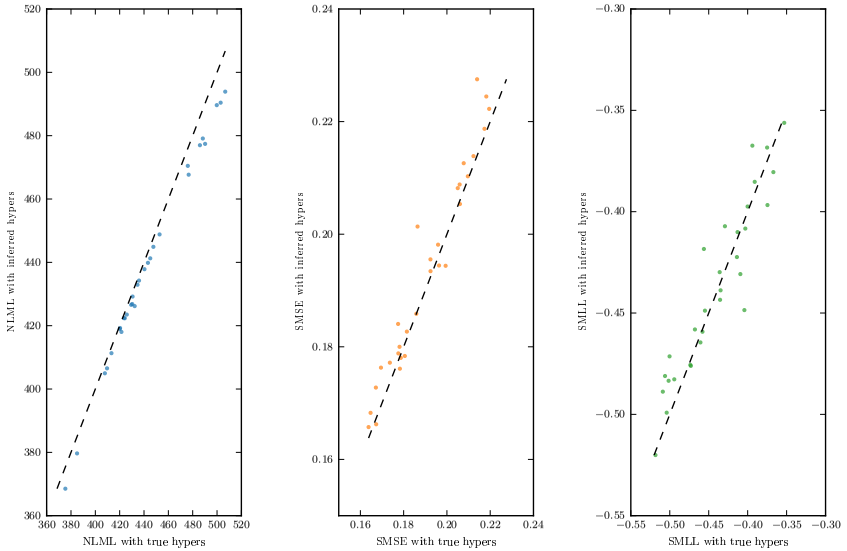


Figure 8: Results on a toy regression problem: Negative log-marginal likelihood, mean squared error and mean log-loss on the test set for full Gaussian process regression on synthetic data sets with *true* hyperparameters and hyperparameters obtained by type-2 ML. Each dot is one trial, i.e. one synthetic data set. The results demonstrate that type-2 maximum likelihood on hyperparameters works well, despite being a little confident on the log-marginal likelihood on the train set.

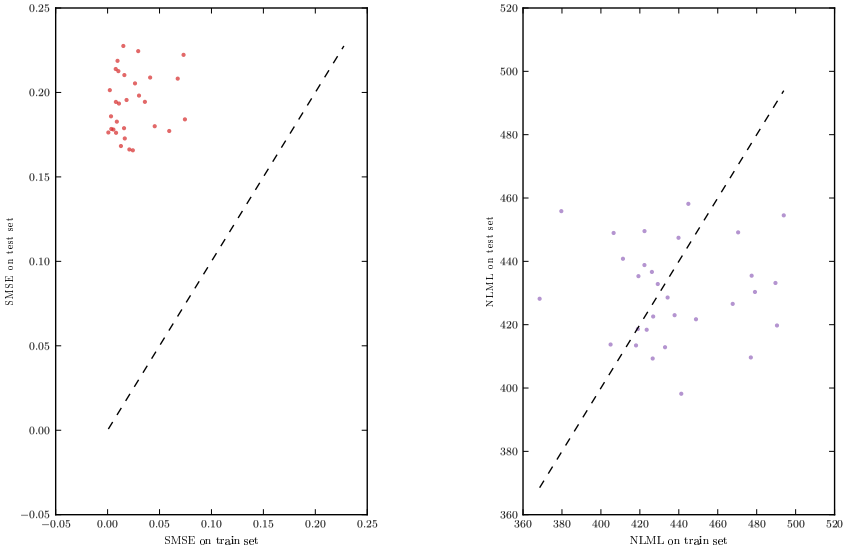


Figure 9: Results on a toy regression problem with 500 training points: Mean squared error and log-likelihood on train and test sets on synthetic data sets with hyperparameters obtained by type-2 ML. In this example, the test error is higher than the training error, as measured by the mean squared error, because the test points and training points are relatively far apart, making the prediction task on the training set easier (interpolation) than on the test set (extrapolation). This is consistent with the results with more training points, shown in Figure 10.

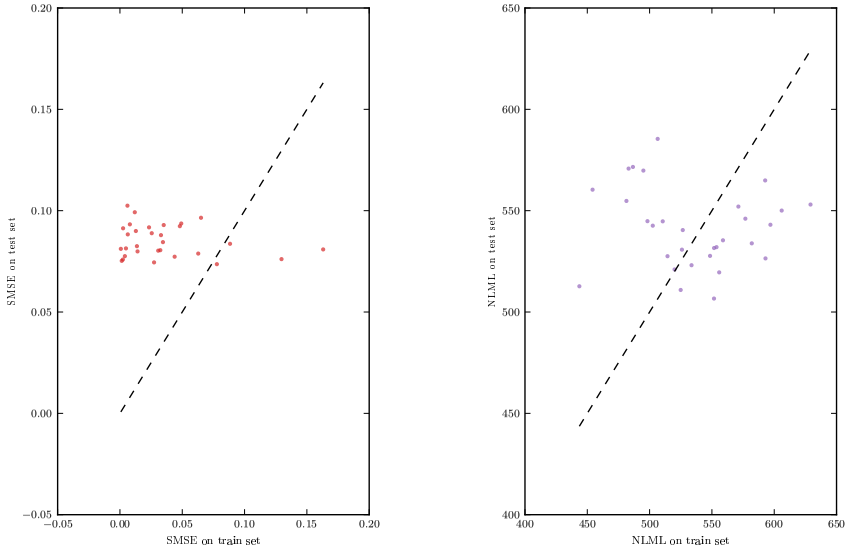


Figure 10: Results on a toy regression problem with 1000 training points: Mean squared error and log-likelihood on train and test sets on synthetic data sets with hyperparameters obtained by type-2 ML. See Figure 9 for a discussion.

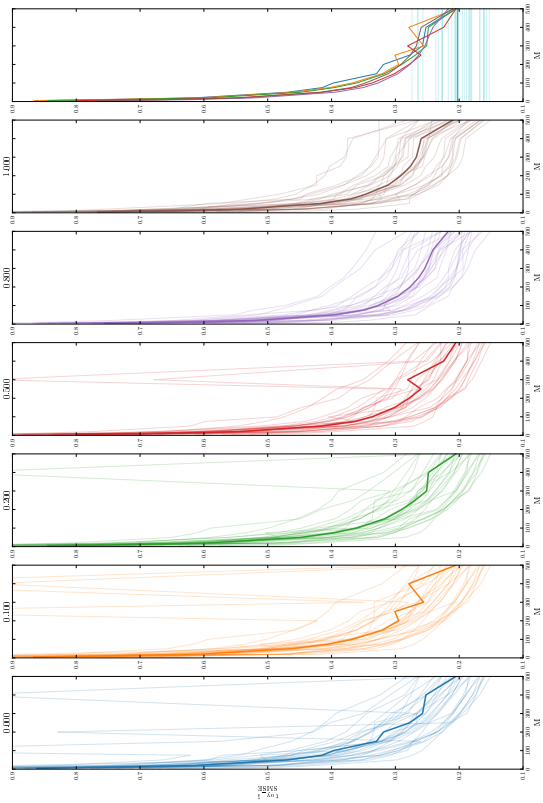


Figure 12: Results on a toy regression problem: Standardsised mean squared error on the test set for various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figure shows the mean for various α, and the results using GP regression.

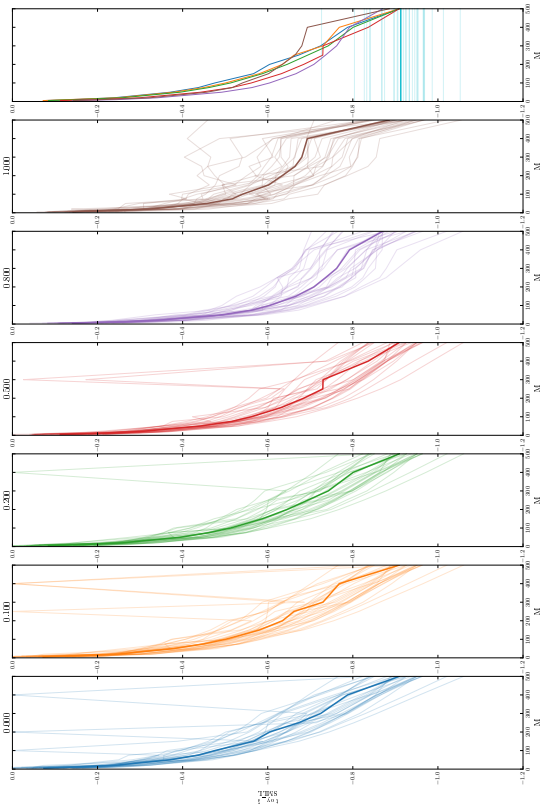


Figure 11: Results on a toy regression problem: Standardsised mean log-loss on the test set for various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figure shows the mean for various α, and the results using GP regression.

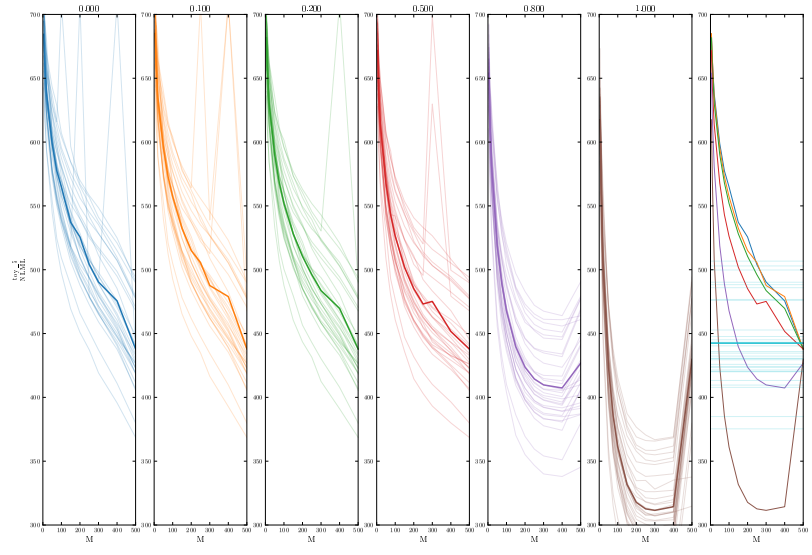


Figure 13: Results on a toy regression problem: The negative log marginal likelihood of the training set after training for various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figure shows the mean for various α, and the results using GP regression. Power EP with α close to 1 over-estimates the marginal-likelihood.

I.2 Real-world Regression

We include the details of the regression data sets in Table 1 and several comparisons of α values in Figures 17 to 22.

data set	N train/test	D
boston	455/51	14
concrete	927/103	9
energy	691/77	9
kin8nm	7373/819	9
naval	10741/1193	18
yacht	277/31	7
power	8611/957	5
red wine	1439/160	12

Table 1: Regression data sets

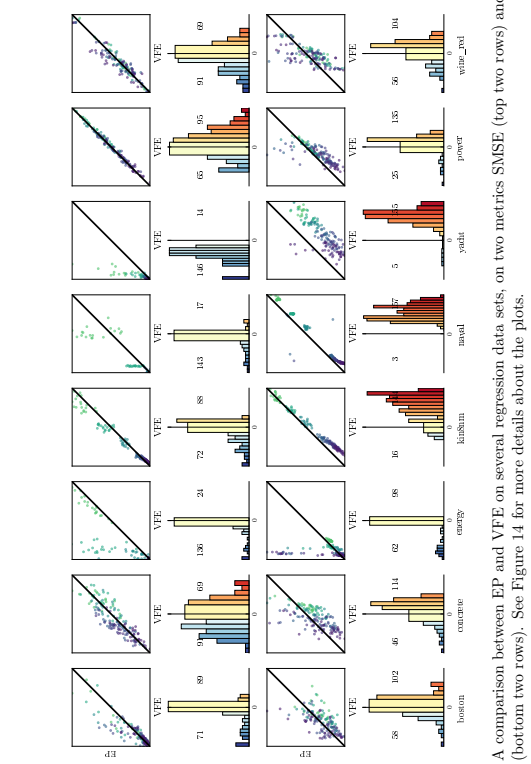


Figure 15: A comparison between EP and VFE on several regression data sets, on two metrics SMSE (top two rows) and SMLL (bottom two rows). See Figure 14 for more details about the plots.

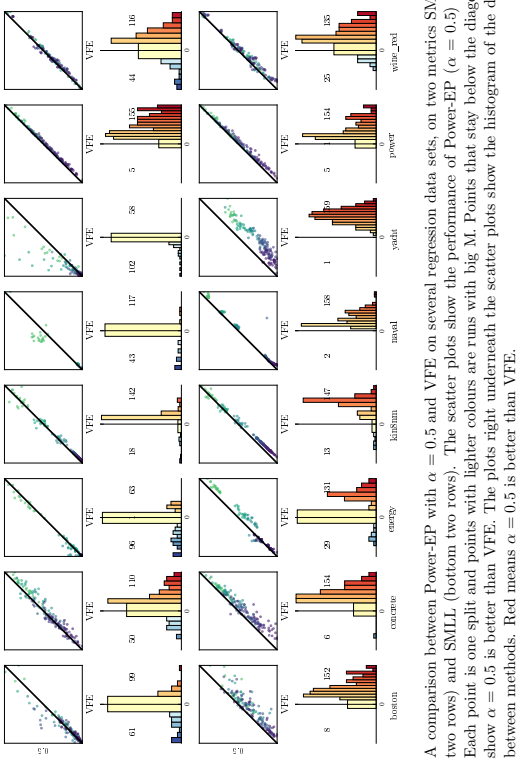


Figure 14: A comparison between Power-EP with $\alpha = 0.5$ and VFE on several regression data sets, on two metrics SMSE (top two rows) and SMLL (bottom two rows). The scatter plots show the performance of Power-EP ($\alpha = 0.5$) vs VFE. Each point is one split and points with lighter colours are runs with big M. Points that stay below the diagonal line show $\alpha = 0.5$ is better than VFE. The plots right underneath the scatter plots show the histogram of the difference between methods. Red means $\alpha = 0.5$ is better than VFE.

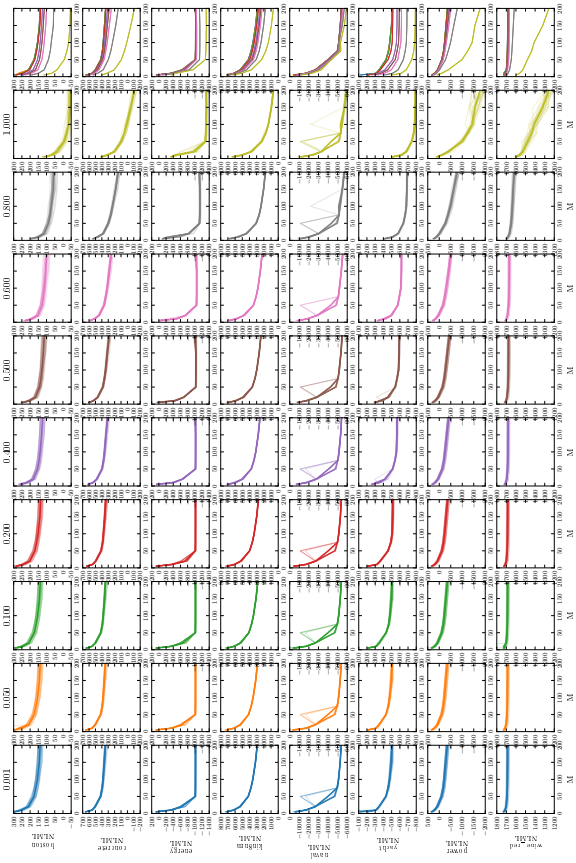


Figure 17: Results on real-world regression problems: Negative training log-marginal likelihood for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better [however, lower could mean overestimation].

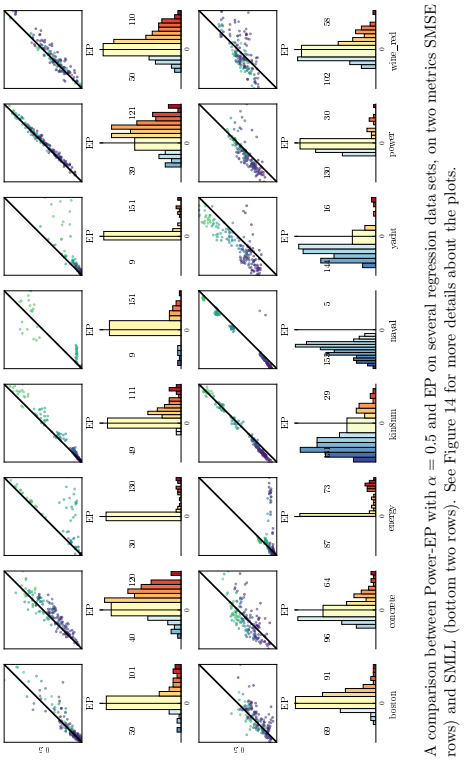


Figure 16: A comparison between Power-EP with $\alpha = 0.5$ and EP on several regression data sets, on two metrics SMSE (top two rows) and SMLL (bottom two rows). See Figure 14 for more details about the plots.

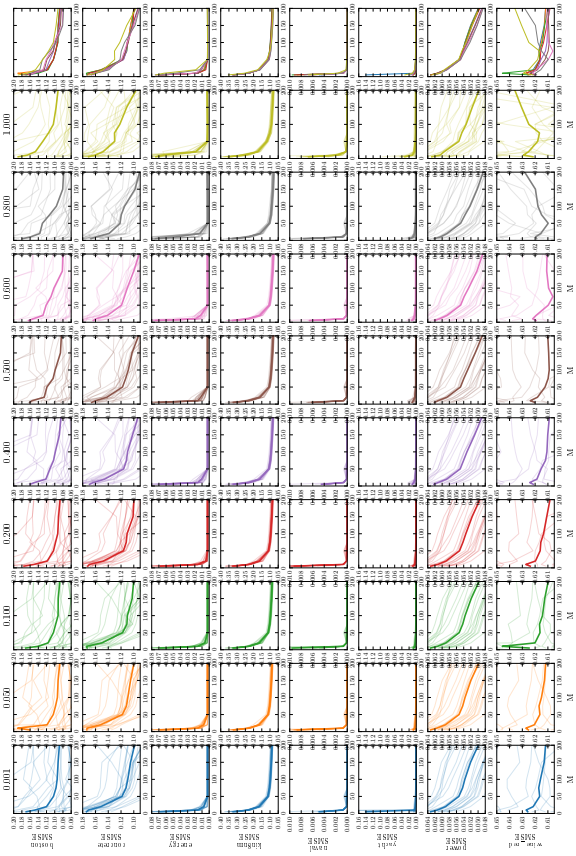


Figure 19: Results on real-world regression problems: Standardised mean squared error on the test set for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better.

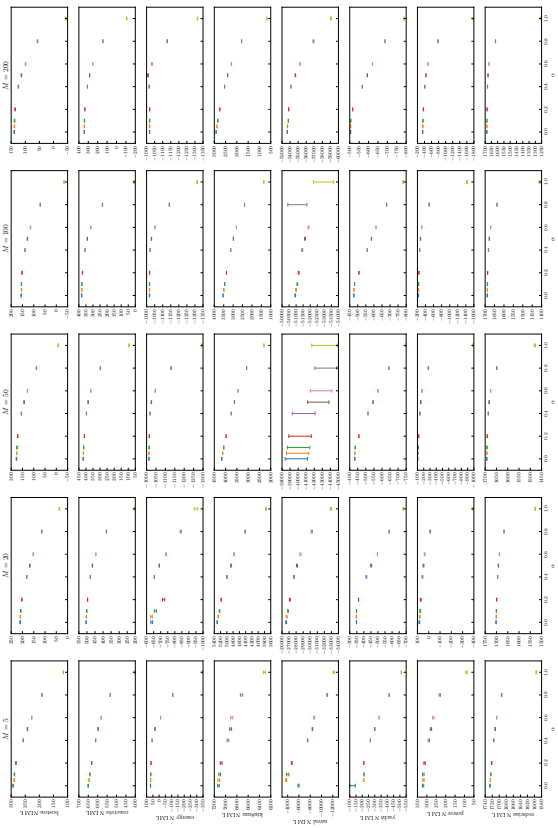


Figure 18: Results on real-world regression problems: Negative training log-marginal likelihood for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better [however, lower could mean overestimation].

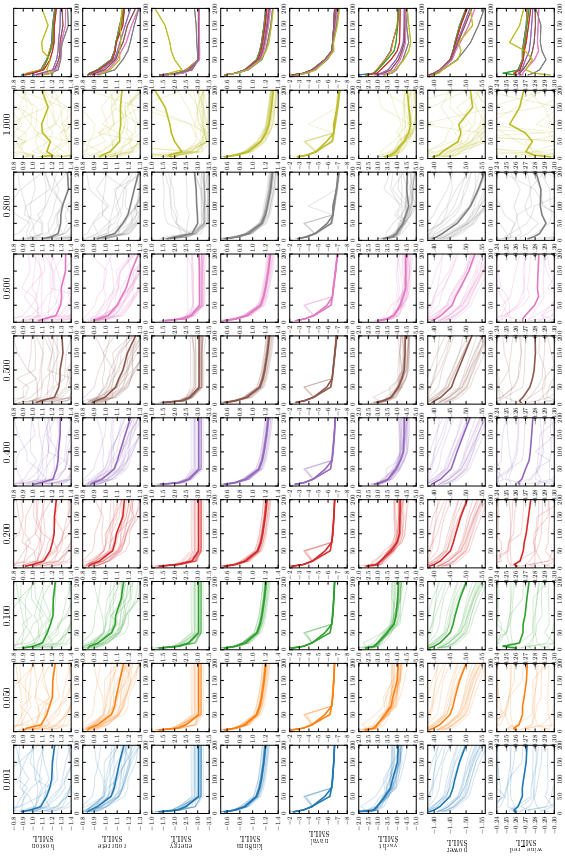


Figure 21: Results on real-world regression problems: Standardised mean log loss on the test set for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better.

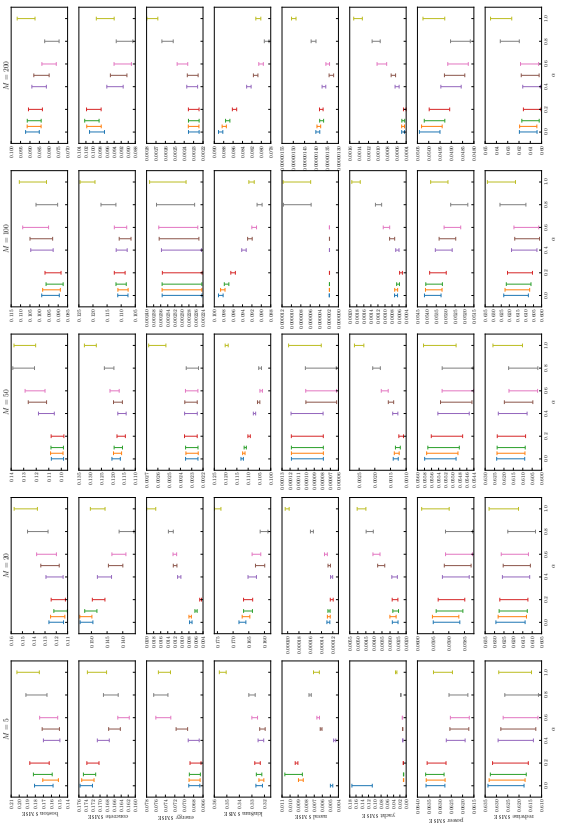


Figure 20: Results on real-world regression problems: Standardised mean squared error on the test set for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better.

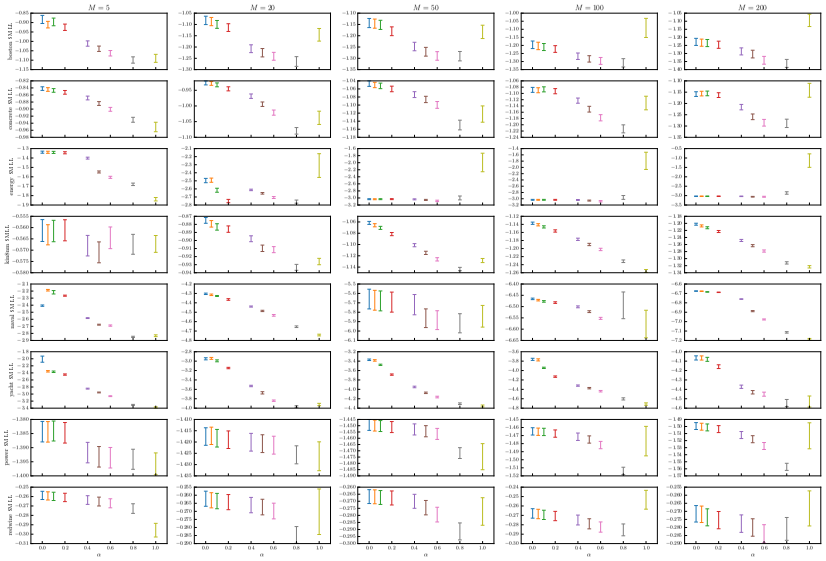


Figure 22: Results on real-world regression problems: Standardised mean log loss on the test set for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better.

I.3 Real-world Classification

It was demonstrated in (Hernández-Lobato and Hernández-Lobato, 2016; Hensman et al., 2015) that, once optimised, the pseudo points tend to concentrate around the decision boundary for VFE, and spread out to cover the data region in EP. Figure 23 illustrates the same effect as α goes from close to 0 (VFE) to 1 (EP).

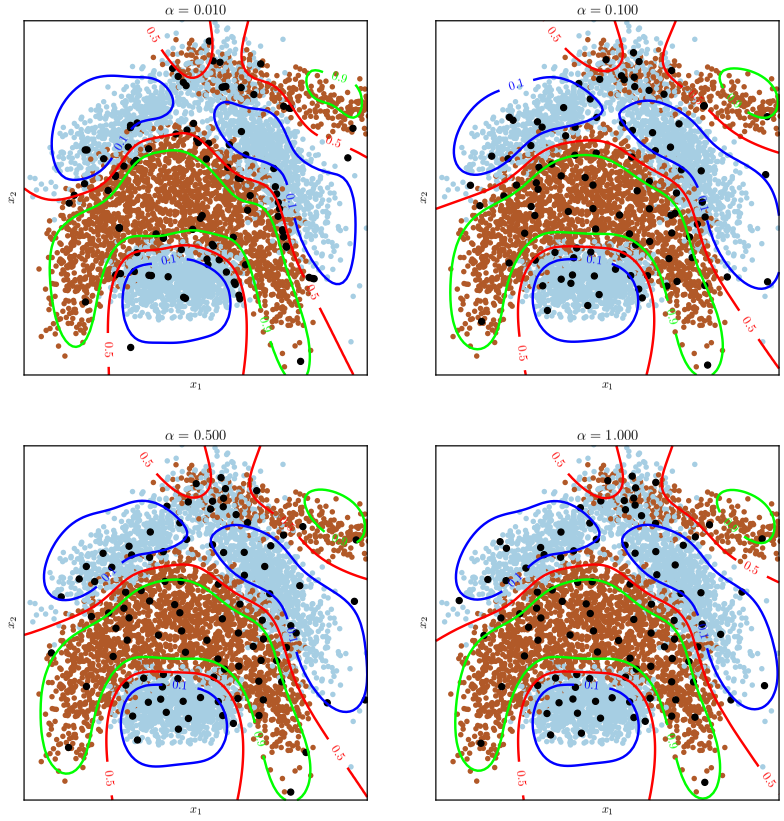


Figure 23: The locations of pseudo data points vary with α.. Best viewed in colour.

We include the details of the classification data sets in Table 2 and several comparisons of α values in Figures 27 to 30.

data set	N train/test	D	N positive/negative
australian	621/69	15	222/468
breast	614/68	11	239/443
crabs	180/20	7	100/100
iono	315/35	35	126/224
pima	690/77	9	500/267
sonar	186/21	61	111/96

Table 2: Classification data sets

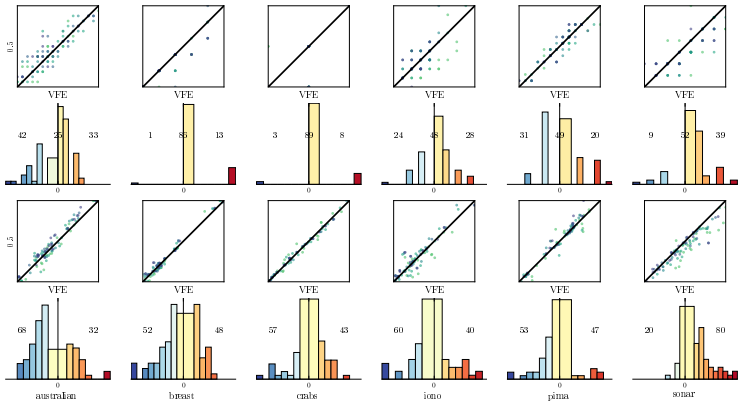


Figure 24: A comparison between Power-EP with $\alpha = 0.5$ and VFE on several classification data sets, on two metrics: classification error (top two rows) and NLL (bottom two rows). See Figure 14 for more details about the plots.

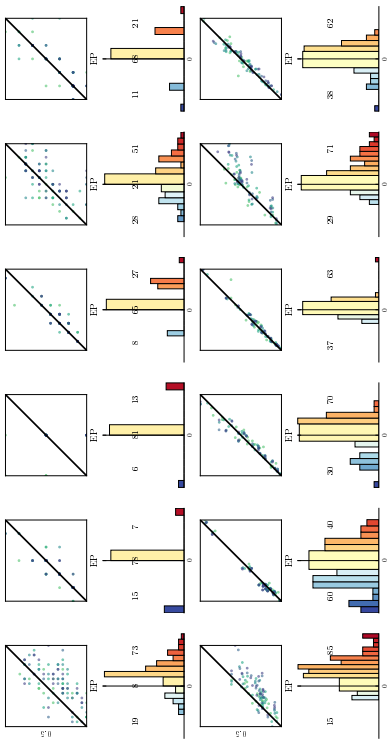


Figure 25: A comparison between EP and VFE on several classification data sets, on two metrics: classification error (top two rows) and NLL (bottom two rows). See Figure 14 for more details about the plots.

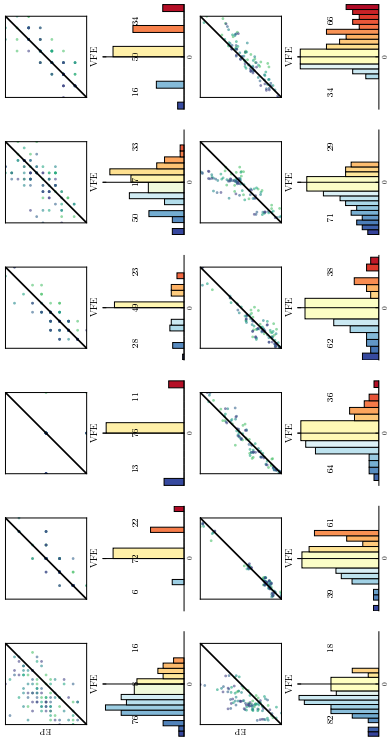


Figure 26: A comparison between Power-EP with $\alpha = 0.5$ and EP on several classification data sets, on two metrics: classification error (top two rows) and NLL (bottom two rows). See Figure 14 for more details about the plots.



Figure 28: Results on real-world classification problems: Classification error rate on the test set for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better.

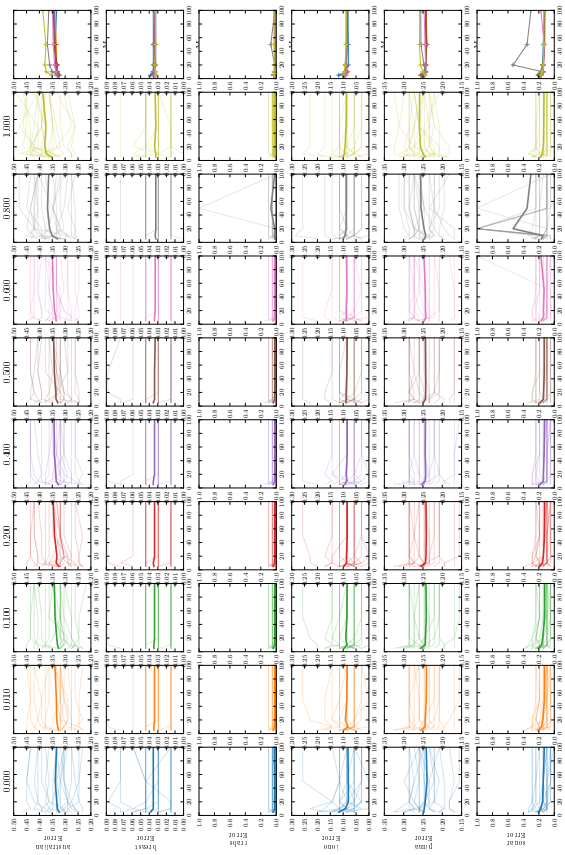


Figure 27: Results on real-world classification problems: Classification error rate on the test set for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better.

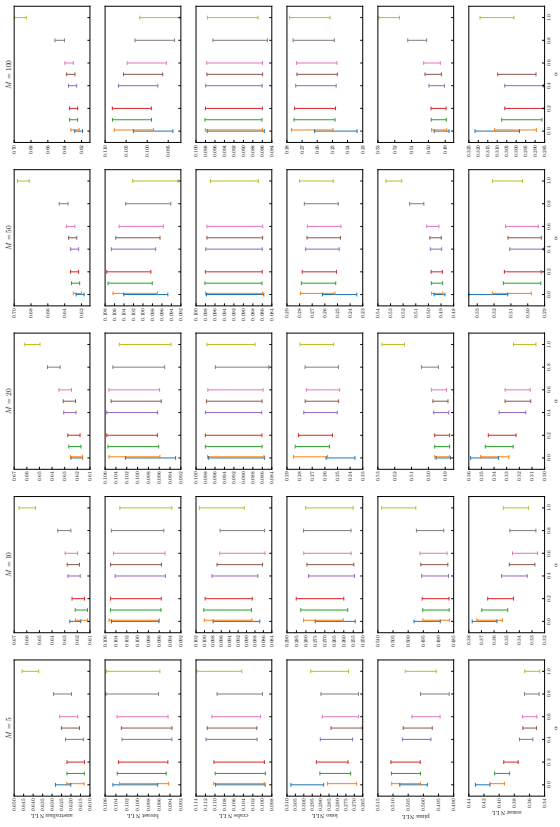


Figure 30: Results on real-world classification problems: Average negative log-likelihood on the test set for different data sets, various values of α and various number of pseudo-points M, averaged over 20 splits. Lower is better.

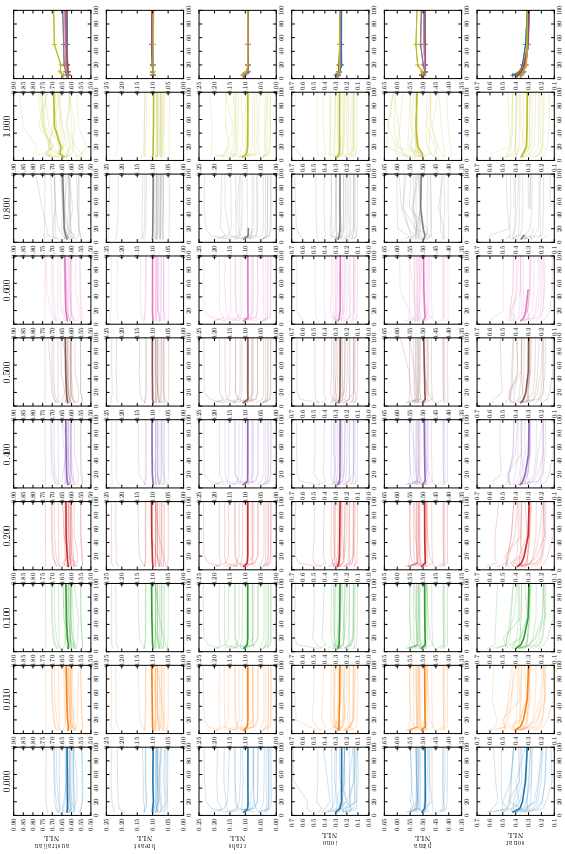


Figure 29: Results on real-world classification problems: Average negative log-likelihood on the test set for different data sets, various values of α and various number of pseudo-points M. Each trace is for one split, bold line is the mean. The rightmost figures show the mean for various α for comparison. Lower is better.

I.4 Binary Classification on Even/Odd MNIST Digits

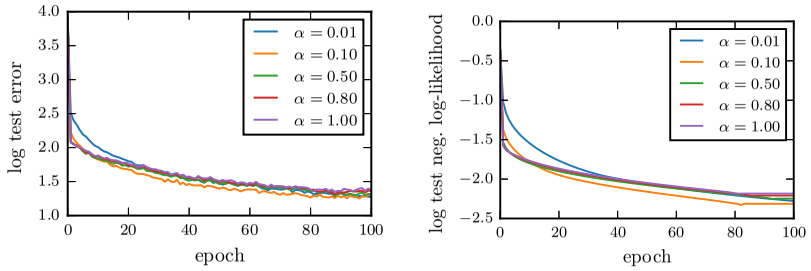


Figure 31: The test error and log-likelihood of the MNIST binary classification task (M=100).

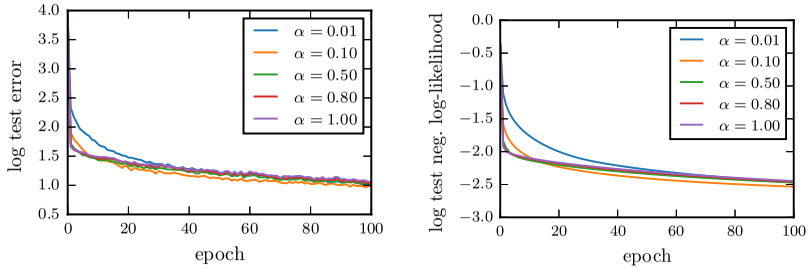


Figure 32: The test error and log-likelihood of the MNIST binary classification task (M=200).

I.5 When $M = N$ and $\alpha = 1$, Do We Recover EP for GPC (Rasmussen and Williams, 2005, sec. 3.6)?

The key difference between the EP method in this manuscript when $M = N$ and the pseudo-inputs and the training inputs are identical, and the standard EP method as described by (Rasmussen and Williams, 2005, sec. 3.6) is the factor representation. While Rasmussen and Williams (2005) used a one dimensional unnormalised Gaussian distribution that touches only *one* function value f_n to approximate each exact factor, the approximate factor used in the EP scheme described in the main text touches *all* M pseudo-points, hence *all* N function values when the pseudo-inputs are placed at the training inputs. However, in

practice both methods give virtually identical results. Figure 33 shows the approximate log marginal likelihood and the negative test log-likelihood, given by running the EP procedure described in the main text on the `ionosphere` data set. We note that these results are similar to that of the standard EP method (see Kuss and Rasmussen, 2005).

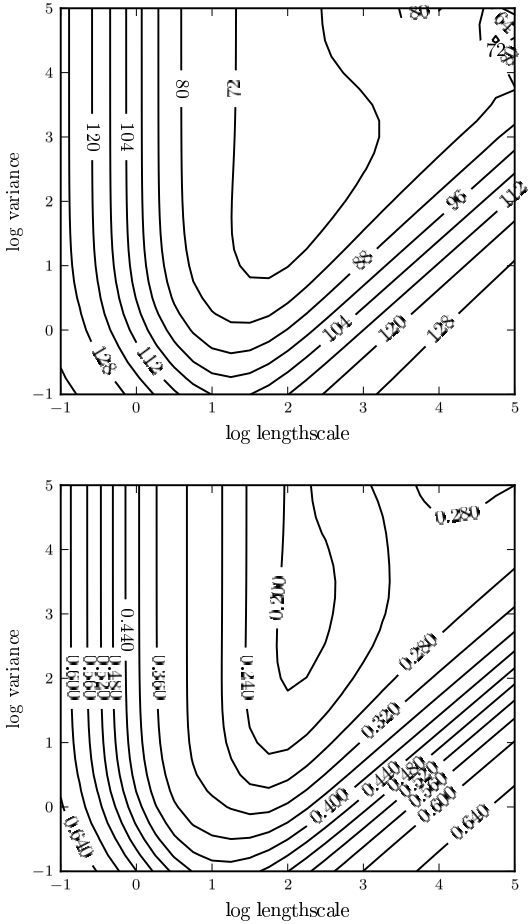


Figure 33: EP energy on the train set [TOP] and the average negative log-likelihood on the test set[BOTTOM] when $M = N$.

References

Mauricio A. Álvarez, David Luengo, Michalis K. Titsias, and Neil D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In *13th International Conference on Artificial Intelligence and Statistics*, pages 25–32, 2010.

Matthias Bauer, Mark van der Wilk, and Carl E. Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems 29*, pages 1525–1533, 2016.

Lawrence D. Brown. *Fundamentals of Statistical Exponential Families with Applications in Statistical Decision Theory*. Institute of Mathematical Statistics, Hayward, CA, 1986.

Thang D. Bui and Richard E. Turner. Tree-structured Gaussian process approximations. In *Advances in Neural Information Processing Systems 27*, pages 2213–2221, 2014.

Thang D. Bui, Cuong V. Nguyen, and Richard E. Turner. Streaming sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems 30*, 2017.

Lehel Csató. *Gaussian Processes — Iterative Sparse Approximations*. PhD thesis, Aston University, 2002.

Lehel Csató and Manfred Opper. Sparse online Gaussian processes. *Neural Computation*, 14(3):641–669, 2002.

Michael R.W. Dawson. *Understanding cognitive science*. Blackwell Publishing, 1998.

Marc P. Deisenroth. *Efficient Reinforcement Learning using Gaussian Processes*. PhD thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2010.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Learning and policy search in stochastic dynamical systems with Bayesian neural networks. In *4th International Conference on Learning Representations*, 2016.

Amir Dezfouli and Edwin V. Bonilla. Scalable inference for Gaussian process models with black-box likelihoods. In *Advances in Neural Information Processing Systems 28*, pages 1414–1422, 2015.

Anibal Figueiras-Vidal and Miguel Lázaro-Gredilla. Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems 22*, pages 1087–1095, 2009.

Roger Frigola, Yutian Chen, and Carl E. Rasmussen. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems 27*, pages 3680–3688, 2014.

Andrew Gelman, Aki Vehtari, Pasi Jylänki, Christian Robert, Nicolas Chopin, and John P Cunningham. Expectation propagation as a way of life. *arXiv preprint arXiv:1412.4869*, 2014.

James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *29th Conference on Uncertainty in Artificial Intellegence*, pages 282–290, 2013.

James Hensman, Alexander G. D. G. Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *18th International Conference on Artificial Intelligence and Statistics*, pages 351–360, 2015.

Daniel Hernández-Lobato and José Miguel Hernández-Lobato. Scalable Gaussian process classification via expectation propagation. In *19th International Conference on Artificial Intelligence and Statistics*, pages 168–176, 2016.

José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang D Bui, and Richard E Turner. Black-box α-divergence minimization. In *33rd International Conference on International Conference on Machine Learning*, pages 1511–1520, 2016.

Trong Nghia Hoang, Quang Minh Hoang, and Bryan Kian Hsiang Low. A distributed variational inference framework for unifying parallel sparse Gaussian process regression models. In *33rd International Conference on Machine Learning*, pages 382–391, 2016.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Kazufumi Ito and Kaiqi Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, 2000.

Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015.

Harold J. Kushner and Amarjit S. Budhiraja. A nonlinear filtering algorithm based on an approximation of the conditional distribution. *IEEE Transactions on Automatic Control*, 45(3):580–585, Mar 2000.

Malte Kuss and Carl E. Rasmussen. Assessing approximate inference for binary Gaussian process classification. *The Journal of Machine Learning Research*, 6:1679–1704, 2005.

Neil D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.

Yingzhen Li, José Miguel Hernández-Lobato, and Richard E. Turner. Stochastic expectation propagation. In *Advances in Neural Information Processing Systems 29*, pages 2323–2331, 2015.

Kian Hsiang Low, Jiangbo Yu, Jie Chen, and Patrick Jaillet. Parallel Gaussian process regression for big data: Low-rank representation meets Markov approximation. In *29th AAAI Conference on Artificial Intelligence*, pages 2821–2827, 2015.

Maren Mahsereci and Philipp Hennig. Probabilistic line searches for stochastic optimization. In *Advances in Neural Information Processing Systems 28*, pages 181–189, 2015.

Alexander G. D. G. Matthews, James Hensman, Richard E. Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *19th International Conference on Artificial Intelligence and Statistics*, pages 231–239, 2016.

Andrew McHutchon. *Nonlinear modelling and control using Gaussian processes*. PhD thesis, University of Cambridge, 2014.

Thomas P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

Thomas P. Minka. Power EP. Technical report, Microsoft Research Cambridge, 2004.

Thomas P. Minka. Divergence measures and message passing. Technical report, Microsoft Research Cambridge, 2005.

Andrew Naish-Guzman and Sean B. Holden. The generalized FITC approximation. In *Advances in Neural Information Processing Systems 20*, pages 1057–1064, 2007.

Hannes Nickisch and Carl E. Rasmussen. Approximations for binary Gaussian process classification. *The Journal of Machine Learning Research*, 9(Oct):2035–2078, 2008.

Juho Piironen and Aki Vehtari. Comparison of Bayesian predictive methods for model selection. *Statistics and Computing*, 27(3):711–735, 2017.

Yuan Qi, Ahmed H. Abdel-Gawad, and Thomas P. Minka. Sparse-posterior Gaussian processes for general likelihoods. In *26th Conference on Uncertainty in Artificial Intelligence*, pages 450–457, 2010.

Joaquin Quiñonero-Candela and Carl E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

Jaakko Riihimäki, Pasi Jylänki, and Aki Vehtari. Nested expectation propagation for Gaussian process classification with a multinomial probit likelihood. *The Journal of Machine Learning Research*, 14(Jan):75–109, 2013.

Alan D. Saul, James Hensman, Aki Vehtari, and Neil D. Lawrence. Chained Gaussian processes. In *19th International Conference on Artificial Intelligence and Statistics*, pages 1431–1440, 2016.

Anton Schwaighofer and Volker Tresp. Transductive and inductive methods for approximate Gaussian process regression. In *Advances in Neural Information Processing Systems 15*, pages 953–960, 2002.

Matthias Seeger. Bayesian inference and optimal design for the sparse linear model. *The Journal of Machine Learning Research*, 9(Apr):759–813, 2008.

Matthias Seeger and Michael I. Jordan. Sparse Gaussian process classification with multiple classes. Technical report, Department of Statistics, University of Berkeley, CA, 2004.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *9th International Conference on Artificial Intelligence and Statistics*, 2003.

Edward Snelson. *Flexible and efficient Gaussian process models for machine learning*. PhD thesis, University College London, 2007.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 19*, pages 1257–1264, 2006.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2951–2959, 2012.

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *12th International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.

Michalis K. Titsias and Neil D. Lawrence. Bayesian Gaussian process latent variable model. In *13th International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.

Felipe Tobar, Thang D. Bui, and Richard E. Turner. Learning stationary time series using Gaussian processes with nonparametric kernels. In *Advances in Neural Information Processing Systems 29*, pages 3501–3509, 2015.

Richard E. Turner and Maneesh Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, T. Cemgil, and S. Chiappa, editors, *Bayesian Time series models*, chapter 5, pages 109–130. Cambridge University Press, 2011.

Jack M. Wang, David J. Fleet, and Aaron Hertzmann. Gaussian process dynamical models. In *Advances in Neural Information Processing Systems 18*, pages 1441–1448, 2005.

Minjie Xu, Balaji Lakshminarayanan, Yee Whye Teh, Jun Zhu, and Bo Zhang. Distributed Bayesian posterior sampling via moment sharing. In *Advances in Neural Information Processing Systems 27*, pages 3356–3364, 2014.

Huaiyu Zhu and Richard Rohwer. Information geometric measurements of generalisation. Technical report, Aston University, 1995.

Huaiyu Zhu and Richard Rohwer. Measurements of generalisation based on information geometry. In *Mathematics of Neural Networks*, pages 394–398. 1997.

Journal of Machine Learning Research 18 (2017) 1-23 Submitted 10/16; Revised 6/17; Published 10/17

Accelerating Stochastic Composition Optimization

Mengdi Wang* MENGDIW@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Ji Liu* JI.LIU.UWISC@GMAIL.COM
Department of Computer Science and Department of Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627, USA

Ethan X. Fang XXF13@PSU.EDU
Department of Statistics and Department of Industrial and Manufacturing Engineering
Pennsyvania State University
University Park, PA 16802, USA

Editor: Inderjit Dhillon

Abstract

We consider the stochastic nested composition optimization problem where the objective is a composition of two expected-value functions. We propose a new stochastic first-order method, namely the accelerated stochastic compositional proximal gradient (ASC-PG) method. This algorithm updates the solution based on noisy gradient queries using a two-timescale iteration. The ASC-PG is the first proximal gradient method for the stochastic composition problem that can deal with nonsmooth regularization penalty. We show that the ASC-PG exhibits faster convergence than the best known algorithms, and that it achieves the optimal sample-error complexity in several important special cases. We demonstrate the application of ASC-PG to reinforcement learning and conduct numerical experiments.

Keywords: Large-scale optimization, stochastic gradient, composition optimization, sample complexity.

1. Introduction

The popular stochastic gradient methods are well suited for minimizing expected-value objective functions or the sum of a large number of loss functions. Stochastic gradient methods find wide applications in estimation, online learning, and training of deep neural networks. Despite their popularity, they do not apply to the minimization of a nonlinear function involving expected values or a composition between two expected-value functions.

In this paper, we consider the *stochastic composition problem*, given by

$$\min_{\mathbf{x}\in X} \quad H(\mathbf{x}) := \underbrace{\mathbb{E}_v(f_v(\mathbb{E}_w(g_w(\mathbf{x}))))}_{=:F(\mathbf{x})} + R(\mathbf{x}) \tag{1}$$

∗. Both authors contributed equally.

©2017 Mengdi Wang, Ji Liu and Ethan X. Fang.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-504.html.

where $f(g(\mathbf{x})) = (f \circ g)(\mathbf{x})$ denotes the function composition, $g_w(\cdot): \Re^n \mapsto \Re^m$ and $f_v(\cdot): \Re^m \mapsto \Re$ are continuously differentiable functions, v, w are random variables, and $R(\mathbf{x}): \Re^n \mapsto \Re \cup \{+\infty\}$ is an extended real-valued closed convex function, X is a convex and closed set. We assume throughout that there exists at least one optimal solution $\mathbf{x}^* \in X$ to problem (1). We focus on the case where f_v and g_w are smooth, but we allow R to be a nonsmooth penalty such as the ℓ_1-norm. We do no require either the outer function f_v or the inner function g_w to be convex or monotone. As a result, the composition problem cannot be reformulated into a saddle point problem in general.

Our algorithmic objective is to develop efficient algorithms for solving problem (1) based on random evaluations of f_v, g_w and their gradients. Our theoretical objective is to analyze the rate of convergence for the stochastic algorithm and to improve it when possible. In the online setting, the iteration complexity of our stochastic methods can be interpreted as a sample-error complexity upper bound for estimating the optimal solution of problem (1).

1.1 Motivating Examples

One motivating example is *reinforcement learning* (Sutton and Barto, 1998). Consider a controllable Markov chain with states $1, \ldots, S$. Estimating the value-per-state of a fixed control policy π is known as on-policy learning. It can be casted into an $S \times S$ system of Bellman equations:

$$\gamma P^\pi V^\pi + r^\pi = V^\pi,$$

where $\gamma \in (0,1)$ is a discount factor, $P^\pi_{s\tilde{s}}$ is the transition probability from state s to state $\tilde{s}$, and r^π_s is the expected state transition reward at state s. The solution V^π to the Bellman equation is the value vector, with $V^\pi(s)$ being the total expected reward starting at state s. In the blackbox simulation environment, P^π, r^π are unknown but can be sampled from a simulator. As a result, solving the Bellman equation becomes a special case of the stochastic composition optimization problem:

$$\min_{\mathbf{x}\in X} \quad \|\mathbb{E}[A]\mathbf{x} - \mathbb{E}[\mathbf{b}]\|^2, \tag{2}$$

where A, B, b are random matrices and random vectors such that $\mathbb{E}[A] = I - \gamma P^\pi$ and $\mathbb{E}[b] = r^\pi$. It can be viewed as the composition of the square norm function $f(\cdot) \equiv f_v(\cdot) = \|\cdot\|$ and the expected linear function $g(\mathbf{x}) = \mathbb{E}[A]\mathbf{x} - \mathbb{E}[\mathbf{b}]$. We will give more details on the reinforcement learning application in Section 4.

Another motivating example is *risk-averse learning*. For example, consider the mean-variance minimization problem

$$\min_{\mathbf{x}\in X} \quad \mathbb{E}_{a,b}[h(\mathbf{x}; a, b)] + \lambda \mathrm{Var}_{a,b}[h(\mathbf{x}; a, b)],$$

where $h(x; a, b)$ is some loss function parameterized by random variables a and b, and $\lambda > 0$ is a regularization parameter. In particular, the mean-variance minimization takes the composition form where $g(\mathbf{x}) = (\mathbb{E}[\mathbf{x}], \mathbb{E}[h(\mathbf{x}; a, b)])$, and $f(y_1, y_2) = \mathbb{E}[h(y_1; a, b)] + \lambda \mathbb{E}\Big(h(y_1; a, b) - y_2\Big)^2$. Its batch version takes the form

$$\min_{\mathbf{x}\in X} \quad \frac{1}{N}\sum_{i=1}^N h(\mathbf{x}; a_i, b_i) + \frac{\lambda}{N}\sum_{i=1}^N \left(h(\mathbf{x}; a_i, b_i) - \frac{1}{N}\sum_{i=1}^N h(\mathbf{x}; a_i, b_i)\right)^2.$$

Here the variance term is the composition of the mean square function and an expected loss function.

Although the stochastic composition problem (1) was barely studied, it actually finds a broad spectrum of emerging applications in estimation and machine learning (see Wang et al. (2017) for a list of applications). Fast optimization algorithms with theoretical guarantees will lead to new computation tools and online learning methods for a broader problem class, no longer limited to the expectation minimization problem.

1.2 Related Works and Contributions

Contrary to the expectation minimization problem, "unbiased" gradient samples are no longer available for the stochastic composition problem (1). The objective is *nonlinear* in the joint probability distribution of (w, v), which substantially complicates the problem. In a recent work by Dentcheva et al. (2016), a special case of the stochastic composition problem, i.e., risk-averse optimization, has been studied. A central limit theorem has been established, showing that the K-sample batch problem converges to the true problem at the rate of $O(1/\sqrt{K})$ in a proper sense. For the case where $R(x) = 0$, Wang et al. (2017) has proposed and analyzed a class of *stochastic compositional gradient/subgradient methods* (SCGD). The SCGD involves two iterations of different time scales, one for estimating x^* by a stochastic quasi-gradient iteration, the other for maintaining a running estimate of $g(x^*)$. Wang and Liu (2016) studies the SCGD in the setting where samples are corrupted with Markov noises (instead of i.i.d. zero-mean noises). Both works establish almost sure convergence of the algorithm and several convergence rate results, which are the best-known convergence rate prior to the current paper.

The idea of using two-timescale quasi-gradient traced back to the earlier work Ermoliev (1976). It proposed for the first time a stochastic iteration that updates the solution and an auxiliary variable using two different stepsizes. The incremental treatment of proximal gradient iteration has been studied extensively for the expectation minimization problem, see for examples Nedić and Bertsekas (2001); Nemirovski et al. (2009); Wang et al. (2015); Wang and Bertsekas (2016) and references therein. However, except for Wang et al. (2017) and Wang and Liu (2016), all of these works focus on variants of the expectation minimization problem and do not apply to the stochastic composition problem (1). Another work partially related to this paper is by Dai et al. (2017). They consider a special case of problem (1) arising in kernel estimation, where they assume that all functions f_v's are convex and their conjugate functions $f_v^\star$'s can be easily obtained/sampled. Under these additional assumptions, they essentially rewrite the problem into a saddle point optimization involving functional variables.

In this paper, we propose a new accelerated stochastic compositional proximal gradient (ASC-PG) method that applies to the penalized problem (1), which is a more general problem than the one considered in Wang et al. (2017). We use a *coupled martingale stochastic analysis* to show that ASC-PG achieves significantly better sample-error complexity in various cases. We also show that ASC-PG exhibits optimal sample-error complexity in two important special cases: the case where the outer function is linear and the case where the inner function is linear. Note that the current work has a preliminary conference version (Wang et al., 2016).

The current journal version provides more complete technical details as well as important theoretical extensions (Theorem 3).

Our major contributions are summarized as follows:

1. We propose the first stochastic *proximal-gradient* method for the stochastic composition problem. This is the first algorithm that is able to address the nonsmooth regularization penalty $R(\cdot)$ without deteriorating the convergence rate.

2. We obtain a convergence rate $O(K^{-4/9})$ for smooth optimization problems that are not necessarily convex, where K is the number of queries to the stochastic first-order oracle. This improves the best known convergence rate and provides a new benchmark for the stochastic composition problem.

3. We provide a comprehensive analysis and results that apply to various special cases. In particular, our results contain as special cases the known optimal rate results for the expectation minimization problem, i.e., $O(1/\sqrt{K})$ for general objectives and $O(1/K)$ for strongly convex objectives.

4. In the special case where the inner function $g(\cdot)$ is a linear mapping, we show that it is sufficient to use one timescale to guarantee convergence. Our result achieves the non-improvable rate of convergence $O(1/K)$ for optimal strongly convex optimization and $O(1/\sqrt{K})$ for convex optimization and nonconvex smooth optimization. It implies that the inner linearity does not bring fundamental difficulty to the stochastic composition problem.

5. We show that the proposed method leads to a new on-policy reinforcement learning algorithm. The new learning algorithm achieves the optimal convergence rate $O(1/\sqrt{K})$ for solving Bellman equations based on K observations of state-to-state transitions.

In comparison with Wang et al. (2017), our analysis is more succinct and leads to stronger results. To the best of our knowledge, Theorems 1,2,3 in this paper provide the best-known rates for the stochastic composition problem.

Paper Organization. Section 2 states the sampling oracle and the accelerated stochastic compositional proximal gradient algorithm (ASC-PG). Section 3 states the convergence rate results in the case of general nonconvex objective and in the case of strongly convex objective, respectively. Section 4 describes an application of ASC-PG to reinforcement learning and gives numerical experiments.

Notations and Definitions. For $\mathbf{x} \in \Re^n$, we denote by $\mathbf{x}'$ its transpose, and by $\|\mathbf{x}\|$ its Euclidean norm (i.e., $\|\mathbf{x}\| = \sqrt{\mathbf{x}'\mathbf{x}}$). For two sequences $\{\mathbf{y}_k\}$ and $\{\mathbf{z}_k\}$, we write $\mathbf{y}_k = O(\mathbf{z}_k)$ if there exists a constant $c > 0$ such that $\|\mathbf{y}_k\| \leq c\|\mathbf{z}_k\|$ for each k. We denote by $\mathbf{I}_{\text{condition}}^{\text{value}}$ the indicator function, which returns "value" if the "condition" is satisfied; otherwise 0. We denote by H^* the optimal objective function value of problem (1), denote by X^* the set of optimal solutions, and denote by $\mathcal{P}_S(\mathbf{x})$ the Euclidean projection of $\mathbf{x}$ onto S for any convex set S. We also denote by short that $f(\mathbf{y}) = \mathbb{E}_v[f_v(\mathbf{y})]$ and $g(\mathbf{x}) = \mathbb{E}_w[g_w(\mathbf{x})]$.

2. Algorithm

We focus on the black-box sampling environment. Suppose that we have access to a stochastic first-order oracle, which returns random realizations of first-order information upon queries. This is a typical simulation oracle that is available in both online and batch learning. More specifically, assume that we are given a **Sampling Oracle** (**SO**) such that

- Given some $\mathbf{x} \in \Re^n$, the **SO** returns a random vector $g_w(x)$ and a noisy subgradient $\nabla g_w(\mathbf{x})$.
- Given some $\mathbf{y} \in \Re^m$, the **SO** returns a noisy gradient $\nabla f_v(\mathbf{y})$.

Now we propose the Accelerated Stochastic Compositional Proximal Gradient (ASC-PG) algorithm, see Algorithm 1. ASC-PG is a generalization of the SCGD proposed by Wang et al. (2017), in which a proximal step is used to replace the projection step.

Algorithm 1 Accelerated Stochastic Compositional Proximal Gradient (ASC-PG)

Require: $\mathbf{x}_1 \in \Re^n$, **SO**, K, stepsize sequences $\{\alpha_k\}_{k=1}^K$, and $\{\beta_k\}_{k=1}^K$.
Ensure: $\{\mathbf{x}_k\}_{k=1}^K$

1: Initialize $\mathbf{z}_1 = \mathbf{x}_1$ and $\mathbf{y}_1 = 0$.
2: **for** $k = 1, \cdots, K$ **do**
3: Query the **SO** and obtain gradient samples $\nabla f_{v_k}(\mathbf{y}_k)$, $\nabla g_{w_k}(\mathbf{z}_k)$.
4: Update the main iterate by

$$\mathbf{x}_{k+1} = \text{prox}_{\alpha_k R(\cdot)}\left(\mathbf{x}_k - \alpha_k \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\right).$$

5: Update auxillary iterates by an *extrapolation-smoothing* scheme:

$$\begin{aligned}\mathbf{z}_{k+1} &= \left(1 - \frac{1}{\beta_k}\right)\mathbf{x}_k + \frac{1}{\beta_k}\mathbf{x}_{k+1},\\ \mathbf{y}_{k+1} &= (1-\beta_k)\mathbf{y}_k + \beta_k g_{w_{k+1}}(\mathbf{z}_{k+1}),\end{aligned}$$

where the sample $g_{w_{k+1}}(\mathbf{z}_{k+1})$ is obtained via querying the **SO**.
6: **end for**

In Algorithm 1, the *extrapolation-smoothing scheme* (i.e., the $(\mathbf{y}, \mathbf{z})$-step) is critical to the acceleration of convergence. The acceleration is due to the fast running estimation of the unknown quantity $g(\mathbf{x}_k) := \mathbb{E}_w[g_w(\mathbf{x}_k)]$. At iteration k, the running estimate $\mathbf{y}_k$ of $g(\mathbf{x}_k)$ is obtained using a weighted smoothing scheme, corresponding to the $\mathbf{y}$-step; while the new query point $\mathbf{z}_{k+1}$ is obtained through extrapolation, corresponding to the $\mathbf{z}$-step. The updates are constructed in a way such that $\mathbf{y}_k$ is a nearly unbiased estimate of $g(\mathbf{x}_k)$. To see how the extrapolation-smoothing scheme works, we let the weights be

$$\xi_t^{(k)} = \begin{cases}\beta_t \prod_{i=t+1}^k (1-\beta_i), & \text{if } k > t \geq 0\\ \beta_k, & \text{if } k = t \geq 0.\end{cases} \tag{3}$$

Then,we can verify the following important relations:

$$\mathbf{x}_{k+1} = \sum_{t=0}^k \xi_t^{(k)} \mathbf{z}_{t+1}, \qquad \mathbf{y}_{k+1} = \sum_{t=0}^k \xi_t^{(k)} g_{w_{t+1}}(\mathbf{z}_{t+1}),$$

which essentially say that $\mathbf{x}_{k+1}$ is a damped weighted average of $\{z_{t+1}\}_0^{k+1}$ and $\mathbf{y}_{k+1}$ is a damped weighted average of $\{g_{w_{t+1}}(z_{t+1})\}_0^{k+1}$.

An Analytical Example of the Extrapolation-Smooth Scheme Now consider the special case where $g_w(\cdot)$ is always a linear mapping $g_w(z) = A_w z + b_z$ and $\beta_k = 1/(k+1)$. We can verify that $\xi_t^{(k)} = 1/(k+1)$ for all t. Then we have

$$g(\mathbf{x}_{k+1}) = \frac{1}{k+1}\sum_{t=0}^k \mathbb{E}[A_w]\mathbf{z}_{t+1} + \mathbb{E}[\mathbf{b}_w], \qquad \mathbf{y}_{k+1} = \frac{1}{k+1}\sum_{t=0}^k A_{w_{t+1}}\mathbf{z}_{t+1} + \frac{1}{k+1}\sum_{t=0}^k \mathbf{b}_{w_{t+1}}.$$

In this way, we can see that the scaled error

$$k(\mathbf{y}_{k+1} - g(\mathbf{x}_{k+1})) = \sum_{t=0}^k (A_{w_{t+1}} - \mathbb{E}[A_w])\mathbf{z}_{t+1} + \sum_{t=0}^k (\mathbf{b}_{w_{t+1}} - \mathbb{E}[\mathbf{b}_w])$$

is a *zero-mean* and *zero-drift* martingale. Under additional technical assumptions, we have

$$\mathbb{E}[\|\mathbf{y}_{k+1} - g(\mathbf{x}_{k+1})\|^2] \leq O\left(\frac{1}{k}\right).$$

Note that the zero-drift property of the error martingale is the key to the fast convergence rate. The zero-drift property comes from the near-unbiasedness of $\mathbf{y}_k$, which is due to the special construction of the extrapolation-smoothing scheme. In the more general case where g_w is not necessarily linear, we can use a similar argument to show that $\mathbf{y}_k$ is a nearly unbiased estimate of $g(\mathbf{x}_k)$. As a result, the extrapolation-smoothing $(\mathbf{y}, \mathbf{z})$-step ensures that $\mathbf{y}_k$ tracks the unknown quantity $g(\mathbf{x}_k)$ efficiently.

3. Main Results

We present our main theoretical results in this section. Note that for ease of presentation, we defer all detailed proofs for the theorems, and technical lemmas to Appendix. Let us begin by stating our assumptions. Note that all assumptions involving random realizations of v, w hold with probability 1.

Assumption 1 *The samples generated by the* **SO** *are unbiased in the following sense:*

1. $\mathbb{E}_{\{w_k, v_k\}}(\nabla g_{w_k}^\top(\mathbf{x})\nabla f_{v_k}(\mathbf{y})) = \nabla g^\top(\mathbf{x})\nabla f(\mathbf{y}) \quad \forall k = 1, 2, \cdots, K, \quad \forall \mathbf{x}, \forall \mathbf{y}.$
2. $\mathbb{E}_{w_k}(g_{w_k}(\mathbf{x})) = g(\mathbf{x}) \quad \forall \mathbf{x}.$

Note that w_k and v_k are not necessarily independent.

Assumption 2 *The sample gradients and values generated by the* **SO** *satisfy*

$$\mathbb{E}_w(\|g_w(\mathbf{x}) - g(\mathbf{x})\|^2) \leq \sigma^2 \quad \forall \mathbf{x}.$$

Assumption 3 *The sample gradients generated by the* **SO** *are uniformly bounded, and the penalty function R has bounded gradients.*

$$\|\nabla f_v(\mathbf{x})\| \leq \Theta(1), \quad \|\nabla g_w(\mathbf{x})\| \leq \Theta(1), \quad \|\partial R(\mathbf{x})\| \leq \Theta(1) \quad \forall \mathbf{x}, \forall w, \forall v$$

Assumption 4 *There exists $L_F, L_f, L_g > 0$ such that*

1. $F(\mathbf{z}) - F(\mathbf{x}) \leq \langle \nabla F(\mathbf{x}), \mathbf{z} - \mathbf{x} \rangle + \frac{L_F}{2}\|\mathbf{z} - \mathbf{x}\|^2 \quad \forall \mathbf{x}\ \forall \mathbf{z}.$
2. $\|\nabla f_v(\mathbf{y}) - \nabla f_v(\mathbf{w})\| \leq L_f \|\mathbf{y} - \mathbf{w}\| \quad \forall \mathbf{y}\ \forall \mathbf{w}\ \forall v.$
3. $\|g(\mathbf{x}) - g(\mathbf{z}) - \nabla g(\mathbf{z})^\top (\mathbf{x} - \mathbf{z})\| \leq \frac{L_g}{2}\|\mathbf{x} - \mathbf{z}\|^2 \quad \forall \mathbf{x}\ \forall \mathbf{z}.$

Condition 1 of Assumption 4 requires that the function $F(\cdot)$ is L_f-smooth. Condition 2 requires that f_v has Lipschitz gradient for all v. Condition 3 requires that the function $g(\cdot)$ is L_g smooth. Note that the case where $L_f = 0$ or $L_g = 0$ coincides with the special case where either f_v or g is linear, respectively. The constants L_F, L_f, L_g jointly characterize the smoothness and complexity of stochastic composition optimization. They do not admit any straightforward dependence relation.

Our first main result concerns with general optimization problems which are not necessarily convex.

Theorem 1 (Smooth (Nonconvex) Optimization) *Let Assumptions 1, 2, 3, and 4 hold. Denote by $F(\mathbf{x}) := (\mathbb{E}_v(f_v) \circ \mathbb{E}_w(g_w))(\mathbf{x})$ for short and suppose that $R(\mathbf{x}) = 0$ in (1) and $\mathbb{E}(F(\mathbf{x}_k))$ is bounded from above. Choose $\alpha_k = k^{-a}$ and $\beta_k = 2k^{-b}$ where $a \in (0,1)$ and $b \in (0,1)$ in Algorithm 1. Then we have*

$$\frac{\sum_{k=1}^K \mathbb{E}(\|\nabla F(\mathbf{x}_k)\|^2)}{K} \leq O(K^{a-1} + L_f^2 L_g K^{4b-4a} \mathbf{I}_{4a-4b=1}^{\log K} + L_f^2 K^{-b} + K^{-a}). \tag{4}$$

If $L_g \neq 0$ and $L_f \neq 0$, choose $a = 5/9$ and $b = 4/9$, yielding

$$\frac{\sum_{k=1}^K \mathbb{E}(\|\nabla F(\mathbf{x}_k)\|^2)}{K} \leq O(K^{-4/9}). \tag{5}$$

If $L_g = 0$ or $L_f = 0$, choose $a = b = 1/2$, yielding

$$\frac{\sum_{k=1}^K \mathbb{E}(\|\nabla F(\mathbf{x}_k)\|^2)}{K} \leq O(K^{-1/2}). \tag{6}$$

The result of Theorem 1 strictly improves the best-known results given by Wang et al. (2017). First the result of (5) improves the finite-sample error bound from $O(k^{-2/7})$ to $O(k^{-4/9})$ for general convex and nonconvex optimization. This improves the best known convergence rate and provides a new benchmark for the stochastic composition problem. Note that it is possible to relax the condition "$\mathbb{E}(F(\mathbf{x}_k))$ is bounded from above" in Theorem 1. However, it would make the analysis more cumbersome and yield an additional term $\log K$ in the error bound.

Our second main result concerns strongly convex objective functions. We say that the objective function H is *optimally strongly convex* with parameter $\lambda > 0$ if

$$H(\mathbf{x}) - H(\mathcal{P}_{X^*}(\mathbf{x})) \geq \lambda \|\mathbf{x} - \mathcal{P}_{X^*}(\mathbf{x})\|^2 \quad \forall \mathbf{x}. \tag{7}$$

(see Liu and Wright (2015)). Note that any strongly convex function is optimally strongly convex, but the reverse does not hold. For example, the objective function (2) in on-policy reinforcement learning is always optimally strongly convex (even if $\mathbb{E}(A)$ is a rank deficient matrix), but not necessarily strongly convex.

We point out that there are other class of functions that are not strongly convex, for which first-order algorithms still achieve linear convergence. See Karimi et al. (2016) and Necoara et al. (2015) for examples. For ease of presentation, in this work, we only consider optimally strongly convex functions.

Theorem 2 *(Strongly Convex Optimization) Suppose that the objective function $H(\mathbf{x})$ in (1) is optimally strongly convex with parameter $\lambda > 0$ defined in (7). Set $\alpha_k = C_a k^{-a}$ and $\beta_k = C_b k^{-b}$ where $C_a > 4\lambda$, $C_b > 2$, $a \in (0,1]$, and $b \in (0,1]$ in Algorithm 1. Under Assumptions 1, 2, 3, and 4, we have*

$$\mathbb{E}(\|\mathbf{x}_K - \mathcal{P}_{X^*}(\mathbf{x}_K)\|^2) \leq O\left(K^{-a} + L_f^2 L_g K^{-4a+4b} + L_f^2 K^{-b}\right). \tag{8}$$

If $L_g \neq 0$ and $L_f \neq 0$, choose $a = 1$ and $b = 4/5$, yielding

$$\mathbb{E}(\|\mathbf{x}_K - \mathcal{P}_{X^*}(\mathbf{x}_K)\|^2) \leq O(K^{-4/5}). \tag{9}$$

If $L_g = 0$ or $L_f = 0$, choose $a = 1$ and $b = 1$, yielding

$$\mathbb{E}(\|\mathbf{x}_K - \mathcal{P}_{X^*}(\mathbf{x}_K)\|^2) \leq O(K^{-1}). \tag{10}$$

Let us discuss the results of Theorem 2. In the general case where $L_f \neq 0$ and $L_g \neq 0$, the convergence rate in (9) is consistent with the result of Wang et al. (2017). Now consider the special case where $L_g = 0$, i.e., the inner mapping is linear. This result finds an immediate application to Bellman error minimization problem (2) which arises from reinforcement learning problem in (and with ℓ_1 norm regularization). The proposed ASC-PG algorithm is able to achieve the optimal rate $O(1/K)$ without any extra assumption on the outer function f_v. To the best of our knowledge, this is the best (also optimal) sample-error complexity for on-policy reinforcement learning.

Theorem 3 *(General Convex Optimization) Suppose that both $F(\mathbf{x})$ and $H(\mathbf{x})$ are convex, and the feasible set X is bounded. Set $\alpha_k = \Theta(k^{-a})$ and $\beta_k = \Theta(k^{-b})$ with $a \in (0,1)$, and $b \in (0,1]$ in Algorithm 1. Under Assumptions 1, 2, 3, and 4, we have for any $c > 0$:*

$$\begin{aligned} H(\bar{x}_K) - H^* \leq O\Big(&K^{a-1} + K^{a-c}\mathbf{I}_{c=1}^{\log K} + K^{-a} \\ &+ L_f^2 L_g K^{-5a+4b+c} \mathbf{I}_{6a-4b-c=1}^{\log K} + L_f^2 K^{-a-b+c} \mathbf{I}_{2a+b-c=1}^{\log K}\Big). \end{aligned} \tag{11}$$

where $\bar{\mathbf{x}}_K := \frac{\sum_{k=2}^{K+1} \alpha_k \mathbf{x}_k}{\sum_{k=2}^{K+1} \alpha_k}$. *If* $L_g \neq 0$ *and* $L_f \neq 0$, *choose* $a = 5/7$, $b = 4/7$, $c = 1$, *yielding*

$$H(\bar{x}_K) - H^* \leq O(K^{-2/7}). \tag{12}$$

If $L_g = 0$ *or* $L_f = 0$, *choose* $a = 1/2$, $b = 1$, *and* $c = 1$, *yielding*

$$H(\bar{x}_K) - H^* \leq O(K^{-1/2} \log K). \tag{13}$$

Theorem 3 gives stronger results than the best-known results of Wang et al. (2017) and requires milder assumptions. In the general case where $L_f \neq 0$ and $L_g \neq 0$, the convergence rate in (12) matches the result of Wang et al. (2017). When either $L_g = 0$ (i.e., the inner mapping is linear or $L_f = 0$ (i.e., the outer mapping is linear), the proposed ASC-PG algorithm is able to achieve the optimal rate $O(1/\sqrt{K})$ up to a logarithmic factor.

Remarks Theorems 1, 2, and 3 give important implications about the special cases where $L_f = 0$ or $L_g = 0$. In these cases, we argue that our convergence rate (10) is "*optimal*" with respect to the sample size K. To see this, it is worth pointing out the $O(1/K)$ rate of convergence is optimal for strongly convex expectation minimization problem. Because the expectation minimization problem is a special case of the problem (1), the $O(1/K)$ convergence rate must be optimal for the stochastic composition problem too. Similarly, the $O(1/\sqrt{K})$ rate of convergence is also optimal for general convex optimization due to the same argument.

- Consider the case where $L_f = 0$, which means that the outer function $f_v(\cdot)$ is linear with probability 1. Then the stochastic composition problem (1) reduces to an expectation minimization problem since $(\mathbb{E}_v f_v \circ \mathbb{E}_w g_w)(\mathbf{x}) = \mathbb{E}_v(f_v(\mathbb{E}_w g_w(\mathbf{x}))) = \mathbb{E}_v \mathbb{E}_w (f_v \circ g_w)(\mathbf{x})$. Therefore, it makes a perfect sense to obtain the optimal convergence rate.
- Consider the case where $L_g = 0$, which means that the inner function $g(\cdot)$ is a linear mapping. The result is quite surprising. Note that even $g(\cdot)$ is a linear mapping, it does not reduce problem (1) to an expectation minimization problem. However, the ASC-PG still achieves the optimal convergence rate. This suggests that, when inner linearity holds, the stochastic composition problem (1) is not fundamentally more difficult than the expectation minimization problem.

The convergence rate results unveiled in Theorems 1 and 2 are the best known results for the composition problem. We believe that they provide important new result which provides insights into the complexity of the stochastic composition problem.

4. Application to Reinforcement Learning

In this section, we apply the proposed ASC-PG algorithm to conduct policy value evaluation in reinforcement learning through attacking Bellman equations. Suppose that there are in total S states. Let the policy of interest be π. Denote the value function of states by $V^\pi \in \Re^S$, where $V^\pi(s)$ denotes the value of being at state s under policy π. The Bellman equation of the problem is

$$V^\pi(s_1) = \mathbb{E}_\pi\{r_{s_1,s_2} + \gamma \cdot V^\pi(s_2)|s_1\} \text{ for all } s_1, s_2 \in \{1, ..., S\},$$

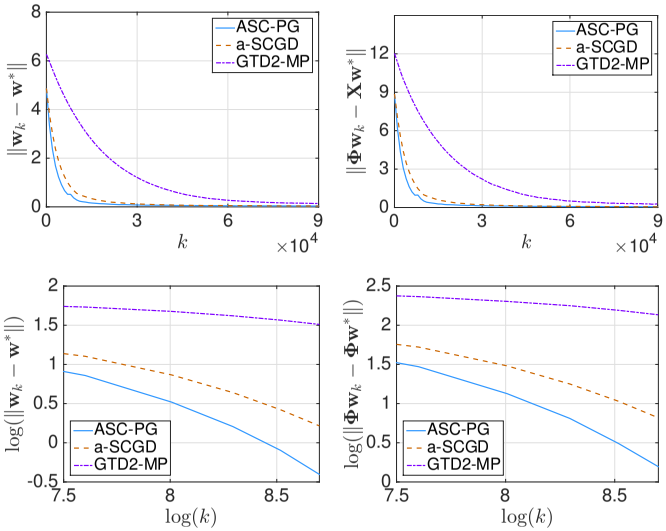


Figure 1: Empirical convergence rate of the ASC-PG algorithm and the GTD2-MP algorithm under Experiment 1 averaged over 100 runs, where $\mathbf{w}_k$ denotes the solution at the k-th iteration.

where r_{s_1,s_2} denotes the reward of moving from state s_1 to s_2, and the expectation is taken over all possible future state s_2 conditioned on current state s_1 and the policy π. We have that the solution $V^* \in \Re^S$ to the above equation satisfies that $V^* = V^\pi$. Here a moderately large S will make solving the Bellman equation directly impractical. To resolve the curse of dimensionality, in many practical applications, we approximate the value of each state by some linear map of its feature $\boldsymbol{\phi}_s \in \Re^d$, where $d < S$ to reduce the dimension. In particular, we assume that $V^\pi(s) \approx \boldsymbol{\phi}_s^T \mathbf{w}^*$ for some $\mathbf{w}^* \in \Re^m$.

To compute $\mathbf{w}^*$, we formulate the problem as a Bellman residual minimization problem that

$$\min_{\mathbf{w}} \sum_{s=1}^{S} (\boldsymbol{\phi}_s^T \mathbf{w} - q_{\pi,s}(\mathbf{w}))^2,$$

where $q_{\pi,s}(\mathbf{w}) = \mathbb{E}_\pi\{r_{s,s'} + \gamma \cdot \boldsymbol{\phi}_{s'}^T \mathbf{w}\} = \sum_{s'} P_{ss'}^\pi(\{r_{s,s'} + \gamma \cdot \boldsymbol{\phi}_{s'}^T \mathbf{w}\}$; $\gamma < 1$ is a discount factor, and $r_{s,s'}$ is the random reward of transition from state s to state s'. It is clearly seen that the proposed ASC-PG algorithm could be directly applied to solve this problem where we take

$$g(\mathbf{w}) = (\boldsymbol{\phi}_1^T \mathbf{w}, q_{\pi,1}(\mathbf{w}), ..., \boldsymbol{\phi}_S^T \mathbf{w}, q_{\pi,S}(\mathbf{w})) \in \Re^{2S},$$

$$f\Big((\boldsymbol{\phi}_1^T \mathbf{w}, q_{\pi,1}(\mathbf{w}), ..., \boldsymbol{\phi}_S^T \mathbf{w}, q_{\pi,S}(\mathbf{w}))\Big) = \sum_{s=1}^{S} (\boldsymbol{\phi}_s \mathbf{w} - q_{\pi,s}(\mathbf{w}))^2 \in \Re.$$

We point out that the $g(\cdot)$ function here is a linear map. By our theoretical analysis, we expect to achieve a faster $O(1/k)$ rate of convergence, which is justified empirically in our later simulation study.

We consider three experiments, where in the first two experiments, we compare our proposed accelerated ASC-PG algorithm with accelerated SCGD (a-SCGD) algorithm (Wang et al., 2017) and the recently proposed GTD2-MP algorithm (Liu et al., 2015). Also, in the first two experiments, we do not add any regularization term, i.e. $R(\cdot) = 0$. We point out here that in experiments 1 and 2, since there is no penalization term, i.e., $R(\cdot) = 0$, the ASC-PG algorithm essentially matches the a-SCGD algorithm in Wang et al. (2017). However, note that the choice of stepsizes $\{\alpha_k\}$ and $\{\beta_k\}$ of ASC-PG are different from the stepsizes of a-SCGD, where for ASC-PG $\alpha_k = k^{-1}$ and $\beta_k = k^{-1}$, and for a-SCGD $\alpha_k = k^{-1}$ and $\beta_k = k^{-4/5}$. In the third experiment, we add an ℓ_1-penalization term $\lambda\|\mathbf{w}\|_1$. In all cases, we choose the step sizes via comparison studies as in Dann et al. (2014):

- Experiment 1: We use the Baird's example (Baird, 1995), which is a well-known example to test the off-policy convergent algorithms. This example contains $S = 6$ states, and two actions at each state. We refer the readers to Baird (1995) for more detailed information of the example.

- Experiment 2: We generate a Markov decision problem (MDP) using similar setup as in White and White (2016). In each instance, we randomly generate an MDP which contains $S = 100$ states, and three actions at each state. The dimension of the Given one state and one action, the agent can move to one of four next possible states. In our simulation, we generate the transition probabilities for each MDP instance uniformly from $[0, 1]$ and normalize the sum of transitions to one, and we generate the reward for each transition also uniformly in $[0, 1]$.

- Experiment 3: We generate the data same as Experiment 2 except that we have a larger $d = 100$ dimensional feature space, where only the first 4 components of $\mathbf{w}^*$ are non-zeros. We add an ℓ_1-regularization term, $\lambda\|\mathbf{w}\|_1$, to the objective function.

Denote by $\mathbf{w}_k$ the solution at the k-th iteration. For the first two experiments, we report the empirical convergence performance $\|\mathbf{w}_k - \mathbf{w}^*\|$ and $\|\mathbf{\Phi}\mathbf{w}_k - \mathbf{\Phi}\mathbf{w}^*\|$, where $\mathbf{\Phi} = (\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_S)^T \in \Re^{S\times d}$ and $\mathbf{\Phi}\mathbf{w}^* = V$, and all $\mathbf{w}_k$'s are averaged over 100 runs, in the first two subfigures of Figures 1 and 2. It is seen that the ASC-PG algorithm achieves the fastest convergence rate empirically in both experiments. To further evaluate our theoretical results, we plot $\log(t)$ vs. $\log(\|\mathbf{w}_k - \mathbf{w}^*\|)$ (or $\log(\|\mathbf{\Phi}\mathbf{w}_k - \mathbf{\Phi}\mathbf{w}^*\|)$ averaged over 100 runs for the first two experiments in the second two subfigures of Figures 1 and 2. The empirical results further support our theoretical analysis that $\|\mathbf{w}_k - \mathbf{w}^*\|^2 = O(1/k)$ for the ASC-PG algorithm when $g(\cdot)$ is a linear mapping.

For Experiment 3, as the optimal solution is unknown, we run the ASC-PG algorithm for one million iterations and take the corresponding solution as the optimal solution $\hat{\mathbf{w}}^*$, and we report $\|\mathbf{w}_k - \hat{\mathbf{w}}^*\|$ and $\|\mathbf{\Phi}\mathbf{w}_k - \mathbf{\Phi}\hat{\mathbf{w}}^*\|$ averaged over 100 runs in Figure 3. It is seen the the ASC-PG algorithm achieves fast empirical convergence rate.

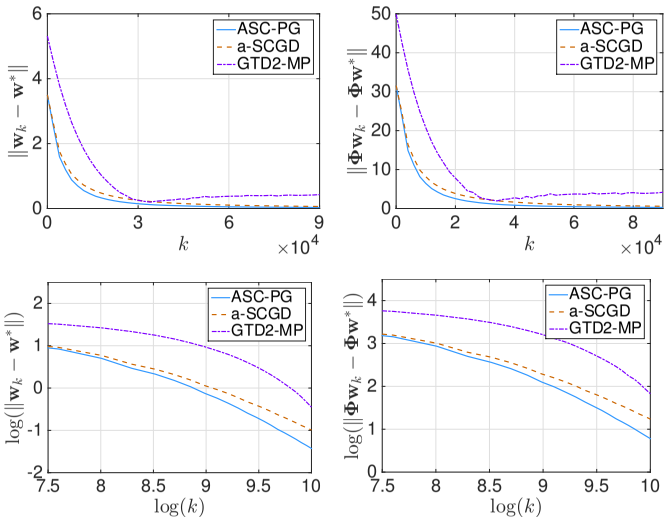


Figure 2: Empirical convergence rate of the ASC-PG algorithm and the GTD2-MP algorithm under Experiment 2 averaged over 100 runs, where $\mathbf{w}_k$ denotes the solution at the k-th iteration.

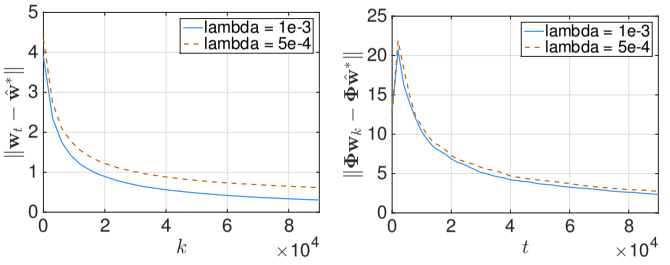


Figure 3: Empirical convergence rate of the ASC-PG algorithm with the ℓ_1-regularization term $\lambda\|\mathbf{w}\|_1$ under Experiment 3 averaged over 100 runs, where $\mathbf{w}_k$ denotes the solution at the t-th iteration.

5. Conclusion

We develop the first proximal gradient method for stochastic composition optimization with nonsmooth penalties. The algorithm updates by interacting with a stochastic first-order

oracle. Finite-sample convergence rates are established under a variety of assumptions, which provide new rate benchmarks that improve the best-known results prior to this paper. Application of the ASC-PG to reinforcement learning leads to a new on-policy learning algorithm, which achieves faster convergence than the best known algorithms. For future research, it remains open whether or under what circumstances the current $O(K^{-4/9})$ can be further improved. Another direction is to customize and adapt the algorithm and analysis to more specific problems arising from reinforcement learning and risk-averse optimization, in order to fully exploit the potential of the proposed method.

Acknowledgments

We thank the Editor, the Action Editor and two anonymous reviewers for the careful consideration and invaluable suggestions, which help us further improve the paper. This project is in part supported by NSF grants CCF1718513, CNS-1548078, DMS-10009141, CMMI-1653435, and by the National Institute on Drug Abuse grant P50 DA039838. The content of this manuscript is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute on Drug Abuse or the National Institutes of Health.

Appendix

This appendix includes all detailed proofs for the theorems in the main text. The key idea of deriving the rate convergence of our algorithm is to find a sequence of random vectors associated with the output solutions at each iteration, which converge at fast rates. To better analyze the convergence process, we have constructed two auxiliary sequences of random variables $\{m_k\}$ and $\{n_k\}$, which are defined in the beginng of Lemma 7, that converge jointly to 0 at a fast rate. The convergence of $\{m_k\}$ and $\{n_k\}$ further imply that $\{\mathbf{x}_k\}$ and $\{\mathbf{y}_k\}$ converge rapidly to their limits respectively. Then we are able to improve the convergence rate of $\mathbf{y}_k$. Intuitively, we show that the improved iterate $\mathbf{y}_k$ tracks the unknown quantity $\mathbb{E}[g(x_k)]$ more closely than in the previous work. This explains why our new algorithm achieves an improved rate of convergence and sample complexity.

A. Proof to Theorem 1

In this section, we provide the detailed proof for Theorem 1. We start by some technical lemmas, The first one provides a convergence result of the generated solutions, where we bound the term $\|\mathbf{x}_k - \mathbf{x}_{k+1}\|^2$ as k increases.

Lemma 4 *Under Assumption 3, two subsequent iterates in Algorithm 1 satisfy*

$$\|\mathbf{x}_k - \mathbf{x}_{k+1}\|^2 \leq \Theta(\alpha_k^2).$$

Proof From the definition of the proximal operation, we have

$$\begin{aligned}\mathbf{x}_{k+1} &= \text{prox}_{\alpha_k R(\cdot)}(\mathbf{x}_k - \alpha_k \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)) \\ &= \arg\min_x \frac{1}{2}\|\mathbf{x} - \mathbf{x}_k + \alpha_k \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\|^2 + \alpha_k R(\mathbf{x}).\end{aligned}$$

The optimality condition suggests the following equality:

$$\mathbf{x}_{k+1} - \mathbf{x}_k = -\alpha_k(\nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k) + \mathbf{s}_{k+1}) \tag{14}$$

where $\mathbf{s}_{k+1} \in \partial R(\mathbf{x}_{k+1})$ is some vector in the sub-differential set of $R(\cdot)$ at $\mathbf{x}_{k+1}$. Then apply the boundedness condition in Assumption 3 to yield

$$\begin{aligned}\|\mathbf{x}_{k+1} - \mathbf{x}_k\| &= \alpha_k\|(\nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k) + \mathbf{s}_{k+1})\| \\ &\leq \alpha_k(\|\nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\| + \|\mathbf{s}_{k+1}\|) \\ &\leq \alpha_k(\|\nabla g_{w_k}^\top(\mathbf{x}_k)\|\|\nabla f_{v_k}(\mathbf{y}_k)\| + \|\mathbf{s}_{k+1}\|) \\ &\overset{\text{(Assumption 3)}}{\leq} \Theta(1)\alpha_k,\end{aligned}$$

which implies the claim. ■

Next, we provide a simple bound which quantifies the difference between the random variable $\|\nabla g_w^\top(\mathbf{x})\nabla_v f(g(\mathbf{x})) - \nabla g_w^\top(\mathbf{x})\nabla_v f(\mathbf{y}))\|$ and its population version $\|\mathbf{y} - g(\mathbf{x})\|$.

Lemma 5 *Under Assumptions 3 and 4, we have*

$$\|\nabla g_w^\top(\mathbf{x})\nabla_v f(g(\mathbf{x})) - \nabla g_w^\top(\mathbf{x})\nabla_v f(\mathbf{y}))\| \leq \Theta(L_f\|\mathbf{y} - g(\mathbf{x})\|).$$

Proof We have

$$\begin{aligned}\|\nabla g_w^\top(\mathbf{x})\nabla f_v(g(\mathbf{x})) - \nabla g_w^\top(\mathbf{x})\nabla f_v(\mathbf{y}))\| &\leq \|\nabla g_w^\top(\mathbf{x})\|\|\nabla f_v(g(\mathbf{x})) - \nabla f_v(\mathbf{y})\| \\ &\overset{\text{(Assumption 3)}}{\leq} \Theta(1)\|\nabla f_v(g(\mathbf{x})) - \nabla f_v(\mathbf{y})\| \\ &\overset{\text{(Assumption 4)}}{\leq} \Theta(L_f)\|\mathbf{y} - g(\mathbf{x})\|.\end{aligned}$$

It completes the proof. ■

Then, we prove two lemmas which provides the convergence rate of $\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2)$.

Lemma 6 *Given two sequences of positive scalars $\{s_k\}_{k=1}^\infty$ and $\{\gamma_k\}_{k=1}^\infty$ satisfying*

$$s_{k+1} \leq (1 - \gamma_k + C_1\gamma_k^2)s_k + C_2 k^{-a} \tag{15}$$

where $C_1 \geq 0$, $C_2 \geq 0$, and $a \geq 0$. Letting $\gamma_k \in \Re$ as $\gamma_k = C_3 k^{-b}$ where $b \in (0, 1]$ and $C_3 > 2$, the sequence can be bounded by

$$v_k \leq Ck^{-c}$$

where C and c are defined as

$$C := \max_{k \leq (C_1 C_3^2)^{1/b}+1} s_k k^c + \frac{C_2}{C_3 - 2} \quad \text{and} \quad c := a - b.$$

In other words, we have

$$s_k \leq \Theta(k^{-a+b}).$$

Proof We prove it by induction. First it is easy to verify that the claim holds for $k \leq (C_1C_3^2)^{1/b}$ from the definition for C. Next we prove from "k" to "$k+1$", that is, given $s_k \leq Ck^{-c}$ for $k > (C_1C_3^2)^{1/b}$, we need to prove $s_{k+1} \leq C(k+1)^{-c}$.

$$\begin{aligned} s_{k+1} &\leq (1-\gamma_k+C_1\gamma_k^2)s_k + C_2k^{-a} \\ &\leq (1-C_3k^{-b}+C_1C_3^2k^{-2b})Ck^{-c} + C_2k^{-a} \\ &= Ck^{-c} - CC_3k^{-b-c} + CC_1C_3^2k^{-2b-c} + C_2k^{-a}. \end{aligned} \tag{16}$$

To prove that (16) is bounded by $C(k+1)^{-c}$, it suffices to show that

$$\Delta := (k+1)^{-c} - k^{-c} + C_3k^{-b-c} - C_1C_3^2k^{-2b-c} > 0 \quad \text{and} \quad C \geq \frac{C_2k^{-a}}{\Delta}.$$

From the convexity of function $h(t) = t^{-c}$, we have the inequality $(k+1)^{-c} - k^{-c} \geq (-c)k^{-c-1}$. Therefore we obtain

$$\begin{aligned} \Delta &\geq -ck^{-c-1} + C_3k^{-b-c} - C_1C_3^2k^{-2b-c} \\ &\overset{(b\leq 1,\ k>(C_1C_3^2)^{1/b})}{\geq} (C_3-2)(k^{-b-c}) \\ &\overset{(C_3>2)}{>} 0. \end{aligned}$$

To verify the second one, we have

$$\frac{C_2k^{-a}}{\Delta} \leq \frac{C_2}{C_3-2}k^{-a+b+c} \overset{(c=a+b)}{=} \frac{C_2}{C_3-2} \leq C$$

where the last inequality is due to the definition of C. It completes the proof. ■

Lemma 7 *Choose β_k to be $\beta_k = C_bk^{-b}$ where $C_b > 2$, $b \in (0,1]$, and $\alpha_k = C_ak^{-a}$. Under Assumptions 1 and 2, we have*

$$\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2) \leq L_g\Theta(k^{-4a+4b}) + \Theta(k^{-b}). \tag{17}$$

Proof Denote by m_{k+1}

$$m_{k+1} := \sum_{t=0}^{k} \xi_t^{(k)} \|\mathbf{x}_{k+1} - \mathbf{z}_{t+1}\|^2$$

and n_{k+1}

$$n_{k+1} := \left\| \sum_{t=0}^{k} \xi_t^{(k)} (g_{w_{t+1}}(\mathbf{z}_{t+1}) - g(\mathbf{z}_{t+1})) \right\|$$

for short.

From Lemma 10 in (Wang et al., 2017), we have

$$\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2 \leq \left(\frac{L_g}{2}m_k + n_k\right)^2 \leq L_gm_k^2 + 2n_k^2. \tag{18}$$

From Lemma 11 in (Wang et al., 2017), m_{k+1} can be bounded by

$$m_{k+1} \leq (1-\beta_k)m_k + \beta_kq_k + \frac{2}{\beta_k}\|\mathbf{x}_k - \mathbf{x}_{k+1}\|^2 \tag{19}$$

where q_k is bounded by

$$\begin{aligned} q_{k+1} &\leq (1-\beta_k)q_k + \frac{4}{\beta_k}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\ &\overset{\text{(Lemma 4)}}{\leq} (1-\beta_k)q_k + \frac{\Theta(1)\alpha_k^2}{\beta_k} \\ &\leq (1-\beta_k)q_k + \Theta(k^{-2a+b}). \end{aligned}$$

Taking $s_k = q_k$, and $\gamma_k = \beta_k$, applying Lemma 6 implies the following decay rate

$$q_k \leq \Theta(k^{-2a+b+b}) = \Theta(k^{-2a+2b}).$$

Together with (19), we have

$$\begin{aligned} m_{k+1} &\leq (1-\beta_k)m_k + \beta_kq_k + \frac{2}{\beta_k}\|\mathbf{x}_k - \mathbf{x}_{k+1}\|^2 \\ &\leq (1-\beta_k)m_k + \Theta(k^{-2a+b}) + \Theta(k^{-2a+b}) \\ &\leq (1-\beta_k)m_k + \Theta(k^{-2a+b}), \end{aligned}$$

which leads to

$$m_k \leq \Theta(k^{-2a+2b}) \quad \text{and} \quad m_k^2 \leq \Theta(k^{-4a+4b}). \tag{20}$$

by using Lemma 6 again. Then we estimate the upper bound for $\mathbb{E}(n_k^2)$. From Lemma 11 in (Wang et al., 2017), we know $\mathbb{E}(n_k^2)$ is bounded by

$$\mathbb{E}(n_{k+1}^2) \leq (1-\beta_k)^2\mathbb{E}(\|n_k\|^2) + \beta_k^2\sigma_g^2 = (1-2\beta_k+\beta_k^2)\mathbb{E}(\|n_k\|^2) + \beta_k^2\sigma_g^2.$$

By using Lemma 6 again, we have

$$\mathbb{E}(n_k^2) \leq \Theta(k^{-b}). \tag{21}$$

Now we are ready to estimate the upper bound of $\|\mathbf{y}_{k+1} - g(\mathbf{x}_{k+1})\|^2$ by following (18)

$$\begin{aligned} \mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2) &\leq L_g\mathbb{E}(m_k^2) + 2\mathbb{E}(n_k^2) \\ &\overset{(20)+(21)}{\leq} L_g\Theta(k^{-4a+4b}) + \Theta(k^{-b}). \end{aligned}$$

It completes the proof. ■

Now we are ready to prove Theorem 1.

Proof to Theorem 1. From the Lipschitzian condition in Assumption 4, we have

$$F(\mathbf{x}_{k+1}) - F(\mathbf{x}_k)$$

$$
\begin{aligned}
&\leq && \langle \nabla F(\mathbf{x}_k),\ \mathbf{x}_{k+1}-\mathbf{x}_k\rangle + \frac{L_F}{2}\|\mathbf{x}_{k+1}-\mathbf{x}_k\|^2 \\
&\text{(Lemma 6)} \\
&\leq && -\alpha_k \langle \nabla F(\mathbf{x}_k),\ \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\rangle + \Theta(\alpha_k^2) \\
&= && -\alpha_k\|\nabla F(\mathbf{x}_k)\|^2 + \alpha_k \underbrace{\langle \nabla F(\mathbf{x}_k),\ \nabla F(\mathbf{x}_k) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\rangle}_{=:T} \\
& && +\Theta(\alpha_k^2) \qquad (22)
\end{aligned}
$$

Next we estimate the upper bound for $\mathbb{E}(T)$:

$$
\begin{aligned}
\mathbb{E}(T) &= && \mathbb{E}(\langle \nabla F(\mathbf{x}_k),\ \nabla F(\mathbf{x}_k) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(g(\mathbf{x}_k))\rangle) \\
& && +\mathbb{E}(\langle \nabla F(\mathbf{x}_k),\ \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(g(\mathbf{x}_k)) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\rangle) \\
&\text{(Assumption 1)} \\
& = && \mathbb{E}(\langle \nabla F(\mathbf{x}_k),\ \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(g(\mathbf{x}_k)) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\rangle) \\
& \leq && \frac{1}{2}\mathbb{E}(\|\nabla F(\mathbf{x}_k)\|^2) + \frac{1}{2}\mathbb{E}(\|\nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(g(\mathbf{x}_k)) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\|^2) \\
&\text{(Lemma 5)} \\
& \leq && \frac{1}{2}\mathbb{E}(\|\nabla F(\mathbf{x}_k)\|^2) + \Theta(L_f^2)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2).
\end{aligned}
$$

Take expectation on both sides of (22) and substitute $\mathbb{E}(T)$ by its upper bound:

$$
\begin{aligned}
& && \frac{\alpha_k}{2}\|\nabla F(\mathbf{x}_k)\|^2 \\
&\leq && \mathbb{E}(F(\mathbf{x}_k)) - \mathbb{E}(F(\mathbf{x}_{k+1})) + \Theta(L_f^2\alpha_k)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^2) \\
&\text{(Lemma 7)} \\
&\leq && \mathbb{E}(F(\mathbf{x}_k)) - \mathbb{E}(F(\mathbf{x}_{k+1})) + L_g\Theta(L_f^2\alpha_k)\Theta(k^{-4a+4b}) + \Theta(L_f^2\alpha_k k^{-b}) + \Theta(\alpha_k^2) \\
&\leq && \mathbb{E}(F(\mathbf{x}_k)) - \mathbb{E}(F(\mathbf{x}_{k+1})) + L_f^2 L_g\Theta(k^{-5a+4b}) + L_f^2\Theta(k^{-a-b}) + \Theta(k^{-2a})
\end{aligned}
$$

which suggests that

$$
\begin{aligned}
& && \mathbb{E}(\|\nabla F(\mathbf{x}_k)\|^2) \\
&\leq && 2\alpha_k^{-1}\mathbb{E}(F(\mathbf{x}_k)) - 2\alpha_k^{-1}\mathbb{E}(F(\mathbf{x}_{k+1})) + L_f^2 L_g\Theta(k^{-4a+4b}) + L_f^2\Theta(k^{-b}) + \Theta(k^{-a}) \\
&\leq && 2k^a\mathbb{E}(F(\mathbf{x}_k)) - 2k^a\mathbb{E}(F(\mathbf{x}_{k+1})) + L_f^2 L_g\Theta(k^{-4a+4b}) + L_f^2\Theta(k^{-b}) + \Theta(k^{-a}). \qquad (23)
\end{aligned}
$$

Sum Eq. (23) from $k=1$ to K and obtain

$$
\begin{aligned}
\frac{\sum_{k=1}^K \mathbb{E}(\|\nabla F(\mathbf{x}_k)\|^2)}{K} &\leq 2K^{-1}\alpha_1^{-1}F(\mathbf{x}_1) + K^{-1}\sum_{k=2}^K((k+1)^a - k^a)\mathbb{E}(F(\mathbf{x}_k)) \\
&\quad + K^{-1}\sum_{k=1}^K L_f^2 L_g\Theta(k^{-4a+4b}) + K^{-1}L_f^2\sum_{k=1}^K\Theta(k^{-b}) + K^{-1}\sum_{k=1}^K\Theta(k^{-a}) \\
&\leq 2K^{-1}F(\mathbf{x}_0) + K^{-1}\sum_{k=2}^K ak^{a-1}\mathbb{E}(F(\mathbf{x}_k)) \\
&\quad + K^{-1}\sum_{k=1}^K L_f^2 L_g\Theta(k^{-4a+4b}) + K^{-1}L_f^2\sum_{k=1}^K\Theta(k^{-b}) + K^{-1}\sum_{k=1}^K\Theta(k^{-a})
\end{aligned}
$$

$$
\leq O(K^{a-1} + L_f^2 L_g K^{4b-4a}\mathbf{I}_{4a-4b=1}^{\log K} + L_f^2 K^{-b} + K^{-a}),
$$

where the second inequality uses the fact that $h(t) = t^a$ is a concave function suggesting $(k+1)^a \leq k^a + ak^{a-1}$, and the last inequality uses the condition $\mathbb{E}(F(\mathbf{x}_k)) \leq \Theta(1)$.

Letting $a^* = 5/9$ and $b^* = 4/9$, we obtain the desired convergence rate $O(K^{-4/9})$. ■

B. Proof to Theorem 2

In this section, we provide the detailed proof for Theorem 2. The proof is based on the following key lemma, which provides a "contraction" property of $(\mathbb{E}(H(\mathbf{x}_{k+1})) - H^*)$ and $\mathbb{E}(\|\mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_{k+1})\|^2)$.

Lemma 8 *Assume that both $F(\mathbf{x})$ and $R(\mathbf{x})$ are convex. Under Assumptions 1, 2, 3, and 4, the iterates generated by Algorithm 1 satisfies for any sequence of positive scalars $\{\phi_k\}$:*

$$
\begin{aligned}
&2\alpha_k(\mathbb{E}(H(\mathbf{x}_{k+1})) - H^*) + \mathbb{E}(\|\mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_{k+1})\|^2) \qquad (24) \\
&\quad \leq (1+\phi_k)\mathbb{E}(\|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^3) + \Theta(L_f^2\alpha_k^2/\phi_k)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^2).
\end{aligned}
$$

Proof Following the line of the proof to Lemma 4, we have

$$
\mathbf{x}_{k+1} - \mathbf{x}_k = -\alpha_k(\nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k) + \mathbf{s}_{k+1}) \qquad (25)
$$

where $\mathbf{s}_{k+1} \in \partial R(\mathbf{x}_{k+1})$ is some vector in the sub-differential set of $R(\cdot)$ at $\mathbf{x}_{k+1}$. Then we consider $\|\mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_{k+1})\|^2$:

$$
\begin{aligned}
& && \|\mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_{k+1})\|^2 \\
&\leq && \|\mathbf{x}_{k+1} - \mathbf{x}_k + \mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2 \\
&= && \|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + 2\langle \mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_k)\rangle \\
&\overset{(25)}{=} && \|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 - 2\alpha_k\langle \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k) + \mathbf{s}_{k+1},\ \mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_k)\rangle \\
&= && \|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + 2\alpha_k\langle \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k), \mathcal{P}_{X^*}(\mathbf{x}_k) - \mathbf{x}_{k+1}\rangle \\
& && +2\alpha_k\langle \mathbf{s}_{k+1},\ \mathcal{P}_{X^*}(\mathbf{x}_k) - \mathbf{x}_{k+1}\rangle \\
&\leq && \|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + 2\alpha_k\langle \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k), \mathcal{P}_{X^*}(\mathbf{x}_k) - \mathbf{x}_{k+1}\rangle \\
& && +2\alpha_k(R(\mathcal{P}_{X^*}(\mathbf{x}_k)) - R(\mathbf{x}_{k+1})) \qquad \text{(due to the convexity of } R(\cdot)) \\
&\leq && \|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2 - \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + 2\alpha_k \underbrace{\langle \nabla F(\mathbf{x}_k), \mathcal{P}_{X^*}(\mathbf{x}_k) - \mathbf{x}_{k+1}\rangle}_{T_1} \\
& && +2\alpha_k \underbrace{\langle \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k) - \nabla F(\mathbf{x}_k),\ \mathcal{P}_{X^*}(\mathbf{x}_k) - \mathbf{x}_{k+1}\rangle}_{T_2} \\
& && +2\alpha_k(R(\mathcal{P}_{X^*}(\mathbf{x}_k)) - R(\mathbf{x}_{k+1})) \qquad (26)
\end{aligned}
$$

where the second equality follows from $\|a+b\|^2 = \|b\|^2 - \|a\|^2 + 2\langle a,\ a+b\rangle$ with $a = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $b = \mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)$. We next estimate the upper bound for T_1 and T_2 respectively:

$$
T_1 = \langle \nabla F(\mathbf{x}_k),\ \mathbf{x}_k - \mathbf{x}_{k+1}\rangle + \langle \nabla F(\mathbf{x}_k),\ -\mathbf{x}_k + \mathcal{P}_{X^*}(\mathbf{x}_k)\rangle
$$

$$\begin{aligned}
&\leq \underbrace{F(\mathbf{x}_k) - F(\mathbf{x}_{k+1}) + \frac{L_F}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2}_{\text{due to Assumption 4}} + \underbrace{F(\mathcal{P}_{X^*}(\mathbf{x}_k)) - F(\mathbf{x}_k)}_{\text{due to the convexity of } F(\cdot)} \\
&= F(\mathcal{P}_{X^*}(\mathbf{x}_k)) - F(\mathbf{x}_{k+1}) + \frac{L_F}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 \\
&\leq F(\mathcal{P}_{X^*}(\mathbf{x}_k)) - F(\mathbf{x}_{k+1}) + \Theta(\alpha_k^2),
\end{aligned}$$

where the last inequality uses Lemma 4.

$$\begin{aligned}
T_2 &= \langle \nabla F(\mathbf{x}_k) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k),\ \mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\rangle \\
&\quad + \langle \nabla F(\mathbf{x}_k) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k),\ \mathbf{x}_{k+1} - \mathbf{x}_k\rangle \\
&\leq \underbrace{\langle \nabla F(\mathbf{x}_k) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(g(\mathbf{x}_k)),\ \mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\rangle}_{T_{2,1}} \\
&\quad + \underbrace{\langle \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(g(\mathbf{x}_k)) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k),\ \mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\rangle}_{T_{2,2}} \\
&\quad + \frac{\alpha_k}{2}\underbrace{\|\nabla F(\mathbf{x}_k) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\|^2}_{T_{2,3}} + \frac{1}{2\alpha_k}\|\mathbf{x}_k - \mathbf{x}_{k+1}\|^2
\end{aligned}$$

where the last line is due to the inequality $\langle a, b\rangle \leq \frac{1}{2\alpha_k}\|a\|^2 + \frac{\alpha_k}{2}\|b\|^2$. For $T_{2,1}$, we have $\mathbb{E}(T_{2,1}) = 0$ due to Assumption 1. For $T_{2,2}$, we have

$$\begin{aligned}
T_{2,2} &\leq \frac{\alpha_k}{2\phi_k}\|\nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(g(\mathbf{x}_k)) - \nabla g_{w_k}^\top(\mathbf{x}_k)\nabla f_{v_k}(\mathbf{y}_k)\|^2 + \frac{\phi_k}{2\alpha_k}\|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2 \\
&\overset{\text{(Lemma 5)}}{\leq} \Theta\left(L_f^2\frac{\alpha_k}{\phi_k}\right)\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2 + \frac{\phi_k}{2\alpha_k}\|\mathbf{x}_k - \mathbf{x}_{k+1}\|^2.
\end{aligned}$$

$T_{2,3}$ can be bounded by a constant

$$T_{2,3} \leq 2\|\nabla F(\mathbf{x}_k)\|^2 + 2\|\nabla g_{w_k}^\top \nabla f_{v_k}(\mathbf{y}_k)\|^2 \overset{\text{(Assumption 3)}}{\leq} \Theta(1).$$

Take expectation on T_2 and put all pieces into it:

$$\mathbb{E}(T_2) \leq \Theta\left(L_f^2\frac{\alpha_k}{\phi_k}\right)\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2 + \frac{1}{2\alpha_k}(\phi_k\|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2 + \|\mathbf{x}_k - \mathbf{x}_{k+1}\|^2) + \Theta(\alpha_k).$$

Taking expectation on both sides of (26) and plugging the upper bounds of T_1 and T_2 into it, we obtain

$$\begin{aligned}
&2\alpha_k(\mathbb{E}(H(\mathbf{x}_{k+1})) - H^*) + \mathbb{E}(\|\mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_{k+1})\|^2) \\
&\quad \leq (1+\phi_k)\mathbb{E}(\|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^3) + \Theta(L_f^2\alpha_k^2/\phi_k)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^2),
\end{aligned}$$

which completes the proof. ■

Then, we prove the Theorem 2 based on the previous lemma.

Proof to Theorem 2 Apply the optimally strong convexity in (7) to Lemma 8, yielding

$$\begin{aligned}
&(1+2\lambda\alpha_k)\mathbb{E}(\|\mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_{k+1})\|^2) \\
&\quad \leq (1+\phi_k)\mathbb{E}(\|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^3) + \Theta(L_f^2\alpha_k^2/\phi_k)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^2).
\end{aligned}$$

It follows by dividing $1 + 2\lambda\alpha_k$ on both sides

$$\begin{aligned}
&\mathbb{E}(\|\mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_{k+1})\|^2) \\
&\quad \leq \frac{1+\phi_k}{1+2\lambda\alpha_k}\mathbb{E}(\|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^3) + \Theta(L_f^2\alpha_k^2/\phi_k)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^2).
\end{aligned}$$

Choosing $\phi_k = \lambda\alpha_k - 2\lambda^2\alpha_k^2 \geq 0.5\lambda\alpha_k$ yields

$$\begin{aligned}
&\mathbb{E}(\|\mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_{k+1})\|^2) \\
&\leq (1-\lambda\alpha_k)\mathbb{E}(\|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^2) + \frac{\Theta(L_f^2\alpha_k)}{\lambda}\mathbb{E}(\|g(\mathbf{x}_k) - \mathbf{y}_k\|^2) \\
&\leq (1-\lambda\alpha_k)\mathbb{E}(\|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2) + \Theta(k^{-2a}) + \Theta(L_g L_f^2 k^{-5a+4b} + L_f^2 k^{-a-b}).
\end{aligned}$$

Apply Lemma 6 and substitute the subscript k by K to obtain the first claim in (8)

$$\mathbb{E}(\|\mathbf{x}_K - \mathcal{P}_{X^*}(\mathbf{x}_K)\|^2) \leq O\left(K^{-a} + L_f^2 L_g K^{-4a+4b} + L_f^2 K^{-b}\right).$$

The followed specification of a and b can easily verified. ■

C. Proof to Theorem 3

In what follows, we provide the detailed proof to Theorem 3.

Proof to Theorem 3 Let $P_k = 2\alpha_k(\mathbb{E}(H(\mathbf{x}_{k+1})) - H^*) + \mathbb{E}(\|\mathbf{x}_{k+1} - \mathcal{P}_{X^*}(\mathbf{x}_{k+1})\|^2)$, which is the left hand side of inequality (24). Summing P_k up from $k = 1$ to $k = K$ and taking $\phi_k = \Theta(k^{-c})$ yields

$$\begin{aligned}
&2\sum_{k=1}^K \alpha_k(\mathbb{E}(H(\mathbf{x}_{k+1})) - H^*) + \mathbb{E}(\|\mathbf{x}_{K+1} - \mathcal{P}_{X^*}(\mathbf{x}_{K+1})\|^2) \\
&\leq (1+\phi_1)\mathbb{E}(\|\mathbf{x}_1 - \mathcal{P}_{X^*}(\mathbf{x}_1)\|^2) + \sum_{k=2}^K \phi_k\mathbb{E}(\|\mathbf{x}_k - \mathcal{P}_{X^*}(\mathbf{x}_k)\|^2) \\
&\quad + \sum_{k=2}^K \Theta(L_f^2\alpha_k^2/\phi_k)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2) + \Theta(\alpha_k^2) \\
&\leq \Theta(1) + \Theta\left(\sum_{k=2}^K \phi_k\right) + \Theta(K^{1-2a}) + \sum_{k=2}^K \Theta(L_f^2\alpha_k^2/\phi_k)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2)
\end{aligned}$$

$$\leq \quad \Theta(1) + \Theta\left(K^{1-c}\mathbf{I}_{c=1}^{\log K}\right) + O(K^{1-2a}) + \sum_{k=2}^{K} \Theta(L_f^2 \alpha_k^2/\phi_k)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2). \quad (27)$$

Note that the second inequality holds by the condition that the feasible set X is bounded, and thus $\|x_k - \mathcal{P}_{X^*}(x_k)\|^2$ is bounded.

We continue to bound the last term on the right hand side of (27):

$$\begin{aligned} & \sum_{k=2}^{K} \Theta(L_f^2 \alpha_k^2/\phi_k)\mathbb{E}(\|\mathbf{y}_k - g(\mathbf{x}_k)\|^2) \\ \text{(Lemma 7)} \quad \leq \quad & \sum_{k=2}^{K} L_f^2 L_g \Theta(k^{-6a+4b-c}) + L_f^2 \Theta(k^{-2a-b-c}) \\ \leq \quad & L_f^2 L_g \Theta(K^{1-6a+4b-c}\mathbf{I}_{6a-4b-c=1}^{\log K}) + L_f^2 \Theta(K^{1-2a-b-c}\mathbf{I}_{2a+b-c=1}^{\log K}). \quad (28) \end{aligned}$$

Plugging (28) into (27) and dividing $2\sum_{k=1}^{K}\alpha_k$ on both sides:

$$\begin{aligned} & \frac{2\sum_{k=1}^{K}\alpha_k H(\mathbf{x}_{k+1}) - H^*}{2\sum_{k=1}^{K}\alpha_k} \\ \leq \quad & \frac{\Theta(1) + \Theta\left(K^{1-c}\mathbf{I}_{c=1}^{\log K}\right) + \Theta(K^{1-2a})}{2\sum_{k=1}^{K}\alpha_k} \\ & + \frac{L_f^2 L_g \Theta(K^{1-6a+4b+c}\mathbf{I}_{6a-4b-c=1}^{\log K}) + L_f^2\Theta(K^{1-2a-b+c}\mathbf{I}_{2a-b+c=1}^{\log K})}{2\sum_{k=1}^{K}\alpha_k} \\ \leq \quad & O(K^{a-1}) + O(K^{a-c}\mathbf{I}_{c=1}^{\log K}) + O(K^{-a}) \\ & + O(L_f^2 L_g K^{-5a+4b+c}\mathbf{I}_{6a-4b-c=1}^{\log K}) + O(L_f^2 K^{-a-b+c}\mathbf{I}_{2a+b-c=1}^{\log K}) \\ = \quad & O\left(K^{a-1} + K^{a-c}\mathbf{I}_{c=1}^{\log K} + K^{-a} + L_f^2 L_g K^{-5a+4b+c}\mathbf{I}_{6a-4b-c=1}^{\log K} + L_f^2 K^{-a-b+c}\mathbf{I}_{2a+b-c=1}^{\log K}\right). \end{aligned}$$

where the last inequality uses the fact $\sum_{k=1}^{K}\alpha_k \geq \Theta(K^{1-a})$. To finish the proof of (11), we use Jensen's inequality

$$\frac{2\sum_{k=1}^{K}\alpha_k H(\mathbf{x}_{k+1})}{2\sum_{k=1}^{K}\alpha_k} \geq H(\bar{\mathbf{x}}_K).$$

■

References

L Baird. Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning (ICML)*, pages 30–37, 1995.

B. Dai, N. He, Y. Pan, B. Boots, and L. Song. Learning from conditional distributions via dual embeddings. In *Artificial Intelligence and Statistics (AISTATS)*, pages 1458–1467, 2017.

C. Dann, G. Neumann, and J. Peters. Policy evaluation with temporal differences: A survey and comparison. *The Journal of Machine Learning Research*, 15(1):809–883, 2014.

D. Dentcheva, S. Penev, and A. Ruszczyński. Statistical estimation of composite risk functionals and risk optimization problems. *Annals of the Institute of Statistical Mathematics*, pages 1–24, 2016.

Y. M. Ermoliev. *Methods of Stochastic Programming*. Monographs in Optimization and OR, Nauka, Moscow, 1976.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (KDD)*, pages 795–811. Springer, 2016.

B. Liu, J. Liu, M. Ghavamzadeh, S. Mahadevan, and M. Petrik. Finite-sample analysis of proximal gradient td algorithms. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

J. Liu and S. J. Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.

I Necoara, Yu Nesterov, and F Glineur. Linear convergence of first order methods for non-strongly convex optimization. *arXiv preprint arXiv:1504.06298*, 2015.

A. Nedić and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12:109–138, 2001.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19:1574–1609, 2009.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT press, 1998.

M. Wang and D. P. Bertsekas. Stochastic first-order methods with random constraint projection. *SIAM Journal on Optimization*, 26(1):681–717, 2016.

M. Wang and J. Liu. A stochastic compositional subgradient method using Markov samples. *Winter Simulation Conference*, 2016.

M. Wang, Y. Chen, J. Liu, and Y. Gu. Random multi-constraint projection: Stochastic gradient methods for convex optimization with many constraints. *arXiv preprint arXiv:1511.03760*, 2015.

M. Wang, J. Liu, and E. X. Fang. Accelerating stochastic composition optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1714–1722, 2016.

M. Wang, E. X. Fang, and H. Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161 (1-2):419–449, 2017.

A. White and M. White. Investigating practical linear temporal difference learning. In *International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, pages 494–502, 2016.

Journal of Machine Learning Research 18 (2017) 1-37 Submitted 9/16; Revised 7/17; Published 10/17

Distributed Bayesian Learning with Stochastic Natural Gradient Expectation Propagation and the Posterior Server

Leonard Hasenclever hasenclever@stats.ox.ac.uk
Department of Statistics, University of Oxford, United Kingdom

Stefan Webb stefanw@robots.ox.ac.uk
Department of Engineering Sciences, University of Oxford, United Kingdom

Thibaut Lienart lienart@stats.ox.ac.uk
Department of Statistics, University of Oxford, United Kingdom

Sebastian Vollmer vollmer@stats.ox.ac.uk
Department of Statistics, University of Oxford, United Kingdom

Balaji Lakshminarayanan balajiln@google.com
DeepMind

Charles Blundell cblundell@google.com
DeepMind

Yee Whye Teh y.w.teh@stats.ox.ac.uk
Department of Statistics, University of Oxford, United Kingdom

Editor: Manfred Opper

Abstract

This paper makes two contributions to Bayesian machine learning algorithms. Firstly, we propose stochastic natural gradient expectation propagation (SNEP), a novel alternative to expectation propagation (EP), a popular variational inference algorithm. SNEP is a black box variational algorithm, in that it does not require any simplifying assumptions on the distribution of interest, beyond the existence of some Monte Carlo sampler for estimating the moments of the EP tilted distributions. Further, as opposed to EP which has no guarantee of convergence, SNEP can be shown to be convergent, even when using Monte Carlo moment estimates. Secondly, we propose a novel architecture for distributed Bayesian learning which we call the posterior server. The posterior server allows scalable and robust Bayesian learning in cases where a data set is stored in a distributed manner across a cluster, with each compute node containing a disjoint subset of data. An independent Monte Carlo sampler is run on each compute node, with direct access only to the local data subset, but which targets an approximation to the global posterior distribution given all data across the whole cluster. This is achieved by using a distributed asynchronous implementation of SNEP to pass messages across the cluster. We demonstrate SNEP and the posterior server on distributed Bayesian learning of logistic regression and neural networks.

Keywords: Distributed Learning, Large Scale Learning, Deep Learning, Bayesian Learning, Variational Inference, Expectation Propagation, Stochastic Approximation, Natural Gradient, Markov chain Monte Carlo, Parameter Server, Posterior Server.

©2017 Leonard Hasenclever, Stefan Webb, Thibaut Lienart, Sebastian Vollmer, Balaji Lakshminarayanan, Charles Blundell and Yee Whye Teh.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-478.html

1. Introduction

Algorithms and systems for enabling machine learning from large scale data sets are becoming increasingly important in the era of Big Data. This has driven many developments, including various forms of stochastic gradient descent, parallel and distributed learning systems, use of GPUs, sketching, random Fourier features, divide-and-conquer methods, as well as various approximation schemes. These large scale machine learning systems have in turn driven significant advances across many data-oriented sciences and technologies, ranging from the biological sciences, neuroscience, social sciences, signal processing, speech processing, natural language processing, computer vision etc.

In this paper we will consider methods for large scale *Bayesian* machine learning. As opposed to the more common empirical risk minimisation or maximum likelihood approaches, where learning is phrased as finding a set of parameters optimal with respect to a data set and to a loss or likelihood function, Bayesian machine learning rests upon probabilistic models which capture the dependencies among all observed and unobserved variables, and where learning is phrased as computing the posterior distribution over unobserved variables (including both latent variables and model parameters) given the observed data.

The Bayesian framework can more fully capture the uncertainty in learnt parameters and prevent overfitting. In principle this allows the use of more complex and larger scale models. However, Bayesian approaches are generally more computationally intensive than optimisation-based ones, and have to date not led to methods which are as scalable.

For complex models, exact computation of the posterior distribution is intractable and approximate schemes such as variational inference (VI) (Wainwright and Jordan, 2008), Markov chain Monte Carlo (MCMC) (Gilks et al., 1996) and sequential Monte Carlo (Doucet et al., 2001) are needed. Scalable methods in both traditions include: stochastic variational inference (Hoffman et al., 2013, Mnih and Gregor, 2014, Rezende et al., 2014) which apply minibatch stochastic gradient descent (Robbins and Monro, 1951) to optimise the variational objective function, stochastic gradient MCMC (Welling and Teh, 2011, Patterson and Teh, 2013, Ding et al., 2014, Teh et al., 2015, Leimkuhler, B. and Shang, X., 2016, Ma et al., 2015, Li et al., 2016) which uses minibatch stochastic gradients within MCMC, austerity MCMC (Korattikara et al., 2014, Bardenet et al., 2014) which uses data subsampling to reduce computational cost of Metropolis-Hastings acceptance steps, and embarrassingly parallel MCMC (Huang and Gelman, 2005, Scott et al., 2013, Wang and Dunson, 2013, Neiswanger et al., 2014) which distribute data across a cluster, runs independent MCMC samplers on each worker and combines samples across the cluster only at the end to reduce network communication costs. In addition, standard learning schemes have also been successfully deployed in large scale settings. A successful example that is particularly relevant to our work is expectation propagation (EP), introduced by (Minka, 2001) and (Opper and Winther, 2000) which iteratively constructs approximations to a factorized distribution. EP is at the heart of the TrueSkill XBox player rating and matching system (Herbrich et al., 2007).

Our work builds upon prior work on using EP for performing distributed Bayesian learning (Xu et al., 2014, Gelman et al., 2014). In this framework, a data set is partitioned into disjoint subsets with each subset stored on a worker node in a cluster. Learning is performed at each worker based on the data subset there using MCMC sampling. As

opposed to embarrassingly parallel MCMC methods which only communicate the samples to the master at the end of learning, EP is used to communicate messages (infrequently) across the cluster. These messages coordinate the samplers such that the target distributions (which coincidentally are the tilted distributions in EP) of all samplers on all workers share certain moments, e.g. means and variances, hence the name sampling via moment sharing (SMS) coined by (Xu et al., 2014). At convergence, it can be shown that the target distributions of the samplers also share moments with the EP approximation to the global posterior distribution given all data, hence the target distributions on the workers can themselves be treated as approximations to the global posterior.

While SMS works well on simpler models like Bayesian logistic regression, we have found that it did not work for more complex, high-dimensional, and non-convex models like Bayesian deep neural networks. This is due to the non-convergence of EP, particularly as the moments of the tilted distributions needed by EP are estimated using MCMC sampling, with estimation noise that further compounds the well-known lack of convergence guarantees for EP, and the fact that extremely long MCMC runs are needed for the samplers to equilibrate due to the complex posterior distribution in these models.

Our first contribution is thus the development of stochastic natural-gradient EP (SNEP), an alternative algorithm to power EP (a generalisation of EP) (Minka, 2004) which optimises the same variational objective function. SNEP is a double-loop algorithm with convergence guarantees. The inner loop is a stochastic natural-gradient descent algorithm which tolerates estimation noise, so that SNEP is convergent even with moments estimated using MCMC samplers. Our derivation of SNEP improves upon the derivation of the convergent EP algorithm of (Heskes and Zoeter, 2002) in that ours works for a more general class of models, we make explicit the underlying variational objective function that is being optimised, and ours uses a natural-gradient descent algorithm (Amari and Nagaoka, 2001) more tolerant of Monte Carlo noise.

Building upon the development of SNEP, our second contribution is a distributed Bayesian learning architecture which we call the posterior server. In analogy to the parameter server (Ahmed et al., 2012) which maintains and serves the parameter vector to a cluster of workers, the posterior server maintains and serves (an approximation to) the posterior distribution. Figure 1 gives a schematic for the steps involved. Each worker has a subset of data, from which we get a likelihood function. It also maintains a tractable approximation of the likelihood and a cavity distribution which is effectively a conditional distribution over the parameters given all data on other workers. An MCMC sampler targets the normalised product of the cavity distribution and the (true) likelihood, and forms a stochastic estimate of the required moments, which is in turn used to update the likelihood approximation using stochastic natural-gradient descent. Each worker communicates with the posterior server asynchronously and in a non-blocking manner, sending the current likelihood approximation and receiving the new cavity distribution. This communication protocol makes more efficient use of computational resources on workers than SMS, which requires either synchronous or blocking asynchronous protocols.

Note that our set-up of the distributed learning problem is that each worker has access to a subset of data, and no worker has access to all data. This situation might occur in situations other than large scale learning. For example, when working with sensitive patient data which cannot be shared directly, we might still want to be able to make use

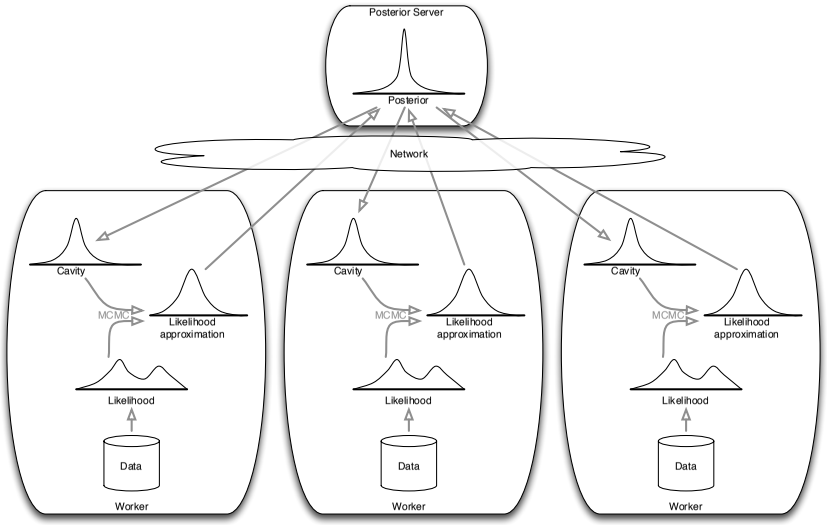


Figure 1: The posterior server.

of all available data across multiple sites to improve inference. Typically known as divide-and-conquer or consensus inference (Zhang et al., 2015b, Zhao et al., 2016, Kleiner et al., 2014, Battey et al., 2015), it is also well-known that this situation is harder than typical distributed learning settings where it is assumed that all data is accessible on all workers, which are settings assumed in DistBelief networks (Dean et al., 2012) and elastic-averaging SGD (Zhang et al., 2015a), two state-of-the-art distributed learning algorithms for neural networks.

In the following, Section 2 describes our set up of distributed Bayesian learning and reviews the necessary background on exponential families and convex duality. Section 3 formulates EP and power EP within the framework of variational inference. Our contributions are contained in Section 4 deriving SNEP and Section 5 describing the posterior server architecture. Section B describes additional techniques we used to make the method work on more complex problems like neural networks. We demonstrate the approach Bayesian logistic regression and Bayesian neural networks in Section 6. Section 7 concludes with a summary and discussion of future work.

2. Problem Set-up and Background

In this section we set-up the problem of distributed Bayesian learning, using the framework of variational inference in exponential families. For an excellent introduction to exponential families and variational inference we refer the interested reader to (Wainwright and Jordan, 2008). Section 2.1 reviews exponential families and convex duality while introducing nota-

tion used throughout the paper. Readers familiar with these concepts can skim ahead to section 2.2.

2.1 Exponential Families and Convex Duality

.

Consider an exponential family described by a d-dimensional sufficient statistics function $s(x)$. A member p_θ of this exponential family is parameterized by a natural parameter $\theta \in \mathbb{R}^d$, and has density (with respect to some base measure, say Lebesgue),

$$p_\theta(x) = \exp\left(\theta^\top s(x) - A(\theta)\right),$$
$$A(\theta) = \log \int \exp\left(\theta^\top s(x)\right) dx.$$

The log partition function $A(\theta)$ is convex and finite on the natural domain of the exponential family,

$$\Theta := \{\theta : A(\theta) < \infty\} \subset \mathbb{R}^d,$$

which is a convex subset of $\mathbb{R}^d$.

Associated with any distribution p and the d-dimensional sufficient statistics function $s(x)$ is a mean parameter,

$$\mu := \mathbb{E}_p[s(x)],$$

where $\mathbb{E}_p$ denotes the expectation operator with respect to p. The set of valid mean parameters $\mathcal{M}$ is a closed convex set, which we refer to as the mean domain,

$$\mathcal{M} = \{\mu : \exists \text{ distribution } p \text{ with } \mu = \mathbb{E}_p[s(x)]\} \subset \mathbb{R}^d$$

Given a natural parameter $\theta \in \Theta$, the exponential family member p_θ is also associated with a mean parameter $\mu = \mathbb{E}_\theta[s(x)]$ (where $\mathbb{E}_\theta$ denotes the expectation with respect to p_θ), which we can write as a function of the natural parameters, $\mu(\theta)$. If the exponential family is minimal[1], then the mapping $\theta \mapsto \nabla A(\theta)$ is one-to-one and onto the interior of $\mathcal{M}$, and maps θ to the mean parameter, $\mu(\theta) = \nabla A(\theta)$.

We will assume that our exponential family of interest is minimal.

The convex conjugate of $A(\theta)$ is,

$$A^*(\mu) := \sup_{\theta \in \Theta} \theta^\top \mu - A(\theta).$$

Evaluated at the mean parameter $\mu(\theta)$, the conjugate is the negative entropy of p_θ,

$$A^*(\mu(\theta)) = \mathbb{E}_\theta[\log p_\theta(x)].$$

1. The exponential family is minimal if the d 1-dimensional functions making up the sufficient statistics function $s(x)$ are linearly independent, i.e. $\theta^\top s(x) = 0$ for all x implies $\theta = 0$.

Conversely, we have

$$A(\theta) = \sup_{\mu \in \mathcal{M}} \theta^\top \mu - A^*(\mu)$$

and that the natural parameter associated with μ is $\theta = \theta(\mu) = \nabla A^*(\mu)$. In the following we will make extensive use of the duality between A and A^* and between θ and μ described above.

It is useful to write down formulae for the KL divergence between two exponential family distributions, parameterized by natural and mean parameter pairs θ, μ and θ', μ' respectively:

$$\begin{aligned}
\mathrm{KL}(p_\theta \| p_{\theta'}) &= \mathbb{E}_\theta[\log p_\theta(x) - \log p_{\theta'}(x)] \\
&= \mathbb{E}_\theta[\theta^\top s(x) - A(\theta) - (\theta')^\top s(x) + A(\theta')] \\
&= \mu^\top(\theta - \theta') - A(\theta) + A(\theta') \\
&= A^*(\mu) + A(\theta') - \mu^\top \theta' \\
&= A^*(\mu) - A^*(\mu') + (\mu' - \mu)^\top \theta'. \qquad (1)
\end{aligned}$$

We will write $\mathrm{KL}(\theta \| \theta'), \mathrm{KL}(\mu \| \theta')$ etc. to refer to the same KL divergence between the same two distributions.

As an example, for a diagonal covariance Gaussian of dimension $d/2$, we have

$$\begin{aligned}
p(x) &= \prod_{j=1}^{d/2} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{1}{2\sigma_j^2}(x_j - u_j)^2\right) \\
&= \exp\left(\sum_{j=1}^{d/2} (u_j \sigma_j^{-2})(x_j) + (-\sigma_j^{-2})(\tfrac{1}{2}x_j^2) - \tfrac{1}{2}(u_j^2 \sigma_j^{-2} + \log(2\pi\sigma_j^2))\right)
\end{aligned}$$

So the sufficient statistics are x_j and $\frac{1}{2}x_j^2$, mean parameters are $\mu_{j1} = u_j$ and $\mu_{j2} = \frac{1}{2}(u_j^2 + \sigma_j^2)$, natural parameters are $\theta_{j1} = u_j \sigma_j^{-2}$ and $\theta_{j2} = -\sigma_j^{-2}$, and

$$A(\theta) = \sum_{j=1}^{d/2} \frac{1}{2}(u_j^2 \sigma_j^{-2} + \log(2\pi\sigma_j^2))$$
$$A^*(\mu) = \sum_{j=1}^{d/2} \frac{-1}{2}(1 + \log(2\pi\sigma_j^2))$$

The conversions between natural and mean parameters are:

$$\begin{aligned}
&u_j = \mu_{j1} = -\theta_{j1}\theta_{j2}^{-1} && \theta_{j1} = \mu_{j1}(2\mu_{j2} - \mu_{j1}^2)^{-1} && \mu_{j1} = -\theta_{j1}\theta_{j2}^{-1} \\
&\sigma_j^2 = 2(\mu_{j2} - \mu_{j1}^2) = -\theta_{j2}^{-1} && \theta_{j2} = -(2\mu_{j2} - \mu_{j1}^2)^{-1} && \mu_{j2} = \frac{1}{2}(\theta_{j1}^2\theta_{j2}^{-2} - \theta_{j2}^{-1})
\end{aligned}$$

We will use a diagonal covariance Gaussian as our exponential family in most of our experiments, due to the high-dimensionality of the models used.

2.2 Distributed Bayesian Learning

We assume that our model is parameterized by a high-dimensional parameter vector x. Let the prior distribution $p_0(x)$ be a member of a tractable and minimal exponential family distribution, with natural parameter $\theta_0 \in \Theta \subset \mathbb{R}^d$, sufficient statistics function $s(x)$ and log partition function $A(\theta_0)$. Specifically, we will take $p_0(x)$ to be a diagonal covariance Gaussian. We refer to this exponential family as the *base exponential family*.

We assume that our training data set is spread across a cluster of n compute nodes or workers, with log likelihood $\ell_i(x)$ on compute node $i = 1, \ldots, n$. For example, if we let $\{S_i\}$ be a partition of the data indices, each compute node i could store the corresponding subset of the data $D_i = \{y_c\}_{c \in S_i}$, so that the log likelihood $\ell_i(x)$ is a sum over terms, each corresponding to the log density of one data point stored on node i,

$$\ell_i(x) = \sum_{c \in S_i} \log p(y_c \mid x).$$

Covariates and input vectors can be easily incorporated in the above. The target posterior distribution is then,

$$\tilde{p}(x) := p(x \mid \{D_i\}_{i=1}^n) \propto p_0(x) \exp\left(\sum_{i=1}^n \ell_i(x)\right). \tag{2}$$

Using neural networks as an example, x corresponds to all learnable weights and biases in a network, $\log p(y_c \mid x)$ gives the probability of the class of data item c given the corresponding input vector, and the Gaussian prior corresponds to weight decay.

The learning task is then to compute the posterior distribution. For example, we may want to estimate the posterior mean or variance of the model parameters x, or we may want to draw samples distributed according to $\tilde{p}(x)$, using these to predict on test data by averaging the predictive densities over the samples as a Monte Carlo estimate of the marginal predictive density. In the rest of the paper we will aim to obtain these efficiently but approximately.

3. Variational Inference in an Extended Exponential Family

In general, the likelihood functions are intractable and approximations are necessary. In this paper we will formulate the learning task as variational inference in an extended exponential family. In particular, we will consider a class of variational methods known as *power expectation propagation* (power EP) (Minka, 2004).

3.1 Extended Exponential Family

To start with, we may trivially formulate the target posterior distribution as an *extended exponential family distribution* with sufficient statistics $\tilde{s}(x) := [s(x), \ell_1(x), \ldots, \ell_n(x)]$ and natural parameters $\tilde{\theta} := [\theta_0, \mathbf{1}_n]$ where $\mathbf{1}_n$ is a vector of 1's of length n:

$$\tilde{p}(x) = \exp\left(\tilde{\theta}^\top \tilde{s}(x) - \tilde{A}(\tilde{\theta})\right).$$

The extended log partition function $\tilde{A}(\tilde{\theta})$ is (up to a constant) the log marginal probability of the data,

$$\begin{aligned}\tilde{A}(\tilde{\theta}) = \log \int \exp\left(\tilde{\theta}^\top \tilde{s}(x)\right) dx &= \log \int \exp\left(\theta_0^\top s(x) + \sum_{i=1}^n \ell_i(x)\right) dx \\ &= \log \mathbb{E}_{\theta_0}\left[\exp\left(\sum_{i=1}^n \ell_i(x)\right)\right] + A(\theta_0) \\ &= \log p(\{D_i\}_{i=1}^n) + A(\theta_0).\end{aligned}$$

Denoting the convex conjugate by $\tilde{A}^*(\tilde{\mu})$ and the extended mean domain by $\tilde{\mathcal{M}} \subset \mathbb{R}^{d+n}$, the problem of posterior computation can be expressed as the following concave variational maximization problem:

$$\max_{\tilde{\mu} \in \tilde{\mathcal{M}}} \tilde{\theta}^\top \tilde{\mu} - \tilde{A}^*(\tilde{\mu}). \tag{3}$$

For example, if the prior exponential family is a diagonal covariance Gaussian, then the optimal mean parameter is $\tilde{\mu}^* := [\mu^*, \nu_1^*, \ldots \nu_n^*]$, where $\mu^* := \mathbb{E}_{\tilde{\theta}}[s(x)] \in \mathbb{R}^d$ corresponds to the posterior means and variances of the model parameters x, while $\nu_i^* := \mathbb{E}_{\tilde{\theta}}[\ell_i(x)]$ is the posterior expectation of the ith log likelihood $\ell_i(x)$. Hence the extended mean parameters capture the important aspects of the posterior distribution.

3.2 Power Expectation Propagation

In this section we derive another class of variational approximations corresponding to a generalization of expectation propagation (EP) (Minka, 2001) called power EP (Minka, 2004). The derivation is a straightforward generalisation of the variational formulation of Wainwright and Jordan (2008) from EP to power EP.

For each worker node i let $\beta_i > 0$ be a given positive real number. Typically, we take $\beta_i = 1$ which corresponds to standard EP, while $\beta_i \to \infty$ corresponds to variational Bayes (Wiegerinck and Heskes, 2003, Minka, 2004). In the formulation of Wainwright and Jordan (2008), EP involves two approximations; both associated with simpler exponential families, which we refer to as *locally extended exponential families* (or sometimes *local exponential families*). For each i, let the ith locally extended exponential family be associated with the sufficient statistics function $s_i(x) := [s(x), \ell_i(x)]$. Let Θ_i, $\mathcal{M}_i$, A_i, A_i^* be the associated (local) natural domain, mean domain, log partition function and negative entropy respectively. A distribution in this locally extended exponential family with natural parameter $[\theta_i, \eta_i] \in \Theta_i$ has the form

$$p_{\theta_i, \eta_i}(x) = \exp\left(\theta_i^\top s(x) + \eta_i \ell_i(x) - A_i(\theta, \eta_i)\right), \tag{4}$$

which is a distribution obtained by tilting a tractable distribution with density proportional to $\exp(\theta_i^\top s(x))$ by a single intractable likelihood $\exp(\ell_i(x))$ raised to the power of η_i. This family can be thought of as treating the ith likelihood term exactly, while approximating all other likelihood terms by projecting them onto the tractable base exponential family,

the hope being that this family is still tractable while being closer to the true posterior distribution $\tilde{p}(x)$.

For the first approximation, the extended negative entropy $\tilde{A}^*$ is approximated using a tree-like approximation constructed using only the locally extended negative entropies,

$$\tilde{A}^*([\mu, \nu_1, \ldots, \nu_n]) \approx A^*(\mu) + \sum_{i=1}^n \beta_i (A_i^*(\mu, \nu_i) - A^*(\mu)).$$

This approximation is related to the Bethe entropy of loopy belief propagation (Yedidia et al., 2001) and fractional belief propagation (Wiegerinck and Heskes, 2003). Secondly, the extended mean domain $\tilde{\mathcal{M}}$ is approximated by a local mean domain,

$$\mathcal{L} := \{[\mu, \nu_1, \ldots, \nu_n] : [\mu, \nu_i] \in \mathcal{M}_i \text{ for all } i = 1, \ldots, n\}. \tag{5}$$

The local mean domain is an outer bound, $\mathcal{L} \supset \tilde{\mathcal{M}}$. Intuitively, the constraints described by $\tilde{\mathcal{M}}$ are replaced by the weaker constraints described by the local mean domains. We assume that working with the local exponential families in these ways will be more tractable than working with the full extended exponential family.

Let $\mu_0 \in \mathcal{M}$ be a mean parameter in the base exponential family. For the ith local exponential family, we denote a mean parameter by $[\mu_i, \nu_i] \in \mathcal{M}_i$, and require the marginalization constraint $\mu_i = \mu_0$. The power EP variational problem, which is not in general concave, is,

$$\begin{aligned} \max_{\mu_0, [\mu_i, \nu_i]_{i=1}^n} \quad & \theta_0^\top \mu_0 + \sum_{i=1}^n 1 \cdot \nu_i - A^*(\mu_0) - \sum_{i=1}^n \beta_i (A_i^*(\mu_i, \nu_i) - A^*(\mu_i)) \\ \text{subject to} \quad & \mu_0 \in \mathcal{M} \\ & [\mu_i, \nu_i] \in \mathcal{M}_i \quad \text{for } i = 1, \ldots, n \\ & \mu_0 = \mu_i \quad \text{for } i = 1, \ldots, n \end{aligned} \tag{6}$$

Note in particular that the entropy and mean domain in the original variational problem (3) have been replaced by their respective approximations.

3.3 Deriving EP and Power EP Updates

In this section, for completeness' sake, we derive the updates for EP and power EP as fixed-point equations that solve the variational problem. Readers familiar with this derivation can skip ahead to section 3.4.

First we introduce Lagrange multipliers λ_i for the equality constraints $\mu_0 = \mu_i$, so that the above is equivalent to,

$$\begin{aligned} \max_{\mu_0, [\mu_i, \nu_i]_{i=1}^n} \min_{[\lambda_i]_{i=1}^n} \quad & \underbrace{\theta_0^\top \mu_0 - A^*(\mu_0) + \sum_{i=1}^n \Big(\nu_i - \lambda_i^\top (\mu_i - \mu_0) - \beta_i (A_i^*(\mu_i, \nu_i) - A^*(\mu_i))\Big)}_{=: L(\mu_0, [\mu_i, \nu_i, \lambda_i]_{i=1}^n)} \\ \text{subject to} \quad & \mu_0 \in \mathcal{M} \\ & [\mu_i, \nu_i] \in \mathcal{M}_i \quad \text{for } i = 1, \ldots, n \end{aligned} \tag{7}$$

where the domain of λ_i is $\mathbb{R}^d$. Let the Lagrangian above be denoted by $L(\mu_0, [\mu_i, \nu_i, \lambda_i]_{i=1}^n)$. The Karush-Kuhn-Tucker (KKT) conditions of the above variational problem have to be satisfied at an optimum, and simply involve setting the derivatives with respect to $\mu_0, \mu_i, \nu_i, \lambda_i$ to zero:

$$\frac{dL}{d\lambda_i} = 0 : \qquad \mu_i = \mu_0 \tag{8a}$$

$$\frac{dL}{d\mu_0} = 0 : \qquad \theta_0 - \nabla A^*(\mu_0) + \sum_{j=1}^n \lambda_j = 0$$

$$\theta_0 + \sum_{j=1}^n \lambda_j = \nabla A^*(\mu_0) \tag{8b}$$

$$\frac{dL}{d\mu_i} = 0 : \qquad -\lambda_i - \beta_i \nabla_{\mu_i} A_i^*(\mu_i, \nu_i) + \beta_i \nabla A^*(\mu_i) = 0$$

$$\theta_0 + \sum_{j=1}^n \lambda_j - \beta_i^{-1} \lambda_i = \nabla_{\mu_i} A_i^*(\mu_i, \nu_i) \tag{8c}$$

$$\frac{dL}{d\nu_i} = 0 : \qquad \beta_i^{-1} = \nabla_{\nu_i} A_i^*(\mu_i, \nu_i) \tag{8d}$$

Equation (8b) shows that an optimal μ_0 has to be the mean parameter corresponding to a (base) exponential family distribution with natural parameter $\theta_0 + \sum_{j=1}^n \lambda_j$. Specifically, the posterior distribution can be approximated as,

$$\tilde{p}(x) \propto p_0(x) \exp\left(\sum_{j=1}^n \ell_j(x)\right) \approx \exp\left(\left(\theta_0 + \sum_{j=1}^n \lambda_j\right)^\top s(x)\right). \tag{9}$$

In other words, we can interpret λ_j as the natural parameter of an exponential family approximation to the likelihood factor $\exp(\ell_j(x))$.

Further, from (8c) and (8d) above, we see that the optimal $[\mu_i, \nu_i]$ is the mean parameter associated with the local posterior distribution,

$$p_i(x) \propto \exp\left(\left(\theta_0 + \sum_{j=1}^n \lambda_j - \beta_i^{-1} \lambda_i\right)^\top s(x) + \beta_i^{-1} \ell_i(x)\right). \tag{10}$$

For standard EP, where $\beta_i = 1$, we get,

$$p_i(x) \propto \exp\left(\left(\theta_0 + \sum_{j \neq i} \lambda_j\right)^\top s(x) + \ell_i(x)\right).$$

The above local posterior distribution is known as the tilted distribution in EP, with the term $(\theta_0 + \sum_{j=1}^n \lambda_j - \beta_i^{-1} \lambda_i)^\top s(x)$ corresponding to the cavity distribution with (the β_i^{-1}th power of) the exponential family approximation to the ith likelihood removed, and replaced

by (the β_i^{-1}th power of) the likelihood factor itself. Each step of EP involves first computing $[\mu_i, \nu_i]$ as the mean parameter of the tilted distribution, then updating λ_i, using (8a) and (8b):

$$\lambda_i^{\text{new}} = \nabla A^*(\mu_i) - \theta_0 - \sum_{j \neq i} \lambda_j, \tag{11}$$

where we recall that $\nabla A^*(\mu_i)$ computes the natural parameter corresponding to the mean parameter μ_i in the base exponential family. The EP update ensures that the expectation of the sufficient statistics function under the tilted distribution $p_i(x)$ and under its exponential family approximation (with natural parameters $\theta_0 + \sum_{j=1}^n \lambda_j - \beta_i^{-1}\lambda_i + \beta_i^{-1}\lambda_i^{\text{new}}$) match. At convergence, this ensures that the expectations under all the tilted distributions and under the approximated posterior distribution (9) match.

Note that the (power) EP updates are derived as fixed point equations and have no guarantees of convergence. In practice, damped updates can be used to avoid oscillations without affecting the fixed points:

$$\lambda_i^{\text{new}} = \alpha\lambda_i + (1-\alpha)\left(\nabla A^*(\mu_i) - \theta_0 - \sum_{j \neq i} \lambda_j\right). \tag{12}$$

where $\alpha \in [0, 1)$ is a damping factor. Another potential issue is that λ_i^{new} as computed may not lie in the natural domain Θ, in which case higher damping factors ensuring that it does can be used, or the update can be discarded.

3.4 Computing Mean Parameters

Assuming that the base exponential family is tractable, the above steps of EP involve additions, subtractions, and conversions between mean and natural parameters so are easy to compute. The only difficulty is in the computation of the mean parameters (expectations of the sufficient statistics function) of the tilted distributions, which involve the log likelihoods $\ell_i(x)$. In typical applications of EP to graphical models, these are either obtained analytically or using numerical quadrature.

In more complex and general scenarios, both analytic or quadrature-based methods are ruled out. A successful and common class of methods for calculating expectations under otherwise intractable distributions is Monte Carlo. A number of papers have proposed such an approach, including Barthelmé and Chopin (2011), Heess et al. (2013), Gelman et al. (2014) using importance sampling and Xu et al. (2014), Gelman et al. (2014) using Markov chain Monte Carlo (MCMC). In addition, Heess et al. (2013), Eslami et al. (2014), Jitkrittum et al. (2015) use learning techniques to speed-up the process by directly predicting the natural parameters given properties of the tilted distribution.

We have explored the sampling via moment sharing (SMS) algorithm (Xu et al., 2014) in the context of distributed Bayesian learning of deep neural networks. Unfortunately, the SMS algorithm did not work even for relatively small neural networks and data sets, partly because of the high dimensionality and partly because the Monte Carlo estimation is inherently stochastic, both of which we found affected the convergence of the EP fixed point equations.

4. Stochastic Natural-gradient Expectation Propagation

In this section we will derive a novel convergent stochastic approximation based alternative to EP and power EP, which optimises the same variational objective but is significantly more tolerant of Monte Carlo noise. Our algorithm is derived using a modified variational objective with additional auxiliary variables but with the same optima as the original problem, and solving the dual problem using a stochastic approximation algorithm (Robbins and Monro, 1951). In the following we will use "EP" to refer to both EP and power EP for simplicity.

4.1 Auxiliary Variational Problem

For each i, we introduce an auxiliary natural parameter vector $\theta_i' \in \Theta$, and introduce a term $-\sum_{i=1}^n \mathrm{KL}(p_{\mu_i} \| p_{\theta_i'})$ into the variational objective (7), where p_{μ_i} and $p_{\theta_i'}$ are the base exponential family distribution given by mean parameter μ_i and natural parameter θ_i', respectively. This results in a lower bound on the original objective and is reminiscent of the EM algorithm (Dempster et al., 1977, Neal and Hinton, 1999) and of typical variational Bayes approximations (Beal, 2003, Wainwright and Jordan, 2008). Plugging the relevant formula for the KL divergence (1) into (7), we have the resulting variational problem

$$\begin{aligned} \max_{\mu_0, [\mu_i, \nu_i, \theta_i']_{i=1}^n} \quad & \theta_0^\top \mu_0 - A^*(\mu_0) + \sum_{i=1}^n \left(\nu_i - \beta_i \left(A_i^*(\mu_i, \nu_i) - \mu_i^\top \theta_i' + A(\theta_i')\right)\right) \\ \text{subject to} \quad & \mu_0 \in \mathcal{M} \\ & [\mu_i, \nu_i] \in \mathcal{M}_i \quad \text{for } i = 1, \ldots, n \\ & \theta_i' \in \Theta \quad \text{for } i = 1, \ldots, n \\ & \mu_0 = \mu_i \quad \text{for } i = 1, \ldots, n \end{aligned} \tag{13}$$

Maximizing over θ_i' while keeping the other variables fixed will simply set $\theta_i' = \nabla A^*(\mu_i)$, so that the KL terms vanish and resulting in the original problem (6). On the other hand, the constrained maximization over the original parameters $\mu_0, [\mu_i, \nu_i]_{i=1}^n$ requires introducing Lagrange multipliers,

$$\begin{aligned} \max_{[\theta_i']_{i=1}^n} \max_{\mu_0, [\mu_i, \nu_i]_{i=1}^n} \min_{[\lambda_i]_{i=1}^n} \quad & \theta_0^\top \mu_0 - A^*(\mu_0) + \sum_{i=1}^n \Big(\nu_i - \lambda_i^\top(\mu_i - \mu_0) - \beta_i \Big(A_i^*(\mu_i, \nu_i) - \mu_i^\top \theta_i' + A(\theta_i')\Big)\Big) \\ \text{subject to} \quad & \mu_0 \in \mathcal{M} \\ & [\mu_i, \nu_i] \in \mathcal{M}_i \quad \text{for } i = 1, \ldots, n \\ & \theta_i' \in \Theta \quad \text{for } i = 1, \ldots, n \end{aligned} \tag{14}$$

We can solve this inner constrained optimization by transforming to the convex dual. Noticing that the Lagrangian is concave in $\mu_0, [\mu_i, \nu_i]_{i=1}^n$ and that Slater's condition holds, the

duality gap is zero and we have,

$$\max_{[\theta_i']_{i=1}^n} \min_{[\lambda_i]_{i=1}^n} \max_{\mu_0,[\mu_i,\nu_i]_{i=1}^n} \theta_0^\top \mu_0 - A^*(\mu_0) + \sum_{i=1}^n \Big(\nu_i - \lambda_i^\top(\mu_i - \mu_0) - \beta_i\Big(A_i^*(\mu_i,\nu_i) - \mu_i^\top\theta_i' + A(\theta_i')\Big)\Big)$$
$$\text{subject to} \quad \mu_0 \in \mathcal{M} \qquad (15)$$
$$[\mu_i,\nu_i] \in \mathcal{M}_i \quad \text{for } i = 1,\ldots,n$$
$$\theta_i' \in \Theta \quad \text{for } i = 1,\ldots,n$$

Now maximizing over $\mu_0, [\mu_i,\nu_i]_{i=1}^n$, we have the equivalent dual problem, whose objective function we will denote with $L([\theta_j',\lambda_j]_{j=1}^n)$,

$$\max_{[\theta_i']_{i=1}^n} \min_{[\lambda_i]_{i=1}^n} \underbrace{A\left(\theta_0 + \sum_{i=1}^n \lambda_i\right) + \sum_{i=1}^n \beta_i \left(A_i\left(\theta_i' - \beta_i^{-1}\lambda_i, \beta_i^{-1}\right) - A(\theta_i')\right)}_{=:L([\theta_j',\lambda_j]_{j=1}^n)}$$
$$\text{subject to} \quad \theta_i' \in \Theta \quad \text{for } i = 1,\ldots,n \qquad (16)$$

In summary, our algorithm is coordinate maximization of (13), where we alternate between maximizing $[\theta_i']_{i=1}^n$ given $\mu_0, [\mu_i,\nu_i]_{i=1}^n$, and maximizing $\mu_0, [\mu_i,\nu_i]_{i=1}^n$ given $[\theta_i']_{i=1}^n$. To solve the second maximization, we transform to the dual and minimize (16) with respect to the Lagrange multipliers $[\lambda_i]_{i=1}^n$. This requires an inner iteration loop involving stochastic natural gradient descent, which is described next. The structure of our derived algorithm is the same as the convergent double loop EP algorithm of (Heskes and Zoeter, 2002). Our derivation is interesting and useful in that a single global variational problem is described, rather than a sequence of variational problems each obtained by minorizing around the current parameters and then maximizing, as in the minorization-maximization (MM) algorithm) (Hunter and Lange, 2004) and the CCCP algorithm (Yuille, 2002). Our algorithm is also different in the choice of stochastic natural gradient descent inner loop.

4.2 Stochastic Natural Gradient Descent

The gradient of the dual objective $L([\theta_j',\lambda_j]_{j=1}^n)$ is,

$$\frac{dL}{d\lambda_i} = \nabla A\left(\theta_0 + \sum_{j=1}^n \lambda_j\right) - \nabla_{\theta_i} A_i\left(\theta_i' - \beta_i^{-1}\lambda_i, \beta_i^{-1}\right) \qquad (17)$$

where we used the notation $\nabla_{\theta_i} A_i$ for the partial derivative of $A_i(\cdot,\cdot)$ with respect to its first (d-dimensional) argument. While this gradient is the direction of steepest descent in a Euclidean geometry it does not take into account the geometry of the base exponential family. In other words, the gradient is not covariant[2] (MacKay, 1999) and the corresponding gradient descent algorithm is not expected to perform well.

2. A covariant gradient descent algorithm roughly means that the size of the gradient in a direction positively covaries with the length scale in that direction hence with the desired step size. An example of a non-covariant gradient descent algorithm is steepest descent with standard Euclidean geometry, say for a 2D quadratic loss function that is long in the x-axis and short in the y-axis. The gradient will be close to pointing vertically up or down, but the best descent direction points horizontally instead.

A better approach is to use natural gradient descent (Amari et al., 1996, Amari and Nagaoka, 2001) or, equivalently, mirror descent (Beck and Teboulle, 2003, Raskutti and Mukherjee, 2015) instead. These methods treat the parameter space as a Riemannian manifold where the metric tensor is given by the Fisher information. The direction of steepest descent in this geometry is given by the gradient preconditioned by the inverse metric. Natural gradient descent has appealing theoretical properties as it incorporates second order information to be more covariant. As we will see below we can take advantage of natural gradient descent at no additional computational cost by reparametrising. As noted in the previous section, the Lagrange multiplier λ_i can be interpreted as the natural parameters of the base exponential family approximation to the likelihood $\exp(\ell_i(x))$. Reparameterising using the corresponding mean parameter γ_i instead, with $\lambda_i = \nabla A^*(\gamma_i)$, the gradient is,

$$\begin{aligned}\frac{dL}{d\gamma_i} &= \frac{d\lambda_i}{d\gamma_i}\left(\nabla A\left(\theta_0 + \sum_{j=1}^n \lambda_j\right) - \nabla_{\theta_i} A_i\left(\theta_i' - \beta_i^{-1}\lambda_i, \beta_i^{-1}\right)\right) \\ &= \nabla^2 A^*(\gamma_i)\left(\nabla A\left(\theta_0 + \sum_{j=1}^n \lambda_j\right) - \nabla_{\theta_i} A_i\left(\theta_i' - \beta_i^{-1}\lambda_i, \beta_i^{-1}\right)\right) \qquad (18)\end{aligned}$$

The appropriate metric in the mean parameter space is simply $\nabla^2 A^*(\gamma_i)$, so that its inverse cancels the $\nabla^2 A^*(\gamma_i)$ term (Raskutti and Mukherjee, 2015), and the natural gradient update is simply,

$$\gamma_i^{(t+1)} = \gamma_i^{(t)} + \epsilon_t\left(\nabla_{\theta_i} A_i\left(\theta_i' - \beta_i^{-1}\lambda_i^{(t)}, \beta_i^{-1}\right) - \nabla A\left(\theta_0 + \sum_{j=1}^n \lambda_j^{(t)}\right)\right) \qquad (19)$$

where the corresponding natural parameter is given as a function of the mean parameter, $\lambda_i^{(t)} := \nabla A^*(\gamma_i^{(t)})$, and ϵ_t is the step size at iteration t. These updates can be performed in series or parallel fashion. In our distributed Bayesian learning setting they are performed in an asynchronous distributed fashion (Section 5).

Recall that derivatives of the log partition function compute the mean parameters from the natural parameters. Hence the first term of the natural gradient step is the mean parameter of the current local tilted distribution,

$$p_i^{(t)}(x) = \exp\left((\theta_i' - \beta_i^{-1}\lambda_i^{(t)})^\top s(x) + \beta_i^{-1}\ell_i(x) - A_i(\theta_i' - \beta_i^{-1}\lambda_i^{(t)}, \beta_i^{-1})\right), \qquad (20)$$

while the second term is the mean parameter of the current exponential family approximation to the global posterior. Their difference gives the update for the mean parameter of the likelihood approximation. The gradient is zero when both terms are equal, precisely the condition from which the EP fixed point equation is derived.

In general, the first term cannot be obtained in closed form, and we instead use a Markov chain Monte Carlo (MCMC) estimate, leading to a stochastic natural-gradient descent algorithm. Specifically, let $\mathcal{K}_i(\cdot \,|\, x; \theta_i' - \beta_i^{-1}\lambda_i^{(t)}, \beta_i)$ be a Markov chain kernel with previous state x whose invariant distribution is the local tilted distribution (20). Let $x_i^{(t)}$ be

the state of the Markov chain at iteration t. The next state $x_i^{(t+1)}$ is obtained by simulating from the Markov chain, using the current values of the parameters,

$$x_i^{(t+1)} \sim \mathcal{K}_i\left(\cdot \mid x_i^{(t)}; \theta_i' - \beta_i^{-1}\lambda_i^{(t)}, \beta_i\right). \tag{21}$$

In summary, the stochastic natural gradient update is,

$$\gamma_i^{(t+1)} = \gamma_i^{(t)} + \epsilon_t \left(s\left(x_i^{(t+1)}\right) - \nabla A\left(\theta_0 + \sum_{j=1}^{n} \lambda_j^{(t)}\right)\right) \tag{22}$$

We refer to the resulting algorithm as "Stochastic Natural-gradient EP", or SNEP for short.

Technically, the stochastic natural-gradient descent requires unbiased estimates of gradients, and the mean parameter estimates obtained using MCMC updates are only unbiased if the Markov chain equilibrates in between gradient updates. In practice, we find that using MCMC samples works well regardless, an observation consistent with past experiences with stochastic approximation algorithms in machine learning (Neal and Hinton, 1999, Tieleman, 2008).

The stochastic natural gradient descent algorithm above serves as the inner loop of our double loop algorithm, and solves for the constrained maximization of the mean parameters $\mu_0, [\mu_i, \nu_i]_{i=1}^n$ given auxiliary parameters $[\theta_i']_{i=1}^n$. Returning now to the outer loop update, maximizing θ_i' involves simply setting it to be $\nabla A^*(\mu_i)$, the natural parameter corresponding to the mean parameter μ_i of the local tilted distribution (see the discussion after Equation 13). Assuming that the inner loop has converged, this and the mean parameters of all other tilted distributions would be equal to μ_0, so that the t'th outer loop update is,

$$(\theta_i')^{(t'+1)} = \nabla A^*(\mu_i) = \nabla A^*(\mu_0) = \theta_0 + \sum_{j=1}^{n} \lambda_j^{(\infty)} \tag{23}$$

where $\lambda_j^{(\infty)}$ is the converged value of the dual parameters in the inner loop. In practice, we simply perform the outer loop update infrequently (and before the inner loop has fully converged). In our experiments we do not see instabilities resulting from this, an observation also consistent with past experiences with double loop algorithms (Teh and Welling, 2002, Yuille, 2002, Heskes and Zoeter, 2002).

In the extreme case where the outer loop update is performed after every inner loop update, we can roll both updates into the following update,

$$\gamma_i^{(t+1)} = \gamma_i^{(t)} + \epsilon_t \left(\nabla_{\theta_i} A_i \left(\theta_0 + \sum_{j=1}^{n} \lambda_j^{(t)} - \beta_i^{-1}\lambda_i^{(t)}, \beta_i^{-1}\right) - \nabla A\left(\theta_0 + \sum_{j=1}^{n} \lambda_j^{(t)}\right)\right) \tag{24}$$

It is interesting to contrast the above with a damped EP update, which in our notation is given by,

$$\lambda_i^{(t+1)} = \lambda_i^{(t)} + \epsilon_t \left(\nabla A^* \left(\nabla_{\theta_i} A_i \left(\theta_0 + \sum_{j=1}^{n} \lambda_j^{(t)} - \beta_i^{-1}\lambda_i^{(t)}, \beta_i^{-1}\right)\right) - \left(\theta_0 + \sum_{j=1}^{n} \lambda_j^{(t)}\right)\right) \tag{25}$$

Note that $\nabla A^*(\cdot)$ converts from mean parameters to natural parameters, so that the first term in parentheses can be read as first computing the moments (mean parameters) of the tilted distribution then converting into the corresponding natural parameter. Hence each term in the damped EP update is obtained by converting the corresponding term in (24) from mean to natural parameters, i.e. applying $\nabla A^*(\cdot)$. In other words the update (24) can be thought of as a mean parameter space version of a damped EP update. When mean parameters are estimated with Monte Carlo noise, updating using mean parameters rather than natural parameters leads to more stable behaviour; see Section 6.1 for empirical results supporting this claim.

4.3 Discussion and Related Works

While we developed SNEP in the context of distributed Bayesian learning, it is clear that it is generally applicable, since the distribution (2) targeted is simply a product of factors, each of which is approximated by a factor in the base exponential family, precisely the setting of EP. SNEP can therefore be used in place of EP in situations where Monte Carlo moment estimates are used, including graphical models (Heess et al., 2013, Eslami et al., 2014, Jitkrittum et al., 2015, Lienart et al., 2015), hierarchical Bayesian models (Gelman et al., 2014), and approximate Bayesian computation (Barthelmé and Chopin, 2011).

Since Minka (2001), there have been a substantial number of extensions and alternatives to EP proposed. Stochastic EP (Li et al., 2015) and averaged EP (Dehaene and Barthelmé, 2015) assume that all factors can be well approximated by the *same* exponential family factor via parameter tying. This saves memory storage and was shown to work well. This is an orthogonal idea to our work, and it is possible to apply it to SNEP as well in the future, although in the distributed learning setting considered in this paper each worker must keep a copy of the parameters anyway, which means it would not reduce memory requirements. Convergent EP (Heskes and Zoeter, 2002) is also a double loop coordinate descent algorithm with convergence guarantees but SNEP differs in that the inner loop is a stochastic natural-gradient descent which is more robust to the use of Monte Carlo estimated moments. We note here that stochastic natural-gradient descent has also been used in Stochastic Variational Inference (Hoffman et al., 2013).

SNEP can be considered a black-box variational inference algorithm, as the only requirement on the model is the existence of MCMC samplers targeting the tilted distributions. Black-box methods have recently been developed for variational Bayes (Ranganath et al., 2014, Black-box Variational Inference or BBVI) and for power EP (Hernandez-Lobato et al., 2016, Black-box Alpha or BB-α). BB-α and SNEP are both methods for Bayesian learning based on power EP and both operate in the mean parameter space. BB-α uses the parameter tying idea of stochastic and averaged EP to construct a further approximated objective function which is then optimised using a convergent single-loop algorithm. As a result the fixed points of the algorithm are different from power EP. Single loop algorithms are generally preferrable as it is well known that double loop algorithms can be slow if the inner loop is run until convergence. In our experiments, we do not run the inner loop of SNEP until convergence, and have found empirically that even a single iteration of the inner loop per outer loop update does not affect convergence or stability. In fact, on a two-layer feedforward neural network, SNEP converges in less than one tenth of the number of epochs

and a fraction of the running times of those quoted by Hernandez-Lobato et al. (2016). In both BBVI and BB-α, naive Monte Carlo estimators are used, with samples drawn from the approximating distribution. The resulting estimators can have high variance, requiring techniques for variance reduction. In contrast, SNEP uses MCMC samplers targeting the tilted distribution. It is generally accepted that, in high-dimensional settings, MCMC samplers often have lower variance than naive Monte Carlo and as a result work better, with the tradeoff being that MCMC samplers need to equilibrate and produce correlated samples.

Our application of SNEP to distributed Bayesian learning applies one exponential family approximation per subset of data on each worker node. This contrasts with typical applications of EP and variational inference in general, which applies one approximation per data item. This is made possible due to the black-box flexibility of our approach, since the likelihood associated with a data subset is more complex than for a single data item. In cases where the subset is itself quite large, and the Bernstein-von Mises theorem holds, the likelihood will be close to a Gaussian, so that if we use a (full-covariance) Gaussian as the base exponential family, the approximation will introduce negligible biases. For a recent study of EP in the large data limit, see (Dehaene and Barthelmé, 2015). We can think of our approach as a hybrid which interpolates between a pure variational approach (when $n = N$) and a pure MCMC approach (when $n = 1$), with smaller n corresponding to less bias introduced by approximations but higher variance/computational costs.

In our experiments, we apply SNEP to deep neural networks. Deep learning has been extremely successful in a large number of different domains. However, one drawback of neural networks trained by stochastic optimisation techniques is that they do not provide uncertainty estimates. Both black-box variational methods and stochastic gradient MCMC methods can be applied to neural networks yielding uncertainty estimates. Probabilistic backpropagation (Hernandez-Lobato and Adams, 2015, PBP) relies on assumed density filtering (ADF), a sequential form of EP. Other approaches are based on mean field variational inference, which optimises a lower bound of the marginal likelihood. Graves (2011) proposed an algorithm using a biased Monte Carlo estimate of this lower bound to optimise it. More recently, this approach has been extended by Blundell et al. (2015, Bayes by Backprop) who obtain unbiased Monte Carlo estimates of the same objective using the reparametrization trick (Kingma and Welling, 2014).

5. The Posterior Server

Our development of SNEP is motivated by the problem of distributed Bayesian learning outlined in Section 2.2, where each log likelihood term $\ell_i(x)$ corresponds to the log probability/density of the data subset on worker node i. Using SNEP, each worker node iteratively learns an exponential family approximation of $\ell_i(x)$, with a master node coordinating the learning across workers. We call the master node the *posterior server*, as it maintains and serves the exponential family approximation of the posterior distribution, obtained by combining the prior with the likelihood approximations at the workers.

In more detail, the posterior server maintains the natural parameter $\theta_{\text{posterior}} := \theta_0 + \sum_{j=1}^{n} \lambda_j$ of the global posterior approximation, while each worker node i maintains the mean and natural parameters γ_i, λ_i of the likelihood approximation and the state of the MCMC

sampler x_i. It also maintains the auxiliary parameter θ_i' used in the outer loop. Learning at the worker proceeds by alternating between the MCMC (21) and inner loop updates (22), with periodic outer loop updates (23). These updates require access to the data subset at the node, as well as the cavity distribution, with natural parameters $\theta_{-i} := \theta_0 + \sum_{j \neq i} \lambda_j$, which is obtained by getting $\theta_{\text{posterior}}$ from the posterior server and subtracting the local λ_i.

Communication between the worker node and the posterior server involves the worker first sending the posterior server the difference $\Delta_i := \lambda_i^{\text{new}} - \lambda_i^{\text{old}}$ between the current natural parameters λ_i^{new} and the one during the previous communication with the server, λ_i^{old}. The posterior server updates its global posterior approximation via $\theta_{\text{posterior}}^{\text{new}} = \theta_{\text{posterior}}^{\text{old}} + \Delta_i$, and sends the new value $\theta_{\text{posterior}}^{\text{new}}$ back to the worker. The worker in turn uses this to update the cavity, $\theta_{-i}^{\text{new}} = \theta_{\text{posterior}}^{\text{new}} - \lambda_i^{\text{new}}$.

The pseudocode for the overall algorithm is given in Algorithm 1. Note that all communications are performed asynchronously and in a non-blocking fashion. In particular, Steps 11-17 are performed in a separate coroutine from the main loop (Steps 7-18), and Step 15 can happen several iterations of the main loop after Step 14. This is so that compute nodes can spend most of their time learning (Steps 8-10) and do not have to wait for network communications to complete. However, this means that the previous copy of the natural parameter λ_i used when the last message was sent to the master (Step 13) needs to be stored, in addition to the most recent one. We also note that faster compute nodes need not wait for slower ones since they each learn their own separate likelihood approximation parameters. It would be interesting for future research to explore adaptive methods to allow faster compute nodes to increase the data subsets that they learn from, and slower ones to decrease, to balance the learning progress across compute nodes more evenly.

5.1 Discussion and Related Works

Our naming of the posterior server contrasts with that of the parameter server (Ahmed et al., 2012) which is typically used for maximum likelihood (or minimum empirical risk) estimation of model parameters. Note however that our algorithmic contribution is effectively orthogonal to (Ahmed et al., 2012), who proposed a generic and robust computational architecture for distributed machine learning. We believe it is possible to implement our algorithm using the parameter server software framework.

One of the difficulties of the parameter server architecture is that learning happens at the level of parameters, with a single set of parameters being maintained across the cluster. Since the data subsets on workers are disjoint, the learning on each worker tends to make the local copy of the parameters diverge from those on the parameter server and on other workers. As a result, frequent synchronisation with the parameter server is necessary to prevent stale parameters and gradients. As an example, in the DistBelief method (Dean et al., 2012), experiments were conducted where the communication with the master was performed after every iteration, which can significantly slow down the learning process. On the other hand, one of the interesting aspects of the posterior server is that it lifts learning from the level of parameters to the level of distributions over parameters. As a result each worker can maintain a distinct parameter set in the MCMC sampler and a distinct likelihood approximation (since the likelihoods on different workers are indeed different as they have different data subsets) without requiring frequent communication with the posterior server.

Algorithm 1 Posterior Server: Distributed Bayesian Learning via SNEP

for each compute node $i = 1, \ldots, n$ **asynchronously do**
  let $\gamma_i^{(1)}$ be the initial mean parameter of local likelihood approximation.
  let $\lambda_i^{\text{old}} := \lambda_i^{(1)} := \nabla A^*(\gamma_i^{(1)})$ be the initial natural parameter of local likelihood approximation.
  let $\theta_{-i} := \theta_0 + \sum_{j \neq i} \lambda_j^{(1)}$ be the initial natural parameter of cavity distribution
  let $\theta_i' := \theta_{-i} + \lambda_i^{(1)}$ be the initial auxiliary parameter.
  let $x_i^{(1)} \sim p_{\theta_{-i} + \lambda_i^{(1)}}$ be the initial state of MCMC sampler.
  for $t = 1, 2, \ldots$ until convergence **do**
    update local state via MCMC:

$$x_i^{(t+1)} \sim \mathcal{K}_i \left(\cdot \mid x_i^{(t)}; \theta_i' - \beta_i^{-1} \lambda_i^{(t)}, \beta_i^{-1} \right)$$

    update local likelihood approximation:

$$\gamma_i^{(t+1)} := \gamma_i^{(t)} + \epsilon_t \left(s(x_i^{(t+1)}) - \nabla A \left(\theta_{-i} + \lambda_i^{(t)} \right) \right)$$
$$\lambda_i^{(t+1)} := \nabla A^*(\gamma_i^{(t+1)})$$

    every N_{outer} iterations **do**: update auxiliary parameter:

$$\theta_i' := \theta_{-i} + \lambda_i^{(t)}$$

    every N_{sync} iterations **asynchronously do:** communicate with posterior server:
      let $\Delta_i := \lambda_i^{(t)} - \lambda_i^{\text{old}}$.
      update $\lambda_i^{\text{old}} := \lambda_i^{(t)}$.
      send Δ_i to posterior server.
      receive $\theta_{\text{posterior}}$ from posterior server.
      update $\theta_{-i} := \theta_{\text{posterior}} - \lambda_i^{\text{old}}$.
  end for
end for
for the posterior server **do**
  let $\theta_{\text{posterior}} := \theta_0 + \sum_{j=1}^n \lambda_i^{(1)}$ be the initial natural parameter of the posterior approximation.
  maintain a queue of messages from workers.
  for each message Δ_i received from some worker i **do**
    update $\theta_{\text{posterior}} := \theta_{\text{posterior}} + \Delta_i$.
    send $\theta_{\text{posterior}}$ to worker i.
  end for
end for

The only role of communication here is for the cavity distributions, which can be thought of as a way for the system to focus the learning happening on the workers on the relevant regions of the parameter space (Xu et al., 2014). Empirically, the precise parameterisation of the cavity distribution is not very important. For an extreme example, suppose both the prior and likelihood terms are close to being Gaussians and the tractable family is also Gaussian. Then the likelihood approximation will not in fact depend on the cavity distribution at all, and will converge to the true likelihood on each worker independently. See also Zhang et al. (2015a) for elastic-averaging SGD, a similar idea of allowing each worker a separate parameter vector, using an ADMM-like methodology.

6. Experiments

In this section we present our experiments on SNEP and the posterior server. Our implementation, which is available at `http://bigbayes.github.com/PosteriorServer`, is in the Julia[3] technical computing language. In our setup, the master and workers are run on separate system processes (that is, in separate memory spaces) on a many-core computing cluster, making use of the high-level parallel programming abstractions provided by Julia. Other architectures are possible; for example, multiple threads sharing an address space, multiple GPUs, or nodes communicating over a network using MPI. For the MCMC sampler on workers, we chose an adaptive version of stochastic gradient Langevin dynamics (SGLD) (Welling and Teh, 2011) related to Adam (Kingma and Ba, 2015), which is more computationally scalable to larger neural network models and data sets than standard MCMC (see Appendix B for details). Our neural network models are implemented in the Mocha[4] deep learning Julia package.

6.1 Comparison to SMS on Bayesian Logistic Regression

In this section we compare SNEP against *Sampling via Moment Sharing* (SMS) (Xu et al., 2014), a related algorithm for distributed Bayesian learning whereby each worker has a separate MCMC sampler and coordination across workers is achieved using plain EP. SMS was originally proposed to scale up MCMC methods and as such it assumes that MCMC chains can be run for many iterations to convergence in between communications with the master. SMS uses the MCMC iterates to estimate the moments required for EP updates and so, as discussed in the introduction, requires a large number of iterations to produce the low Monte Carlo noise for EP updates to work properly.

We illustrate below the differing dynamics of SMS and SNEP when applied to a Bayesian logistic regression model with simulated data. We took a similar setup as in the SMS paper (Xu et al., 2014), generating a dataset $\mathcal{D} = \{(z_c, y_c)\}_{c=1}^N$ where covariates $z_c \in \mathbb{R}^d$ and response $y_c \in \{0, 1\}$. The conditional distribution of y_c given z_c and weights x is

$$p(y_c = 1 \mid z_c, x) = \sigma(z_c^\top x) \tag{26}$$

where σ denotes the logistic function. We used a Gaussian prior $p_0(x) = \mathcal{N}(x; 0_d, 10 I_d)$ on x and the aim is to construct an approximation to the posterior $p(x \mid \mathcal{D})$. We generated

3. `http://julialang.org`.
4. `https://github.com/pluskid/Mocha.jl`.

$N = 50000$ data points with $d = 50$ using iid draws for the covariates, $z_c \sim \mathcal{N}(\mu, \Sigma)$ where $\mu \in [0, 1]^d$ and $\Sigma = PP^\top$ with $P \in [-1, 1]^{d\times d}$, with entries drawn uniformly at random for both P and μ. The generating weight vector x^* is drawn from the prior $\mathcal{N}(x; 0_d, 10I_d)$. The labels y_c are then sampled according to the model.

Both algorithms were run with three workers each with one third of the data. SNEP is run with 1 inner loop iteration per synchronisation with the master (for the purpose of comparing against SMS). Varying numbers of MCMC samples per inner loop iteration were used for both algorithms, to investigate the effect of Monte Carlo noise on the performances of the algorithms (low number of samples meaning high noise and both lower performance and lower computational cost). The damping for SMS and the learning rate for SNEP were tuned for best performances. As the base exponential family, we used a full-covariance Gaussian. We compared the predictive RMSE $\sqrt{\sum_c |\hat{p}_c - y_c|^2/N}$ obtained with both methods over time where $\hat{p}_c = \sigma(z_c^T \overline{x})$ where $\overline{x}$ is the currently estimated posterior mean. We also compared the relative difference between the estimated posterior mean and that estimated from a long run of the No-U-Turn sampler (Hoffman and Gelman, 2014) as implemented in Stan (Carpenter et al., 2017) with 4 chains and 50000 iterations.

Figure 2 illustrates the performances of both algorithms and how they are influenced by the number of MCMC samples used per iteration. It can be observed that SNEP is more robust and better performing than SMS in the presence of noise. In general, we found that the number of samples needed for SMS to perform well grows with the dimensionality of the problem. In models with very high dimensionality and multi-modality (e.g., neural networks), the number of samples needed per step is very large leading to iterations that are computationally impractical, whereas SNEP can afford to use many fewer samples.

6.2 Bayesian Neural Networks

In this section we report experimental results applying SNEP and the posterior server to distributed Bayesian learning of neural networks. We have found that the SMS algorithm exhibited significant instabilities and was not suitable for these larger scale problems. Instead our aim here is to explore the behaviour of SNEP when varying various key hyperparameters. We used diagonal covariance Gaussians as the approximating exponential family due to computational cost considerations. While a diagonal Gaussian approximation can be quite poor for the full Bayesian posterior, past works have shown that they can produce good predictive performances (Hernandez-Lobato and Adams, 2015, Blundell et al., 2015, Kirkpatrick et al., 2017). Hence our aim is to evaluate the predictive performance of SNEP, treated as a distributed learning algorithm for neural networks, and comparing it to Adam (Kingma and Ba, 2015), a state-of-the-art stochastic gradient descent (SGD) algorithm with access to the whole data set on a single computer, as well as several state-of-the-art distributed SGD algorithms: asynchronous SGD (A-SGD) (Dean et al., 2012) and elastic averaging SGD (EASGD) (Zhang et al., 2015a).

In our experiments we used stochastic gradient Langevin dynamics with an preconditioning scheme reminiscent of Adam as the MCMC sampler. The preconditioning scheme is the same as the one proposed by (Li et al., 2016) except for the addition of a debiasing reminiscent of Adam. We found that sometimes this adaptive scheme lead to large injected noise, especially at the start of training which was detrimental to learning. To mitigate

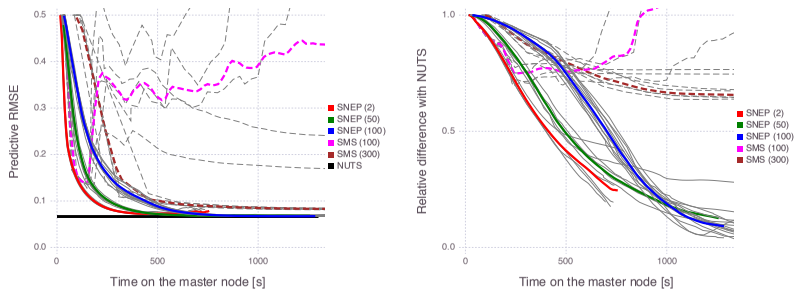


Figure 2: (left) Comparison of the predictive RMSE obtained with SMS (dashed-lines) and SNEP (full lines) versus the time on the master node when varying the number of MCMC samples per iteration (numbers reported in the legend). When too few samples are used for the moment estimates, SMS becomes unstable whilst SNEP does not suffer from this. SNEP is also consistently faster and more accurate than SMS. The accuracy obtained with the posterior mean estimated with a long run of Stan/NUTS is also displayed as comparison (horizontal line). (right) Relative difference between the posterior means estimated using Stan/NUTS and using SMS (dashed lines) or SNEP (full lines). Each coloured line corresponds to an average over several runs (represented in light grey).

this effect we limited the standard deviation of the injected noise to be at most ϵ in our experiments. For further details see Appendix B.

6.2.1 MNIST, Two Layer Fully Connected Network

In this section we look at training a fully connected neural network on the MNIST data set[5], which consists of 60000 training images of handwritten digits of size 28×28 and 10000 test images, the task being to classify each image into one of ten classes. The network has two hidden layers with 500 and 300 rectified linear units (ReLUs) respectively and softmax output units.

In the first set of experiments, we varied a number of hyperparameters of the learning regime while keeping the rest kept at default values, to investigate the sensitivity of the learning to these hyperparameters. The default values were chosen by hand in a rough initial round of experimentation as follows: the minibatch size is 100, the learning rate for SNEP was 0.02 and step size for the adaptive SGLD sampler was 0.001, the number of inner loop iterations per communication with master is $N_{sync} = 10$, the number of inner loop iterations per outer loop update is $N_{outer} = 10$, and the parameters were initialised with the popular Xavier initialisation (Glorot and Bengio, 2010). We set a minimum variance of 0.01 for the likelihood approximations, as we found that the estimated variances are otherwise too small due to the diagonal Gaussian approximation. Unless otherwise stated, we used four workers to explore the distributed behaviour. Test curves were produced by evaluating

5. `http://yann.lecun.com/exdb/mnist/`.

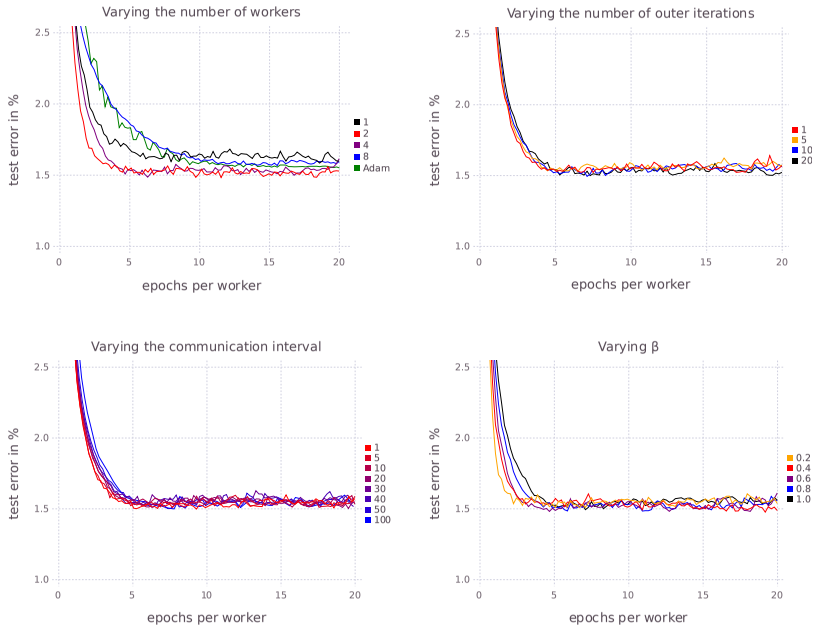


Figure 3: Results on a two-layer densely connected model on MNIST varying the parameters of SNEP and comparing against SGD, A-SGD, and EASGD. Four workers were used in the distributed methods in the experiments that held the number of workers fixed. This demonstrates that SNEP is insensitive to the full convergence of the outer loop, is robust to large communication intervals, that convergence is faster for a lower β (on this data set), and that SNEP produces competitive results quickly. Detailed analyses of sub-figures (left to right, top to bottom) are reported in the main text.

networks based on the approximate posterior means at the master. Each reported curve in the figures is an average over ten repetitions. In our experiments we used one CPU core per worker process. See Figure 3 for the results of this set of experiments.

First, we examined the behaviour of SNEP as N_{worker} is varied. We see that two workers perform significantly better than one worker or Adam. However, performance deteriorates as the number of workers is increased further. This is possibly due to smaller number of data items per worker which makes the variational approximation more severe. Another factor we found is that because we imposed a minimum variance in for the likelihood approximations, using more workers gives tighter cavity distributions which can affect the scale of the natural-gradients and slow convergence significantly.

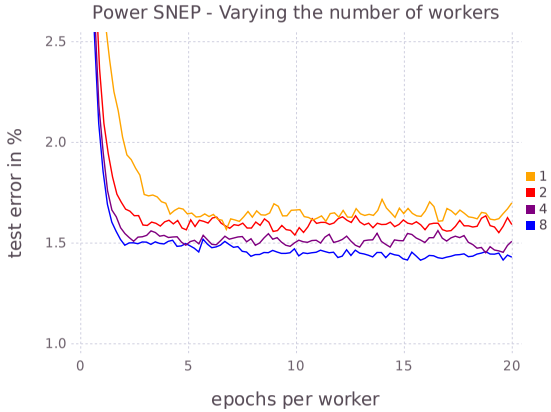


Figure 4: Results on a two-layer densely connected model on MNIST for power SNEP ($\beta = 1/N_{workers}$) varying the number of workers. There is a clear improvement in terms of convergence speed and final accuracy for more workers.

Second, we investigated the effect of varying N_{sync} and N_{outer}. Figure 3 demonstrates that SNEP is insensitive to these hyperparameters over reasonably large range of values. Note in particular that infrequent communications with the posterior server (up to once every 100 iterations in this experiment) did not significantly deteriorate the learning process at all. In fact, the related SMS algorithm (Xu et al., 2014) effectively involves communications with the master once every thousands of iterations.

Third, we investigated the effect of varying the β parameter. Interestingly, lower values of β sped up convergence significantly. The literature on power EP links the β parameter with locally minimising $\alpha-$divergences (where $\beta = 1/\alpha$). Lower values of β correspond to local approximations that are likely to capture more of the full posterior rather than matching one mode (Minka, 2005). Another factor is that given the minimum variance setting, lower values of β increase the variance of the cavity distribution allowing for more local exploration. Generally, In our experiments we found that a heuristic of setting $\beta = 1/N_{workers}$ worked particularly well for MNIST. We call the resulting algorithm power SNEP (p-SNEP). With this parameter setting we see a clear advantage for more workers (see figure 4). More workers give better results in terms of the test error and convergence is fastest for eight workers. Note that eight workers reach a test error of 1.5% after about two epochs, which is much faster than our experiments with $\beta = 1.0$.

Finally, we compared SNEP with two distributed SGD algorithms, asynchronous SGD (A-SGD, also known as Downpour) (Dean et al., 2012) and elastic averaging SGD (EASGD) (Zhang et al., 2015a). Our implementations differ slightly from those given in the papers. For A-SGD, both the workers and master performed Adam updates with independently

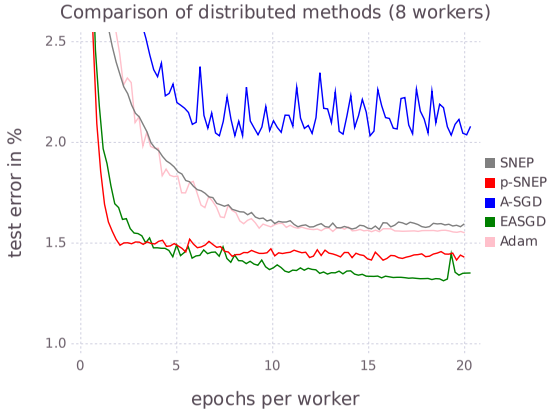


Figure 5: Results on a two-layer densely connected model on MNIST comparing SNEP and p-SNEP to asychronous SGD and elastic averaging SGD.

chosen step size constants. The authors of EASGD do not address the case of distributing the data among the workers. Nonetheless, we found this posed no problem to the algorithm. All hyperparameters were tuned to optimise performances. We used the same length of communication intervals $N_{sync} = 10$ and four workers for all algorithms. As shown in figure 5, SNEP (with $\beta = 1$) and p-SNEP (with $\beta = 1/N_{workers}$) achieve performances comparable to these state-of-the-art methods, both in terms of speed and final test accuracies. SNEP is comparable to Adam and better than A-SGD. p-SNEP outperforms A-SGD, Adam and SNEP, and is slightly faster initially than EASGD but achieves slightly higher test errors. It is important to note that we are tackling a harder problem with SNEP/p-SNEP, that of Bayesian learning rather than optimisation.

6.2.2 MNIST, Very Deep Fully Connected Network

Deeper models pose a much harder learning problem—as the depth increases, the number of saddle points increases exponentially (Dauphin et al., 2014). Moreover, learning often gets stuck not at critical points but in large plateaus of the error surface (Lipton, 2016). It is, therefore, interesting to evaluate (p-)SNEP against standard methods on a deeper fully-connected network, and we chose the architecture from Neelakantan et al. (2016), which has 20 hidden layers of 50 units each.

The experimental setup was similar to the previous section. The distributed methods used a communication intervals of $N_{sync} = 10$ and a varying number of workers. For p-SNEP, we optimized the prior variance, scaling of the injected noise, learning rate, step size, and beta hyperparameters by a course grid search over five orders of magnitude.

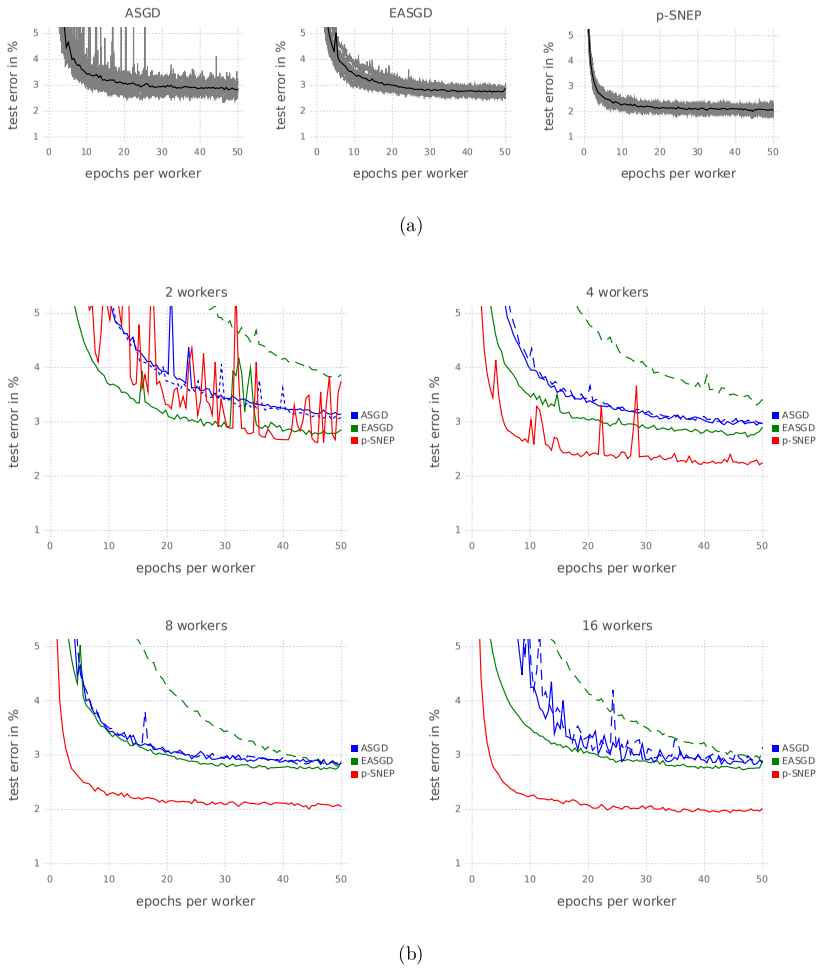


Figure 6: Results on deep narrow model for MNIST. (a) Comparing distributed methods for 8 workers. p-SNEP attains an extra percent of accuracy with less variability in the runs. The solid line is the average over 10 runs. (b) Varying the number of workers, averaged over 10 runs. The dashed line indicates result with no prelearning phase for A-SGD and EASGD, whereas for those methods the solid line indicates an Adam pre-learning phase of a third of an epoch.

Similarly, for A-SGD we optimized the Adam step size constants, separately on the workers and master, for EASGD the moving rate and Adam step size constants, and for SGD the Adam step size constants and scaling of injected noise, from no noise to three orders of magnitude. All methods were initialized by the Xavier method, and run with and without a non-distributed pre-learning phase of a third of an epoch. We found that pre-learning was essential for EASGD, while it was unnecessary for A-SGD and p-SNEP. In fact, for SNEP, too long a pre-learning phase is actually harmful to the final solution reached, which we suggest is due to the reduced exploration between workers. We only report results for p-SNEP without pre-learning (as per the previous section).

The same Adam step size constant was found to be optimal across all methods, 0.00065. For p-SNEP, the prior variance was set to 0.05, β to 1/8, and the learning rate to 0.04. For EASGD, the moving rate was set to 0.25. For A-SGD, the worker step sizes were set to 0.00065 and the master step size to a third of this; it must be slower to account for the staleness of the parameters. For SGD, we discovered that a small amount of noise, the same constant as is optimal for p-SNEP, reduces the variance of the runs around their mean by helping some runs avoid getting stuck in bad solutions. However, SGD did not converge to within the plot limits within the time set for experiments and so is omitted in the results shown in Figure 6a.

p-SNEP was found to significantly outperform A-SGD and EASGD, obtaining an extra percent of accuracy relative to other methods after fifty epochs for 8 and 16 workers. From the perspective of traditional neural network learning, our algorithm has two advantages for learning deep models: the addition of noise, and a principled method of regularizing the parameters. As in Neelakantan et al. (2016), we found it was important to add noise to the SGD methods. Although with a good initialization, added noise only improved the worst runs and not the best, smoothing out convergence. We conjecture that the added noise helps learning escape the difficult saddle points and plateaus. For this model, a strong prior variance was found to be important for p-SNEP to extract the full capacity of the model. We conjecture that this allows the gradients to flow back through the network more easily by forcing the parameters of the top layers to be small. Surprisingly, we found L2 regularization did not help with A-SGD and EASGD. p-SNEP also clearly benefits from additional workers in this setting, in contrast to A-SGD and EASGD (see Figure 6b).

It is interesting to note that this model has about one sixth the parameters of the previous two layer model. Yet, due to the deeper architecture and ability to successfully navigate the difficult learning landscape, p-SNEP is able to obtain similar test accuracies.

6.2.3 CIFAR-10, Convolutional Network

We also experimented with distributed Bayesian learning of convolutional neural networks, applying these to the CIFAR-10 data set (Krizhevsky, 2009) which consists of 50000 training instances and 10000 test instances from 10 classes, each instance being a 32x32 colour natural image. The network used is the one described in Alex Krizhesky's CIFAR tutorial[6] and consists of 8 layers: a first convolutional layer followed by max-pooling and local response normalization, a second convolutional layer also followed by max-pooling and local response normalization, and a third convolutional layer followed by a fully connected layer.

6. `https://code.google.com/p/cuda-convnet/wiki/Methodology`

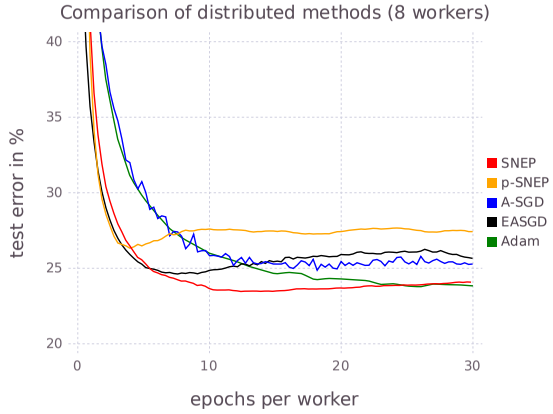


Figure 7: Learning curves varying the number of workers (averaged over three runs). The green line shows Adam with default parameters for comparison. Best viewed in colour.

For SNEP, we used the following settings: $N_{sync} = 10$, $N_{outer} = 10$, mini-batch size 100, weights initialised with the Xavier initialisation, and SNEP learning rate and adaptive SGLD step sizes of 0.02. We did not investigate improving performance by building in invariances using data perturbations or having multiple learning phases with different learning rates. Learning curves averaged over 3 runs are shown in Figure 7. All distributed algorithms are shown with hand-tuned optimal hyperparameter settings. On this data set SNEP outperforms other distributed algorithms. It converges as quickly as EASGD and p-SNEP but reaches a better final test accuracy. p-SNEP converges very fast but to a worse local optimum. It is not clear why p-SNEP performed better on previous experiments but not here; understanding the optimal choice of β is an interesting area for future research as we have observed that this parameter has a strong influence on the performance of the algorithm.

7. Discussion

In this paper, we have proposed a novel alternative to expectation propagation called stochastic natural-gradient expectation propagation (SNEP). SNEP is demonstrably convergent, even when using Monte Carlo estimates of the moments/mean parameters of tilted distributions. Experimentally, we find that SNEP converges more efficiently and more stably than other methods considered, particularly when Monte Carlo noise is high. Using SNEP, we have proposed the posterior server architecture for distributed Bayesian learning using an asynchronous non-blocking message-passing protocol. The architecture uses a

separate MCMC sampler on each worker, and SNEP to coordinate the samplers across the cluster so that the target distributions agree on the moments which characterise the base exponential family. In contrast with typical maximum likelihood parameter server architectures, the posterior server allows each worker to learn separate variational parameters, and as a result requires less frequent synchronisation across the cluster. We believe that this insight can allow for significant advances to distributed learning, although more work is still needed to make this reality. Finally, we applied SNEP and the posterior server to distributed Bayesian learning of both fully-connected and convolutional neural networks, where we showed performances on par with or better than state-of-the-art distributed and non-distributed optimisation algorithms. In conclusion, we have demonstrated that Bayesian learning of large complex models can be achieved efficiently and effectively in a distributing computing setting, and can achieve state-of-the-art performances on learning neural networks.

Our work leaves open a number of interesting avenues of future research, which we have discussed throughout the paper. It would be interesting to develop other novel convergent alternatives to EP with potentially faster convergence (e.g. using a single loop instead of double loop algorithm), and to apply such methods to other settings where Monte Carlo estimates are used within EP. It would also be interesting to further investigate combining SNEP with ideas from other projects such as the parameter tying idea of stochastic and averaged EP. We also believe that further explorations of learning regimes and hyperparameters, as well as of larger data sets, is called for, and will demonstrate further improvements to the method. These explorations can include using samples obtained at workers to form predictive probabilities, evaluating the algorithms using test log probabilities and on their abilities to quantify uncertainties, understanding the role of the β parameter, and combining SNEP with simulated annealing for optimisation. Another interesting area for application is on-device machine learning which falls naturally into our distributed data set-up. It would be interesting to compare SNEP to existing approaches such as federated learning (McMahan et al., 2016) We have also noted that our application of SNEP to neural network learning necessitates a diagonal covariance Gaussian approximation for computational reasons. Further work can consider richer posterior approximations, such as block-diagonal covariances or the use of matrix variate Gaussian distributions (Louizos and Welling, 2016). Ultimately, we believe these explorations will demonstrate the utility of a distributed Bayesian approach to learning.

Acknowledgments

LH is funded by the EPSRC OxWaSP CDT through grant EP/L016710/1. SW gratefully acknowledges support from the EPSRC AIMS CDT through grant EP/L015987/2. TL gratefully acknowledges funding from the Scatcherd European scholarship and EPSRC Grant EP/L505031/1. SJV thanks EPSRC for funding through EPSRC Grants EP/N000188/1 and EP/K009850/1. YWT gratefully acknowledges EPSRC for research funding through grant EP/K009362/1, and the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617071. The authors thank Daniel Hernández-Lobato, Yingzhen Li, Thang Bui and Rich Turner for valuable feedback on the preprint and suggestions about tied EP factors.

Appendix A. Relationship of Ideal Variational Problem in the Extended Exponential Family to Variational Inference

As expected, the ideal variational problem in the extended exponential family in (3) is intractable and approximations are needed for tractability, with different approximations leading to different variational approaches. For readers unfamiliar with this framework, it might be illuminating to link the above framework to standard mean field variational inference, which corresponds to approximating the mean domain $\tilde{\mathcal{M}}$ by the set of moments achievable by some family of factorised distributions q over x. The standard evidence lower bound on the log marginal probability (also known as the variational free energy) is,

$$\begin{aligned} p(\{D_i\}_{i=1}^n) &\geq \mathbb{E}_q[\log p(x, \{D_i\}_{i=1}^n) - \log q(x)] \\ &= \mathbb{E}_q\left[\theta_0^\top s(x) - A(\theta_0) + \sum_{i=1}^n \ell_i(x)\right] - \mathbb{E}_q\left[\log q(x)\right] \\ &= \tilde{\theta}^\top \tilde{\mu} - A(\theta_0) - \mathbb{E}_q\left[\log q(x)\right] \end{aligned}$$

where we have introduced $\mu = \mathbb{E}_q[s(x)]$ and $\nu_i = \mathbb{E}_q[\ell_i(x)]$ as the moments of q, and we have used the definition of the extended natural and mean parameters $\tilde{\theta}$, $\tilde{\mu}$. The second term is a constant not dependent on q, while the third term above is the negative entropy, so that the above is equivalent to (3) except that the optimisation is only over a family of factorised distributions q.

Appendix B. Additional Techniques for Bayesian Neural Networks

In our experiments we investigated the use of SNEP and the posterior server for distributed Bayesian learning of neural networks. To get the method working well on the notoriously complicated posterior distributions for neural networks, a number of additional techniques are needed, which we describe here.

B.1 Adaptive Stochastic Gradient Langevin Dynamics

Most of the computational costs associated with the algorithm involve the MCMC updates to the state x_i. When the number of data points stored on each compute node is large, standard MCMC updates are infeasible as each update requires computations involving every data point. In our experiments we used the stochastic gradient Langevin dynamics (SGLD) algorithm proposed by Welling and Teh (2011) which scales well to large data sets.

SGLD uses mini-batches of data to provide unbiased estimates of gradients which are used in a time discretized Langevin dynamics simulation whose stationary distribution is the desired tilted distribution (20). SGLD injects noise in every step. As the discretization stepsize decreases the noise due to the stochastic gradients is eventually dominated by the injected noise and hence neglibible. Likewise the discretization introduces errors which tend to zero as the discretization step sizes decreases to zero; see (Teh et al., 2015, Vollmer

et al., 2016). Recall that the data points on compute node i is $D_i = \{y_c\}_{c \in S_i}$, and the log likelihood is

$$\ell_i(x_i) = \sum_{c \in S_i} \log p(y_c \mid x_i).$$

Let $B^{(t)} \subset S_i$ be a mini-batch of data, chosen uniformly at random with fixed size. Each SGLD update is,

$$x_i^{(t+1)} = x_i^{(t)} + \kappa_t (M^{(t)})^{-1} \left(\nabla s(x_i^{(t)})^\top (\theta_i' - \beta_i^{-1} \lambda_i^{(t)}) + \beta_i^{-1} \frac{|S_i|}{|B^{(t)}|} \sum_{c \in B^{(t)}} \nabla \log p(y_c | x_i^{(t)}) \right) + \eta_t,$$
$$\eta_t \sim \mathcal{N}(0, 2\kappa_t (M^{(t)})^{-1}). \tag{27}$$

The term inside the parentheses is an unbiased estimate of the gradient of the log density of the tilted distribution (20), κ_t is the discretization step size, $M^{(t)}$ is (an adaptive) diagonal mass matrix, while η_t is an injected normally-distributed noise, which prevents SGLD from converging to a mode of the distribution and distinguishes it from stochastic gradient descent. See Welling and Teh (2011) for details.

In (27), $M^{(t)}$ is a mass matrix which effectively controls the length scale of updates to each dimension of x_i. It is well known that in neural networks the length scales of gradients differ significantly across different parameters and adaptation of learning rates specific to each parameter is crucial for successful deployment of stochastic gradient descent learning. We have found that this is the case for SGLD as well, and used an adaptation scheme for the mass matrix reminiscent of Adam (Kingma and Ba, 2015). The adaptation scheme is the same as the one proposed by (Li et al., 2016) except for the addition of a debiasing reminiscent of Adam. At iteration t let g_t be the stochastic gradient estimate in (27). We use a diagonal mass matrix M_t that is updated according to the following update equations:

$$v_t = \beta v_{t-1} + (1 - \beta) g_t \odot g_t$$
$$\mathrm{diag}(M_t) = \frac{v_t}{1 - \beta^2},$$

where $\odot$ denotes the elementwise product. We used $\beta = 0.999$ in our experiments. The actual adaptive stepsize is then given by

$$\kappa_t = \frac{\epsilon}{\sqrt{M_t} + \delta},$$

where $\epsilon = 10^{-3}$ and $\delta = 10^{-8}$ is added for numerical stability. We found that sometimes this adaptive scheme lead to large injected noise, specially at the start of training which was detrimental to learning. To mitigate this effect we limited the standard deviation of injected noise to be at most ϵ in our experiments. An alternative solution to this problem could be the approach proposed by Lu et al. (2017).

B.2 Shifting MCMC States After Communication with Posterior Server

After each communication with the posterior server, the target distribution of the MCMC sampler on the worker, say i, is changed, because of θ_{-i} being updated in Step 16 of

Algorithm 1. Assuming that the MCMC sampler had previously converged, it will now not be so anymore because of this shift in the target distribution, and a number of burn-in iterations may be needed before the mean parameter estimates can be used for SNEP updates again.

For Gaussian base exponential families, we can shift the MCMC state along with the target distribution when θ_{-i} is updated in the following way. Suppose μ_i^{old}, Σ_i^{old}, μ_i^{new}, Σ_i^{new} are the means and covariances of the approximate Gaussian posterior before and after the update to θ_{-i} (with natural parameters $\lambda_i + \theta_{-i}^{\text{old}}$, $\lambda_i + \theta_{-i}^{\text{new}}$ respectively where λ_i is the current natural parameter of the Gaussian likelihood approximation). Suppose x_i^{old} is the MCMC state before the update. Then we shift the MCMC state as follows:

$$x_i^{\text{new}} = \mu_i^{\text{new}} + (\Sigma_i^{\text{new}})^{\frac{1}{2}} (\Sigma_i^{\text{old}})^{-\frac{1}{2}} (x_i^{\text{old}} - \mu_i^{\text{old}}). \tag{28}$$

The idea is that x_i^{new} should be at the same location relative to the new Gaussian approximation to the posterior as x_i^{old} is relative to the old Gaussian approximation. We have found that no burn-in is needed with this shift in the MCMC state after each communication.

References

A. Ahmed, M. Aly, J. Gonzalez, S. Narayanamurthy, and A. J. Smola. Scalable inference in latent variable models. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2012.

S. Amari and H. Nagaoka. *Methods of Information Geometry*. American Mathematical Society, 2001.

S. Amari, A. Cichocki, and H. Yang. A new algorithm for blind signal separation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 8, pages 757–763, 1996.

R. Bardenet, A. Doucet, and C. Holmes. Towards scaling up Markov chain Monte Carlo: an adaptive subsampling approach. In *International Conference on Machine Learning (ICML)*, 2014.

S. Barthelmé and N. Chopin. ABC-EP: Expectation propagation for likelihood-free Bayesian computation. In *International Conference on Machine Learning (ICML)*, 2011.

H. Battey, J. Fan, H. Liu, J. Lu, and Z. Zhu. Distributed estimation and inference with statistical guarantees. *ArXiv:1509.05457*, 2015.

M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight Uncertainty in Neural Network. In *International Conference on Machine Learning (ICML)*, pages 1613–1622, 2015.

B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software, Articles*, 76(1):1–32, 2017.

Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2933–2941, 2014.

J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, and A. Y. Ng. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1223–1231, 2012.

G. Dehaene and S. Barthelmé. Expectation propagation in the large-data limit. *ArXiv:1503.08060*, 2015.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

A. Doucet, N. de Freitas, and N. J. Gordon. *Sequential Monte Carlo Methods in Practice*. Statistics for Engineering and Information Science. New York: Springer-Verlag, May 2001.

S. M. A. Eslami, D. Tarlow, P. Kohli, and J. Winn. Just-in-time learning for fast and flexible inference. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

A. Gelman, A. Vehtari, P. Jylänki, T. Sivula, D. Tran, S. Sahai, P. Blomstedt, J. P. Cunningham, D. Schiminovich, and C. Robert. Expectation propagation as a way of life: A framework for Bayesian inference on partitioned data. ArXiv:1412.4869, 2014.

W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, 1996.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

A. Graves. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2348–2356, 2011.

N. Heess, D. Tarlow, and J. Winn. Learning to pass expectation propagation messages. In *Advances In Neural Information Processing Systems (NIPS)*, 2013.

R. Herbrich, T. Minka, and T. Graepel. TrueskillTM: A Bayesian skill rating system. *Advances in Neural Information Processing Systems (NIPS)*, 19:569, 2007.

J. M. Hernandez-Lobato and Ryan Adams. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *International Conference on Machine Learning (ICML)*, pages 1861–1869, 2015.

J. M. Hernandez-Lobato, Y. Li, M. Rowland, T. Bui, D. Hernandez-Lobato, and R. Turner. Black-Box Alpha Divergence Minimization. In *International Conference on Machine Learning (ICML)*, pages 1511–1520, 2016.

T. Heskes and O. Zoeter. Expectation propagation for approximate inference in dynamic Bayesian networks. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 18, 2002.

M. Hoffman, D. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

M. D. Hoffman and A. Gelman. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15, 2014.

Z. Huang and A. Gelman. Sampling for Bayesian computation with large datasets. Technical report, Department of Statistics, Columbia University, 2005.

D. R. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 2004.

W. Jitkrittum, A. Gretton, N. Heess, S. M. Ali Eslami, B. Lakshminarayanan, D. Sejdinovic, and Z. Szabó. Kernel-based just-in-time learning for passing expectation propagation messages. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society B*, 76, 2014.

A. Korattikara, Y. Chen, and M. Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *International Conference on Machine Learning (ICML)*, 2014.

A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

Leimkuhler, B. and Shang, X. Adaptive Thermostats for Noisy Gradient Systems. *SIAM Journal on Scientific Computing*, 38(2):A712–A736, 2016.

C. Li, C. Chen, D. Carlson, and L. Carin. Preconditioned stochastic gradient Langevin dynamics for deep neural networks. In *AAAI Conference on Artificial Intelligence*, 2016.

Y. Li, J. M. Hernandez-Lobato, and R. E. Turner. Stochastic expectation propagation. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

T. Lienart, Y. W. Teh, and A. Doucet. Expectation particle belief propagation. In *Advances In Neural Information Processing Systems (NIPS)*, 2015.

Z. C. Lipton. Stuck in a what? adventures in weight space. *arXiv preprint arXiv:1602.07320*, 2016.

C. Louizos and M. Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning (ICML)*, volume 48, pages 1708–1716, 2016.

X. Lu, V. Perrone, L. Hasenclever, Y. W. Teh, and S. J. Vollmer. Relativistic Monte Carlo . In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1236–1245, 2017.

Y.-A. Ma, T. Chen, and E. B. Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

D. J. C. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. http://www.inference.org.uk/mackay/abstracts/ica.html, 1999.

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

T. Minka. Power EP. Technical report, Microsoft Research, 2004.

T. Minka. Divergence measures and message passing. Technical report, Microsoft Research Cambridge, January 2005. URL `https://www.microsoft.com/en-us/research/publication/divergence-measures-and-message-passing/`.

T. P. Minka. *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2001.

A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning (ICML)*, 2014.

R. M. Neal and G.E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1999.

A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens. Adding gradient noise improves learning for very deep networks. In *International Conference on Learning Representations (ICLR) Workshop*, 2016.

W. Neiswanger, C. Wang, and E. P. Xing. Asymptotically exact, embarrassingly parallel MCMC. In *International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.

M. Opper and O. Winther. Gaussian Processes for Classification: Mean-Field Algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

S. Patterson and Y. W. Teh. Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014.

G. Raskutti and S. Mukherjee. The information geometry of mirror descent. *IEEE Transactions on Information Theory*, 61:1451–1457, 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, 2014.

H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407, 1951.

S. L. Scott, A. W. Blocker, F. V. Bonassi, H. Chipman, E. George, and R. McCulloch. Bayes and big data: the consensus Monte Carlo algorithm. In *Bayes 250*, 2013.

Y. W. Teh and M. Welling. The unified propagation and scaling algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, volume 14, 2002.

Y. W. Teh, A. H. Thiéry, and S. J. Vollmer. Consistency and fluctuations for stochastic gradient Langevin dynamics. *Journal of Machine Learning Research*, 2015.

T. Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *International Conference on Machine Learning (ICML)*, 2008.

S. J. Vollmer, K. C. Zygalakis, and Y. W. Teh. Exploration of the (non-)asymptotic bias and variance of stochastic gradient Langevin dynamics. *Journal of Machine Learning Research*, 2016.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

X. Wang and D. B. Dunson. Parallelizing MCMC via Weierstrass sampler, 2013. arXiv:1312.4605.

M. Welling and Y. W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning (ICML)*, 2011.

W. Wiegerinck and T. Heskes. Fractional belief propagation. *Advances in Neural Information Processing Systems (NIPS)*, 2003.

M. Xu, B. Lakshminarayanan, Y. W. Teh, J. Zhu, and B. Zhang. Distributed Bayesian posterior sampling via moment sharing. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized belief propagation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 13, pages 689–695, 2001.

A.L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722, 2002.

S. Zhang, A. Choromanska, and Y. LeCun. Deep learning with elastic averaging SGD. *Advances in Neural Information Processing Systems (NIPS)*, 2015a.

Y. Zhang, J. C. Duchi, and M. J. Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 2015b.

T. Zhao, G. Cheng, and H. Liu. A partially linear framework for massive heterogeneous data. *The Annals of Statistics*, 44(4):1400–1437, 08 2016.

Journal of Machine Learning Research 18 (2017) 1-28 Submitted 1/16; Revised 5/17; Published 10/17

Optimal Dictionary for Least Squares Representation

Mohammed Rayyan Sheriff MOHAMMEDRAYYAN@SC.IITB.AC.IN
Debasish Chatterjee DCHATTER@IITB.AC.IN
Systems and Control Engineering
IIT Bombay
Mumbai 400076, India

Editor: Benjamin Recht

Abstract

Dictionaries are collections of vectors used for the representation of a class of vectors in Euclidean spaces. Recent research on optimal dictionaries is focused on constructing dictionaries that offer sparse representations, i.e., ℓ_0-optimal representations. Here we consider the problem of finding optimal dictionaries with which representations of a given class of vectors is optimal in an ℓ_2-sense: optimality of representation is defined as attaining the minimal average ℓ_2-norm of the coefficients used to represent the vectors in the given class. With the help of recent results on rank-1 decompositions of symmetric positive semidefinite matrices, we provide an explicit description of ℓ_2-optimal dictionaries as well as their algorithmic constructions in polynomial time.

Keywords: ℓ_2-optimal dictionary, rank-1 decomposition, finite tight frames

1. Introduction

A *dictionary* is a collection of vectors in a finite-dimensional vector space over $\mathbb{R}$, with which other vectors of the vector space are represented. A dictionary is a generalization of a basis: While the number of vectors in a basis is exactly equal to the dimension of the vector space, a dictionary may contain more elements. In this article we consider a problem of finding an optimal dictionary, where optimality is interpreted as the minimum expected average size of the coefficients required to represent a certain collection of vectors drawn from a given probability distribution.

We begin with a toy example to motivate the problems treated in this article. Let V be a random vector that attains values 'close' to $\begin{pmatrix}0 & 2\end{pmatrix}^\top$ with high probability; the situation is demonstrated below:

©2017 Mohammed Rayyan Sheriff and Debasish Chatterjee.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-046.html.

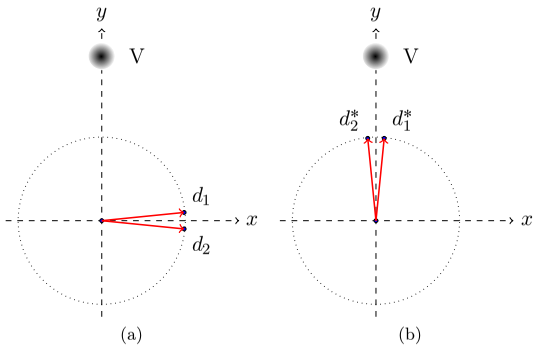


Figure 1: Comparison of two dictionaries.

Suppose that our dictionary consists of the vectors $d_1 = \begin{pmatrix}1 & -\epsilon\end{pmatrix}^\top$ and $d_2 = \begin{pmatrix}1 & \epsilon\end{pmatrix}^\top$ in $\mathbb{R}^2$, with a small positive value of ϵ. Since we must represent V using d_1 and d_2, the corresponding coefficients α_1 and α_2 must be such that $\alpha_1 \begin{pmatrix}1 & \epsilon\end{pmatrix}^\top + \alpha_2 \begin{pmatrix}1 & -\epsilon\end{pmatrix}^\top = V \approx \begin{pmatrix}0 & 2\end{pmatrix}^\top$. A quick calculation shows that the magnitudes of the coefficients α_1 and α_2 should then be approximately equal to $1/(\epsilon)$ with high probability. To wit, the magnitudes of these coefficients are large for small values of ϵ. It is therefore more appropriate in this situation to consider a dictionary consisting of vectors $d_1^* = \begin{pmatrix}\epsilon & 1\end{pmatrix}^\top$ and $d_2^* = \begin{pmatrix}-\epsilon & 1\end{pmatrix}^\top$ to represent the samples of V, in which case, the magnitudes of the coefficients of the representations are closer to 1 with high probability. The latter values are comparatively far smaller compared to the values close to $1/(\epsilon)$ obtained with the preceding dictionary. This simple example shows that given some statistical information about the random vectors to be represented, the question of designing a dictionary that minimizes the average cost of representation can be better addressed.

Let us now turn to a situation in which considering the average cost of representations is natural. Our motivation comes from a control theoretic ideas perspective. Consider a linear time-invariant control system modeled by the recursion

$$x(t+1) = Ax(t) + Bu(t), \quad t = 0, 1, \dots, \tag{1}$$

where the 'system matrix' $A \in \mathbb{R}^{n\times n}$ and the 'control matrix' $B \in \mathbb{R}^{n\times m}$ are given, with the initial boundary condition $x(0) = \bar{x} \in \mathbb{R}^n$ fixed. For an arbitrarily selected $\hat{x} \in \mathbb{R}^n$, consider the standard *reachability problem* for (1), that is:

$$\begin{aligned}&\text{If possible, find a sequence } (u(t))_t \subset \mathbb{R}^m \text{ of control vectors}\\ &\text{that steer the system states to } \hat{x}.\end{aligned} \tag{2}$$

A necessary and sufficient condition for such a sequence to exist for every pair $(\bar{x}, \hat{x})$ is that the rank of the matrix $\mathfrak{R}_K(A,B) := \begin{pmatrix}B & AB & \cdots & A^{n-1}B\end{pmatrix}$ is equal to n, which we impose for the moment. Letting $K := \min\left\{k \geqslant 0 \;\middle|\; \operatorname{rank}\left(\mathfrak{R}_K(A,B)\right) = n\right\}$ denote the 'reachability

index' of (1), we see at once that the control vectors $(u(t))_{t=0}^{K-1}$ needed to execute the transfer of the states of (1) from $\bar{x}$ to $\hat{x}$ must be a solution to the linear equation

$$\hat{x} - A^K \bar{x} = \sum_{t=0}^{K-1} A^t B u(t) = \mathfrak{R}_K(A,B) \begin{pmatrix} u(K-1) \\ \vdots \\ u(1) \\ u(0) \end{pmatrix}.$$

It is now natural to consider the 'control cost' of transferring $\bar{x}$ to $\hat{x}$, for which, a natural candidate is the associated ℓ_2 performance index $\sum_{i=0}^{K-1} \|u(t)\|^2$. Since in practice, the ℓ_2 performance index is analogous to the amount of energy spent to control the system, its practical importance can hardly be overstated in the context of control. Let us list three examples:

- In attitude control/orientation problems of space vehicles, one must execute most of the rapid manoeuvre using the energy from the limited amount of fuel on board, or with the energy available from on-board batteries; minimizing the energy expenditure, therefore, is crucial.
- In controlled automated mobile robots (e.g., automated cars) designed to reach a given location within a certain time, reduction of energy consumption leads directly to reduction in fuel consumed.
- In control of electronic systems such as power electronic drives, the associated ℓ_2 performance index involves information of the amount of power drawn from the electricity grid to control the system, leading directly to minimization of power consumption and thereby heating.

Minimization of control effort has been an integral part of control theory, and is generally studied under the class of Linear Quadratic problems; see, e.g., (Bertsekas, 1995), (Anderson and Moore, 2007), (Clarke, 2013), (Liberzon, 2012), or any standard book on optimal control. It is evident that the task of designing control systems that require minimum control energy for their typical manoeuvres is of great importance.

It is a standard practice to study the reachability problem (2), for $\bar{x} = 0$ and $\hat{x}$ on the unit sphere; due to linearity of (1), this special case provides sufficient insight into the general case. Let us consider the following optimal control problem:

$$\begin{aligned} \underset{(u(t))_t}{\text{minimize}} \quad & \mathsf{E}\left[\sum_{t=0}^{K-1} \|u(t)\|^2\right] \\ \text{subject to} \quad & \begin{cases} x(t+1) = Ax(t) + Bu(t) \quad \text{for all } t = 0, \dots, K-1, \\ x(0) = 0, \\ x(K) = \hat{x} \text{ distributed according to } \mu, \end{cases} \end{aligned} \tag{3}$$

where μ is a probability distribution on $\mathbb{R}^d$. It is known that if $\hat{x}$ is uniformly distributed over the unit sphere, then the optimal control problem (3) admits an unique optimal solution and the optimum value is proportional to $\mathrm{tr}\big(W_{A,B}^{-1}\big)$, where $W_{A,B} \coloneqq \mathfrak{R}_K(A,B)\mathfrak{R}_K(A,B)^\top$ is the *controllability grammian* of the system; for details see, e.g., (Müller and Weber, 1972) and (Pasqualetti et al., 2014). It can be readily shown that if $\Sigma \coloneqq \mathsf{E}[\hat{x}\hat{x}^\top]$ is well defined, then the optimum value of (3) is equal to $\mathrm{tr}\big(\Sigma W_{A,B}^{-1}\big)$. Evidently, for a given distribution of

$\hat{x}$, different linear systems (1) — described completely by the pair (A,B) — incur different optimum values $\mathrm{tr}\big(\Sigma W_{A,B}^{-1}\big)$ of (3).

Against the above backdrop, consider the question of *designing* the linear control system (1) such that the value of (3) is as low as possible. Since most control problems involve designing control sequences to execute a class of desired manoeuvres, for a given distribution of $\hat{x}$ it is then natural to design the linear systems in order to minimize the optimum value $\mathrm{tr}\big(\Sigma W_{A,B}^{-1}\big)$ of the optimal control problem (3). In this case, the system design problem is similar to the one of finding an ℓ_2-optimal dictionary as described above: here the matrices A and B are to be designed, within a feasible region, such that the column vectors constituting the matrix $\mathfrak{R}_K(A,B)$ lead to minimal expected average cost of reachability, i.e., minimal value of (3). Such problems routinely arise in networked control, where the pair (A,B) is a function of the constituent systems and the connectivity of the network. From an operational standpoint, it is good for a networked system to have its components connected in a way such that the resulting system incurs small expected average state transfer costs. Indeed, control systems are typically designed (Müller and Weber, 1972) by optimizing a *figure of merit / measure of quality / measure of controllability*; in particular, networked control systems are designed in (Pasqualetti et al., 2014) using a measure of quality defined there. Based on this work on ℓ_2-optimal dictionaries, we have proposed a novel measure of quality in (Sheriff and Chatterjee, 2017), and further developments for algorithmic synthesis of large-scale control systems will be reported elsewhere. Besides these applications in control theory and practice, one of the key objective of our work here is to investigate and understand the physical nature of the ℓ_2-optimal dictionaries independent of their connection with control theory. Such a study will shed light on other control theoretic properties of observability and estimation.

There has been significant recent research into finding optimal dictionaries, briefly outlined in (Tošić and Frossard, 2011); current research centers around the development of learning algorithms for finding optimal dictionaries. Much of the thrust is on arriving at dictionaries that offer sparse representations of sample vectors. One of the first learning algorithms to develop a dictionary that offers sparse representation of images was given in (Olshausen and Field, 1997). Since then many learning algorithms have been developed to obtain dictionaries that offer sparse representation along with other special properties such as online computation capability (Mairal et al., 2009a), better classification property (Mairal et al., 2009b; Yang et al., 2011), better adaptive properties (Skretting and Engan, 2010); several other algorithms are given in (K. Delgado et al., 2003; Yaghoobi et al., 2009; Mallat and Zhang, 1993).

The problem addressed in this article differs from the mainstream research of finding dictionaries offering sparse (ℓ_0-optimal) representations in the sense that our objective is to find dictionaries that give minimum average ℓ_2-norm of the coefficient vector used for representation. Intuitively, optimization of the ℓ_2-norm of the representation vector tends to 'distribute' the information of the data being represented among all components of the representation vector; this makes the representation robust to accidental changes in the coefficients.

- An advantage of considering the ℓ_2-cost is that it involves a norm arising from an inner product; consequently, it comes with a rich set of properties associated with it. These properties are crucially employed in this article to modify the intrinsically non-convex

problem of finding an ℓ_2-optimal dictionary into an *equivalent* convex optimization problem,[1] allowing us to compute an optimal dictionary in *polynomial time* and arrive at *analytical expressions of the optimal costs*. We provide these algorithms in Sections 4.1 and 4.3.

- One more advantage of considering optimization in the ℓ_2-sense is related to the fact that the ℓ_2-cost involves the natural notion of *energy* which is extremely important in practice, especially in control theoretic applications.
- The results presented here also add to the recent developments in the advantages of representing signals/vectors using tight frames for finite-dimensional Hilbert spaces.

This article unveils as follows: In Section 2 we formally introduce our problem of finding an optimal dictionary which offers least square representation. Section 2 is the heart of this article, where we solve the problem of finding an ℓ_2-optimal dictionary, and arrive at an explicit solution. Algorithms to construct ℓ_2-optimal dictionaries are given in Section 4, where we present the proofs of our main results. The case of representing random vectors distributed uniformly on the unit sphere is treated in Subsection 2.4; we demonstrate that the ℓ_2-optimal dictionaries in this case are *finite tight frames*. The intermediate Section 3 contains results related to rank-1 decomposition of positive semidefinite matrices; these constitute essential tools for the solutions of our main results. We conclude in Section 5 with a summary of this work and future directions.

Notations

We employ standard notations in this article. As usual, $\|\cdot\|$ is the standard Euclidean norm. The $n \times n$ identity and $m \times n$ zero matrices are denoted by I_n and $O_{m\times n}$, respectively. For a matrix M we let $\mathrm{tr}(M)$ and M^+ denote its trace and Moore-Penrose pseudo-inverse, respectively. The set of $n \times n$ symmetric and positive (semi-)definite matrices with real entries is denoted by $\mathbb{S}^{n\times n}_{++}$ ($\mathbb{S}^{n\times n}_{+}$), and the set of $n \times n$ symmetric matrices with real entries is denoted by $\mathbb{S}^{n\times n}$. For a Borel probability measure μ defined on $\mathbb{R}^n$, we let $\mathsf{E}_\mu[\cdot]$ denote the corresponding mathematical expectation. The image of a map f is written as $\mathrm{image}(f)$. The gradient of a continuously differentiable function f is denoted by ∇f. For finite ordered sets A and B, we let $A \uplus B$ denote the ordered set consisting of the elements (in their order) of A followed by the elements (in their order) of B; for instance, if $A = (1, 2)$ and $B = (-5, -7)$, then $A \uplus B = (1, 2, -5, -7)$. Suppose that A and B are two ordered sets such that $B \subset A$ as sets, then $A\backslash B$ is the ordered sub-collection in A after deleting the elements of the set B. Finally, given an ordered collection of vectors $(x_i)_{i=1}^n$ in $\mathbb{R}^\nu$ with $\nu \geqslant n$ and equipped with the standard inner product, $\mathrm{Ortho}\big((x_i)_{i=1}^n\big)$ gives the result of Gram-Schmidt orthonormalization of the collection $(x_i)_{i=1}^n$ considered in the order in which they appear i.e., $x_1, x_2, \ldots, x_n$.

2. The ℓ_2-optimal dictionary problem and its solution

Let V denote an $\mathbb{R}^n$-valued random vector defined on some probability space, and having distribution (i.e., Borel probability measure,) μ. We assume that V has finite variance. Let

1. By equivalence of two optimization problems we mean that an optimal solution to either of the problems can be obtained from an optimal solution to the other problem.

R_V denote the support of μ,[2] and let X_V be the smallest subspace of $\mathbb{R}^n$ containing R_V. Our goal is to represent the instances/samples of V with the help of a *dictionary* of vectors:

$$D_K := \big\{d_i \in \mathbb{R}^n \;\big|\; \|d_i\| = 1 \text{ for } i = 1, \ldots, K\big\} \quad \text{with a given } K \geqslant n,$$

in an optimal fashion. A *representation* of an instance v of the random vector V is given by the coefficient vector $\alpha = (\alpha_1 \;\ldots\; \alpha_K)^\top$, such that

$$v = \sum_{i=1}^K \alpha_i d_i. \tag{4}$$

A *reconstruction* of the sample v from the representation α is carried out by taking the linear combination $\sum_{i=1}^K \alpha_i d_i$. We define the *cost* associated with representing v in terms of the coefficient vector α as $\sum_{i=1}^K \alpha_i^2$. Since the dictionary vectors $\{d_i\}_{i=1}^K$ must be able to represent any sample of V, the property that $\mathrm{span}\{d_i\}_{i=1}^K \supset R_V$ is essential. A dictionary $D_K = \{d_i\}_{i=1}^K \subset \mathbb{R}^n$ is said to be *feasible* if $\mathrm{span}\{d_i\}_{i=1}^K \supset R_V$. We denote by $\mathcal{D}_K$ the set of all feasible dictionaries.

For a feasible dictionary $D_K = \{d_i\}_{i=1}^K$, with $m := \dim\big(\mathrm{span}\{d_i\}_{i=1}^K\big)$, and for any $v \in R_V$, the linear equation (4) is satisfied by infinitely many values of α whenever $K > m$. In fact, the solution set of (4) constitutes a $(K-m)$-dimensional affine subspace of $\mathbb{R}^K$. Therefore, in order to represent a given v uniquely, one must define a mechanism of selecting a particular point from this affine subspace, thus making the coefficient vector $\alpha = (\alpha_1 \;\ldots\; \alpha_K)^\top$ a function of v. Let f denote such a function; to wit, $f(v) := \alpha$ is the coefficient vector used to represent the sample v. We call such a map $R_V \ni v \longmapsto f(v) \in \mathbb{R}^K$ a *scheme of representation*. Representation of samples of the random vector V using a dictionary D_K and a scheme f is said to be *proper* if any vector $v \in R_V$ can be uniquely represented and then exactly reconstructed back. It is clear that for proper representation of V with a dictionary D_K consisting of vectors $\{d_i\}_{i=1}^K$, the mapping $R_V \ni v \longmapsto f(v) \in \mathbb{R}^K$ should be an injection that satisfies

$$V = \begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix} f(V) \quad \mu\text{-almost surely.} \tag{5}$$

A scheme f of representation is said to be *feasible* if for some feasible dictionary $D_K := \{d_i\}_{i=1}^K \in \mathcal{D}_K$ the equality $\begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix} f(V) = V$ is satisfied almost surely. We denote by $\mathcal{F}$ the set of all feasible schemes of representation.

Given a scheme f of representation, the (random) cost associated with representing V is given by $\|f(V)\|^2$. The problem of finding an ℓ_2-optimal dictionary can now be posed as:

> Find a pair consisting of a dictionary $D_K^* \in \mathcal{D}_\mathcal{K}$ and a feasible scheme f^* of representation such that the average cost $\mathsf{E}_\mu\big[\|f^*(V)\|^2\big]$ of representation is minimal.

Here the subscript μ indicates the distribution of random vector V with respect to which the expectation is evaluated. In other words, we have the following optimization problem:

$$\begin{aligned} &\underset{D_K, f}{\text{minimize}} && \mathsf{E}_\mu\big[\|f(V)\|^2\big] \\ &\text{subject to} && \begin{cases} D_K \in \mathcal{D}_K, \\ f \in \mathcal{F}. \end{cases} \end{aligned} \tag{6}$$

2. Recall (Parthasarathy, 2005, Theorem 2.1, Definition 2.1, pp. 27-28) that the support of μ is the set of points $z \in \mathbb{R}^n$ such that the μ-measure of every open neighbourhood of z is positive.

The problem given in (6) will be referred to as the ℓ_2-*optimal dictionary* problem. It should be noted that the ℓ_2-optimal dictionary problem is non-convex due to the constraint that the dictionary vectors $\{d_i\}_{i=1}^K$ of a feasible dictionary must be of unit length. Even if we change this constraint to $\{\|d_i\| \leqslant 1\}$ from $\{\|d_i\| = 1\}$, which makes the feasible region of dictionary vectors convex, the set of feasible schemes of representation is not known to be a convex set a priori.

In this article we solve the ℓ_2-optimal dictionary problem given in (6) in two steps:
(Step I) We let $X_V = \mathbb{R}^n$.
(Step II) We let X_V be any proper nontrivial subspace of $\mathbb{R}^n$.[3]
The remainder of this section is devoted to describing Steps I and II by exposing our main results, followed by discussions, a numerical example, and a treatment of the important case of the uniform distribution on the unit sphere of $\mathbb{R}^n$.

2.1 Step I: $X_V = \mathbb{R}^n$

If $X_V = \mathbb{R}^n$, a dictionary of vectors $D_K = \{d_i\}_{i=1}^K \subset \mathbb{R}^n$ is feasible if and only if $\|d_i\| = 1$ for all $i = 1, \ldots, K$, and $\operatorname{span}\{d_i\}_{i=1}^K = \mathbb{R}^n$. Thus, the ℓ_2-optimization problem (6) reduces to:

$$\begin{aligned} &\underset{\{d_i\}_{i=1}^K, f}{\text{minimize}} && \mathsf{E}_\mu\big[\|f(V)\|^2\big] \\ &\text{subject to} && \begin{cases} \|d_i\| = 1 \text{ for all } i = 1, \ldots, K, \\ \operatorname{span}\{d_i\}_{i=1}^K = \mathbb{R}^n, \\ \begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix} f(V) = V \quad \mu\text{-almost surely.} \end{cases} \end{aligned} \tag{7}$$

Let $\Sigma_V \coloneqq \mathsf{E}_\mu[VV^\top]$. We claim that Σ_V is positive definite. Indeed, if not, then there exists a nonzero vector $x \in \mathbb{R}^n$ such that $x^\top V = 0$ almost surely, which contradicts the assumption that $X_V = \mathbb{R}^n$.

Existence and characterization of the optimal solutions to (7) is done by the following:

Theorem 1. *Consider the optimization problem* (7)*, and let* $\Sigma_V \coloneqq \mathsf{E}_\mu\big[VV^\top\big]$.
- (7) *admits an optimal solution.*
- *The optimal value corresponding to* (7) *is* $\dfrac{\big(\operatorname{tr}(\Sigma_V^{1/2})\big)^2}{K}$.
- *Optimal solutions of* (7) *are characterized by:*
  - *a dictionary* $D_K^* = \{d_i^*\}_{i=1}^K$ *that is feasible for* (7) *and that satisfies*

$$\sum_{i=1}^K d_i^* d_i^{*\top} = M^* \coloneqq \frac{K}{\operatorname{tr}\big(\Sigma_V^{1/2}\big)}\, \Sigma_V^{1/2}, \tag{8}$$

  and
  - *a scheme* $f_{D_K^*}^*(v) \coloneqq \begin{pmatrix} d_1^* & d_2^* & \cdots & d_K^* \end{pmatrix}^+ v$.

Moreover, all optimal dictionary-scheme pairs can be obtained via the procedure described in Algorithm 2 on p. 22.

3. The trivial case of $X_V = \{0\}$ is discarded because then there is nothing to prove; we therefore limit ourselves to 'nontrivial' subspaces of $\mathbb{R}^n$.

2.2 Step II: X_V is a strict nontrivial subspace of $\mathbb{R}^n$

Let X_V be any proper nontrivial subspace of $\mathbb{R}^n$. In this situation it is reasonable to expect that no optimal dictionary that solves (6) contains elements that do not belong to X_V. That this indeed happens is the assertion of the following Lemma, whose proof is provided in Section 4:

Lemma 2. *Optimal solutions, if any exists, of problem* (6) *are such that the optimal dictionary vectors* $\{d_i^*\}_{i=1}^K$ *satisfy* $d_i^* \in X_V$ *for all* $i = 1, \ldots, K$.

Lemma 2 guarantees that if the problem (6) admits a solution, then the corresponding optimal dictionary vectors must be elements of X_V. This means that it is enough to optimize over dictionaries with their elements in X_V instead of the whole of $\mathbb{R}^n$. Therefore, the constraint $\operatorname{span}\{d_i\}_{i=1}^K \supset R_V$ can be equivalently stated as $\operatorname{span}\{d_i\}_{i=1}^K = X_V$.

Let the dimension of X_V be m with $m < n$, and let $\mathcal{B} = \{b_i\}_{i=1}^m$ be a basis for X_V. It should be noted that $X_V = \operatorname{image}(\Sigma_V)$, and therefore, a basis of X_V can be obtained by computing a basis of the subspace $\operatorname{image}(\Sigma_V)$. An example of such a basis of X_V is the collection of unit eigenvectors of Σ_V corresponding to its non-zero eigenvalues.

Fix a basis $\mathcal{B} = \{b_i\}_{i=1}^m$ of X_V. Let B be a matrix containing the vectors $\{b_i\}_{i=1}^m$ as its columns:

$$B \coloneqq \begin{pmatrix} b_1 & b_2 & \cdots & b_m \end{pmatrix}.$$

If δ_i is the representation of the dictionary vector d_i in the basis $\mathcal{B}$, i.e., $d_i = B\delta_i$, then the constraints on the family $\{d_i\}_{i=1}^K$ get transformed to the following ones on $\{\delta_i\}_{i=1}^K$:
- $\|d_i\|^2 = 1 \quad \Rightarrow \quad \delta_i^\top \big(B^\top B\big)\delta_i = 1$, and
- $\operatorname{span}\{d_i\}_{i=1}^K \supset R_V \quad \Rightarrow \quad \operatorname{span}\{d_i\}_{i=1}^K = X_V \quad \Rightarrow \quad \operatorname{span}\{\delta_i\}_{i=1}^K = \mathbb{R}^m$.

We define the random vector

$$V_X \coloneqq \big((B^\top B)^{-1} B^\top\big) V.$$

Then V_X is an $\mathbb{R}^m$ valued random vector which is the representation of random vector V in the basis $\mathcal{B}$. For every scheme f that is feasible for (6), let us define an associated scheme for representing samples of the random vector V_X by

$$\mathbb{R}^m \ni v \longmapsto f_X(v) \coloneqq f(Bv) \in \mathbb{R}^K.$$

The conditions on feasibility of f in (6) imply that the scheme f_X is feasible if for a feasible dictionary of vectors $\{\delta_i\}_{i=1}^K$,

$$\begin{pmatrix} \delta_1 & \delta_2 & \cdots & \delta_K \end{pmatrix} f_X(V_X) = V_X \quad \mu\text{-almost surely.}$$

In other words, in contrast to the problem (6), where the optimization is carried out over vectors in $\mathbb{R}^n$, we can equivalently consider the same problem in $\mathbb{R}^m$, but with the following modified constraints:

$$\begin{aligned} &\underset{\{\delta_i\}_{i=1}^K, f_X}{\text{minimize}} && \mathsf{E}_\mu\big[\|f_X(V_X)\|^2\big] \\ &\text{subject to} && \begin{cases} \delta_i^\top \big(B^\top B\big)\delta_i = 1 \text{ for all } i = 1, \ldots, K, \\ \operatorname{span}\{\delta_i\}_{i=1}^K = \mathbb{R}^m, \\ \begin{pmatrix} \delta_1 & \delta_2 & \cdots & \delta_K \end{pmatrix} f_X(V_X) = V_X \quad \mu\text{-almost surely.} \end{cases} \end{aligned} \tag{9}$$

In relation to the problem (9) let us define the following quantities

$$\begin{cases} \Sigma_V \coloneqq \mathsf{E}_\mu[VV^\top] \\ \Sigma \coloneqq (B^\top B)^{-1/2}\big(B^\top \Sigma_V B\big)(B^\top B)^{-1/2} \\ H^* \coloneqq \dfrac{K}{\operatorname{tr}\big(\Sigma^{1/2}\big)}\big((B^\top B)^{-1/2}\Sigma^{1/2}(B^\top B)^{-1/2}\big). \end{cases} \tag{10}$$

Since the support of V_X is m-dimensional, we conclude from previous discussion that $\Sigma_{V_X} \coloneqq \mathsf{E}_\mu\big[V_X V_X^\top\big]$ is positive definite. Since $\Sigma = (B^\top B)^{1/2}\Sigma_{V_X}(B^\top B)^{1/2}$, it follows that Σ is positive definite, which in turn implies that H^* is positive definite.

To summarize, an ℓ_2-optimal dictionary-scheme pair that solves the optimization problem (6) is equivalently obtained from an optimal solution of the problem (9), and is characterized by the following:

Theorem 3. *Consider the optimization problem* (9)*.*
- ∘ (9) *admits an optimal solution.*
- ∘ *The optimal value corresponding to* (9) *is* $\dfrac{\big(\operatorname{tr}(\Sigma^{1/2})\big)^2}{K}$.
- ∘ *Optimal solutions of* (9) *are characterized by:*
  - ▻ *a dictionary* $D_K^* = \{\delta_i^*\}_{i=1}^K$ *that is feasible for* (9) *and that satisfies*

$$\sum_{i=1}^K \delta_i^* \delta_i^{*\top} = H^*, \tag{11}$$

  and
  - ▻ *a scheme* $f_X^*(u) \coloneqq \begin{pmatrix}\delta_1^* & \delta_2^* & \cdots & \delta_K^*\end{pmatrix}^+ u$.

Consequently, an optimal solution of the ℓ_2-optimal dictionary problem (6) *consisting of an ℓ_2-optimal dictionary-scheme pair is given by*
- ∘ *A collection of vectors* $\{d_i^*\}_{i=1}^K$ *defined as* $d_i^* \coloneqq B\delta_i^*$ *for* $i = 1, 2, \ldots, K$*, and*
- ∘ *the scheme* $f^*(v) \coloneqq \begin{pmatrix}d_1^* & d_2^* & \cdots & d_K^*\end{pmatrix}^+ v$.

Moreover, all optimal dictionary-scheme pairs can be obtained via the procedure given in Algorithm 3 on p. 26.

2.3 Discussion and a numerical example

Remark 4. The problem (6) does not a priori hypothesize an affine/linear structure of candidate schemes. The fact that linear schemes are optimal in (6) is one of the crucial assertions of both Theorem 1 and Theorem 3.

Remark 5. Algorithmic computation of an ℓ_2-optimal dictionary relies on the second moment Σ_V of the random vector V. Complete knowledge of the distribution μ is, therefore, unnecessary. This is an advantage since in practical situations, learning/estimating Σ_V from data is comparatively less demanding than getting a description of the distribution μ itself.

Remark 6. Let $M \in \mathbb{S}_+^{n\times n}$ be such that $\operatorname{image}(M) = X_V$, let $\mathcal{B} = \{b_i\}_{i=1}^m$ be a basis for X_V evaluated as a basis for $\operatorname{image}(M)$. Let

$$B \coloneqq \begin{pmatrix}b_1 & b_2 & \cdots & b_m\end{pmatrix}$$

$$\Sigma(M) \coloneqq (B^\top B)^{-1/2}\big(B^\top M B\big)(B^\top B)^{-1/2}$$
$$H(M) \coloneqq \frac{K}{\operatorname{tr}\Big(\big(\Sigma(M)\big)^{1/2}\Big)}\big((B^\top B)^{-1/2}\big(\Sigma(M)\big)^{1/2}(B^\top B)^{-1/2}\big).$$

Suppose that $\{d_i\}_{i=1}^K$ and $f(\cdot)$ are the dictionary and the scheme obtained using the procedure given in Algorithm 3 using M and K as inputs. By simplifying the pseudo-inverse $\begin{pmatrix}d_1 & d_2 & \cdots & d_K\end{pmatrix}^+$ in $f(\cdot)$, the average cost $J(M)$ of representing V using the scheme $f(\cdot)$ turns out to be

$$\begin{aligned} J(M) &= \mathsf{E}_\mu\Big[V^\top B(B^\top B)^{-1}\big(H(M)\big)^{-1}(B^\top B)^{-1}B^\top V\Big] \\ &= \operatorname{tr}\Big(\big(H(M)\big)^{-1}(B^\top B)^{-1}B^\top \Sigma_V B(B^\top B)^{-1}\Big) \\ &= \operatorname{tr}\Big(\big(H(M)\big)^{-1}(B^\top B)^{-1/2}\,\Sigma\,(B^\top B)^{-1/2}\Big) \\ &= \operatorname{tr}\Big((B^\top B)^{-1/2}\big(H(M)\big)^{-1}(B^\top B)^{-1/2}\,\Sigma\Big) \\ &= \frac{1}{K}\,\operatorname{tr}\Big(\big(\Sigma(M)\big)^{1/2}\Big)\,\operatorname{tr}\Big(\big(\Sigma(M)\big)^{-1/2}\Sigma\Big). \end{aligned} \tag{12}$$

Let $S \coloneqq \big\{T \in \mathbb{S}_+^{n\times n} \;\big|\; \operatorname{image}(T) = X_V\big\}$. Since the sequence of maps

$$\begin{aligned} S \ni T &\longmapsto \Sigma(T) \in \mathbb{S}_{++}^{m\times m}, \\ \mathbb{S}_{++}^{m\times m} \ni T &\longmapsto T^{1/2} \in \mathbb{S}_{++}^{m\times m}, \\ \mathbb{S}_{++}^{m\times m} \ni T &\longmapsto T^{-1}\Sigma \in \mathbb{S}_{++}^{m\times m}, \\ \mathbb{S}_{+}^{m\times m} \ni T &\longmapsto \operatorname{tr}(T) \in \mathbb{R}, \end{aligned}$$

are, evidently, continuous, it follows at once that the map $S \ni M \longmapsto J(M) \in \mathbb{R}$ is also continuous. If $\widehat{\Sigma}_V$ denotes the estimated second moment of V, and the estimation is carried out with a large enough number of samples of V, with probability one we have $\operatorname{image}(\widehat{\Sigma}_V) = X_V$. Therefore, by continuity of $M \longmapsto J(M)$, we see at once that

$$J(\widehat{\Sigma}_V) \xrightarrow[\widehat{\Sigma}_V \longrightarrow \Sigma_V]{} J(\Sigma_V) = \frac{\big(\operatorname{tr}(\Sigma^{1/2})\big)^2}{K}.$$

Remark 7. The optimal average cost of representation of a random vector V is inversely proportional to the size K of the optimal dictionary, as is evident from the optimal costs in Theorems 1 and 3. To wit, the optimal average cost of representation decreases monotonically with K, which is expected.

Remark 8. ℓ_2-optimal dictionaries for representing a random vector V are also optimal for representing any scalar multiple αV of V for any $0 \neq \alpha \in \mathbb{R}$. Indeed, it is clear that H^* defined in (10) is invariant under nonzero scalar multiplications of V. Therefore, ℓ_2-optimal dictionaries are also invariant under nonzero scalar multiplications of the random vector V. This fact also follows from the observation made in Remark 4.

Remark 9. An ℓ_2-optimal dictionary as characterized by Theorem 3 appears there in the form of what is known as a *rank*-1 *decomposition* of the positive definite matrix H^*. Elements

of the theory of rank-1 decompositions of positive definite matrices is discussed below in Section 3. This particular decomposition plays a crucial rôle in transforming the search space of the ℓ_2-optimal dictionary problem (7) from the set of dictionaries to the set of symmetric positive definite matrices with real entries, and translating the non-convex ℓ_2-optimal dictionary problem into a tractable convex one.

Remark 10. All ℓ_2-optimal dictionaries are unique upto rank-1 decompositions of a unique positive definite matrix that is obtained from the second moment $\mathsf{E}[VV^\top]$ of the random vector V. That is, for a given random vector whose samples are to be optimally represented, every ℓ_2-optimal dictionary is obtained from a rank-1 decomposition of a unique positive definite matrix.

Remark 11. Looking ahead at Algorithm 3, it becomes evident that non-uniqueness of optimal dictionaries can be attributed to the non-uniqueness in the selection of C in Step 5 of Algorithm 3, and the element of choice associated to the selection of p_j and p_k in Step 2 of Algorithm 1. The number of optimal solutions may be infinite depending on the distribution of the random vector V. For instance, if V is uniformly distributed over the unit sphere of $\mathbb{R}^n$ and $K = n$, then the elements in an ℓ_2-optimal dictionary form an orthonormal basis of $\mathbb{R}^n$. (The special case of uniform distribution of V over spheres is discussed in Section 2.4.) Of course, there are infinitely many orthonormal bases of $\mathbb{R}^n$ for $n \geqslant 2$.

Remark 12. From Algorithm 3 on p. 26 we can infer that by calculating the matrix B there, consisting of the eigenvectors of Σ_V corresponding to its non-zero eigenvalues, the computations of $(B^\top B)^{-1/2}$, $\Sigma^{1/2}$, and C in the decomposition given in Step 5 become straightforward. Therefore, the chief computational load in Algorithm 3 consists of eigen-decomposition of Σ_V and that in Algorithm 1 (in Step 6), both of which can be performed in polynomial time.

Example 1. Let $V = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}$ be a random vector taking values in $\mathbb{R}^2$, with V_1 and V_2 being independent random variables. Let the density functions of V_1 and V_2 be

$$\rho_{V_1}(v) = 2(v-1)\mathbb{1}_{[1,2]}(v) \quad \text{and} \quad \rho_{V_2}(v) = 2(2-v)\mathbb{1}_{[1,2]}(v),$$

respectively. The support of V is, therefore, the square $[1,2] \times [1,2]$. Elementary calculations lead to $\Sigma_V := \mathsf{E}_\rho[VV^\top] = \begin{pmatrix} 17/6 & 20/9 \\ 20/9 & 11/6 \end{pmatrix}$. We employed the procedure described in Algorithm 2 for the given matrix Σ_V and $K = 3$ in MATLAB. An optimal dictionary $\{y_1^*, y_2^*, y_3^*\}$ was obtained, with

$$y_1^* = \begin{pmatrix} 0.9789 \\ 0.2045 \end{pmatrix}, \quad y_2^* = \begin{pmatrix} 0.6792 \\ 0.7339 \end{pmatrix}, \quad y_3^* = \begin{pmatrix} 0.5870 \\ 0.8096 \end{pmatrix};$$

the optimum value of the objective function was reported to be 1.8930. This collection $\{y_i^*\}_{i=1}^3$ of optimal vectors are marked with crosses on the circumference of the unit circle shown in Figure 2. A second optimal dictionary $\{z_1^*, z_2^*, z_3^*\}$ was obtained, also using Algorithm 2, with dictionary vectors

$$z_1^* = \begin{pmatrix} 0.4214 \\ 0.9069 \end{pmatrix}, \quad z_2^* = \begin{pmatrix} 0.9284 \\ 0.3717 \end{pmatrix}, \quad z_3^* = \begin{pmatrix} 0.8513 \\ 0.5247 \end{pmatrix},$$

with an identical optimal value as in the former case. The vectors $\{z_i^*\}_{i=1}^3$ are marked with dark circles on the circumference of the unit circle in Figure 2.

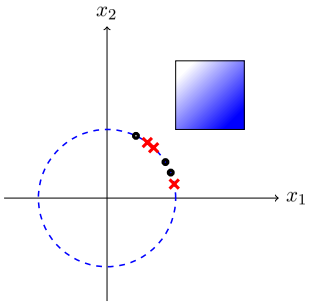


Figure 2: The two optimal dictionaries in Example 1.

It is expected that the optimal dictionary vectors are concentrated towards the bottom right corner of the support $[1,2] \times [1,2]$ (the region with strong shading in figure 2). In the optimal solution $\{z_i^*\}_{i=1}^3$, two vectors z_2^* and z_3^* point to the region where density of V is concentrated the most. Also, for the solution $\{y_i^*\}_{i=1}^3$, two vectors y_2^* and y_3^* are oriented towards the center of the square $[1,2] \times [1,2]$, with the remaining vector pointing towards the region of higher density. These results correlate positively with what may be expected out of ℓ_2-optimal dictionaries.

2.4 Uniform distribution over the unit sphere

We shall test our results on the important case of μ being the uniform distribution on the unit sphere. Note that due to (rigid) rotational symmetry of the distribution, it follows that rigid rotations of optimal dictionaries in this case are also optimal.

Let us consider a dictionary consisting of (unit) vectors that are 'close' to each other, i.e., the inner product between any two elements of the dictionary is close to 1. It is quite evident that such a dictionary is not optimal for representing uniformly distributed samples due to the fact that samples of V that are almost orthogonal to the dictionary vectors carry equal priority as any other vector but require large coefficients for their representation. It is, therefore, more natural to search for dictionaries in which the constituent vectors are 'maximally spaced out'.

Several examples of collections of vectors that are 'maximally spaced out' may be found in (Benedetto and Fickus, 2003, Section 4). Collections of vectors that are maximally far apart from each other are known to attain 'equilibria' under the actions of different kinds of forces defined and explained in (Benedetto and Fickus, 2003, Section 4) and (Saff and Kuijlaars, 1997, p. 6). Such collections of vectors are generalized by the ideas of *tight frames* as explained in (Benedetto and Fickus, 2003); see also (Christensen, 2016; Daubechies et al., 1986; Benedetto and Fickus, 2003; Zimmermann, 2001) for related information.

We recall here some standard definitions for completeness and to provide the necessary substratum for our next result. Let n, K be positive integers such that $K \geqslant n$. We say that a collection of vectors $\{x_i\}_{i=1}^K$ is a *frame* for $\mathbb{R}^n$ if there exist some constants $c, C > 0$ such that

$$c\,\|x\|^2 \leqslant \sum_{i=1}^{K} \langle x_i, x\rangle^2 \leqslant C\,\|x\|^2 \quad \text{for all } x \in \mathbb{R}^n.$$

We say that a frame $\{x_i\}_{i=1}^K \subset \mathbb{R}^n$ is *tight* if $c = C$. In addition, if $\{x_i\}_{i=1}^K \subset \mathbb{R}^n$ is a tight frame and $\|x_i\| = 1$ for all $i = 1, 2, \ldots, K$, we say that the collection $\{x_i\}_{i=1}^K$ is a *c-unit norm tight frame* (a c-UNTF).

We have the following connection between ℓ_2-optimal dictionaries and UNTFs:

Proposition 13. *A dictionary $D_K = \{d_i\}_{i=1}^K$ is optimal for representing samples of a random vector V that is uniformly distributed over the surface of the unit sphere of $\mathbb{R}^n$ if and only if the collection $\{d_i\}_{i=1}^K$ of vectors constitute a $\frac{K}{n}$-UNTF.*

Proof If V is uniformly distributed over the unit sphere, we have $\Sigma_V = \mathsf{E}[VV^\top] = \frac{1}{n} I_n$. According to Theorem 1 the collection $\{d_i\}_{i=1}^K$ is an optimal dictionary if and only if

$$\sum_{i=1}^{K} d_i d_i^\top = \frac{K}{\operatorname{tr}\left(\frac{1}{\sqrt{n}} I_n\right)} \left(\frac{1}{\sqrt{n}} I_n\right) = \frac{K}{n} I_n. \tag{13}$$

Since the family $\{d_i\}_{i=1}^K$ must span $\mathbb{R}^n$ by definition, it is a frame. The *frame operator* for the frame $\{d_i\}_{i=1}^K$ is given by (Benedetto and Fickus, 2003, Section 2)

$$\mathbb{R}^n \ni y \longmapsto S(y) \coloneqq \sum_{i=1}^{K} \langle d_i, y\rangle\, d_i = \left(\sum_{i=1}^{K} d_i d_i^\top\right) y \in \mathbb{R}^n,$$

where $\langle v, w\rangle = v^\top w$ is the standard inner product in $\mathbb{R}^n$. (Benedetto and Fickus, 2003, Theorem 3.1) asserts that a collection of unit norm vectors $\{d_i\}_{i=1}^K$ forms a tight frame in $\mathbb{R}^n$ if and only if the collection is a $\frac{K}{n}$-UNTF. From (Benedetto and Fickus, 2003, Theorem 2.1) it follows that a collection of vectors $\{d_i\}_{i=1}^K$ is a $\frac{K}{n}$-UNTF if and only if

$$S = \sum_{i=1}^{K} d_i d_i^\top = \frac{K}{n} I_n. \tag{14}$$

The assertion follows from (13) and (14). ∎

3. A particular class of rank-1 decompositions of matrices

We collect and establish here some results on the theory of rank-1 decompositions of matrices. While these facts will be needed for our main results, they are also of independent interest.

A standard result in matrix theory (Bhatia, 2009, p. 2) states that a symmetric positive semidefinite matrix with real entries $M \in \mathbb{S}_+^{n\times n}$, can be decomposed as $YY^\top$ for some

$Y \in \mathbb{R}^{n\times r}$, where $r \coloneqq \operatorname{rank}(M)$. Let y_i indicate the i th column of the matrix Y. Then the equality $M = YY^\top$ is equivalent to

$$M = \sum_{i=1}^{r} y_i y_i^\top.$$

More generally for $K \geqslant r$, let

$$\overline{M} \coloneqq \begin{pmatrix} M & O_{n\times(K-r)} \\ O_{(K-r)\times n} & I_{K-r} \end{pmatrix},$$

where O is a zero matrix of order $n \times (K - r)$. If we consider the decomposition of $\overline{M}$ as $\overline{M} = \overline{Y}\,\overline{Y}^\top$ with $\overline{Y} \in \mathbb{R}^{(n+K-r)\times K}$, and indicate by Y the upper $n \times K$ matrix block of $\overline{Y}$, we get $M = YY^\top$. In other words

$$M = \sum_{i=1}^{K} y_i y_i^\top. \tag{15}$$

There are numerous ways of decomposing positive semidefinite matrices; some of them are discussed in (Zhang, 2011, Theorem 7.3). The speciality of a particular decomposition lies in the characteristics exhibited by the vectors y_i's. A particular rank-1 decomposition which we will use to solve the ℓ_2-optimal dictionary problem is the one where for every $M \in \mathbb{S}_+^{n\times n}$ and $K \geqslant r \coloneqq \operatorname{rank}(M)$ there exists a collection of vectors $\{y_i\}_{i=1}^K \subset \mathbb{R}^n$ that satisfy

$$M = \sum_{i=1}^{K} y_i y_i^\top \quad \text{and} \quad y_i^\top y_i = \frac{\operatorname{tr}(M)}{K} \quad \text{for all } i = 1, \ldots, K. \tag{16}$$

We are now in a position to present Algorithm 1 and its associated Theorem 14, whose corollaries will give us the needed rank-1 decomposition of (16). We mention that Algorithm 1 is, in principle, similar to Procedure 1 of (Sturm and Zhang, 2003), and in particular, the assertions of Theorem 14 and its corollaries can be obtained by applying (Sturm and Zhang, 2003, Proposition 3 and Corollary 4) via some straightforward modifications. However, we provide the complete proofs here for the sake of completeness.

Theorem 14. *For any matrix $\Lambda \in \mathbb{R}^{n\times n}$ there exists an orthonormal collection $(x_i)_{i=1}^n \subset \mathbb{R}^n$ of vectors satisfying*

$$x_i^\top \Lambda x_i = \frac{\operatorname{tr}(\Lambda)}{n} \quad \textit{for all } i = 1, \ldots, n,$$

Moreover, such a collection can be obtained from Algorithm 1.

Proof First we establish that the collection of vectors $(x_i)_{i=1}^{n-1}$ contained in S_{n-1} (recall that S_{n-1} is generated in the **for** loop in the Algorithm 1,) are orthonormal, and satisfy $x_i^\top \Lambda x_i = \frac{\operatorname{tr}(\Lambda)}{n}$ for $i = 1, \ldots, n-1$. We shall prove this by induction on i.

The induction base: For $i = 1$, we have $P_1 = (e_1, e_2, \ldots, e_n)$. Since $\sum_{m=1}^n e_m^\top \Lambda e_m = \operatorname{tr}(\Lambda)$, vectors $p_j, p_k \in P_1$ exist such that $p_j^\top \Lambda p_j \leqslant \frac{\operatorname{tr}(\Lambda)}{n} \leqslant p_k^\top \Lambda p_k$. We solve for θ in the equation

$$g_{p_j;p_k}(\theta) \coloneqq \frac{\big((1-\theta)p_j + \theta p_k\big)^\top \Lambda \big((1-\theta)p_j + \theta p_k\big)}{\big((1-\theta)^2 + \theta^2\big)} = \frac{\operatorname{tr}(\Lambda)}{n}. \tag{17}$$

Algorithm 1: Calculation of orthonormal bases à la Theorem 14

Input: A matrix $\Lambda \in \mathbb{R}^{n\times n}$.

Output: An orthonormal collection of vectors $(x_i)_{i=1}^n \subset \mathbb{R}^n$ such that $x_i^\top \Lambda x_i = \frac{\operatorname{tr}(\Lambda)}{n}$ for all $i = 1, \ldots, n$.

Initialize quantities by $S_0 = \varnothing$, $i = 1$.

for i from 1 to $(n-1)$

do

$S_i' = S_{i-1} \uplus (e_1, e_2, \ldots, e_n)$.

$P_i = \operatorname{Ortho}(S_i')\backslash S_{i-1}$.

Find $p_j, p_k \in P_i$ such that $p_j^\top \Lambda p_j \leqslant \frac{\operatorname{tr}(\Lambda)}{n} \leqslant p_k^\top \Lambda p_k$.

Let $\Theta \in [0,1]$ be a solution of the equation (in θ)

$$\big((1-\theta)p_j + \theta p_k\big)^\top \Lambda \big((1-\theta)p_j + \theta p_k\big) = \frac{\operatorname{tr}(\Lambda)}{n}\big((1-\theta)^2 + \theta^2\big)$$

Define $x_i \coloneqq \frac{(1-\Theta)p_j + \Theta p_k}{((1-\Theta)^2+\Theta^2)^{1/2}}$.

Define $S_i \coloneqq S_{i-1} \uplus (x_i)$.

end for loop

$S_n' = S_{n-1} \uplus (e_1, e_2, \ldots, e_n)$.

Output $S_n \coloneqq \operatorname{Ortho}(S_n')$.

We know that a solution exists in $[0,1]$ because for $\theta = 0$ we have

$$g_{p_j;p_k}(0) = \left[\frac{((1-\theta)p_j + \theta p_k)^\top \Lambda((1-\theta)p_j + \theta p_k)}{((1-\theta)^2+\theta^2)}\right]_{\theta=0} = p_j^\top \Lambda p_j \leqslant \frac{\operatorname{tr}(\Lambda)}{n},$$

for $\theta = 1$ we have

$$g_{p_j;p_k}(1) = \left[\frac{((1-\theta)p_j + \theta p_k)^\top \Lambda((1-\theta)p_j + \theta p_k)}{((1-\theta)^2+\theta^2)}\right]_{\theta=1} = p_k^\top \Lambda p_k \geqslant \frac{\operatorname{tr}(\Lambda)}{n},$$

and $g_{p_j;p_k}(\cdot)$ is a continuous function of θ. Let Θ be such a solution. Then, following the notation in Algorithm 1, we have

$$x_1 \coloneqq \frac{(1-\Theta)p_j + \Theta p_k}{\sqrt{(1-\Theta)^2+\Theta^2}}.$$

Since p_j, p_k are elements of P_1, they are orthonormal; therefore,

$$\|x_1\| = \frac{\sqrt{(1-\Theta)^2\|p_j\|^2 + \Theta^2\|p_k\|^2}}{\sqrt{(1-\Theta)^2+\Theta^2}} = 1,$$

and since Θ is a solution of equation (17) we have

$$x_1^\top \Lambda x_1 = \frac{\operatorname{tr}(\Lambda)}{n}.$$

Induction hypothesis: Assume that for some i between 1 and $n-1$ the collection $S_i = (x_\ell)_{\ell=1}^i$ is orthonormal, and satisfies

$$x_\ell^\top \Lambda x_\ell = \frac{\operatorname{tr}(\Lambda)}{n} \quad \text{for all } \ell = 1, \ldots, i.$$

Induction step: In view of the induction hypothesis, we define

$$S_{i+1}' \coloneqq S_i \uplus (e_1, e_2, \ldots, e_n) = (x_1, x_2, \ldots x_i, e_1, e_2, \ldots, e_n),$$

and compute

$$\begin{aligned}\operatorname{Ortho}(S_{i+1}') &= (x_1, x_2, \ldots, x_i, p_1, p_2, \ldots, p_{n-i}),\\ P_{i+1} &= (p_1, p_2, \ldots, p_{n-i})\end{aligned}$$

as in Algorithm 1. Since the collection $(x_\ell)_{\ell=1}^i \uplus (p_\ell)_{\ell=1}^{n-i}$ is an orthonormal basis for $\mathbb{R}^n$, we have

$$\sum_{\ell=1}^{i} x_\ell^\top \Lambda x_\ell + \sum_{\ell=1}^{n-i} p_\ell^\top \Lambda p_\ell = \operatorname{tr}(\Lambda),$$

leading to

$$\sum_{\ell=1}^{n-i} p_\ell^\top \Lambda p_\ell = \frac{(n-i)}{n}\operatorname{tr}(\Lambda).$$

Thus, there exist vectors $p_j, p_k \in P_{i+1}$ such that $p_j^\top \Lambda p_j \leqslant \frac{\operatorname{tr}(\Lambda)}{n} \leqslant p_k^\top \Lambda p_k$. Let us consider the equation

$$g_{p_j,p_k}(\theta) \coloneqq \frac{((1-\theta)p_j + \theta p_k)^\top \Lambda((1-\theta)p_j + \theta p_k)}{((1-\theta)^2+\theta^2)} = \frac{\operatorname{tr}(\Lambda)}{n} \tag{18}$$

in θ. From arguments given in the case of $i = 1$, we know that a solution Θ of (18) exists on $[0,1]$. We define

$$x_{i+1} \coloneqq \frac{(1-\Theta)p_j + \Theta p_k}{\sqrt{(1-\Theta)^2+\Theta^2}}.$$

Since p_j, p_k are orthogonal to the vectors $(x_\ell)_{\ell=1}^i$, so is any linear combination of p_j, p_k. Therefore, x_{i+1} is orthogonal to the vectors $(x_\ell)_{\ell=1}^i$, which, along with the fact that

$$\|x_{i+1}\| = \frac{\sqrt{(1-\Theta)^2\|p_j\|^2 + \Theta^2\|p_k\|^2}}{\sqrt{(1-\Theta)^2+\Theta^2}} = 1,$$

makes the collection $(x_\ell)_{\ell=1}^{i+1}$ orthonormal. Also, since Θ is a solution of (18), we get

$$x_{i+1}^\top \Lambda x_{i+1} = \frac{\operatorname{tr}(\Lambda)}{n}.$$

Therefore, by mathematical induction, we conclude that the collection $(x_i)_{i=1}^{n-1}$ contained in S_{n-1} has the required properties.

Finally, in the 4th and 5th steps of Algorithm 1, we get

$$S'_n = (x_1, x_2, \ldots, x_{n-1}, e_1, e_2, \ldots, e_n),$$

and

$$\mathrm{Ortho}(S'_n) = (x_1, x_2, \ldots, x_{n-1}, x_n).$$

By construction, $(x_\ell)_{\ell=1}^n$ is an orthonormal collection, implying that $\sum_{i=1}^n x_i^\top \Lambda x_i = \mathrm{tr}(\Lambda)$. In turn, this leads to

$$\begin{aligned} x_n^\top \Lambda x_n &= \sum_{i=1}^{n} x_i^\top \Lambda x_i - \sum_{i=1}^{n-1} x_i^\top \Lambda x_i \\ &= \mathrm{tr}(\Lambda) - \left(\frac{n-1}{n}\right)\mathrm{tr}(\Lambda) \\ &= \frac{\mathrm{tr}(\Lambda)}{n}. \end{aligned}$$

Thus, Algorithm 1 yields a collection of orthonormal vectors $(x_i)_{i=1}^n$ such that

$$x_i^\top \Lambda x_i = \frac{\mathrm{tr}(\Lambda)}{n} \quad \text{for all } i = 1, 2, \ldots, n,$$

thereby completing the proof. ∎

Corollary 15 (Rank-1 decomposition)**.** *Let $X \in \mathbb{S}_+^{n\times n}$, define $r \coloneqq \mathrm{rank}(X)$, and let $T \in \mathbb{S}^{n\times n}$. There exists a collection of vectors $\{x_i\}_{i=1}^r \subset \mathbb{R}^n$ such that*

$$X = \sum_{j=1}^{r} x_j x_j^\top, \quad \textit{and} \quad x_i^\top T x_i = \frac{1}{r}\,\mathrm{tr}(XT) \ \textit{for all } i = 1, \ldots, r.$$

Proof We know (Bhatia, 2009, p. 2) that any symmetric positive semidefinite matrix X with real entries and of rank r can be decomposed as $CC^\top$ where $C \in \mathbb{R}^{n\times r}$. Let us define $\Lambda \in \mathbb{R}^{r\times r}$ as $\Lambda \coloneqq C^\top T C$. According to Theorem 14 a collection of orthonormal vectors $\{y_i\}_{i=1}^r \subset \mathbb{R}^r$ can be obtained such that

$$y_i^\top C^\top T\ C y_i = y_i^\top \Lambda y_i = \frac{\mathrm{tr}(\Lambda)}{r}.$$

We define a collection $\{x_i\}_{i=1}^r \subset \mathbb{R}^r$ by $x_i \coloneqq C y_i$ for $i = 1, \ldots, r$. Then

$$\sum_{i=1}^{r} x_i x_i^\top = C\left(\sum_{i=1}^{r} y_i y_i^\top\right)C^\top = C I_r C^\top = X.$$

Moreover, for every $i = 1, \ldots, r$,

$$x_i^\top T x_i = y_i^\top C^\top T C y_i = \frac{\mathrm{tr}(\Lambda)}{r} = \frac{1}{r}\,\mathrm{tr}(C^\top T C) = \frac{1}{r}\,\mathrm{tr}(XT).$$

The assertion follows. ∎

Corollary 15 is generalized slightly by the following one; we shall employ this particular form to solve the ℓ_2-optimal dictionary problem in Theorem 1.

Corollary 16. *Let $M \in \mathbb{S}_+^{n\times n}$ and define $r \coloneqq \mathrm{rank}(M)$. Let $A \in \mathbb{S}^{n\times n}$ and $K \geqslant r$ be given. There exists a collection of vectors $\{y_i\}_{i=1}^K \subset \mathbb{R}^n$ such that*

$$M = \sum_{j=1}^{K} y_j y_j^\top, \quad \textit{and} \quad y_i^\top A y_i = \frac{1}{K}\,\mathrm{tr}(MA) \ \textit{for all } i = 1, \ldots, K. \tag{19}$$

Proof Let us consider the square matrices X, T of order $K + n - r$ in Corollary 15 to be

$$X \coloneqq \begin{pmatrix} M & O_{n\times(K-r)} \\ O_{(K-r)\times n} & I_{K-r} \end{pmatrix} \quad \text{and} \quad T \coloneqq \begin{pmatrix} A & O_{n\times(K-r)} \\ O_{(K-r)\times n} & O_{(K-r)\times(K-r)} \end{pmatrix}.$$

Then $\mathrm{rank}(X) = K$ by construction. Therefore, vectors $\{x_i\}_{i=1}^K \subset \mathbb{R}^{n+K-r}$ exist satisfying the properties in Corollary 15. Let us denote $\mathbb{R}^n \ni y_i \coloneqq \begin{pmatrix} x_{i1} & \ldots & x_{in} \end{pmatrix}^\top$ for $i = 1, \ldots, K$; in other words, y_i is the vector formed by the first n components of x_i. Then

$$\sum_{i=1}^{K} y_i y_i^\top = M,$$

and for any $i = 1, \ldots, K$,

$$y_i^\top A y_i = x_i^\top T x_i = \frac{1}{K}\,\mathrm{tr}(XT) = \frac{1}{K}\,\mathrm{tr}(MA).$$

The assertion follows at once. ∎

4. Proofs of Theorem 1, Lemma 2, and Theorem 3

4.1 Proof of Theorem 1

Proof For a given dictionary $D_K \in \mathcal{D}_K$ of vectors $\{d_i\}_{i=1}^K$ that is feasible for (7), let us define a scheme of representation

$$\mathbb{R}^n \ni v \longmapsto f_{D_K}^*(v) \coloneqq \begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix}^+ v \in \mathbb{R}^K.$$

Quite clearly, $\begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix} f_{D_K}^*(v) = v$ for any $v \in \mathbb{R}^n$ by the definition of the pseudo-inverse because if $\mathrm{span}\{d_i\}_{i=1}^K = \mathbb{R}^n$, then $\begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix}^+ v$ solves the equation $\begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix} x = v$. Therefore,

$$\begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix} f_{D_K}^*(V) = V \ \ \mu\text{-almost surely.}$$

We know that $f_{D_K}^*(v) = \begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix}^+ v$ is the solution of the least squares problem

$$\begin{aligned} &\underset{x\in\mathbb{R}^K}{\text{minimize}} && \|x\|^2 \\ &\text{subject to} && \begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix} x = v. \end{aligned}$$

Therefore, for an arbitrary $f \in \mathcal{F}$ such that $\begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix} f(v) = v$ for all $v \in \mathbb{R}^n$, we must have

$$\left\| f^*_{D_K}(v) \right\|^2 \leqslant \|f(v)\|^2 \quad \text{for all } v \in \mathbb{R}^n.$$

Therefore,

$$\left\| f^*_{D_K}(V) \right\|^2 \leqslant \|f(V)\|^2 \quad \mu\text{-almost surely,}$$

and hence,

$$\mathsf{E}_\mu\left[\left\| f^*_{D_K}(V) \right\|^2\right] \leqslant \mathsf{E}_\mu\left[\|f(V)\|^2\right].$$

Minimizing over all feasible dictionaries and schemes, we get

$$\inf_{D_K \in \mathcal{D}_\mathcal{K}} \mathsf{E}_\mu\left[\left\| f^*_{D_K}(V) \right\|^2\right] \leqslant \inf_{\substack{D_K \in \mathcal{D}_K, \\ f \in \mathcal{F}}} \mathsf{E}_\mu\left[\|f(V)\|^2\right] \tag{20}$$

The problem on the left-hand side of the inequality (20) is

$$\begin{aligned} \underset{\{d_i\}_{i=1}^K}{\text{minimize}} \quad & \mathsf{E}_\mu\left[\left\| f^*_{D_K}(V) \right\|^2\right] \\ \text{subject to} \quad & \begin{cases} \|d_i\| = 1 \text{ for all } i = 1, \ldots, K, \\ \operatorname{span}\{d_i\}_{i=1}^K = \mathbb{R}^n. \end{cases} \end{aligned} \tag{21}$$

From (20) we can conclude that the optimal value, if it exists, of problem (7) is bounded below by the optimal value, if it exists, of the one given in (21). Our strategy is to demonstrate that optimization problem (21) admits a solution, and we shall furnish a feasible solution of (7) that achieves a value of the objective function that is equal to the optimal value of the problem (21). This will solve (7).

Let $D \coloneqq \begin{pmatrix} d_1 & d_2 & \cdots & d_K \end{pmatrix}$. The objective function in (21) can be computed as

$$\begin{aligned} \mathsf{E}_\mu\left[\left\| f^*_{D_K}(V) \right\|^2\right] &= \mathsf{E}_\mu\left[\left\| D^+ V \right\|^2\right] \\ &= \mathsf{E}_\mu\left[V^\top (D^+)^\top D^+ V\right] \\ &= \mathsf{E}_\mu\left[V^\top \left(D^\top (DD^\top)^{-1}\right)^\top \left(D^\top (DD^\top)^{-1}\right) V\right] \\ &= \mathsf{E}_\mu\left[V^\top (DD^\top)^{-1} DD^\top (DD^\top)^{-1} V\right] \\ &= \mathsf{E}_\mu\left[V^\top (DD^\top)^{-1} V\right] \\ &= \mathsf{E}_\mu\left[\operatorname{tr}(V^\top (DD^\top)^{-1} V)\right] \\ &= \mathsf{E}_\mu\left[\operatorname{tr}(VV^\top (DD^\top)^{-1})\right] \\ &= \operatorname{tr}\left(\mathsf{E}_\mu\left[VV^\top\right](DD^\top)^{-1}\right). \end{aligned}$$

Letting $\Sigma_V \coloneqq \mathsf{E}_\mu\left[VV^\top\right]$ and writing $DD^\top = \sum_{i=1}^K d_i d_i^\top$ the optimization problem (21) is rephrased as

$$\begin{aligned} \underset{\{d_i\}_{i=1}^K}{\text{minimize}} \quad & \operatorname{tr}\left(\Sigma_V \left(\sum_{i=1}^K d_i d_i^\top\right)^{-1}\right) \\ \text{subject to} \quad & \begin{cases} \|d_i\| = 1 \text{ for all } i = 1, \ldots, K, \\ \operatorname{span}\{d_i\}_{i=1}^K = \mathbb{R}^n. \end{cases} \end{aligned} \tag{22}$$

Let S be the feasible set for the problem in (22). At first (22) appears to be non-convex. Let us demonstrate that the objective function of (22) is convex in $DD^\top$. We know that whenever Σ_V is a positive definite matrix,

$$\operatorname{tr}(\Sigma_V M^{-1}) = \operatorname{tr}\left(\Sigma_V^{1/2} M^{-1} \Sigma_V^{1/2}\right) = \operatorname{tr}\left(\left(\Sigma_V^{-1/2} M \Sigma_V^{-1/2}\right)^{-1}\right).$$

From (Bhatia, 1997, p. 113 and Exercise V.1.15, p. 117) we know that inversion of a matrix is a *matrix convex* map on the set of positive definite matrices. Therefore, for any $\theta \in [0,1]$ and $M_1, M_2 \in \mathbb{S}^{n\times n}_{++}$ we have

$$\begin{aligned} &\left(\Sigma_V^{-1/2}\big((1-\theta)M_1 + \theta M_2\big)\Sigma_V^{-1/2}\right)^{-1} \\ &\qquad = \left((1-\theta)\left(\Sigma_V^{-1/2} M_1 \Sigma_V^{-1/2}\right) + \theta\left(\Sigma_V^{-1/2} M_2 \Sigma_V^{-1/2}\right)\right)^{-1} \\ &\qquad\qquad \preceq (1-\theta)\left(\Sigma_V^{-1/2} M_1 \Sigma_V^{-1/2}\right)^{-1} + \theta\left(\Sigma_V^{-1/2} M_2 \Sigma_V^{-1/2}\right)^{-1}, \end{aligned} \tag{23}$$

where $A \preceq B$ implies that $B - A$ is positive semidefinite. Since $\operatorname{tr}(\cdot)$ is a *linear functional* over the set of $n \times n$ matrices we have

$$\begin{aligned} \operatorname{tr}\left(\Sigma_V\big((1-\theta)M_1 + \theta M_2\big)^{-1}\right) &= \operatorname{tr}\left(\left(\Sigma_V^{-1/2}\big((1-\theta)M_1 + \theta M_2\big)\Sigma_V^{-1/2}\right)^{-1}\right) \\ &\leqslant (1-\theta)\operatorname{tr}\left(\left(\Sigma_V^{-1/2} M_1 \Sigma_V^{-1/2}\right)^{-1}\right) + \theta \operatorname{tr}\left(\left(\Sigma_V^{-1/2} M_2 \Sigma_V^{-1/2}\right)^{-1}\right) \\ &\leqslant (1-\theta)\operatorname{tr}(\Sigma_V M_1^{-1}) + \theta \operatorname{tr}(\Sigma_V M_2^{-1}). \end{aligned}$$

In other words, the function $M \longmapsto \operatorname{tr}(\Sigma_V M^{-1})$ is a convex function on the set of symmetric and positive definite matrices. Moreover, we know that for a collection $\{d_i\}_{i=1}^K$ that is feasible for (22),

$$\mathcal{D}_K \ni \{d_i\}_{i=1}^K \longmapsto h(d_1, \ldots, d_K) \coloneqq \sum_{i=1}^K d_i d_i^\top$$

maps into the set of positive definite matrices. Therefore, the objective function in (22) is a convex function on image(h). This allows us to translate the feasible set of the optimization problem (22) to the set of matrices M formed by all feasible collections $\{d_i\}_{i=1}^K$, i.e., on $h(\mathcal{D}_K)$.

Let $R \coloneqq \left\{M \in \mathbb{S}^{n\times n}_{++} \;\middle|\; \operatorname{tr}(M) = K\right\}$. On the one hand, from Corollary 16 with $A = I_n$, we know that any symmetric and positive definite matrix $M \in R$ can be decomposed as

$$M = \sum_{i=1}^K d_i d_i^\top \quad \text{with } \|d_i\| = \sqrt{\frac{\operatorname{tr}(M)}{K}} = 1 \text{ for all } i = 1, \ldots, K.$$

The fact that M is positive definite implies that $\operatorname{span}\{d_i\}_{i=1}^K = \mathbb{R}^n$. Therefore, $\{d_i\}_{i=1}^K \in \mathcal{D}_K$ and $M = h(d_1, \ldots, d_K)$, which implies that

$$R \subset h(S). \tag{24}$$

On the other hand, for any collection of vectors $\{d_i\}_{i=1}^K \in \mathcal{D}_K$, we have $h(d_1,\ldots,d_K) = \sum_{i=1}^K d_i d_i^\top \in \mathbb{S}_{++}^{n\times n}$ and $\operatorname{tr}\big(h(d_1,\ldots,d_K)\big) = \sum_{i=1}^K d_i^\top d_i = K$. Therefore, by definition of R,

$$h(S) \subset R. \tag{25}$$

From (24) and (25) we conclude that $h(\mathcal{D}_K) = R$. The optimization problem (22) is, therefore, equivalent to the one where the feasible set is the set of positive definite matrices with trace K, i.e., from (22),

$$\begin{aligned} &\underset{M\in\mathbb{S}_{++}^{n\times n}}{\text{minimize}} && \operatorname{tr}\big(\Sigma_V M^{-1}\big) \\ &\text{subject to} && \operatorname{tr}(M) - K = 0. \end{aligned} \tag{26}$$

The optimization problem in (26) is convex since its objective function is convex (as a function of M) and the feasible region is the intersection of a convex cone $\mathbb{S}_{++}^{n\times n}$ and the affine space $\big\{M \in \mathbb{R}^{n\times n} \;\big|\; \operatorname{tr}(M) - K = 0\big\}$. In the light of (Boyd and Vandenberghe, 2004, p. 244) it follows that (26) can be solved by considering just the first order optimality conditions. These first order optimality conditions are expressed in terms of a Lagrangian

$$L(M,\gamma) \coloneqq \operatorname{tr}(M^{-1}\Sigma_V) + \gamma\big(\operatorname{tr}(M) - K\big),$$

containing a KKT multiplier γ at an optimal point M^* as

$$\begin{aligned} 0 = \nabla_M L(M^*,\gamma) &= \nabla_M \Big(\operatorname{tr}(M^{-1}\Sigma_V) + \gamma\big(\operatorname{tr}(M) - K\big)\Big)\Big|_{M=M^*} \\ &= -\big((M^*)^{-1}\Sigma_V(M^*)^{-1}\big)^\top + \gamma I_n. \end{aligned} \tag{27}$$

But since $M^*, \Sigma_V \in \mathbb{S}_{++}^{n\times n}$, by symmetry it follows that $(M^*)^{-1}\Sigma_V(M^*)^{-1} = \gamma I_n$, leading to

$$\Sigma_V = \gamma (M^*)^2. \tag{28}$$

Since $\Sigma_V \neq O_{n\times n}$, we get $\gamma \neq 0$, and write M^* as

$$M^* = \frac{1}{\sqrt{\gamma}}\Sigma_V^{1/2}.$$

To evaluate γ we use the fact that by construction $K = \operatorname{tr}(M^*) = \frac{1}{\sqrt{\gamma}}\operatorname{tr}\big(\Sigma_V^{1/2}\big)$, which gives

$$\gamma = \left(\frac{\operatorname{tr}\big(\Sigma_V^{1/2}\big)}{K}\right)^2.$$

In other words, the final expression of the optimizer M^* in the problem (26) is

$$M^* = \frac{K}{\operatorname{tr}\big(\Sigma_V^{1/2}\big)}\Sigma_V^{1/2}. \tag{29}$$

It follows that the optimal value of the problem (26) (and therefore of (22)) is $\dfrac{\big(\operatorname{tr}(\Sigma_V^{1/2})\big)^2}{K}$. Therefore, this value must be a lower bound of the optimal value, if it exists, for the problem (7).

Employing Corollary 16 with $A = I_n$, we decompose M^* as

$$M^* = \sum_{i=1}^K d_i^* d_i^{*\top} \quad \text{with } \|d_i^*\| = 1 \text{ for each } i = 1,\ldots,K. \tag{30}$$

Let us consider the dictionary D_K^* consisting of the vectors $\{d_i^*\}_{i=1}^K$ obtained above. Since $X_V = \mathbb{R}^n$, the matrices $\Sigma_V, \Sigma_V^{1/2}$, and M^* are of rank n, and therefore, $\operatorname{span}\{d_i^*\}_{i=1}^K = \mathbb{R}^n$. Along with the fact that $\|d_i^*\| = 1$, we see that the dictionary D_K^* of vectors $\{d_i^*\}_{i=1}^K$ is feasible for the problem (7).

Let us define the scheme

$$\mathbb{R}^n \ni v \longmapsto f_{D_K^*}^*(v) \coloneqq \begin{pmatrix} d_1^* & d_2^* & \cdots & d_K^* \end{pmatrix}^+ v \in \mathbb{R}^K.$$

It is evident that this scheme $f_{D_K^*}^*$ is feasible for (7). But then the objective function in (7) evaluated at $D_K = D_K^*$ and $f = f_{D_K^*}^*$ must be equal to $\dfrac{\big(\operatorname{tr}(\Sigma_V^{1/2})\big)^2}{K}$. Since this particular value is also a lower bound for the optimal value of (7), the problem (7) is solvable. An optimal dictionary-scheme pair is given by

$$\begin{cases} D_K^* = \{d_i^*\}_{i=1}^K \text{ obtained from the decomposition (30), and} \\ \mathbb{R}^n \ni v \longmapsto f^*(v) \coloneqq \begin{pmatrix} d_1^* & d_2^* & \cdots & d_K^* \end{pmatrix}^+ v \in \mathbb{R}^K. \end{cases} \tag{31}$$

The proof is now complete. ∎

We provide the Algorithm 2 that computes optimal dictionary-scheme pairs for the case $X_V = \mathbb{R}^n$. The inputs to the algorithm are the matrix Σ_V and the size K of a dictionary:

Algorithm 2: ℓ_2-optimal dictionary for the case $X_V = \mathbb{R}^n$.

Input: A matrix $\Sigma_V \in \mathbb{S}_{++}^{n\times n}$ and a number $K \geqslant n$.
Output: An ℓ_2-optimal dictionary-scheme pair $\big(\{d_i^*\}_{i=1}^K, f^*\big)$.

1 Define $M_1 \coloneqq \frac{K}{\operatorname{tr}\left(\Sigma_V^{1/2}\right)}\,\Sigma_V^{1/2}$.
2 Define $M_2 \coloneqq \begin{pmatrix} M_1 & O_{n\times(K-n)} \\ O_{(K-n)\times n} & I_{K-n} \end{pmatrix}$, $A \coloneqq \begin{pmatrix} I_n & O_{n\times(K-n)} \\ O_{(K-n)\times n} & O_{(K-n)\times(K-n)} \end{pmatrix}$
3 Compute $C \in \mathbb{R}^{K\times K}$ such that $M_2 = CC^\top$.
4 Define $\Lambda \in \mathbb{R}^{K\times K}$ by $\Lambda \coloneqq C^\top A C$, and apply Algorithm 1 to get a collection of vectors $\{x_i\}_{i=1}^K \subset \mathbb{R}^K$.
5 Define the collection $\{v_i\}_{i=1}^K \subset \mathbb{R}^K$ by $v_i \coloneqq Cx_i$ for $i = 1,\ldots,K$.
6 Define the ℓ_2-optimal dictionary $\{d_i^*\}_{i=1}^K \subset \mathbb{R}^n$ such that the jth component of d_i^* is given by $d_i^*(j) \coloneqq v_i(j)$ for $j = 1,\ldots,n$ and for every $i = 1,\ldots,K$.
7 Define the optimal scheme $\mathbb{R}^n \ni v \longmapsto f^*(v) \coloneqq \begin{pmatrix} d_1^* & d_2^* & \cdots & d_K^* \end{pmatrix}^+ v$.

4.2 Proof of Lemma 2

Proof We argue by contradiction. Suppose that the assertion of the Lemma is false. If we denote by x_i the orthogonal projection of d_i on X_V and by y_i the orthogonal projection of

d_i on the orthogonal complement of X_V, we must have $\|x_i\| < 1$ for at least one value of i. If f is an optimal scheme of representation, feasibility of f gives, for any $v \in R_V$,

$$\begin{aligned} v &= \sum_{i=1}^{K} d_i f_i(v) = \left(\sum_{i=1}^{K} x_i f_i(v)\right) + \left(\sum_{i=1}^{K} y_i f_i(v)\right) \\ &= \sum_{\substack{i=1,\\ \|x_i\|\neq 0}}^{K} x_i f_i(v) + 0. \end{aligned} \tag{32}$$

Fix a unit vector $x \in X_V$, and define a dictionary $\{d_i^*\}_{k=1}^K$ by

$$d_i^* := \begin{cases} \dfrac{x_i}{\|x_i\|} & \text{if } \|x_i\| \neq 0, \\ x & \text{otherwise.} \end{cases}$$

Then clearly

$$\operatorname{span}\{d_i^*\}_{i=1}^K \supset \operatorname{span}\{x_i\}_{i=1}^K \supset R_V \quad \text{and} \quad \|d_i^*\| = 1 \text{ for all } i = 1, \ldots, K.$$

In other words, the dictionary of vectors $\{d_i^*\}_{i=1}^K$ is feasible for the problem (6). Let us now define a scheme f^* by

$$\mathbb{R}^n \ni v \longmapsto f^*(v) := \operatorname{diag}\{\|x_1\|, \|x_2\|, \ldots, \|x_K\|\} f(v) \in \mathbb{R}^K.$$

For any $v \in R_V$, using the dictionary consisting of vectors $\{d_i^*\}_{i=1}^K$ we get

$$\sum_{i=1}^{K} d_i^* f_i^*(v) = \sum_{i=1}^{K} d_i^* \|x_i\| f_i(v) = \sum_{\substack{i=1,\\ \|x_i\|\neq 0}}^{K} \frac{x_i}{\|x_i\|} \|x_i\| f_i(v) = v, \tag{33}$$

where the last equality follows from (32). Thus, $f^*(\cdot)$ along with the dictionary of vectors $\{d_i^*\}_{i=1}^K$ is feasible for problem (6). But for any $v \in R_V$ we have

$$\|f^*(v)\|^2 = \sum_{i=1}^{K} \big(f_i^*(v)\big)^2 = \sum_{i=1}^{K} \|x_i\|^2 \big(f_i(v)\big)^2 < \sum_{i=1}^{K} \big(f_i(v)\big)^2 = \|f(v)\|^2,$$

where the inequality is due to the fact that $\|x_i\| < 1$ for at least one i. This contradicts the assumption that the pair $\{d_i\}_{i=1}^K$ along with the scheme f is optimal for (6). ■

4.3 Proof of Theorem 3

Proof The problem (9) is similar to problem (7) except for the first constraint. In (7) we optimize over vectors taking values on the surface of the unit sphere, whereas in (9) we optimize over vectors taking values on the surface of the ellipsoid $\{x \in \mathbb{R}^m \mid x^\top (B^\top B) x = 1\}$. Following the arguments in the proof of Theorem 1 till (22), one can conclude that the

optimal value, if it exists, of problem (9) is bounded below by the optimal value, if it exists, of the problem

$$\begin{aligned} \underset{\{\delta_i\}_{i=1}^K}{\text{minimize}} \quad & \operatorname{tr}\left(\Sigma_{V_X}\left(\sum_{i=1}^{K} \delta_i \delta_i^\top\right)^{-1}\right) \\ \text{subject to} \quad & \begin{cases} \delta_i^\top (B^\top B) \delta_i = 1 \text{ for all } i = 1, 2, \ldots, K, \\ \operatorname{span}\{\delta_i\}_{i=1}^K = \mathbb{R}^m, \end{cases} \end{aligned} \tag{34}$$

where $\Sigma_{V_X} := \mathsf{E}_\mu\big[V_X V_X^\top\big] = \big((B^\top B)^{-1} B^\top\big)\, \mathsf{E}_\mu\big[V V^\top\big] \big((B^\top B)^{-1} B^\top\big)^\top$.

Let us define:

- S to be the feasible region of the problem (34),
- $R := \big\{H \in \mathbb{S}_{++}^{m\times m} \;\big|\; \operatorname{tr}\big(H(B^\top B)\big) = K\big\}$, and
- the map $\big(\mathbb{R}^m\big)^K \ni (\delta_1, \delta_2, \ldots, \delta_K) \longmapsto h(\delta_1, \delta_2, \ldots, \delta_K) := \sum_{i=1}^K \delta_i \delta_i^\top \in \mathbb{S}_+^{m\times m}$.

From Corollary 16 we see that for every $H \in R$ there exists a collection of vectors $\{\delta_i\}_{i=1}^K$ such that

$$\sum_{i=1}^{K} \delta_i \delta_i^\top = H \quad \text{and} \quad \delta_i^\top (B^\top B) \delta_i = \frac{\operatorname{tr}\big(H(B^\top B)\big)}{K} = 1,$$

which, along with the fact that $\operatorname{rank}(H) = m \Rightarrow \operatorname{span}\{\delta_i\}_{i=1}^K = \mathbb{R}^m$, imply that

$$R \subset h(S). \tag{35}$$

Moreover, for any collection $\{\delta_i\}_{i=1}^K \in S$, we have

$$\operatorname{tr}\big(h(\delta_1, \delta_2, \ldots, \delta_K)(B^\top B)\big) = \sum_{i=1}^{K} \delta_i^\top (B^\top B) \delta_i = K \quad \text{and} \quad h(\delta_1, \delta_2, \ldots, \delta_K) \in \mathbb{S}_{++}^{m\times m},$$

which implies that

$$h(S) \subset R. \tag{36}$$

From (35) and (36) we conclude that $R = h(S)$. In other words, instead of optimizing over the feasible collection of vectors in S in (34), one can equivalently optimize over the set of symmetric positive definite matrices in R. This consideration leads us to the problem:

$$\begin{aligned} \underset{H \in \mathbb{S}_{++}^{m\times m}}{\text{minimize}} \quad & \operatorname{tr}\big(\Sigma_{V_X} H^{-1}\big) \\ \text{subject to} \quad & \operatorname{tr}\big(H(B^\top B)\big) - K = 0. \end{aligned} \tag{37}$$

Letting $M := (B^\top B)^{1/2} H (B^\top B)^{1/2}$, we write the optimization problem (37) with M as the variable instead of H. Due to this change of variables, the constraint and the objective function become

$$\operatorname{tr}\big(H(B^\top B)\big) = \operatorname{tr}\big((B^\top B)^{1/2} H (B^\top B)^{1/2}\big) = \operatorname{tr}(M),$$

and

$$\begin{aligned} \operatorname{tr}\big(\Sigma_{V_X} H^{-1}\big) &= \operatorname{tr}\big(\Sigma_{V_X} (B^\top B)^{1/2} M^{-1} (B^\top B)^{1/2}\big) \\ &= \operatorname{tr}\big((B^\top B)^{1/2} \Sigma_{V_X} (B^\top B)^{1/2} M^{-1}\big) \\ &= \operatorname{tr}\big(\Sigma M^{-1}\big), \end{aligned} \tag{38}$$

where

$$\begin{aligned}\Sigma &\coloneqq (B^\top B)^{1/2}\Sigma_{V_X}(B^\top B)^{1/2}\\ &= (B^\top B)^{1/2}\big((B^\top B)^{-1}B^\top\big)\,\mathsf{E}_\mu\big[VV^\top\big]\big((B^\top B)^{-1}B^\top\big)^\top(B^\top B)^{1/2}\\ &= (B^\top B)^{-1/2}\big(B^\top\Sigma_V B\big)(B^\top B)^{-1/2}.\end{aligned} \tag{39}$$

Using (38) we write the problem (37) equivalently as:

$$\begin{aligned}\underset{M\in\mathbb{S}^{n\times n}_{++}}{\text{minimize}}\quad & \operatorname{tr}\big(\Sigma M^{-1}\big)\\ \text{subject to}\quad & \operatorname{tr}(M)-K=0.\end{aligned} \tag{40}$$

The problem (40) is identical to (26), which implies that the problem (40) is solvable, and an optimizer is

$$M^* \coloneqq \frac{K}{\operatorname{tr}\big(\Sigma^{1/2}\big)}\Sigma^{1/2}.$$

Therefore, the problem (37) is solvable, and an optimizer is

$$\begin{aligned}H^* &\coloneqq (B^\top B)^{-1/2}M^*(B^\top B)^{-1/2}\\ &= \frac{K}{\operatorname{tr}\big(\Sigma^{1/2}\big)}\big((B^\top B)^{-1/2}\Sigma^{1/2}(B^\top B)^{-1/2}\big).\end{aligned} \tag{41}$$

From Corollary 16 it follows that there exists a collection $\{\delta_i^*\}_{i=1}^K$ of vectors such that

$$\sum_{i=1}^K \delta_i^*\delta_i^{*\top} = H^* \quad\text{and}\quad \delta_i^{*\top}\big(B^\top B\big)\delta_i^* = \frac{\operatorname{tr}\big(H^*(B^\top B)\big)}{K} = 1.$$

Employing arguments similar to those given in the proof of Theorem 1, we now conclude that the pair

- the collection of vectors $\{\delta_i^*\}_{i=1}^K$, and
- the scheme $f_X^*(u) = \begin{pmatrix}\delta_1^* & \delta_2^* & \cdots & \delta_K^*\end{pmatrix}^+ u$,

is optimal for the problem (9). Using the optimal solution of (9), we define a dictionary-scheme pair as:

$$\begin{cases} d_i^* \coloneqq B\delta_i^* \quad \text{for } i=1,\dots,K,\\ \mathbb{R}^n \ni v \longmapsto f^*(v) \coloneqq f_X^*\Big(\big((B^\top B)^{-1}B^\top\big)v\Big) = \begin{pmatrix}\delta_1^* & \delta_2^* & \cdots & \delta_K^*\end{pmatrix}^+\big((B^\top B)^{-1}B^\top\big)v.\end{cases} \tag{42}$$

It is clear that the pair in (42) is feasible for the problem (6), and that the corresponding objective function evaluates to the optimal value of the problem (34). Therefore, along with the assertion of Lemma 2 we can conclude that the problem (6) is solvable, and in fact an optimal solution is given by (42) with the optimal value of $\frac{\left(\operatorname{tr}(\Sigma^{1/2})\right)^2}{K}$. This completes the proof. ■

As in the case $X_V = \mathbb{R}^n$, we now provide the Algorithm 3 to obtain an optimal dictionary-scheme pair for the general ℓ_2-optimal dictionary problem (6). The algorithm takes the

matrix Σ_V and the size of the dictionary K as its inputs. From Σ_V we extract a matrix $B \in \mathbb{R}^{n\times m}$ containing a set of basis vectors for image(Σ_V) in its columns, these vectors form a basis for X_V.

Algorithm 3: A procedure to obtain ℓ_2-optimal dictionary.

Input: A matrix $\Sigma_V \in \mathbb{S}_+^{n\times n}$ and a number $K \geqslant m \coloneqq \dim(X_V) = \operatorname{rank}(\Sigma_V)$.
Output: An ℓ_2-optimal dictionary-scheme pair $\big(\{y_i^*\}_{i=1}^K, f^*\big)$.

1. Compute a basis $\{b_i\}_{i=1}^m$ for image(Σ_V) and define $B \coloneqq \begin{pmatrix}b_1 & b_2 & \cdots & b_m\end{pmatrix}$.
2. Define $\Sigma \coloneqq (B^\top B)^{-1/2}\big(B^\top\Sigma_V B\big)(B^\top B)^{-1/2}$.
3. Compute $H \coloneqq \frac{K}{\operatorname{tr}\left(\Sigma^{1/2}\right)}\big((B^\top B)^{-1/2}\Sigma^{1/2}(B^\top B)^{-1/2}\big)$.
4. Define $M \coloneqq \begin{pmatrix}H & O_{m\times(K-m)}\\ O_{(K-m)\times m} & I_{K-m}\end{pmatrix}$, $A \coloneqq \begin{pmatrix}B^\top B & O_{m\times(K-m)}\\ O_{(K-m)\times m} & O_{(K-m)\times(K-m)}\end{pmatrix}$
5. Compute $C \in \mathbb{R}^{K\times K}$ such that $M = CC^\top$.
6. Define $\Lambda \in \mathbb{R}^{K\times K}$ by $\Lambda \coloneqq C^\top AC$, and apply Algorithm 1 to get a collection of vectors $\{x_i\}_{i=1}^K \subset \mathbb{R}^K$.
7. Define the collection $\{v_i\}_{i=1}^K \subset \mathbb{R}^K$ as $v_i \coloneqq Cx_i$ for $i = 1,\dots,K$.
8. Define the collection $\{\delta_i^*\}_{i=1}^K \subset \mathbb{R}^m$ such that the jth component of δ_i^* is given by $\delta_i^*(j) \coloneqq v_i(j)$ for $j = 1,\dots,m$ and for every $i = 1,\dots,K$.
9. Define the ℓ_2-optimal dictionary $\{d_i^*\}_{i=1}^K \subset \mathbb{R}^n$ as $d_i^* \coloneqq B\delta_i^*$ for $i = 1,\dots,K$.
10. Define the optimal scheme $\mathbb{R}^n \ni v \longmapsto f^*(v) \coloneqq \begin{pmatrix}d_1^* & d_2^* & \cdots & d_K^*\end{pmatrix}^+ v \in \mathbb{R}^K$.

5. Conclusion and future directions

In this article we have provided an explicit solution of the ℓ_2-optimal dictionary problem in the form of a *rank*-1 *decomposition* of a specific positive definite matrix derived from given data, together with algorithms to compute the corresponding ℓ_2-optimal dictionaries.

The analysis in this article assumes that the second moment of the random vector whose samples are to be represented is known. An online algorithm which estimates the second moment of the random vector and computes the dictionary vectors in parallel is being developed, and will be reported in subsequent articles.

Acknowledgments

We sincerely thank Prof. V. S. Borkar for his valuable pointers on convexity of the ℓ_2-optimal dictionary problem, Prof. K. S. Mallikarjuna Rao for pointing us to the literature on rank-1 decompositions of matrices, and Prof. N. Khaneja for helpful discussions.

References

B. D. O. Anderson and J. B. Moore. *Optimal Control: Linear Quadratic Methods*. Courier Corporation, 2007.

J. J. Benedetto and M. Fickus. Finite Normalized Tight Frames. *Advances in Computational Mathematics*, 18(2-4):357–385, 2003.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1. Athena Scientific Belmont, MA, 1995.

R. Bhatia. *Matrix Analysis*, volume 169. Springer-Verlag, New York, 1997.

R. Bhatia. *Positive Definite Matrices*. Princeton University Press, 2009.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

O. Christensen. *An Introduction to Frames and Riesz Bases*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, [Cham], 2nd edition, 2016.

F. Clarke. *Functional Analysis, Calculus of Variations and Optimal Control*, volume 264. Springer Science & Business Media, 2013.

I. Daubechies, A. Grossmann, and Y. Meyer. Painless Nonorthogonal Expansions. *Journal of Mathematical Physics*, 27(5):1271–1283, 1986.

K. K. Delgado, J. F. Murray, B. D. Rao, K. Engan, T. Lee, and T. J. Sejnowski. Dictionary Learning Algorithms for Sparse Representation. *Neural Computation*, 15(2):349–396, 2003.

D. Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2012.

J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online Dictionary Learning for Sparse Coding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 689–696. ACM, 2009a.

J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, and F. R. Bach. Supervised Dictionary Learning. In *Advances in Neural Information Processing Systems*, pages 1033–1040, 2009b.

S. G. Mallat and Z. Zhang. Matching Pursuits With Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.

P. C. Müller and H. I. Weber. Analysis and optimization of certain qualities of controllability and observability for linear dynamical systems. *Automatica*, 8(3):237–246, 1972.

B. A. Olshausen and D. J. Field. Sparse Coding With an Overcomplete Basis Set: A Strategy Employed by v1? *Vision Research*, 37(23):3311–3325, 1997.

K. R. Parthasarathy. *Probability Measures on Metric Spaces*. AMS Chelsea Publishing, Providence, RI, 2005. Reprint of the 1967 original.

F. Pasqualetti, S. Zampieri, and F. Bullo. Controllability metrics, limitations and algorithms for complex networks. *IEEE Transactions on Control of Network Systems*, 1(1):40–52, 2014.

E. B. Saff and A. Kuijlaars. Distributing Many Points on a Sphere. *The Mathematical Intelligencer*, 19(1):5–11, 1997.

M. R. Sheriff and D. Chatterjee. On a Frame Theoretic Measure of Quality of LTI Systems. *arXiv preprint arXiv:1703.07539*, 2017.

K. Skretting and K. Engan. Recursive Least Squares Dictionary Learning Algorithm. *IEEE Transactions on Signal Processing*, 58(4):2121–2130, 2010.

J. F. Sturm and S. Zhang. On Cones of Nonnegative Quadratic Functions. *Mathematics of Operations Research*, 28(2):246–267, 2003.

I. Tošić and P. Frossard. Dictionary Learning. *Signal Processing Magazine, IEEE*, 28(2): 27–38, 2011.

M. Yaghoobi, T. Blumensath, and M. E. Davies. Dictionary Learning for Sparse Approximations With the Majorization Method. *IEEE Transactions on Signal Processing*, 57(6): 2178–2191, 2009.

M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher Discrimination Dictionary Learning for Sparse Representation. In *IEEE International Conference on Computer Vision (ICCV), 2011*, pages 543–550. IEEE, 2011.

F. Zhang. *Matrix Theory: Basic Results and Techniques*. Springer Science & Business Media, 2011.

G. Zimmermann. Normalized Tight Frames in Finite Dimensions. In *Recent Progress in Multivariate Approximation*, pages 249–252. Springer, 2001.

Journal of Machine Learning Research 18 (2017) 1-37 Submitted 6/16; Revised 7/17; Published 10/17

Computational Limits of A Distributed Algorithm for Smoothing Spline

Zuofeng Shang SHANGZF@IU.EDU
Department of Mathematical Sciences
Indiana University-Purdue University at Indianapolis
Indianapolis, IN 46202, USA

Guang Cheng CHENGG@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47907, USA

Editor: Michael Mahoney

Abstract

In this paper, we explore statistical versus computational trade-off to address a basic question in the application of a distributed algorithm: what is the minimal computational cost in obtaining statistical optimality? In smoothing spline setup, we observe a phase transition phenomenon for the number of deployed machines that ends up being a simple proxy for computing cost. Specifically, a *sharp* upper bound for the number of machines is established: when the number is below this bound, statistical optimality (in terms of nonparametric estimation or testing) is achievable; otherwise, statistical optimality becomes impossible. These sharp bounds partly capture intrinsic computational limits of the distributed algorithm considered in this paper, and turn out to be fully determined by the smoothness of the regression function. We name the asymptotic analysis on such split-and-aggregation estimation/inference as "splitotic" theory. As a side remark, we argue that sample splitting may be viewed as an alternative form of regularization, playing a similar role as smoothing parameter.

Keywords: divide-and-conquer, computational limits, smoothing spline, splitotic theory

1. Introduction

In the parallel computing environment, divide-and-conquer (D&C) method distributes data to multiple machines, and then aggregates local estimates computed from each machine to produce a global one. Such a distributed algorithm often requires a *growing* number of machines in order to process an increasingly large dataset. A practically relevant question is "how many processors do we really need in this parallel computing?" or "shall we allocate all our computational resources in the data analysis?" Such questions are related to the minimal computational cost of this distributed method (which will be defined more precisely later).

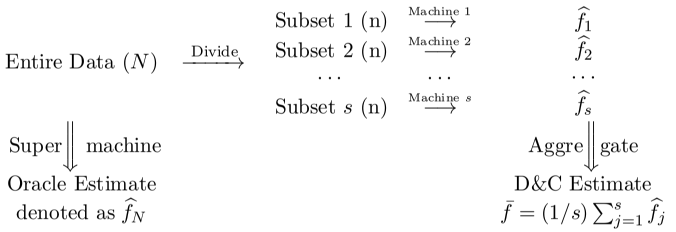
The major goal of this paper is to provide some "theoretical" insights, namely, a splitotic theory, for the above questions from a statistical perspective. Specifically, we consider a classical nonparametric regression setup:

$$y_l = f(l/N) + \epsilon_l,\ l = 0, 1, \ldots, N-1, \tag{1}$$

©2017 Zuofeng Shang and Guang Cheng.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-289.html.

where ϵ_l's are *iid* random errors with $E\{\epsilon_l\} = 0$ and $Var(\epsilon_l) = 1$, in the following distributed algorithm:



We assume that the total sample size is N, the number of machines is s and the size of each sub-sample is n. Hence, $N = s \times n$. Each machine produces an individual smoothing spline estimate $\widehat{f}_j$ to be defined in (3) (Wahba (1990)).

A known property of the above D&C strategy is that it can preserve statistical efficiency for a wide-ranging choice of s (as demonstrated in Figure 1), say $\log s/\log N \in [0, 0.4]$, while largely reducing computational burden as $\log s/\log N$ increases (as demonstrated in Figure 2). An important observation from Figure 1 is that there is an obvious blowup for mean squared errors of $\bar{f}$ when the above ratio is beyond some threshold, e.g, 0.8 for $N = 10000$. Hence, we are interested in knowing whether there exists a critical value of $\log s/\log N$ in theory, beyond which statistical optimality no longer exists. For example, mean squared errors will never achieve minimax optimal lower bound (at rate level) no matter how smoothing parameters are tuned. Such a sharpness result partly captures the computational limit of the particular D&C algorithm considered in this paper, also complementing the upper bound results in Shang and Cheng (2015); Zhang et al. (2015); Zhao et al. (2016).

Our first contribution is to establish a sharp upper bound of s under which $\bar{f}$ achieves the minimax optimal rate $N^{m/(2m+1)}$, where m represents the smoothness of f_0. By "sharp" upper bound, we mean the largest possible upper bound for s to gain statistical optimality. This result is established by directly computing (non-asymptotic) upper and lower bounds of mean squared error of $\bar{f}$. These two bounds hold *uniformly* as s diverges, and thus imply that the rate of mean squared error transits once s reaches the rate $N^{2m/(2m+1)}$, which we call as phase transition in divide-and-conquer estimation. In fact, the choice of smoothing parameter, denoted as λ, also plays a very subtle role in the above phase transition. For example, λ is not necessarily chosen at an optimal level when s attains the above bound as illustrated in Figure 3.

Our second contribution is a sharp upper bound of s under which a simple Wald-type testing method based on $\bar{f}$ is minimax optimal in the sense of Ingster (1993). It is not surprising that our testing method is consistent no matter s is fixed or diverges at any rate. Rather, this sharp bound is entirely determined by analyzing its (non-asymptotic) power. Specifically, we find that our testing method is minimax optimal if and only if s does not grow faster than $N^{(4m-1)/(4m+1)}$. Again, we observe a subtle interplay between s and λ as depicted in Figure 3.

One theoretical insight obtained in our setup is that a more smooth regression function can be optimally estimated or tested at a shorter time. In addition, the above Figure 3

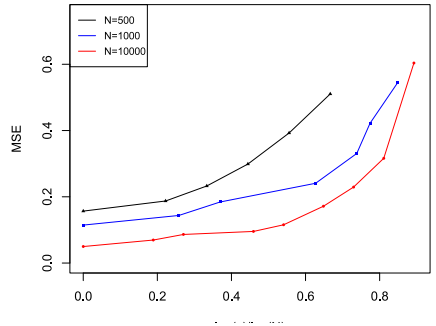


Figure 1: *Mean-square errors (MSE) of $\bar{f}$ based on 500 independent replications under different choices of N and s. The values of MSE stay at low levels for various choice of s with $\log s/\log N \in [0, 0.7]$. True regression function is $f_0(z) = 0.6b_{30,17}(z) + 0.4b_{3,11}(z)$ with b_{a_1,a_2} the density function for $Beta(a_1, a_2)$.*

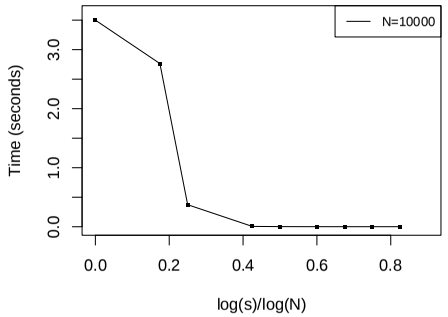


Figure 2: *Computing time of $\bar{f}$ based on a single replication under different choices of s when $N = 10,000$. The larger the s, the smaller the computing time.*

implies that s and λ play an interchangeable role in obtaining statistical optimality. Therefore, we argue that it might be attempting to view sample splitting as an alternative form of regularization, complementing the use of penalization in smoothing spline. In practice,



Figure 3: *Two lines indicate the choices of $s \asymp N^a$ and $\lambda \asymp N^{-b}$, leading to minimax optimal estimation rate (left) and minimax optimal testing rate (right). Whereas (a, b)'s outside these two lines lead to suboptimal rates. Results are based on smoothing spline regression with regularity $m \geq 1$.*

we propose to select λ via a distributed version of generalized cross validation (GCV); see Xu et al. (2017).

In the end, we want to mention that our theoretical results are developed in one-dimensional models under fixed design. This setting allows us to develop proofs based on exact analysis of various Fourier series, coupled with properties of circulant Bernoulli polynomial kernel matrix. The major goal of this work is to provide some theoretical insights in a relatively simple setup, which are useful in extending our results to more general setup such as random or multi-dimensional design. Efforts toward this direction have been made by Liu et al. (2017) who derived upper bounds of s for optimal estimation or testing in various nonparametric models when design is random and multi-dimensional.

2. Smoothing Spline Model

Suppose that we observe samples from model (1). The regression function f is smooth in the sense that it belongs to an m-order ($m \geq 1$) periodic Sobolev space:

$$S^m(\mathbb{I}) = \left\{ \sum_{\nu=1}^{\infty} f_\nu \varphi_\nu(\cdot) : \sum_{\nu=1}^{\infty} f_\nu^2 \gamma_\nu < \infty \right\},$$

where $\mathbb{I} := [0, 1]$ and for $k = 1, 2, \ldots,$

$$\varphi_{2k-1}(t) = \sqrt{2}\cos(2\pi kt), \quad \varphi_{2k}(t) = \sqrt{2}\sin(2\pi kt),$$

$$\gamma_{2k-1} = \gamma_{2k} = (2\pi k)^{2m}.$$

The entire dataset is distributed to each machine in a uniform manner as follows. For $j = 1, \ldots, s$, the jth machine is assigned with samples $(Y_{i,j}, t_{i,j})$, where

$$Y_{i,j} = y_{is-s+j-1} \text{ and } t_{i,j} = \frac{is - s + j - 1}{N}$$

for $i = 1, \ldots, n$. Obviously, $t_{1,j}, \ldots, t_{n,j}$ are evenly spaced points (with a gap $1/n$) across $\mathbb{I}$. At the jth machine, we have the following sub-model:

$$Y_{i,j} = f(t_{i,j}) + \epsilon_{i,j}, \ i = 1, \ldots, n, \tag{2}$$

where $\epsilon_{i,j} = \epsilon_{is-s+j-1}$, and obtain the jth sub-estimate as

$$\widehat{f_j} = \arg\min_{f \in S^m(\mathbb{I})} \ell_{j,n,\lambda}(f).$$

Here, $\ell_{j,n,\lambda}$ represents a penalized square criterion function based on the jth subsample:

$$\ell_{j,n,\lambda}(f) = \frac{1}{2n}\sum_{i=1}^{n}(Y_{i,j} - f(t_{i,j}))^2 + \frac{\lambda}{2}J(f,f), \tag{3}$$

with $\lambda > 0$ being a smoothing parameter and $J(f,g) = \int_{\mathbb{I}} f^{(m)}(t)g^{(m)}(t)dt$[1]

3. Minimax Optimal Estimation

In this section, we investigate the impact of the number of machines on the mean squared error of $\bar{f}$. Specifically, Theorem 3.1 provides an (non-asymptotic) upper bound for this mean squared error, while Theorem 3.2 provides a (non-asymptotic) lower bound. Notably, both bounds hold uniformly as s diverges. From these bounds, we observe an interesting phase transition phenomenon that $\bar{f}$ is minimax optimal if s does not grow faster than $N^{2m/(2m+1)}$ and an optimal $\lambda \asymp N^{-2m/(2m+1)}$ is chosen, but the minimax optimality breaks down if s grows even slightly faster (no matter how λ is chosen). Hence, the upper bound of s is sharp. Moreover, λ does not need to be optimal when this bound is attained. In some sense, a proper sample splitting can compensate a sub-optimal choice of λ.

In this section, we assume that ϵ_l's are *iid* zero-mean random variables with unit variance. Denote mean squared error as

$$\mathrm{MSE}_{f_0}(f) := E_{f_0}\{\|f - f_0\|_2^2\},$$

where $\|f\|_2 = \sqrt{\int_{\mathbb{I}} f(t)^2 dt}$. For simplicity, we write E_{f_0} as E later. Define $h = \lambda^{1/(2m)}$.

Theorem 3.1 *(Upper Bounds of Variance and Squared Bias) Suppose $h > 0$, and N is divisible by n. Then there exist absolute positive constants $b_m, c_m \geq 1$ (depending on m only) such that*

$$E\{\|\bar{f} - E\{\bar{f}\}\|_2^2\} \leq b_m\left(N^{-1} + (Nh)^{-1}\int_0^{\pi nh}\frac{1}{(1+x^{2m})^2}dx\right), \tag{4}$$

$$\|E\{\bar{f}\} - f_0\|_2 \leq c_m\sqrt{J(f_0)(\lambda + n^{-2m} + N^{-1})} \tag{5}$$

for any fixed $1 \leq s \leq N$.

1. For simplicity, we denote $J(f,f) = J(f)$ later.

From (17) and (18) in Appendix, we can tell that $\bar{f} - E\{\bar{f}\}$ is irrelevant to f_0. So is the upper bound for the (integrated) variance in (4). However, this is not the case for the (integrated) bias $\|E\{\bar{f}\} - f_0\|_2$, whose upper bound depends on f_0 through its norm $J(f_0)$. In particular, the (integrated) bias becomes zero if f_0 is in the null space, i.e., $J(f_0) = 0$, according to (5).

Since

$$\mathrm{MSE}_{f_0}(\bar{f}) = E\{\|\bar{f} - E\{\bar{f}\}\|_2^2\} + \|E\{\bar{f}\} - f_0\|_2^2, \tag{6}$$

Theorem 3.1 says that

$$\mathrm{MSE}_{f_0}(\bar{f}) \leq b_m\left(N^{-1} + (Nh)^{-1}\int_0^{\pi nh}\frac{1}{(1+x^{2m})^2}dx\right) + c_m^2 J(f_0)(\lambda + n^{-2m} + N^{-1}). \tag{7}$$

When we choose $h \asymp N^{-1/(2m+1)}$ and $n^{-2m} = O(\lambda)$, it can be seen from (7) that $\bar{f}$ is minimax optimal, i.e., $\|\bar{f} - f_0\|_2 = O_P(N^{-m/(2m+1)})$. Obviously, the above two conditions hold if

$$\lambda \asymp N^{-2m/(2m+1)} \text{ and } s = O(N^{2m/(2m+1)}). \tag{8}$$

From now on, we define the optimal choice of λ as $N^{-2m/(2m+1)}$, denoted as λ_*; according to Zhang et al. (2015). Alternatively, the minimax optimality can be achieved if $s \asymp N^{2m/(2m+1)}$ and $nh = o(1)$, i.e., $\lambda = o(\lambda_*)$. In other words, a sub-optimal choice of λ can be compensated by a proper sampling splitting strategy. See Figure 3 for the subtle relation between s and λ. It should be mentioned that λ_* depends on N (rather than n) for achieving optimal estimation rate. In practice, we propose to select λ via a distributed version of GCV; see Xu et al. (2017).

Remark 3.1 *Under random design and uniformly bounded eigenfunctions, Corollary 4 in Zhang et al. (2015) showed that the above rate optimality is achieved under the following upper bound on s (and $\lambda = \lambda_*$)*

$$s = O(N^{(2m-1)/(2m+1)}/\log N).$$

For example, when $m = 2$, their upper bound is $N^{0.6}/\log N$ (versus $N^{0.8}$ in our case). We improve their upper bound by applying a more direct proof strategy.

To understand whether our upper bound can be further improved, we prove a lower bound result in a "worst case" scenario. Specifically, Theorem 3.2 implies that once s is beyond the above upper bound, the rate optimality will break down for at least one true f_0.

Theorem 3.2 *(Lower Bound of Squared Bias) Suppose $h > 0$, and N is divisible by n. Then for any constant $C > 0$, it holds that*

$$\sup_{\substack{f_0 \in S^m(\mathbb{I}) \\ J(f_0) \leq C}} \|E\{\bar{f}\} - f_0\|_2^2 \geq C(a_m n^{-2m} - 8N^{-1}),$$

where $a_m \in (0,1)$ is an absolute constant depending on m only, for any fixed $1 < s < N$.

It follows by (6) that

$$\sup_{\substack{f_0\in S^m(\mathbb{I})\\ J(f_0)\le C}} \mathrm{MSE}_{f_0}(\bar f) \ge \sup_{\substack{f_0\in S^m(\mathbb{I})\\ J(f_0)\le C}} \|E\{\bar f\}-f_0\|_2^2 \ge C(a_m n^{-2m}-8N^{-1}). \tag{9}$$

It is easy to check that the above lower bound is strictly slower than the optimal rate $N^{-2m/(2m+1)}$ if s grows faster than $N^{2m/(2m+1)}$ no matter how λ is chosen. Therefore, we claim that $N^{2m/(2m+1)}$ is a sharp upper bound of s for obtaining an averaged smoothing spline estimate.

In the end, we provide a graphical interpretation for our sharp bound result. Let $s=N^a$ for $0\le a\le 1$ and $\lambda=N^{-b}$ for $0<b<2m$. Define $\rho_1(a),\rho_2(a)$ and $\rho_3(a)$ as

$$\begin{aligned}
\text{Upper bound of squared bias: } N^{-\rho_1(a)} &\asymp \lambda+n^{-2m}+N^{-1},\\
\text{Lower bound of squared bias: } N^{-\rho_2(a)} &\asymp \max\{n^{-2m}-N^{-1},0\},\\
\text{Upper bound of variance: } N^{-\rho_3(a)} &\asymp N^{-1}+(Nh)^{-1}\int_0^{\pi nh}\frac{1}{(1+x^{2m})^2}dx,
\end{aligned}$$

based on Theorems 3.1 and 3.2. A direct examination reveals that

$$\begin{aligned}
\rho_1(a) &= \min\{2m(1-a),1,b\}\\
\rho_2(a) &= \begin{cases} 2m(1-a), & a>(2m-1)/(2m)\\ \infty, & a\le(2m-1)/(2m)\end{cases}\\
\rho_3(a) &= \max\{a,(2m-b)/(2m)\}
\end{aligned}$$

Figure 4 displays ρ_1,ρ_2,ρ_3 for $\lambda=N^{-2m/(2m+1)}$. It can be seen that when $a\in[0,2m/(2m+1)]$, upper bounds of squared bias and variance maintain at the same optimal rate $N^{-2m/(2m+1)}$, while the exact bound of squared bias increases above $N^{-2m/(2m+1)}$ when $a\in(2m/(2m+1),1)$. This explains why transition occurs at the critical point $a=2m/(2m+1)$ (even when the upper bound of variance decreases below $N^{-2m/(2m+1)}$ when $a\in(2m/(2m+1),1)$).

It should be mentioned that when $\lambda\neq N^{-2m/(2m+1)}$, i.e., $b\neq 2m/(2m+1)$, suboptimal estimation almost always occurs. More explicitly, $b<2m/(2m+1)$ yields $\rho_1(a)<2m/(2m+1)$ for any $0\le a\le 1$. While $b>2m/(2m+1)$ yields $\rho_2(a)<2m/(2m+1)$ for any $2m/(2m+1)<a\le 1$; yields $\rho_3(a)<2m/(2m+1)$ for any $0\le a<2m/(2m+1)$. The only exception is $a=2m/(2m+1)$ which yields $\rho_1=\rho_2=\rho_3=2m/(2m+1)$ for any $b>2m/(2m+1)$.

Remark 3.2 *As a side remark, we notice that each machine is assigned with $n\asymp N^{1/(2m+1)}$ samples when s attains its upper bound in the estimation regime. This is very similar as the local polynomial estimation where approximately $N^{1/(2m+1)}$ local points are used for obtaining optimal estimation (although we realize that our data is distributed in a global manner).*

Remark 3.3 *Under repeated curves with a common design, Cai and Yuan (2011) observed a similar phase transition phenomenon for the minimax rate of a two-stage estimate, where the rate transits when the number of sample curves is nearly $N^{2m/(2m+1)}$. This coincides with our observation for s. However, the common design assumption, upon which their results crucially rely, clearly does not apply to our divide-and-conquer setup, and our proof techniques are significantly different. Rather, Theorems 3.1 and 3.2 imply that the results in Cai and Yuan (2011) may still hold for a non-common design.*

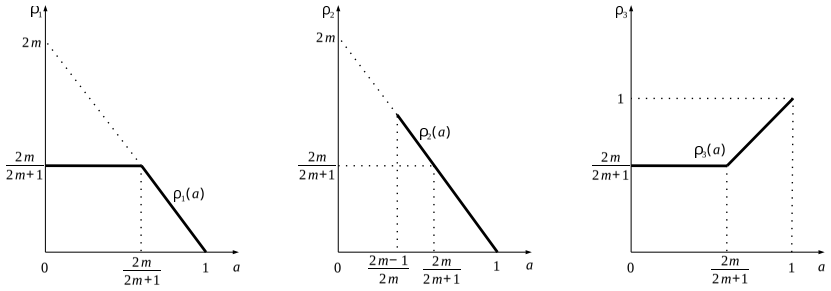


Figure 4: *Plots of $\rho_1(a),\rho_2(a),\rho_3(a)$ versus a, indicated by thick solid lines, under $\lambda=N^{-2m/(2m+1)}$. $\rho_1(a)$, $\rho_2(a)$ and $\rho_3(a)$ indicate upper bound of squared bias, lower bound of squared bias and upper bound of variance, respectively. $\rho_2(a)$ is plotted only for $(2m-1)/(2m)<a\le 1$; when $0\le a\le(2m-1)/(2m)$, $\rho_2(a)=\infty$, which is omitted.*

4. Minimax Optimal Testing

In this section, we consider nonparametric testing:

$$H_0: f=0 \ \text{ v.s. } \ H_1: f\in S^m(\mathbb{I})\backslash\{0\}. \tag{10}$$

In general, testing $f=f_0$ (for a known f_0) is equivalent to testing $f_*\equiv f-f_0=0$. So (10) has no loss of generality. Inspired by the classical Wald test (Shao (2003)), we propose a simple test statistic based on the $\bar f$ as

$$T_{N,\lambda}:=\|\bar f\|_2^2.$$

We find that testing consistency essentially requires no condition on the number of machines no matter it is fixed or diverges at any rate. However, our power analysis, which is non-asymptotically valid, depends on the number of machines in a nontrivial way. Specifically, we discover that our test method is minimax optimal in the sense of Ingster (Ingster (1993)) when s does not grow faster than $N^{(4m-1)/(4m+1)}$ and λ is chosen optimally (different from λ_*, though), but it is no longer optimal once s is beyond the above threshold (no matter how λ is chosen). This is a similar phase transition phenomenon as we observe in the estimation regime. Again, we notice an optimal choice of λ may not be necessary if the above upper bound of s is achieved.

In this section, we assume that the model errors $\epsilon_{i,j}$'s are *iid* standard normal for technical convenience. In fact, our results can be generalized to likelihood ratio test without

assuming Gaussian errors. This extension is possible (technically tedious, though) since likelihood ratio statistic can be approximated by $T_{N,\lambda}$ through quadratic expansion; see Shang and Cheng (2013).

Theorem 4.1 implies the consistency of our proposed test method with the following testing rule:

$$\phi_{N,\lambda} = I(|T_{N,\lambda} - \mu_{N,\lambda}| \geq z_{1-\alpha/2}\sigma_{N,\lambda}),$$

where $\mu_{N,\lambda} := \mathrm{E}_{H_0}\{T_{N,\lambda}\}$, $\sigma^2_{N,\lambda} := \mathrm{Var}_{H_0}\{T_{N,\lambda}\}$ and $z_{1-\alpha/2}$ is the $(1-\alpha/2)\times 100$ percentile of $N(0,1)$. The conditions required in Theorem 4.1 are so mild that our proposed testing is consistent no matter the number of machines is fixed or diverges at any rate.

Theorem 4.1 *(Testing Consistency) Suppose that $h \to 0$, $n \to \infty$ when $N \to \infty$, and $\lim_{N\to\infty} nh$ exists (which could be infinity). Then, we have under H_0,*

$$\frac{T_{N,\lambda} - \mu_{N,\lambda}}{\sigma_{N,\lambda}} \xrightarrow{d} N(0,1), \quad \text{as } N \to \infty.$$

Our next theorem analyzes the non-asymptotic power of $T_{N,\lambda}$, in which we pay particular attention to the impact of s on the separation rate of testing, defined as

$$d_{N,\lambda} = \sqrt{\lambda + n^{-2m} + \sigma_{N,\lambda}}.$$

Let $\mathcal{B} = \{f \in S^m(\mathbb{I}) : J(f) \leq C\}$ for a positive constant C.

Theorem 4.2 *(Upper Bound) Suppose that $h \to 0$, $n \to \infty$ when $N \to \infty$, and $\lim_{N\to\infty} nh$ exists (which could be infinity). Then for any $\varepsilon > 0$, there exist $C_\varepsilon, N_\varepsilon > 0$ s.t. for any $N \geq N_\varepsilon$,*

$$\inf_{\substack{f\in\mathcal{B} \\ \|f\|_2 \geq C_\varepsilon d_{N,\lambda}}} P_f\left(\phi_{N,\lambda} = 1\right) \geq 1 - \varepsilon. \tag{11}$$

Under assumptions of Theorem 4.1, it can be shown that (see (55) in Appendix)

$$\sigma^2_{N,\lambda} \asymp \begin{cases} \frac{n}{N^2}, & \text{if } \lim_{N\to 0} nh = 0, \\ \frac{1}{N^2 h}, & \text{if } \lim_{N\to\infty} nh > 0. \end{cases} \tag{12}$$

Given a range of λ leading to $\lim_{N\to\infty} nh > 0$, we have by (12) that $d_{N,\lambda} = \sqrt{\lambda + (Nh^{1/2})^{-1}}$. An optimal choice of λ (satisfying the above requirement) is $\lambda_{**} := N^{-4m/(4m+1)}$ since it leads to the optimal separating rate $d^*_{N,\lambda} := N^{-2m/(4m+1)}$; see Ingster (1993). Meanwhile, the constraint $\lim_{N\to\infty} nh > 0$ (together with the choice of λ_{**}) implies that

$$s = O(N^{(4m-1)/(4m+1)}). \tag{13}$$

The above discussions illustrate that we can always choose λ_{**} to obtain a minimax optimal testing (just as in the single dataset Shang and Cheng (2013)) as long as s does not grow faster than $N^{(4m-1)/(4m+1)}$. In the case that $\lim_{N\to\infty} nh = 0$, the minimax optimality can be maintained if $s \asymp N^{(4m-1)/(4m+1)}$, $h = o(1)$ and $nh = o(1)$. Such a selection of s gives us a lot of freedom in choosing λ that needs to satisfy $\lambda = o(\lambda_{**})$. A complete picture in depicting the relation between s and λ is given in Figure 3.

We further discover in Theorem 4.3 that the upper bound (13) turns out to be sharp.

Theorem 4.3 *(Lower Bound) Suppose that $s \gg N^{(4m-1)/(4m+1)}$, $h \to 0$, $n \to \infty$ when $N \to \infty$, and $\lim_{N\to\infty} nh$ exists (which could be infinity). Then there exists a positive sequence $\beta_{N,\lambda}$ with $\lim_{N\to\infty} \beta_{N,\lambda} = \infty$ s.t.*

$$\limsup_{N\to\infty} \inf_{\substack{f\in\mathcal{B} \\ \|f\|_2 \geq \beta_{N,\lambda} d^*_{N,\lambda}}} P_f\left(\phi_{N,\lambda} = 1\right) \leq \alpha. \tag{14}$$

Recall that $1-\alpha$ is the pre-specified significance level.

Theorem 4.3 says that when $s \gg N^{(4m-1)/(4m+1)}$, the test $\phi_{N,\lambda}$ is no longer powerful even when $\|f\|_2 \gg d^*_{N,\lambda}$. In other words, our test method fails to be optimal. Therefore, we claim that $N^{(4m-1)/(4m+1)}$ is a sharp upper bound of s to ensure our testing to be minimax optimal.

Remark 4.1 *As a side remark, the existence of $\lim_{N\to\infty} nh$ can be replaced by the following weaker condition under which the results in Theorems 4.1, 4.2 and 4.3 still hold:*

$$\textit{Condition } (\boldsymbol{R}) : \textit{either } \lim_{N\to\infty} nh = 0 \textit{ or } \inf_{N\geq 1} nh > 0.$$

Condition ($\boldsymbol{R}$) aims to exclude irregularly behaved s such as in the following case where s vibrates too much along with N:

$$s = \begin{cases} N^{b_1}, & N \textit{ is odd}, \\ N^{b_2}, & N \textit{ is even}, \end{cases} \tag{15}$$

where $h \asymp N^{-c}$ for some $c > 0$, $b_1, b_2 \in [0,1]$ satisfy $b_1 + c \geq 1$ and $b_2 + c < 1$. Clearly, Condition ($\boldsymbol{R}$) fails under (15).

5. Discussions

This paper offers "theoretical" suggestions on the allocation of data. In a relatively simple distributed algorithm, i.e., in m-order periodic splines with evenly spaced design, our recommendation proceeds as follows:

- Distribute to
$$s \asymp N^{2m/(2m+1)}$$
machines for obtaining an optimal estimate;
- Distribute to
$$s \asymp N^{(4m-1)/(4m+1)}$$
machines for performing an optimal test.

However, data-dependent formulae are still needed in picking a right number of machines in practice. This might be possible in light of Figure 3 indicating that sample splitting could be an alternative form of tuning. As for the choice of λ, we prove that it should be chosen in the order of N even when each subsample has size n. Hence, a distributed version of the generalized cross validation method is applied to each sub-sample; see Xu et al. (2017).

Another theoretically interesting direction is how much adaptive estimation (where m is unknown) can affect the computational limits.

Acknowledgments

We thank PhD student Meimei Liu at Purdue for the simulation study. Zuofeng Shang's research is sponsored by NSF DMS-1764280. Guang Cheng's research is sponsored by NSF (CAREER Award DMS-1151692, DMS-1418042), Simmons Fellowship in Mathematics and OCE of Naval Research (ONR N00014-15-1-2331). Guang Cheng gratefully acknowledges Statistical and Applied Mathematical Sciences Institute (SAMSI) for the hospitality and support during his visit in the 2013-Massive Data Program.

6. Appendix

Proofs of our results are included in this section.

6.1 Proofs in Section 3

Proof [Proof of Theorem 3.1] We do a bit preliminary analysis before proving (4) and (5). It follows from Wahba (1990) that $(S^m(\mathbb{I}), J)$ is a reproducing kernel Hilbert space with reproducing kernel function

$$K(x,y)=\sum_{\nu=1}^{\infty}\frac{\varphi_\nu(x)\varphi_\nu(y)}{\gamma_\nu}=2\sum_{k=1}^{\infty}\frac{\cos(2\pi k(x-y))}{(2\pi k)^{2m}},\ x,y\in\mathbb{I}.$$

For convenience, define $K_x(\cdot)=K(x,\cdot)$ for any $x\in\mathbb{I}$. It follows from the representer theorem (Wahba (1990)) that the optimization to problem (3) has a solution

$$\widehat{f}_j=\sum_{i=1}^{n}\widehat{c}_{i,j}K_{t_{i,j}},\ j=1,2,\ldots,s, \tag{16}$$

where $\widehat{c}_j=(\widehat{c}_{1,j},\ldots,\widehat{c}_{n,j})^T=n^{-1}(\Sigma_j+\lambda I_n)^{-1}Y_j$, $Y_j=(Y_{1,j},\ldots,Y_{n,j})^T$, I_n is $n\times n$ identity matrix, and $\Sigma_j=[K(t_{i,j},t_{i',j})/n]_{1\le i,i'\le n}$. It is easy to see that $\Sigma_1=\Sigma_2=\cdots=\Sigma_s$. For convenience, denote $\Sigma=\Sigma_1$. Similarly, define

$$K'(x,y)=\sum_{\nu=1}^{\infty}\frac{\varphi_\nu(x)\varphi_\nu(y)}{\gamma_\nu^2}=2\sum_{k=1}^{\infty}\frac{\cos(2\pi k(x-y))}{(2\pi k)^{4m}},\ x,y\in\mathbb{I}.$$

For $1\le j\le s$, let $\Omega_j=[K'(t_{i,j},t_{i',j})/n]_{1\le i,i'\le n}$. It is easy to see that $\Omega_1=\Omega_2=\cdots=\Omega_s$. For convenience, denote $\Omega=\Omega_1$, and let $\Phi_{\nu,j}=(\varphi_\nu(t_{1,j}),\ldots,\varphi_\nu(t_{n,j}))$.

It is easy to examine that

$$\begin{aligned}\bar{f}&=\sum_{\nu=1}^{\infty}\frac{\sum_{j=1}^{s}\Phi_{\nu,j}(\Sigma+\lambda I_n)^{-1}Y_j}{N\gamma_\nu}\varphi_\nu\\&=\sum_{\nu=1}^{\infty}\frac{\sum_{j=1}^{s}\Phi_{\nu,j}(\Sigma+\lambda I_n)^{-1}(\mathbf{f}_{0,j}+\boldsymbol{\epsilon}_j)}{N\gamma_\nu}\varphi_\nu,\end{aligned} \tag{17}$$

and

$$E\{\bar{f}\}=\sum_{\nu=1}^{\infty}\frac{\sum_{j=1}^{s}\Phi_{\nu,j}(\Sigma+\lambda I_n)^{-1}\mathbf{f}_{0,j}}{N\gamma_\nu}\varphi_\nu, \tag{18}$$

where $\mathbf{f}_{0,j}=(f_0(t_{1,j}),\ldots,f_0(t_{n,j}))^T$ and $\boldsymbol{\epsilon}_j=(\epsilon_{1,j},\ldots,\epsilon_{n,j})^T$.

We now look at Σ and Ω. For $0\le l\le n-1$, let

$$\begin{aligned}c_l&=\frac{2}{n}\sum_{k=1}^{\infty}\frac{\cos(2\pi kl/n)}{(2\pi k)^{2m}},\\d_l&=\frac{2}{n}\sum_{k=1}^{\infty}\frac{\cos(2\pi kl/n)}{(2\pi k)^{4m}}.\end{aligned}$$

Since $c_l=c_{n-l}$ and $d_l=d_{n-l}$ for $l=1,2,\ldots,n-1$, Σ and Ω are both symmetric circulant of order n. Let $\varepsilon=\exp(2\pi\sqrt{-1}/n)$. Ω and Σ share the same normalized eigenvectors as

$$x_r=\frac{1}{\sqrt{n}}(1,\varepsilon^r,\varepsilon^{2r},\ldots,\varepsilon^{(n-1)r})^T,\ r=0,1,\ldots,n-1.$$

Let $M=(x_0,x_1,\ldots,x_{n-1})$. Denote M^* as the conjugate transpose of M. Clearly, $MM^*=I_n$ and Σ,Ω admits the following decomposition

$$\Sigma=M\Lambda_cM^*,\ \Omega=M\Lambda_dM^*, \tag{19}$$

where $\Lambda_c=\mathrm{diag}(\lambda_{c,0},\lambda_{c,1},\ldots,\lambda_{c,n-1})$ and $\Lambda_d=\mathrm{diag}(\lambda_{d,0},\lambda_{d,1},\ldots,\lambda_{d,n-1})$ with $\lambda_{c,l}=c_0+c_1\varepsilon^l+\ldots+c_{n-1}\varepsilon^{(n-1)l}$ and $\lambda_{d,l}=d_0+d_1\varepsilon^l+\ldots+d_{n-1}\varepsilon^{(n-1)l}$.

Direct calculations show that

$$\lambda_{c,l}=\begin{cases}2\sum_{k=1}^{\infty}\frac{1}{(2\pi kn)^{2m}}, & l=0,\\ \sum_{k=1}^{\infty}\frac{1}{[2\pi(kn-l)]^{2m}}+\sum_{k=0}^{\infty}\frac{1}{[2\pi(kn+l)]^{2m}}, & 1\le l\le n-1.\end{cases} \tag{20}$$

$$\lambda_{d,l}=\begin{cases}2\sum_{k=1}^{\infty}\frac{1}{(2\pi kn)^{4m}}, & l=0,\\ \sum_{k=1}^{\infty}\frac{1}{[2\pi(kn-l)]^{4m}}+\sum_{k=0}^{\infty}\frac{1}{[2\pi(kn+l)]^{4m}}, & 1\le l\le n-1.\end{cases} \tag{21}$$

It is easy to examine that

$$\lambda_{c,0}=2\bar{c}_m(2\pi n)^{-2m},\ \ \lambda_{d,0}=2\bar{d}_m(2\pi n)^{-4m}, \tag{22}$$

and for $1\le l\le n-1$,

$$\begin{aligned}\lambda_{c,l}&=\frac{1}{[2\pi(n-l)]^{2m}}+\frac{1}{(2\pi l)^{2m}}\\&\quad+\sum_{k=2}^{\infty}\frac{1}{[2\pi(kn-l)]^{2m}}+\sum_{k=1}^{\infty}\frac{1}{[2\pi(kn+l)]^{2m}},\\\lambda_{d,l}&=\frac{1}{[2\pi(n-l)]^{4m}}+\frac{1}{(2\pi l)^{4m}}\\&\quad+\sum_{k=2}^{\infty}\frac{1}{[2\pi(kn-l)]^{4m}}+\sum_{k=1}^{\infty}\frac{1}{[2\pi(kn+l)]^{4m}},\end{aligned} \tag{23}$$

and for $\bar{c}_m := \sum_{k=1}^{\infty} k^{-2m}$, $\underline{c}_m := \sum_{k=2}^{\infty} k^{-2m}$, $\bar{d}_m := \sum_{k=1}^{\infty} k^{-4m}$, $\underline{d}_m := \sum_{k=2}^{\infty} k^{-4m}$,

$$\underline{c}_m(2\pi n)^{-2m} \le \sum_{k=2}^{\infty} \frac{1}{[2\pi(kn-l)]^{2m}} \le \bar{c}_m(2\pi n)^{-2m},$$
$$\underline{c}_m(2\pi n)^{-2m} \le \sum_{k=1}^{\infty} \frac{1}{[2\pi(kn+l)]^{2m}} \le \bar{c}_m(2\pi n)^{-2m},$$

$$\underline{d}_m(2\pi n)^{-4m} \le \sum_{k=2}^{\infty} \frac{1}{[2\pi(kn-l)]^{4m}} \le \bar{d}_m(2\pi n)^{-4m},$$
$$\underline{d}_m(2\pi n)^{-4m} \le \sum_{k=1}^{\infty} \frac{1}{[2\pi(kn+l)]^{4m}} \le \bar{d}_m(2\pi n)^{-4m}.$$

For simplicity, we denote $I = E\{\|\bar{f} - E\{\bar{f}\}\|_2^2\}$ and $II = \|E\{\bar{f}\} - f_0\|_2^2$. Hence, $\mathrm{MSE}_{f_0}(\bar{f}) = I + II$.

Proof of (4)

Using (19) – (23), we get that

$$\begin{aligned}
I &= \sum_{\nu=1}^{\infty} \frac{\sum_{j=1}^{s} E\{|\Phi_{\nu,j}(\Sigma+\lambda I_n)^{-1}\boldsymbol{\epsilon}_j|^2\}}{N^2\gamma_\nu^2} \\
&= \sum_{\nu=1}^{\infty} \frac{\sum_{j=1}^{s} \mathrm{trace}((\Sigma+\lambda I_n)^{-1}\Phi_{\nu,j}^T\Phi_{\nu,j}(\Sigma+\lambda I_n)^{-1})}{N^2\gamma_\nu^2} \\
&= \frac{n}{N^2}\sum_{j=1}^{s} \mathrm{trace}\left((\Sigma+\lambda I_n)^{-1}\sum_{\nu=1}^{\infty}\frac{\Phi_{\nu,j}^T\Phi_{\nu,j}/n}{\gamma_\nu^2}(\Sigma+\lambda I_n)^{-1}\right) \\
&= \frac{n}{N^2}\sum_{j=1}^{s} \mathrm{trace}\left((\Sigma+\lambda I_n)^{-1}\Omega(\Sigma+\lambda I_n)^{-1}\right) \\
&= \frac{1}{N}\mathrm{trace}\left(M(\Lambda_c+\lambda I_n)^{-1}\Lambda_d(\Lambda_c+\lambda I_n)^{-1}M^*\right) \\
&= \frac{1}{N}\sum_{l=0}^{n-1}\frac{\lambda_{d,l}}{(\lambda+\lambda_{c,l})^2} \\
&\le \frac{2\bar{d}_m}{N(2\bar{c}_m+(2\pi n)^{2m}\lambda)^2} \\
&\quad +(1+\bar{d}_m)N^{-1}\sum_{l=1}^{n-1}\frac{(2\pi(n-l))^{-4m}+(2\pi l)^{-4m}}{(\lambda+(2\pi(n-l))^{-2m}+(2\pi l)^{-2m})^2} \\
&\le \frac{2\bar{d}_m}{N(2\bar{c}_m+(2\pi n)^{2m}\lambda)^2} \\
&\quad +2(1+\bar{d}_m)N^{-1}\sum_{1\le l\le n/2}\frac{(2\pi l)^{-4m}+(2\pi(n-l))^{-4m}}{(\lambda+(2\pi l)^{-2m}+(2\pi(n-l))^{-2m})^2} \\
&\le \frac{2\bar{d}_m}{N(2\bar{c}_m+(2\pi n)^{2m}\lambda)^2}+4(1+\bar{d}_m)N^{-1}\sum_{1\le l\le n/2}\frac{(2\pi l)^{-4m}}{(\lambda+(2\pi l)^{-2m})^2} \\
&\le \frac{2\bar{d}_m}{N(2\bar{c}_m+(2\pi n)^{2m}\lambda)^2}+\frac{2(1+\bar{d}_m)}{\pi Nh}\int_0^{\pi nh}\frac{1}{(1+x^{2m})^2}dx \\
&\le b_m\left(\frac{1}{N}+\frac{1}{Nh}\int_0^{\pi nh}\frac{1}{(1+x^{2m})^2}dx\right),
\end{aligned}$$

where $b_m \ge 1$ is an absolute constant depending on m only. This proves (4).

Proof of (5)

Throughout, let $\eta = \exp(2\pi\sqrt{-1}/N)$. For $1 \le j, l \le s$, define

$$\begin{aligned}
\Sigma_{j,l} &= \frac{1}{n}\sum_{\nu=1}^{\infty}\frac{\Phi_{\nu,j}^T\Phi_{\nu,l}}{\gamma_\nu}, \\
\sigma_{j,l,r} &= \frac{2}{n}\sum_{k=1}^{\infty}\frac{\cos\left(2\pi k\left(\frac{r}{n}-\frac{j-l}{N}\right)\right)}{(2\pi k)^{2m}},\ r=0,1,\ldots,n-1.
\end{aligned}$$

It can be shown that $\Sigma_{j,l}$ is a circulant matrix with elements $\sigma_{j,l,0}, \sigma_{j,l,1}, \ldots, \sigma_{j,l,n-1}$, therefore, by Brockwell and Davis (1987) we get that

$$\Sigma_{j,l} = M \Lambda_{j,l} M^*, \tag{24}$$

where M is the same as in (19), and $\Lambda_{j,l} = \mathrm{diag}(\lambda_{j,l,0}, \lambda_{j,l,1}, \ldots, \lambda_{j,l,n-1})$, with $\lambda_{j,l,r}$, for $r = 1, \ldots, n-1$, given by the following

$$\begin{aligned}
\lambda_{j,l,r} &= \sum_{t=0}^{n-1} \sigma_{j,l,t} \varepsilon^{rt} \\
&= \frac{2}{n} \sum_{t=0}^{n-1} \sum_{k=1}^{\infty} \frac{\cos\left(2\pi k \left(\frac{t}{n} - \frac{j-l}{N}\right)\right)}{(2\pi k)^{2m}} \varepsilon^{rt} \\
&= \frac{1}{n} \sum_{k=1}^{\infty} \frac{\eta^{-k(j-l)} \sum_{t=0}^{n-1} \varepsilon^{(k+r)t} + \eta^{k(j-l)} \sum_{t=0}^{n-1} \varepsilon^{(r-k)t}}{(2\pi k)^{2m}} \\
&= \sum_{q=1}^{\infty} \frac{\eta^{-(qn-r)(j-l)}}{[2\pi(qn-r)]^{2m}} + \sum_{q=0}^{\infty} \frac{\eta^{(qn+r)(j-l)}}{[2\pi(qn+r)]^{2m}},
\end{aligned} \tag{25}$$

and for $r = 0$, given by

$$\begin{aligned}
\lambda_{j,l,0} &= \sum_{t=0}^{n-1} \sigma_{j,l,t} \\
&= \frac{1}{n} \sum_{k=1}^{\infty} \frac{\sum_{t=0}^{n-1} \varepsilon^{kt} \eta^{k(j-l)} + \sum_{t=0}^{n-1} \varepsilon^{-kt} \eta^{-k(j-l)}}{(2\pi k)^{2m}} \\
&= \sum_{q=1}^{\infty} \frac{\eta^{qn(j-l)} + \eta^{-qn(j-l)}}{(2\pi qn)^{2m}}.
\end{aligned} \tag{26}$$

For $p \geq 0$, $1 \leq v \leq n$, $0 \leq r \leq n-1$ and $1 \leq j \leq s$, define

$$A_{p,v,r,j} = \frac{1}{s} \sum_{l=1}^{s} \lambda_{j,l,r} x_r^* \Phi_{2(pn+v)-1,l}^T, \quad B_{p,v,r,j} = \frac{1}{s} \sum_{l=1}^{s} \lambda_{j,l,r} x_r^* \Phi_{2(pn+v),l}^T.$$

By direct calculation, we have for $1 \leq v \leq n-1$,

$$\begin{aligned}
\Phi_{2(pn+v)-1,l} x_r &= \sqrt{n/2} \left(\eta^{(pn+v)(l-1)} I(r+v=n) + \eta^{-(pn+v)(l-1)} I(v=r)\right), \\
\Phi_{2(pn+v),l} x_r &= \sqrt{-n/2} \left(\eta^{(pn+v)(l-1)} I(r+v=n) - \eta^{-(pn+v)(l-1)} I(v=r)\right),
\end{aligned} \tag{27}$$

and

$$\begin{aligned}
\Phi_{2(pn+n)-1,l} x_r &= \sqrt{n/2} I(r=0) \left(\eta^{(p+1)n(l-1)} + \eta^{-(p+1)n(l-1)}\right), \\
\Phi_{2(pn+n),l} x_r &= \sqrt{-n/2} I(r=0) \left(\eta^{(p+1)n(l-1)} - \eta^{-(p+1)n(l-1)}\right).
\end{aligned} \tag{28}$$

Let $I(\cdot)$ be an indicator function. Then we have for $p \geq 0$, $1 \leq j \leq s$ and $1 \leq v, r \leq n-1$,

$$\begin{aligned}
B_{p,v,r,j} &= \frac{1}{s} \sum_{l=1}^{s} \lambda_{j,l,r} x_r^* \Phi_{2(pn+v),l}^T \\
&= -\frac{1}{s} \sqrt{-n/2} \sum_{l=1}^{s} \left(\sum_{q=1}^{\infty} \frac{\eta^{-(qn-r)(j-l)}}{[2\pi(qn-r)]^{2m}} + \sum_{q=0}^{\infty} \frac{\eta^{(qn+r)(j-l)}}{[2\pi(qn+r)]^{2m}} \right) \\
&\quad \times \left(\eta^{-(pn+v)(l-1)} I(r+v=n) - \eta^{(pn+v)(l-1)} I(r=v) \right) \\
&= -\sqrt{-n/2} \left(\sum_{u \geq -p/s} \frac{\eta^{-(pn+v)(j-1)}}{[2\pi(uN+pn+v)]^{2m}} I(r+v=n) \right. \\
&\quad - \sum_{u \geq (p+1)/s} \frac{\eta^{(pn+v)(j-1)}}{[2\pi(uN-pn-v)]^{2m}} I(r=v) \\
&\quad + \sum_{u \geq (p+1)/s} \frac{\eta^{-(pn+v)(j-1)}}{[2\pi(uN-pn-v)]^{2m}} I(r+v=n) \\
&\quad \left. - \sum_{u \geq -p/s} \frac{\eta^{(pn+v)(j-1)}}{[2\pi(uN+pn+v)]^{2m}} I(r=v) \right) = a_{p,v} x_r^* \Phi_{2(pn+v),j}^T,
\end{aligned} \tag{29}$$

where $a_{p,v} = \sum_{u \geq -p/s} \frac{1}{[2\pi(uN+pn+v)]^{2m}} + \sum_{u \geq (p+1)/s} \frac{1}{[2\pi(uN-pn-v)]^{2m}}$, for $p \geq 0$, $1 \leq v \leq n-1$.

For $v = n$, similar calculations give that

$$\begin{aligned}
B_{p,n,r,j} &= -\sqrt{-n/2} I(r=0) \left(\sum_{u \geq -p/s} \frac{\eta^{-(pn+n)(j-1)}}{[2\pi(uN+pn+n)]^{2m}} \right. \\
&\quad - \sum_{u \geq (p+2)/s} \frac{\eta^{(pn+n)(j-1)}}{[2\pi(uN-pn-n)]^{2m}} \\
&\quad \left. + \sum_{u \geq (p+2)/s} \frac{\eta^{-(pn+n)(j-1)}}{[2\pi(uN-pn-n)]^{2m}} - \sum_{u \geq -p/s} \frac{\eta^{(pn+n)(j-1)}}{[2\pi(uN+pn+n)]^{2m}} \right) \\
&= a_{p,n} x_r^* \Phi_{2(pn+n),j}^T,
\end{aligned} \tag{30}$$

where $a_{p,n} = \sum_{u \geq -p/s} \frac{1}{[2\pi(uN+pn+n)]^{2m}} + \sum_{u \geq (p+2)/s} \frac{1}{[2\pi(uN-pn-n)]^{2m}}$, for $p \geq 0$.

Similarly, we have $p \geq 0$, $1 \leq j \leq s$ and $1 \leq v, r \leq n-1$,

$$\begin{aligned} A_{p,v,r,j} &= \sqrt{n/2}\left(\sum_{u\geq -p/s}\frac{\eta^{-(pn+v)(j-1)}}{[2\pi(uN+pn+v)]^{2m}}I(r+v=n)\right. \\ &+\sum_{u\geq (p+1)/s}\frac{\eta^{(pn+v)(j-1)}}{[2\pi(uN-pn-v)]^{2m}}I(r=v) \\ &+\sum_{u\geq (p+1)/s}\frac{\eta^{-(pn+v)(j-1)}}{[2\pi(uN-pn-v)]^{2m}}I(r+v=n) \\ &\left.+\sum_{u\geq -p/s}\frac{\eta^{(pn+v)(j-1)}}{[2\pi(uN+pn+v)]^{2m}}I(r=v)\right) = a_{p,v}x_r^*\Phi^T_{2(pn+v)-1,j}, \end{aligned} \tag{31}$$

and for $v = n$,

$$\begin{aligned} A_{p,n,r,j} &= \sqrt{n/2}I(r=0)\left(\sum_{u\geq -p/s}\frac{\eta^{-(pn+n)(j-1)}}{[2\pi(uN+pn+n)]^{2m}}\right. \\ &+\sum_{u\geq (p+2)/s}\frac{\eta^{(pn+n)(j-1)}}{[2\pi(uN-pn-n)]^{2m}}+\sum_{u\geq (p+2)/s}\frac{\eta^{-(pn+n)(j-1)}}{[2\pi(uN-pn-n)]^{2m}} \\ &\left.+\sum_{u\geq -p/s}\frac{\eta^{(pn+n)(j-1)}}{[2\pi(uN+pn+n)]^{2m}}\right) = a_{p,n}x_r^*\Phi^T_{2(pn+n)-1,j}. \end{aligned} \tag{32}$$

It is easy to check that both (29) and (31) hold for $r = 0$. Summarizing (29)–(32), we have that for $p \geq 0$, $1 \leq j \leq s$, $1 \leq v \leq n$ and $0 \leq r \leq n-1$,

$$\begin{aligned} A_{p,v,r,j} &= a_{p,v}x_r^*\Phi^T_{2(pn+v)-1,j}, \\ B_{p,v,r,j} &= a_{p,v}x_r^*\Phi^T_{2(pn+v),j}. \end{aligned} \tag{33}$$

To show (5), let $\bar{\boldsymbol{f}}_j = (E\{\bar{f}(t_{1,j})\},\ldots,E\{\bar{f}(t_{n,j})\})^T$, for $1 \leq j \leq s$. It follows by (18) that

$$\begin{aligned} \bar{\boldsymbol{f}}_j &= \sum_{\nu=1}^{\infty}\frac{\sum_{l=1}^s \Phi_{\nu,l}(\Sigma+\lambda I_n)^{-1}\mathbf{f}_{0,l}}{N\gamma_\nu}\Phi^T_{\nu,j} \\ &= \frac{1}{s}\sum_{l=1}^{s}\left(\frac{1}{n}\sum_{\nu=1}^{\infty}\frac{\Phi^T_{\nu,j}\Phi_{\nu,l}}{\gamma_\nu}\right)(\Sigma+\lambda I_n)^{-1}\mathbf{f}_{0,l} \\ &= \frac{1}{s}\sum_{l=1}^{s}\Sigma_{j,l}(\Sigma+\lambda I_n)^{-1}\mathbf{f}_{0,l} \\ &= \frac{1}{s}\sum_{l=1}^{s}M\Lambda_{j,l}(\Lambda_c+\lambda I_n)^{-1}M^*\mathbf{f}_{0,l}, \end{aligned}$$

together with (33), leading to that

$$\begin{aligned} M^*\bar{\boldsymbol{f}}_j &= \frac{1}{s}\sum_{l=1}^{s}\Lambda_{j,l}(\Lambda_c+\lambda I_n)^{-1}M^*\mathbf{f}_{0,l} \\ &= \sum_{\mu=1}^{\infty}f_\mu^0\begin{pmatrix}\frac{\frac{1}{s}\sum_{l=1}^s\lambda_{j,l,0}x_0^*\Phi^T_{\mu,l}}{\lambda+\lambda_{c,0}}\\ \vdots \\ \frac{\frac{1}{s}\sum_{l=1}^s\lambda_{j,l,n-1}x_{n-1}^*\Phi^T_{\mu,l}}{\lambda+\lambda_{c,n-1}}\end{pmatrix} \\ &= \sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)-1}\begin{pmatrix}\frac{\frac{1}{s}\sum_{l=1}^s\lambda_{j,l,0}x_0^*\Phi^T_{2(pn+v)-1,l}}{\lambda+\lambda_{c,0}}\\ \vdots \\ \frac{\frac{1}{s}\sum_{l=1}^s\lambda_{j,l,n-1}x_{n-1}^*\Phi^T_{2(pn+v)-1,l}}{\lambda+\lambda_{c,n-1}}\end{pmatrix} \\ &+\sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)}\begin{pmatrix}\frac{\frac{1}{s}\sum_{l=1}^s\lambda_{j,l,0}x_0^*\Phi^T_{2(pn+v),l}}{\lambda+\lambda_{c,0}}\\ \vdots \\ \frac{\frac{1}{s}\sum_{l=1}^s\lambda_{j,l,n-1}x_{n-1}^*\Phi^T_{2(pn+v),l}}{\lambda+\lambda_{c,n-1}}\end{pmatrix} \\ &= \sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)-1}\begin{pmatrix}\frac{A_{p,v,0,j}}{\lambda+\lambda_{c,0}}\\ \vdots \\ \frac{A_{p,v,n-1,j}}{\lambda+\lambda_{c,n-1}}\end{pmatrix}+\sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)}\begin{pmatrix}\frac{B_{p,v,0,j}}{\lambda+\lambda_{c,0}}\\ \vdots \\ \frac{B_{p,v,n-1,j}}{\lambda+\lambda_{c,n-1}}\end{pmatrix} \\ &= \sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)-1}\begin{pmatrix}\frac{a_{p,v}}{\lambda+\lambda_{c,0}}x_0^*\Phi^T_{2(pn+v)-1,j}\\ \vdots \\ \frac{a_{p,v}}{\lambda+\lambda_{c,n-1}}x_{n-1}^*\Phi^T_{2(pn+v)-1,j}\end{pmatrix} \\ &+\sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)}\begin{pmatrix}\frac{a_{p,v}}{\lambda+\lambda_{c,0}}x_0^*\Phi^T_{2(pn+v),j}\\ \vdots \\ \frac{a_{p,v}}{\lambda+\lambda_{c,n-1}}x_{n-1}^*\Phi^T_{2(pn+v),j}\end{pmatrix}. \end{aligned}$$

On the other hand,

$$\begin{aligned} M^*\mathbf{f}_{0,j} &= \sum_{\mu=1}^{\infty}f_\mu^0M^*\Phi^T_{\mu,j} \\ &= \sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)-1}M^*\Phi^T_{2(pn+v)-1,j}+\sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)}M^*\Phi^T_{2(pn+v),j} \\ &= \sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)-1}\begin{pmatrix}x_0^*\Phi^T_{2(pn+v)-1,j}\\ \vdots \\ x_{n-1}^*\Phi^T_{2(pn+v)-1,j}\end{pmatrix}+\sum_{p=0}^{\infty}\sum_{v=1}^{n}f^0_{2(pn+v)}\begin{pmatrix}x_0^*\Phi^T_{2(pn+v),j}\\ \vdots \\ x_{n-1}^*\Phi^T_{2(pn+v),j}\end{pmatrix}. \end{aligned}$$

Therefore,

$$M^*(\bar{\boldsymbol{f}}_j - \mathbf{f}_{0,j}) = \sum_{p=0}^{\infty}\sum_{v=1}^{n} f^0_{2(pn+v)-1} \begin{pmatrix} b_{p,v,0}x_0^*\Phi^T_{2(pn+v)-1,j} \\ \vdots \\ b_{p,v,n-1}x_{n-1}^*\Phi^T_{2(pn+v)-1,j} \end{pmatrix} + \sum_{p=0}^{\infty}\sum_{v=1}^{n} f^0_{2(pn+v)} \begin{pmatrix} b_{p,v,0}x_0^*\Phi^T_{2(pn+v),j} \\ \vdots \\ b_{p,v,n-1}x_{n-1}^*\Phi^T_{2(pn+v),j} \end{pmatrix}, \quad (34)$$

where $b_{p,v,r} = \frac{a_{p,v}}{\lambda+\lambda_{c,r}} - 1$, for $p \geq 0$, $1 \leq v \leq n$ and $0 \leq r \leq n-1$.

It holds the trivial observation $b_{ks+g,v,r} = b_{g,v,r}$ for $k \geq 0$, $0 \leq g \leq s-1$, $1 \leq v \leq n$ and $0 \leq r \leq n-1$. Define $C_{g,r} = \sum_{k=0}^{\infty}(f^0_{2(kN+gn+n-r)-1} - \sqrt{-1}f^0_{2(kN+gn+n-r)})$ and $D_{g,r} = \sum_{k=0}^{\infty}(f^0_{2(kN+gn+r)-1} + \sqrt{-1}f^0_{2(kN+gn+r)})$, for $0 \leq g \leq s-1$ and $0 \leq r \leq n-1$. Also denote $\overline{C_{g,r}}$ and $\overline{D_{g,r}}$ as their conjugate. By (27) and (28), and direct calculations we get that, for $1 \leq j \leq s$ and $1 \leq r \leq n-1$,

$$\begin{aligned}\delta_{j,r} &\equiv \sum_{p=0}^{\infty}\left(\sum_{v=1}^{n} f^0_{2(pn+v)-1}b_{p,v,r}x_r^*\Phi^T_{2(pn+v)-1,j} + \sum_{v=1}^{n} f^0_{2(pn+v)}b_{p,v,r}x_r^*\Phi^T_{2(pn+v),j}\right) \\ &= \sqrt{\frac{n}{2}}\sum_{p=0}^{\infty}\Big[(f^0_{2(pn+n-r)-1} - \sqrt{-1}f^0_{2(pn+n-r)})b_{p,n-r,r}\eta^{-(pn+n-r)(j-1)} \\ &\quad +(f^0_{2(pn+r)-1} + \sqrt{-1}f^0_{2(pn+r)})b_{p,r,r}\eta^{(pn+r)(j-1)}\Big], \quad (35)\end{aligned}$$

leading to that

$$\begin{aligned}\sum_{j=1}^{s}|\delta_{j,r}|^2 &= \frac{n}{2}\sum_{j=1}^{s}\left|\sum_{p=0}^{\infty}\Big[(f^0_{2(pn+n-r)-1} - \sqrt{-1}f^0_{2(pn+n-r)})b_{p,n-r,r}\eta^{-(pn+n-r)(j-1)} \right.\\ &\quad \left. +(f^0_{2(pn+r)-1} + \sqrt{-1}f^0_{2(pn+r)})b_{p,r,r}\eta^{(pn+r)(j-1)}\Big]\right|^2 \\ &= \frac{n}{2}\sum_{j=1}^{s}\left|\sum_{g=0}^{s-1}\left(C_{g,r}b_{g,n-r,r}\eta^{-(gn+n-r)(j-1)} + D_{g,r}b_{g,r,r}\eta^{(gn+r)(j-1)}\right)\right|^2 \\ &= \frac{n}{2}\sum_{g,g'=0}^{s-1}\sum_{j=1}^{s}(C_{g,r}b_{g,n-r,r}\eta^{-(gn+n-r)(j-1)} + D_{g,r}b_{g,r,r}\eta^{(gn+r)(j-1)}) \\ &\quad \times(\overline{C_{g',r}}b_{g',n-r,r}\eta^{(g'n+n-r)(j-1)} + \overline{D_{g',r}}b_{g',r,r}\eta^{-(g'n+r)(j-1)}) \\ &= \frac{N}{2}\sum_{g=0}^{s-1}(|C_{g,r}|^2b^2_{g,n-r,r} + C_{g,r}\overline{D_{s-1-g,r}}b_{g,n-r,r}b_{s-1-g,r,r} \\ &\quad +D_{g,r}\overline{C_{s-1-g,r}}b_{g,r,r}b_{s-1-g,n-r,r} + |D_{g,r}|^2b^2_{g,r,r}) \\ &= \frac{N}{2}\sum_{g=0}^{s-1}|C_{g,r}b_{g,n-r,r} + D_{s-1-g,r}b_{s-1-g,r,r}|^2 \quad (36) \\ &\leq N\sum_{g=0}^{s-1}(|C_{g,r}|^2b^2_{g,n-r,r} + |D_{s-1-g,r}|^2b^2_{s-1-g,r,r}) \\ &= N\sum_{g=0}^{s-1}(|C_{g,r}|^2b^2_{g,n-r,r} + |D_{g,r}|^2b^2_{g,r,r}).\end{aligned}$$

It is easy to see that for $0 \leq g \leq s-1$ and $1 \leq r \leq n-1$,

$$\begin{aligned}|C_{g,r}|^2 &= (\sum_{k=0}^{\infty} f^0_{2(kN+gn+n-r)-1})^2 + (\sum_{k=0}^{\infty} f^0_{2(kN+gn+n-r)})^2 \\ &\leq \sum_{k=0}^{\infty}(|f^0_{2(kN+gn+n-r)-1}|^2 + |f^0_{2(kN+gn+n-r)}|^2)(kN+gn+n-r)^{2m} \\ &\quad \times\sum_{k=0}^{\infty}(kN+gn+n-r)^{-2m} \\ &\leq \sum_{k=0}^{\infty}(|f^0_{2(kN+gn+n-r)-1}|^2 + |f^0_{2(kN+gn+n-r)}|^2)(kN+gn+n-r)^{2m} \\ &\quad \times\frac{2m}{2m-1}(gn+n-r)^{-2m}, \quad (37)\end{aligned}$$

and

$$\begin{aligned}|D_{g,r}|^2 &\leq \sum_{k=0}^{\infty}(|f^0_{2(kN+gn+r)}|^2 + |f^0_{2(kN+gn+r)-1}|^2)(kN+gn+r)^{2m} \\ &\quad \times\frac{2m}{2m-1}(gn+r)^{-2m}. \quad (38)\end{aligned}$$

For $1 \leq g \leq s-1$, we have $a_{g,n-r} \leq \lambda_{c,r}$, which further leads to $|b_{g,n-r,r}| \leq 2$. Meanwhile, by (20), we have

$$0 \leq \lambda_{c,r} - a_{0,r} \leq (2\pi(n-r))^{-2m} + 2\bar{c}_m(2\pi n)^{-2m} \leq (1+2\bar{c}_m)(2\pi(n-r))^{-2m}.$$

Then we have

$$\begin{aligned}|b_{0,r,r}| &= \frac{\lambda+\lambda_{c,r}-a_{0,r}}{\lambda+\lambda_{c,r}} \\ &\leq \frac{\lambda+(1+2\bar{c}_m)(2\pi(n-r))^{-2m}}{\lambda+(2\pi r)^{-2m}+(2\pi(n-r))^{-2m}} \\ &\leq (1+2\bar{c}_m)\frac{\lambda+(2\pi(n-r))^{-2m}}{\lambda+(2\pi r)^{-2m}+(2\pi(n-r))^{-2m}},\end{aligned}$$

leading to

$$\begin{aligned}r^{-2m}b^2_{0,r,r} &\leq r^{-2m}(1+2\bar{c}_m)^2\left(\frac{\lambda+(2\pi(n-r))^{-2m}}{\lambda+(2\pi(n-r))^{-2m}+(2\pi r)^{-2m}}\right)^2 \\ &\leq r^{-2m}(1+2\bar{c}_m)^2\left(\frac{\lambda+(2\pi(n-r))^{-2m}}{\lambda+(2\pi(n-r))^{-2m}+(2\pi r)^{-2m}}\right) \\ &\leq (2\pi)^{2m}(1+2\bar{c}_m)^2(\lambda+(\pi n)^{-2m}). \quad (39)\end{aligned}$$

The last inequality can be proved in two different cases: $2r \leq n$ and $2r > n$. Similarly, it can be shown that $(n-r)^{-2m}b^2_{0,n-r,r} \leq (2\pi)^{2m}(1+2\bar{c}_m)^2(\lambda+(\pi n)^{-2m})$.

Then we have by (37)–(39) that

$$\begin{aligned}\sum_{r=1}^{n-1}\sum_{g=0}^{s-1}|D_{g,r}|^2b_{g,r,r}^2 &\le \sum_{r=1}^{n-1}\sum_{g=1}^{s-1}\sum_{k=0}^{\infty}(|f^0_{2(kN+gn+r)}|^2+|f^0_{2(kN+gn+r)-1}|^2)(kN+gn+r)^{2m}\\ &\quad\times\frac{2m}{2m-1}(gn+r)^{-2m}2^2+\sum_{r=1}^{n-1}\sum_{k=0}^{\infty}(|f^0_{2(kN+r)}|^2+|f^0_{2(kN+r)-1}|^2)(kN+r)^{2m}\\ &\quad\times\frac{2m}{2m-1}r^{-2m}b_{0,r,r}^2\\ &\le c'_m(\lambda+n^{-2m})\sum_{r=1}^{n-1}\sum_{g=0}^{s-1}\sum_{k=0}^{\infty}(|f^0_{2(kN+gn+r)}|^2+|f^0_{2(kN+gn+r)-1}|^2)\\ &\quad\times(2\pi(kN+gn+r))^{2m}, \qquad (40)\end{aligned}$$

where $c'_m=\max\{(2\pi)^{-2m}\frac{8m}{2m-1},(1+2\bar{c}_m)^2\frac{2m}{2m-1}\}$. Similarly, one can show that

$$\begin{aligned}\sum_{r=1}^{n-1}\sum_{g=0}^{s-1}|C_{g,r}|^2b_{g,n-r,r}^2 &\le c'_m(\lambda+n^{-2m})\sum_{r=1}^{n-1}\sum_{g=0}^{s-1}\sum_{k=0}^{\infty}(|f^0_{2(kN+gn+r)}|^2+|f^0_{2(kN+gn+r)-1}|^2)\\ &\quad\times(2\pi(kN+gn+r))^{2m}. \qquad (41)\end{aligned}$$

Combining (40) and (41) we get that

$$\begin{aligned}\sum_{r=1}^{n-1}\sum_{j=1}^{s}|\delta_{j,r}|^2 &\le 2c'_m(\lambda+n^{-2m})N\sum_{r=1}^{n-1}\sum_{g=0}^{s-1}\sum_{k=0}^{\infty}(|f^0_{2(kN+gn+r)}|^2+|f^0_{2(kN+gn+r)-1}|^2)\\ &\quad\times(2\pi(kN+gn+r))^{2m}. \qquad (42)\end{aligned}$$

To the end of proof of (5), by (34) we have for $1\le j\le s$,

$$\begin{aligned}\delta_{j,0} &\equiv \sum_{p=0}^{\infty}\left(\sum_{v=1}^{n}f^0_{2(pn+v)-1}b_{p,v,0}x_0^*\Phi^T_{2(pn+v)-1,j}+\sum_{v=1}^{n}f^0_{2(pn+v)}b_{p,v,0}x_0^*\Phi^T_{2(pn+v),j}\right)\\ &= \sum_{p=0}^{\infty}\left(f^0_{2(pn+n)-1}b_{p,n,0}x_0^*\Phi^T_{2(pn+n)-1,j}+f^0_{2(pn+n)}b_{p,n,0}x_0^*\Phi^T_{2(pn+n),j}\right)\\ &= \sqrt{\frac{n}{2}}\sum_{p=0}^{\infty}\Big[(f^0_{2(pn+n)-1}-\sqrt{-1}f^0_{2(pn+n)})b_{p,n,0}\eta^{-(p+1)n(j-1)}\\ &\quad+(f^0_{2(pn+n)-1}+\sqrt{-1}f^0_{2(pn+n)})b_{p,n,0}\eta^{(p+1)n(j-1)}\Big]\\ &= \sqrt{\frac{n}{2}}\sum_{g=0}^{s-1}\Big[\sum_{k=0}^{\infty}(f^0_{2(kN+gn+n)-1}-\sqrt{-1}f^0_{2(kN+gn+n)})b_{g,n,0}\eta^{-(gn+n)(j-1)}\\ &\quad+\sum_{k=0}^{\infty}(f^0_{2(kN+gn+n)-1}+\sqrt{-1}f^0_{2(kN+gn+n)})b_{g,n,0}\eta^{(gn+n)(j-1)}\Big]\\ &= \sqrt{\frac{n}{2}}\sum_{g=0}^{s-1}\Big[C_{g,0}b_{g,n,0}\eta^{-(gn+n)(j-1)}+D_{g,n}b_{g,n,0}\eta^{(gn+n)(j-1)}\Big], \qquad (43)\end{aligned}$$

which, together with Cauchy-Schwartz inequality, (37)–(38), and the trivial fact $|b_{g,n,0}|\le 2$ for $0\le g\le s-1$, leads to

$$\begin{aligned}\sum_{j=1}^{s}|\delta_{j,0}|^2 &\le n\sum_{j=1}^{s}\Big|\sum_{g=0}^{s-1}C_{g,0}b_{g,n,0}\eta^{-(gn+n)(j-1)}\Big|^2+n\sum_{j=1}^{s}\Big|\sum_{g=0}^{s-1}D_{g,n}b_{g,n,0}\eta^{(gn+n)(j-1)}\Big|^2\\ &= N\left(\sum_{g=0}^{s-1}|C_{g,0}|^2b_{g,n,0}^2+\sum_{g=0}^{s-1}|D_{g,n}|^2b_{g,n,0}^2\right)\\ &\le 2c'_mn^{-2m}N\sum_{g=0}^{s-1}\sum_{k=0}^{\infty}(|f^0_{2(kN+gn+n)-1}|^2+|f^0_{2(kN+gn+n)}|^2)\times(2\pi(kN+gn+n))^{2m}. \qquad (44)\end{aligned}$$

Combining (42) and (44) we get that

$$\begin{aligned}\sum_{j=1}^{s}\sum_{i=1}^{n}(E\{\bar{f}\}(t_{i,j})-f_0(t_{i,j}))^2 &= \sum_{r=0}^{n-1}\sum_{g=0}^{s-1}|\delta_{j,r}|^2\\ &\le 2c'_m(\lambda+n^{-2m})N\sum_{i=1}^{n}\sum_{g=0}^{s-1}\sum_{k=0}^{\infty}(|f^0_{2(kN+gn+i)}|^2+|f^0_{2(kN+gn+i)-1}|^2)\\ &\quad\times(2\pi(kN+gn+i))^{2m}=2c'_m(\lambda+n^{-2m})NJ(f_0). \qquad (45)\end{aligned}$$

Next we will apply (45) to show (5). Since $\widehat{f}_j$ is the minimizer of $\ell_{j,n,\lambda}(f)$, it satisfies for $1\le j\le s$,

$$-\frac{1}{n}\sum_{i=1}^{n}(Y_{i,j}-\widehat{f}_j(t_{i,j}))K_{t_{i,j}}+\lambda\widehat{f}_j=0.$$

Taking expectations, we get that

$$\frac{1}{n}\sum_{i=1}^{n}(E\{\widehat{f}_j\}(t_{i,j})-f_0(t_{i,j}))K_{t_{i,j}}+\lambda E\{\widehat{f}_j\},$$

therefore, $E\{\widehat{f}_j\}$ is the minimizer to the following functional

$$\ell_{0j}(f)=\frac{1}{2n}\sum_{i=1}^{n}(f(t_{i,j})-f_0(t_{i,j}))^2+\frac{\lambda}{2}J(f).$$

Define $g_j=E\{\widehat{f}_j\}$. Since $\ell_{0j}(g_j)\le\ell_{0j}(f_0)$, we get

$$\frac{1}{2n}\sum_{i=1}^{n}(g_j(t_{i,j})-f_0(t_{i,j}))^2+\frac{\lambda}{2}J(g_j)\le\frac{\lambda}{2}J(f_0).$$

This means that $J(g_j)\le J(f_0)$, leading to

$$\|\frac{1}{s}\sum_{j=1}^{s}g_j^{(m)}\|_2\le\frac{1}{s}\sum_{j=1}^{s}\|g_j^{(m)}\|_2\le\sqrt{J(f_0)}. \qquad (46)$$

Note that $E\{\bar{f}\} = \frac{1}{s}\sum_{j=1}^{s} g_j$. Define $g(t) = (E\{\bar{f}\}(t) - f_0(t))^2$. By (Eggermont and LaRiccia, 2009, Lemma (2.24), pp. 58), (46) and $m \geq 1$ we get that

$$\begin{aligned}\Big|\frac{1}{N}\sum_{l=0}^{N-1} g(l/N) - \int_0^1 g(t)dt\Big| &\leq \frac{2}{N}\int_0^1 \big|\frac{1}{s}\sum_{j=1}^{s} g_j(t) - f_0(t)\big| \times \big|\frac{1}{s}\sum_{j=1}^{s} g_j'(t) - f_0'(t)\big|dt \\ &\leq \frac{2}{N}\|\frac{1}{s}\sum_{j=1}^{s} g_j - f_0\|_2 \times \|\frac{1}{s}\sum_{j=1}^{s} g_j' - f_0'\|_2 \\ &\leq \frac{2}{N}\|\frac{1}{s}\sum_{j=1}^{s} g_j^{(m)} - f_0^{(m)}\|_2^2 \leq \frac{8J(f_0)}{N}. \end{aligned} \tag{47}$$

Combining (45) and (47) we get that

$$\|E\{\bar{f}\} - f_0\|_2^2 \leq c_m^2 J(f_0)(\lambda + n^{-2m} + N^{-1}),$$

where $c_m^2 = \max\{8, 2c_m'\}$. This completes the proof of (5). ■

Proof [Proof of Theorem 3.2] Suppose $f_0 = \sum_{\nu=1}^{\infty} f_\nu^0 \varphi_\nu$ with f_ν^0 satisfying

$$|f_\nu^0|^2 = \begin{cases} Cn^{-1}(2\pi(n+r))^{-2m}, & \nu = 2(n+r)-1, 1 \leq r \leq n/2, \\ 0, & \text{otherwise.} \end{cases} \tag{48}$$

It is easy to see that $J(f_0) = \sum_{1\leq r\leq n/2} |f_{2(n+r)-1}^0|^2 (2\pi(n+r))^{2m} \leq C$.

Consider the decomposition (34) and let $\delta_{j,r}$ be defined as in (35) and (43). It can be easily checked that $C_{g,r} = 0$ for $1 \leq r \leq n/2$ and $0 \leq g \leq s-1$. Furthermore, for $1 \leq r \leq n/2$,

$$\begin{aligned}\lambda_{c,r} - a_{1,r} &= \sum_{u=0}^{\infty}(2\pi(un+r))^{-2m} + \sum_{u=1}^{\infty}(2\pi(un-r))^{-2m} - \sum_{u=0}^{\infty}(2\pi(uN+n+r))^{-2m} \\ &\quad - \sum_{u=1}^{\infty}(2\pi(uN-n-r))^{-2m} \geq (2\pi r)^{-2m}.\end{aligned}$$

Therefore,

$$\begin{aligned} b_{1,r,r}^2 &= \left(\frac{\lambda + \lambda_{c,r} - a_{1,r}}{\lambda + \lambda_{c,r}}\right)^2 \\ &\geq \left(\frac{\lambda + (2\pi r)^{-2m}}{\lambda + 2(1+\bar{c}_m)(2\pi r)^{-2m}}\right)^2 \geq \frac{1}{4(1+\bar{c}_m)^2}. \end{aligned} \tag{49}$$

Using (36) and (49), we have

$$\begin{aligned}\sum_{j=1}^{s}(\bar{\boldsymbol{f}}_j - \mathbf{f}_{0,j})^T(\bar{\boldsymbol{f}}_j - \mathbf{f}_{0,j}) &= \sum_{j=1}^{s}\sum_{r=0}^{n-1}|\delta_{j,r}|^2 \\ &\geq \sum_{1\leq r\leq n/2}\sum_{j=1}^{s}|\delta_{j,r}|^2 \\ &= \sum_{1\leq r\leq n/2}\frac{N}{2}\sum_{g=0}^{s-1}|C_{g,r}b_{g,n-r,r} + D_{s-1-g,r}b_{s-1-g,r,r}|^2 \\ &= \sum_{1\leq r\leq n/2}\frac{N}{2}\sum_{g=0}^{s-1}|D_{s-1-g,r}|^2 b_{s-1-g,r,r}^2 \\ &= \sum_{1\leq r\leq n/2}\frac{N}{2}\sum_{g=0}^{s-1}|D_{g,r}|^2 b_{g,r,r}^2 \\ &\geq \sum_{1\leq r\leq n/2}\frac{N}{2}|D_{1,r}|^2 b_{1,r,r}^2 \\ &\geq \frac{N}{8(1+\bar{c}_m)^2}\sum_{1\leq r\leq n/2}|f_{2(n+r)-1}^0|^2 \\ &\geq \frac{NC}{16(3\pi)^{2m}(1+\bar{c}_m)^2}n^{-2m} \equiv a_m NC n^{-2m}, \end{aligned}$$

where $a_m = \frac{1}{16(3\pi)^{2m}(1+\bar{c}_m)^2} < 1$ is an absolute constant depending on m only. Then the conclusion follows by (47). Proof is completed. ■

6.2 Proofs in Section 4

Proof [Proof of Theorem 4.1] For $1 \leq j, l \leq s$, define

$$\begin{aligned}\Omega_{j,l} &= \frac{1}{n}\sum_{\nu=1}^{\infty}\frac{\Phi_{\nu,j}^T\Phi_{\nu,l}}{\gamma_\nu^2}, \\ \widetilde{\sigma}_{j,l,r} &= \frac{2}{n}\sum_{k=1}^{\infty}\frac{\cos\left(2\pi k\left(\frac{r}{n} - \frac{j-l}{N}\right)\right)}{(2\pi k)^{4m}},\ r = 0, 1, \ldots, n-1. \end{aligned}$$

Clearly $\Omega_{j,l}$ is a circulant matrix with elements $\widetilde{\sigma}_{j,l,0}, \widetilde{\sigma}_{j,l,1}, \ldots, \widetilde{\sigma}_{j,l,n-1}$. Furthermore, by arguments (24)–(26) we get that

$$\Omega_{j,l} = M\Gamma_{j,l}M^*, \tag{50}$$

where M is the same as in (19), and $\Gamma_{j,l} = \mathrm{diag}(\delta_{j,l,0}, \delta_{j,l,1}, \ldots, \delta_{j,l,n-1})$, with $\delta_{j,l,r}$, for $r = 1, \ldots, n-1$, given by the following

$$\delta_{j,l,r} = \sum_{q=1}^{\infty}\frac{\eta^{-(qn-r)(j-l)}}{[2\pi(qn-r)]^{4m}} + \sum_{q=0}^{\infty}\frac{\eta^{(qn+r)(j-l)}}{[2\pi(qn+r)]^{4m}}, \tag{51}$$

and for $r=0$, given by

$$\delta_{j,l,0}=\sum_{q=1}^{\infty}\frac{\eta^{qn(j-l)}+\eta^{-qn(j-l)}}{(2\pi qn)^{4m}}. \tag{52}$$

Define $A=\mathrm{diag}(\underbrace{(\Sigma+\lambda I_n)^{-1},\ldots,(\Sigma+\lambda I_n)^{-1}}_{s})$ and

$$B=\begin{pmatrix}\Omega_{1,1} & \Omega_{1,2} & \cdots & \Omega_{1,s}\\ \Omega_{2,1} & \Omega_{2,2} & \cdots & \Omega_{2,s}\\ \cdots & \cdots & \cdots & \cdots\\ \Omega_{s,1} & \Omega_{s,2} & \cdots & \Omega_{s,s}\end{pmatrix}.$$

Note that B is $N\times N$ symmetric. Under H_0, it can be shown that

$$\begin{aligned}\|\bar f\|_2^2 &= \sum_{\nu=1}^{\infty}\left(\frac{\sum_{l=1}^{s}\Phi_{\nu,l}(\Sigma+\lambda I_n)^{-1}\boldsymbol{\epsilon}_l}{N\gamma_\nu^2}\right)^2\\ &= \frac{1}{Ns}\sum_{j,l=1}^{s}\boldsymbol{\epsilon}_j^T(\Sigma+\lambda I_n)^{-1}\left(\frac{1}{n}\sum_{\nu=1}^{\infty}\frac{\Phi_{\nu,j}^T\Phi_{\nu,l}}{\gamma_\nu^2}\right)(\Sigma+\lambda I_n)^{-1}\boldsymbol{\epsilon}_l\\ &= \frac{1}{Ns}\sum_{j,l=1}^{s}\boldsymbol{\epsilon}_j^T(\Sigma+\lambda I_n)^{-1}\Omega_{j,l}(\Sigma+\lambda I_n)^{-1}\boldsymbol{\epsilon}_l\\ &= \frac{1}{Ns}\boldsymbol{\epsilon}^T ABA\boldsymbol{\epsilon}=\frac{1}{Ns}\boldsymbol{\epsilon}^T\Delta\boldsymbol{\epsilon},\end{aligned}$$

where $\boldsymbol{\epsilon}=(\boldsymbol{\epsilon}_1^T,\ldots,\boldsymbol{\epsilon}_s^T)^T$ and $\Delta\equiv ABA$.

This implies that $T_{N,\lambda}=\boldsymbol{\epsilon}^T\Delta\boldsymbol{\epsilon}/(Ns)$ with $\mu_{N,\lambda}=\mathrm{trace}(\Delta)/(Ns)$ and $\sigma_{N,\lambda}^2=2\mathrm{trace}(\Delta^2)/(Ns)^2$. Define $U=(T_{N,\lambda}-\mu_{N,\lambda})/\sigma_{N,\lambda}$. Then for any $t\in(-1/2,1/2)$,

$$\begin{aligned}\log E\{\exp(tU)\} &= \log E\{\exp(t\boldsymbol{\epsilon}^T\Delta\boldsymbol{\epsilon}/(Ns\sigma_{N,\lambda}))\}-t\mu_{N,\lambda}/\sigma_{N,\lambda}\\ &= -\frac{1}{2}\log\det(I_N-2t\Delta/(Ns\sigma_{N,\lambda}))-t\mu_{N,\lambda}/\sigma_{N,\lambda}\\ &= t\mathrm{trace}(\Delta)/(Ns\sigma_{N,\lambda})+t^2\mathrm{trace}(\Delta^2)/((Ns)^2\sigma_{N,\lambda}^2)\\ &\quad +O(t^3\mathrm{trace}(\Delta^3)/((Ns)^3\sigma_{N,\lambda}^3))-t\mu_{N,\lambda}/\sigma_{N,\lambda}\\ &= t^2/2+O(t^3\mathrm{trace}(\Delta^3)/((Ns)^3\sigma_{N,\lambda}^3)).\end{aligned}$$

It remains to show that $\mathrm{trace}(\Delta^3)/((Ns)^3\sigma_{N,\lambda}^3)=o(1)$ in order to conclude the proof.

In other words, we need to study $\mathrm{trace}(\Delta^2)$ (used in $\sigma_{N,\lambda}^2$) and $\mathrm{trace}(\Delta^3)$. We start from the former. By direct calculations, we get

$$\begin{aligned}\mathrm{trace}(\Delta^2) &= \mathrm{trace}(A^2BA^2B)\\ &= \sum_{l=1}^{s}\mathrm{trace}\left(\sum_{j=1}^{s}M(\Lambda_c+\lambda I_n)^{-2}\Gamma_{l,j}(\Lambda_c+\lambda I_n)^{-2}\Gamma_{j,l}M^*\right)\\ &= \sum_{j,l=1}^{s}\mathrm{trace}\left((\Lambda_c+\lambda I_n)^{-2}\Gamma_{l,j}(\Lambda_c+\lambda I_n)^{-2}\Gamma_{j,l}\right)=\sum_{j,l=1}^{s}\sum_{r=0}^{n-1}\frac{|\delta_{j,l,r}|^2}{(\lambda+\lambda_{c,r})^4}.\end{aligned}$$

For $1\le g\le s$ and $0\le r\le n-1$, define

$$A_{g,r}=\sum_{p=0}^{\infty}\frac{1}{[2\pi(pN+gn-r)]^{4m}}.$$

Using (51) and (52), it can be shown that for $r=1,2,\ldots,n-1$,

$$\begin{aligned}\sum_{j,l=1}^{s}|\delta_{j,l,r}|^2 &= \sum_{j,l=1}^{s}\left|\sum_{g=1}^{s}A_{g,r}\eta^{-gn(j-l)}+\sum_{g=1}^{s}A_{g,n-r}\eta^{(g-1)n(j-l)}\right|^2\\ &= \sum_{j,l=1}^{s}\left(\sum_{g,g'=1}^{s}A_{g,r}A_{g',r}\eta^{-(g-g')n(j-l)}+\sum_{g,g'=1}^{s}A_{g,n-r}A_{g',n-r}\eta^{(g-g')n(j-l)}\right.\\ &\quad \left.+\sum_{g,g'=1}^{s}A_{g,r}A_{g',n-r}\eta^{-(g+g'-1)n(j-l)}+\sum_{g,g'=1}^{s}A_{g,n-r}A_{g',r}\eta^{(g+g'-1)n(j-l)}\right)\\ &= s^2\sum_{g=1}^{s}A_{g,r}^2+s^2\sum_{g=1}^{s}A_{g,n-r}^2+2s^2\sum_{g=1}^{s}A_{g,r}A_{s+1-g,n-r}\\ &\ge s^2\sum_{g=1}^{s}A_{g,r}^2+s^2\sum_{g=1}^{s}A_{g,n-r}^2.\end{aligned}\tag{53}$$

Since

$$\sum_{g=1}^{s}A_{g,r}^2=\sum_{g=1}^{s}\left(\sum_{p=0}^{\infty}\frac{1}{[2\pi(pN+gn-r)]^{4m}}\right)^2\ge\frac{1}{[2\pi(n-r)]^{8m}}, \tag{54}$$

we get that

$$\begin{aligned}\mathrm{trace}(\Delta^2) &\ge \sum_{r=1}^{n-1}\frac{s^2(\sum_{g=1}^{s}A_{g,r}^2+\sum_{g=1}^{s}A_{g,n-r}^2)}{(\lambda+\lambda_{c,r})^4}\\ &\ge s^2\sum_{r=1}^{n-1}\frac{\frac{1}{[2\pi(n-r)]^{8m}}+\frac{1}{(2\pi r)^{8m}}}{(\lambda+\lambda_{c,r})^4}\\ &\ge \frac{2s^2}{(2+2\bar c_m)^4}\sum_{1\le r\le n/2}\frac{\frac{1}{(2\pi r)^{8m}}}{(\lambda+\frac{1}{(2\pi r)^{2m}})^4}\\ &= \frac{s^2}{8(1+\bar c_m)^4}\sum_{1\le r\le n/2}\frac{1}{(1+(2\pi rh)^{2m})^4}\\ &\ge \frac{s^2}{8(1+\bar c_m)^4}h^{-1}\int_h^{nh/2}\frac{1}{(1+(2\pi x)^{2m})^4}dx.\end{aligned}$$

Meanwhile, (53) indicates that for $1\le r\le n-1$,

$$\sum_{j,l=1}^{s}|\delta_{j,l,r}|^2\le 2s^2\sum_{g=1}^{s}A_{g,r}^2+2s^2\sum_{g=1}^{s}A_{g,n-r}^2.$$

From (54) we get that for $1 \le r \le n-1$,

$$\sum_{g=1}^{s} A_{g,r}^2 \le \frac{c_m}{(2\pi(n-r))^{8m}},$$

where $c_m > 0$ is a constant depending on m only.

Similar analysis to (53) shows that

$$\begin{aligned}\sum_{j,l=1}^{s} |\delta_{j,l,0}|^2 &= \sum_{j,l=1}^{s} \left| \sum_{g=1}^{s} A_{g,0}(\eta^{gn(j-l)} + \eta^{-gn(j-l)}) \right|^2 \\ &= 2s^2 \sum_{g=1}^{s} A_{g,0}^2 + 2s^2 \sum_{g=1}^{s-1} A_{g,0} A_{s-g,0} + 2s^2 A_{s,0}^2 \\ &\le 4s^2 \sum_{g=1}^{s} A_{g,0}^2 \le c_m s^2 (2\pi n)^{-8m}.\end{aligned}$$

Therefore,

$$\begin{aligned}\mathrm{trace}(\Delta^2) &\le \frac{4s^2 \sum_{g=1}^{s} A_{g,0}^2}{(\lambda+\lambda_{c,0})^4} + 2s^2 \sum_{r=1}^{n-1} \frac{\sum_{g=1}^{s} A_{g,r}^2 + \sum_{g=1}^{s} A_{g,n-r}^2}{(\lambda+\lambda_{c,r})^4} \\ &\le 4c_m s^2 \sum_{r=1}^{n} \frac{1}{(1+(2\pi r h)^{2m})^4} \\ &\le 4c_m s^2 h^{-1} \int_0^{nh} \frac{1}{(1+(2\pi x)^{2m})^4} dx.\end{aligned}$$

By the above statements, we get that

$$\sigma_{N,\lambda}^2 = 2\mathrm{trace}(\Delta^2)/(Ns)^2 \asymp \begin{cases} \frac{n}{N^2}, & \text{if } nh \to 0, \\ \frac{1}{N^2 h}, & \text{if } \lim_N nh > 0. \end{cases} \tag{55}$$

To the end, we look at the trace of Δ^3. By direct examinations, we have

$$\begin{aligned}\mathrm{trace}(\Delta^3) &= \mathrm{trace}(ABA^2BA^2BA) \\ &= \sum_{j,k=1}^{s} \mathrm{trace}\Big([\sum_{l=1}^{s} M(\Lambda_c+\lambda I_n)^{-2}\Gamma_{j,l}(\Lambda_c+\lambda I_n)^{-2}\Gamma_{l,k}M^*] \\ &\quad \times M(\Lambda_c+\lambda I_n)^{-2}\Gamma_{k,j}M^* \Big) \\ &= \sum_{j,k,l=1}^{s} \mathrm{trace}\left((\Lambda_c+\lambda I_n)^{-2}\Gamma_{j,l}(\Lambda_c+\lambda I_n)^{-2}\Gamma_{l,k}(\Lambda_c+\lambda I_n)^{-2}\Gamma_{k,j} \right) \\ &= \sum_{j,k,l=1}^{s} \sum_{r=0}^{n-1} \frac{\delta_{j,l,r}\delta_{l,k,r}\delta_{k,j,r}}{(\lambda+\lambda_{c,r})^6}.\end{aligned}$$

For $r = 1, 2, \ldots, n-1$, it can be shown that

$$\begin{aligned}\delta_{j,l,r}\delta_{l,k,r}\delta_{k,j,r} &= \left(\sum_{q=1}^{\infty} \frac{\eta^{-qn(j-l)}}{(2\pi(qn-r))^{4m}} + \sum_{q=0}^{\infty} \frac{\eta^{qn(j-l)}}{(2\pi(qn+r))^{4m}} \right) \\ &\quad \times \left(\sum_{q=1}^{\infty} \frac{\eta^{-qn(l-k)}}{(2\pi(qn-r))^{4m}} + \sum_{q=0}^{\infty} \frac{\eta^{qn(l-k)}}{(2\pi(qn+r))^{4m}} \right) \\ &\quad \times \left(\sum_{q=1}^{\infty} \frac{\eta^{-qn(k-j)}}{(2\pi(qn-r))^{4m}} + \sum_{q=0}^{\infty} \frac{\eta^{qn(k-j)}}{(2\pi(qn+r))^{4m}} \right).\end{aligned} \tag{56}$$

We next proceed to show that for $1 \le r \le n-1$,

$$\sum_{l,j,k=1}^{s} \delta_{j,l,r}\delta_{l,k,r}\delta_{k,j,r} \le \frac{96m}{12m-1} \left(\frac{4m}{4m-1} \right)^3 s^3 \left(\frac{1}{(2\pi(n-r))^{12m}} + \frac{1}{(2\pi r)^{12m}} \right). \tag{57}$$

Using the trivial fact that $A_{g,r} \le \frac{4m}{4m-1} \times \frac{1}{(2\pi(gn-r))^{4m}}$, the first term in (56) satisfies

$$\begin{aligned}&\sum_{j,l,k=1}^{s} \sum_{q_1=1}^{\infty} \frac{\eta^{-q_1 n(j-l)}}{(2\pi(q_1 n-r))^{4m}} \sum_{q_2=1}^{\infty} \frac{\eta^{-q_2 n(j-l)}}{(2\pi(q_2 n-r))^{4m}} \sum_{q_3=1}^{\infty} \frac{\eta^{-q_3 n(j-l)}}{(2\pi(q_3 n-r))^{4m}} \\ &= \sum_{j,l,k=1}^{s} \sum_{g_1=1}^{s} A_{g_1,r}\eta^{-g_1 n(j-l)} \sum_{g_2=1}^{s} A_{g_2,r}\eta^{-g_2 n(l-k)} \sum_{g_3=1}^{s} A_{g_3,r}\eta^{-g_3 n(k-j)} \\ &= \sum_{g_1,g_2,g_3=1}^{s} A_{g_1,r}A_{g_2,r}A_{g_3,r} \sum_{j,l,k=1}^{s} \eta^{-g_1 n(j-l)}\eta^{-g_2 n(l-k)}\eta^{-g_3 n(k-j)} \\ &= \sum_{g_1,g_2,g_3=1}^{s} A_{g_1,r}A_{g_2,r}A_{g_3,r} \sum_{j=1}^{s} \eta^{(g_3-g_1)n(j-1)} \sum_{l=1}^{s} \eta^{(g_1-g_2)n(l-1)} \sum_{k=1}^{s} \eta^{(g_2-g_3)n(k-1)} \\ &= s^3 \sum_{g=1}^{s} A_{g,r}^3 \\ &\le \left(\frac{4m}{4m-1} \right)^3 s^3 \sum_{g=1}^{s} \frac{1}{(2\pi(gn-r))^{12m}} \\ &\le \frac{12m}{12m-1} \left(\frac{4m}{4m-1} \right)^3 s^3 \frac{1}{(2\pi(n-r))^{12m}}.\end{aligned}$$

Similarly, one can show that all other terms in (56) are upper bounded by

$$\frac{12m}{12m-1} \left(\frac{4m}{4m-1} \right)^3 s^3 \left(\frac{1}{(2\pi(n-r))^{12m}} + \frac{1}{(2\pi r)^{12m}} \right).$$

Therefore, (57) holds. It can also be shown by (52) and similar analysis that

$$\sum_{j,l,k=1}^{s} \delta_{j,l,0}\delta_{l,k,0}\delta_{k,j,0} \le s^3 (2\pi n)^{-12m}. \tag{58}$$

Using (57) and (58), one can get that

$$\begin{aligned}\mathrm{trace}(\Delta^3) &= \sum_{l,j,k=1}^{s}\sum_{r=0}^{n-1}\frac{\delta_{j,l,r}\delta_{l,k,r}\delta_{k,j,r}}{(\lambda+\lambda_{c,r})^6}\\ &\lesssim s^3\sum_{r=1}^{n-1}\frac{\frac{1}{(2\pi(n-r))^{12m}}+\frac{1}{(2\pi r)^{12m}}}{(\lambda+\lambda_{c,r})}+s^3\frac{\frac{1}{(2\pi n)^{12m}}}{(\lambda+\lambda_{c,0})^{12m}}\\ &\lesssim s^3\sum_{r=1}^{n}\frac{1}{(1+(2\pi rh)^{2m})^6}\\ &\lesssim s^3h^{-1}\int_0^{nh}\frac{1}{(1+(2\pi x)^{2m})^6}dx \asymp \begin{cases} s^3n, & \text{if } nh\to 0,\\ s^3h^{-1}, & \text{if } \lim_N nh>0.\end{cases}\end{aligned} \tag{59}$$

Combining (55) and (59), and using the assumptions $n\to\infty$, $h\to 0$, we get that

$$\mathrm{trace}(\Delta^3)/((Ns)^3\sigma_{N,\lambda}^3) \lesssim \begin{cases} n^{-1/2}, & \text{if } nh\to 0,\\ h^{1/2}, & \text{if } \lim_N nh>0.\end{cases} = o(1).$$

Proof is completed. ■

Proof [Proof of Theorem 4.2] Throughout the proof, we assume that data $Y_1,\ldots,Y_N$ are generated from the sequence of alternative hypotheses: $f\in\mathcal{B}$ and $\|f\|_2\geq C_\varepsilon d_{N,\lambda}$. Define $\mathbf{f}_j=(f(t_{1,j}),\ldots,f(t_{n,j}))^T$ for $1\leq j\leq s$. Then it can be shown that

$$\begin{aligned}NsT_{N,\lambda} &= Ns\sum_{\nu=1}^{\infty}\bar{f}_\nu^2\\ &= \sum_{j,l=1}^{s}Y_j^T(\Sigma+\lambda I_n)^{-1}\Omega_{j,l}(\Sigma+\lambda I_n)^{-1}Y_l\\ &= \sum_{j,l=1}^{s}Y_j^TM(\Lambda_c+\lambda I_n)^{-1}\Gamma_{j,l}(\Lambda_c+\lambda I_n)^{-1}M^*Y_l\\ &= \sum_{j,l=1}^{s}\mathbf{f}_j^TM(\Lambda_c+\lambda I_n)^{-1}\Gamma_{j,l}(\Lambda_c+\lambda I_n)^{-1}M^*\mathbf{f}_l\\ &\quad+\sum_{j,l=1}^{s}\mathbf{f}_j^TM(\Lambda_c+\lambda I_n)^{-1}\Gamma_{j,l}(\Lambda_c+\lambda I_n)^{-1}M^*\boldsymbol{\epsilon}_l\\ &\quad+\sum_{j,l=1}^{s}\boldsymbol{\epsilon}_j^TM(\Lambda_c+\lambda I_n)^{-1}\Gamma_{j,l}(\Lambda_c+\lambda I_n)^{-1}M^*\mathbf{f}_l\\ &\quad+\sum_{j,l=1}^{s}\boldsymbol{\epsilon}_j^TM(\Lambda_c+\lambda I_n)^{-1}\Gamma_{j,l}(\Lambda_c+\lambda I_n)^{-1}M^*\boldsymbol{\epsilon}_l\\ &\equiv T_1+T_2+T_3+T_4.\end{aligned} \tag{60}$$

Next we will analyze all the four terms in the above. Let $f=\sum_{\nu=1}^{\infty}f_\nu\varphi_\nu$. For $0\leq r\leq n-1$ and $1\leq l\leq s$, define $d_{l,r}=x_r^*\mathbf{f}_l$. Then it holds that

$$d_{l,r}=\sum_{p=0}^{\infty}\sum_{v=1}^{n}f_{2(pn+v)-1}x_r^*\Phi_{2(pn+v)-1,l}^T+\sum_{p=0}^{\infty}\sum_{v=1}^{n}f_{2(pn+v)}x_r^*\Phi_{2(pn+v),l}^T.$$

Using (27) and (28), we get that for $1\leq r\leq n-1$,

$$\begin{aligned}d_{l,r} &= \sum_{p=0}^{\infty}\sum_{v=1}^{n-1}f_{2(pn+v)-1}\left(\sqrt{\frac{n}{2}}\right)\left(\eta^{-(pn+v)(l-1)}I(r+v=n)+\eta^{(pn+v)(l-1)}I(r=v)\right)\\ &\quad+\sum_{p=0}^{\infty}\sum_{v=1}^{n-1}f_{2(pn+v)}\left(-\sqrt{-\frac{n}{2}}\right)\left(\eta^{-(pn+v)(l-1)}I(r+v=n)-\eta^{(pn+v)(l-1)}I(r=v)\right)\\ &= \sqrt{\frac{n}{2}}\sum_{p=0}^{\infty}\Big[(f_{2(pn+n-r)-1}-\sqrt{-1}f_{2(pn+n-r)})\eta^{-(pn+n-r)(l-1)}\\ &\quad+(f_{2(pn+r)-1}+\sqrt{-1}f_{2(pn+r)})\eta^{(pn+r)(l-1)}\Big],\end{aligned} \tag{61}$$

and for $r=0$,

$$\begin{aligned}d_{l,0} &= \sum_{p=0}^{\infty}f_{2(pn+n)-1}x_0^*\Phi_{2(pn+n)-1,l}^T+\sum_{p=0}^{\infty}f_{2(pn+n)}x_0^*\Phi_{2(pn+n),l}^T\\ &= \sqrt{\frac{n}{2}}\sum_{p=0}^{\infty}\Big[(f_{2(pn+n)-1}-\sqrt{-1}f_{2(pn+n)})\eta^{-(pn+n)(l-1)}\\ &\quad+(f_{2(pn+n)-1}+\sqrt{-1}f_{2(pn+n)})\eta^{(pn+n)(l-1)}\Big].\end{aligned} \tag{62}$$

We first look at T_1. It can be examined directly that

$$\begin{aligned}T_1 &= \sum_{j,l=1}^{s}(\overline{d_{j,0}},\ldots,\overline{d_{j,n-1}})\mathrm{diag}\left(\frac{\delta_{j,l,0}}{(\lambda+\lambda_{c,0})^2},\ldots,\frac{\delta_{j,l,n-1}}{(\lambda+\lambda_{c,n-1})^2}\right)\times(d_{l,0},\ldots,d_{l,n-1})^T\\ &= \sum_{r=0}^{n-1}\frac{\sum_{j,l=1}^{s}\delta_{j,l,r}\overline{d_{j,r}}d_{l,r}}{(\lambda+\lambda_{c,r})^2}.\end{aligned} \tag{63}$$

Using similar arguments as (29)–(33), one can show that for $p\geq 0$, $1\leq v\leq n$, $0\leq r\leq n-1$ and $1\leq j\leq s$,

$$\begin{aligned}\frac{1}{s}\sum_{l=1}^{s}\delta_{j,l,r}x_r^*\Phi_{2(pn+v)-1,l}^T &= b_{p,v}x_r^*\Phi_{2(pn+v)-1,j}^T,\\ \frac{1}{s}\sum_{l=1}^{s}\delta_{j,l,r}x_r^*\Phi_{2(pn+v),l}^T &= b_{p,v}x_r^*\Phi_{2(pn+v),j}^T,\end{aligned} \tag{64}$$

where

$$b_{p,v}=\begin{cases}\sum_{u\geq -p/s}\frac{1}{(2\pi(uN+pn+v))^{4m}}+\sum_{u\geq (p+1)/s}\frac{1}{(2\pi(uN-pn-v))^{4m}}, & \text{for } 1\leq v\leq n-1,\\ \sum_{u\geq -p/s}\frac{1}{(2\pi(uN+pn+n))^{4m}}+\sum_{u\geq (p+2)/s}\frac{1}{(2\pi(uN-pn-n))^{4m}}, & \text{for } v=n.\end{cases}$$

By (64), we have

$$\begin{aligned}
\sum_{j,l=1}^{s}\delta_{j,l,r}\overline{d_{j,r}}d_{l,r} &= \sum_{j=1}^{s}\overline{d_{j,r}}\sum_{l=1}^{s}\delta_{j,l,r}d_{l,r}\\
&= \sum_{j=1}^{s}\overline{d_{j,r}}\sum_{l=1}^{s}\delta_{j,l,r}\left(\sum_{p=0}^{\infty}\sum_{v=1}^{n}f_{2(pn+v)-1}x_r^*\Phi_{2(pn+v)-1,l}\right.\\
&\quad\left.+\sum_{p=0}^{\infty}\sum_{v=1}^{n}f_{2(pn+v)}x_r^*\Phi^T_{2(pn+v),l}\right)\\
&= \sum_{j=1}^{s}\overline{d_{j,r}}\left(\sum_{p=0}^{\infty}\sum_{v=1}^{n}f_{2(pn+v)-1}\sum_{l=1}^{s}\delta_{j,l,r}x_r^*\Phi^T_{2(pn+v)-1,l}\right.\\
&\quad\left.+\sum_{p=0}^{\infty}\sum_{v=1}^{n}f_{2(pn+v)}\sum_{l=1}^{s}\delta_{j,l,r}x_r^*\Phi^T_{2(pn+v),l}\right)\\
&= s\sum_{j=1}^{s}\overline{d_{j,r}}\left(\sum_{p=0}^{\infty}\sum_{v=1}^{n}f_{2(pn+v)-1}b_{p,v}x_r^*\Phi^T_{2(pn+v)-1,j}+\sum_{p=0}^{\infty}\sum_{v=1}^{n}f_{2(pn+v)}b_{p,v}x_r^*\Phi^T_{2(pn+v),j}\right).
\end{aligned}$$

It then follows from (61) and (62), trivial facts $b_{s-1-g,r}=b_{g,n-r}$ and $C_{g,n-r}=\overline{D_{g,r}}$ (both $C_{g,r}$ and $D_{g,r}$ are defined similarly as those in the proof of Theorem 3.1, but with f_0 therein replaced by f), and direct calculations that for $1\le r\le n-1$

$$\begin{aligned}
\sum_{j,l=1}^{s}\delta_{j,l,r}\overline{d_{j,r}}d_{l,r} &= \frac{sn}{2}\sum_{j=1}^{s}\sum_{p=0}^{\infty}\Big[(f_{2(pn+n-r)-1}+\sqrt{-1}f_{2(pn+n-r)})\eta^{(pn+n-r)(j-1)}\\
&\quad+(f_{2(pn+r)-1}-\sqrt{-1}f_{2(pn+r)})\eta^{-(pn+r)(j-1)}\Big]\\
&\quad\times\sum_{p=0}^{\infty}\Big[(f_{2(pn+n-r)-1}-\sqrt{-1}f_{2(pn+n-r)})b_{p,n-r}\eta^{-(pn+n-r)(j-1)}\\
&\quad+(f_{2(pn+r)-1}+\sqrt{-1}f_{2(pn+r)})b_{p,r}\eta^{(pn+r)(j-1)}\Big]\\
&= \frac{N}{2}\sum_{j=1}^{s}\sum_{g=0}^{s-1}\sum_{k=0}^{\infty}\Big[(f_{2(kN+gn+n-r)-1}+\sqrt{-1}f_{2(kN+gn+n-r)})\eta^{(gn+n-r)(j-1)}\\
&\quad+(f_{2(kN+gn+r)-1}-\sqrt{-1}f_{2(kN+gn+r)})\eta^{-(gn+r)(j-1)}\Big]\\
&\quad\times\sum_{g=0}^{s-1}\sum_{k=0}^{\infty}\Big[(f_{2(kN+gn+n-r)-1}-\sqrt{-1}f_{2(kN+gn+n-r)})b_{ks+g,n-r}\eta^{-(gn+n-r)(j-1)}\\
&\quad+(f_{2(kN+gn+r)-1}+\sqrt{-1}f_{2(kN+gn+r)})b_{ks+g,r}\eta^{(gn+r)(j-1)}\Big]\\
&= \frac{N}{2}\sum_{j=1}^{s}\left[\sum_{g=0}^{s-1}\overline{C_{g,r}}\eta^{(gn+n)(j-1)}+\sum_{g=0}^{s-1}\overline{D_{g,r}}\eta^{-gn(j-1)}\right]\\
&\quad\times\left[\sum_{g=0}^{s-1}b_{g,n-r}C_{g,r}\eta^{-(gn+n)(j-1)}+\sum_{g=0}^{s-1}b_{g,r}D_{g,r}\eta^{gn(j-1)}\right]\\
&= \frac{Ns}{2}\left(\sum_{g=0}^{s-1}b_{g,n-r}|C_{g,r}|^2+\sum_{g=0}^{s-1}b_{s-1-g,r}\overline{C_{g,r}}D_{s-1-g,r}\right.\\
&\quad\left.+\sum_{g=0}^{s-1}b_{s-1-g,n-r}\overline{D_{g,r}}C_{s-1-g,r}+\sum_{g=0}^{s-1}b_{g,r}|D_{g,r}|^2\right),
\end{aligned}$$

which leads to

$$\sum_{r=1}^{n-1}\frac{\sum_{j,l=1}^{s}\delta_{j,l,r}\overline{d_{j,r}}d_{l,r}}{(\lambda+\lambda_{c,r})^2}=\frac{Ns}{2}\sum_{r=1}^{n-1}\frac{\sum_{g=0}^{s-1}b_{g,r}|C_{s-1-g,r}+D_{g,r}|^2}{(\lambda+\lambda_{c,r})^2}. \tag{65}$$

Since $J(f)\le C$, equivalently, $\sum_{\nu=1}^{\infty}(f_{2\nu-1}^2+f_{2\nu}^2)(2\pi\nu)^{2m}\le C$, we get that

$$\sum_{1\le r\le n/2}(f_{2r-1}^2+f_{2r}^2)\ge\|f\|_2^2-C(2\pi n)^{-2m}. \tag{66}$$

Meanwhile, for $1\le r\le n/2$, using similar arguments as (40) and (41) one can show that there exists a constant c'_m relying on C and m s.t.

$$\begin{aligned}
|C_{s-1,r}+D_{0,r}|^2 &= \left(f_{2r-1}+\sum_{k=0}^{\infty}f_{2(kN+N-r)-1}+\sum_{k=1}^{\infty}f_{2(kN+r)-1}\right)^2\\
&\quad+\left(f_{2r}+\sum_{k=1}^{\infty}f_{2(kN+r)}-\sum_{k=0}^{\infty}f_{2(kN+N-r)}\right)^2\\
&\ge \frac{1}{2}(f_{2r-1}^2+f_{2r-1}^2)-c'_mN^{-2m},
\end{aligned}\tag{67}$$

and

$$\begin{aligned}
&|C_{s-1,r}+D_{0,r}|^2(2\pi r)^{2m}\\
\le\ & 4\left[(\sum_{k=0}^{\infty}f_{2(kN+N-r)-1})^2+(\sum_{k=0}^{\infty}f_{2(kN+N-r)})^2\right.\\
&\left.+(\sum_{k=0}^{\infty}f_{2(kN+r)-1})^2+(\sum_{k=0}^{\infty}f_{2(kN+r)})^2\right](2\pi r)^{2m}\\
\le\ & \left(4\sum_{k=0}^{\infty}f_{2(kN+N-r)-1}^2(2\pi(kN+N-r))^{2m}\sum_{k=0}^{\infty}(2\pi(kN+N-r))^{-2m}\right.\\
&+4\sum_{k=0}^{\infty}f_{2(kN+N-r)}^2(2\pi(kN+N-r))^{2m}\sum_{k=0}^{\infty}(2\pi(kN+N-r))^{-2m}\\
&+4\sum_{k=0}^{\infty}f_{2(kN+r)-1}^2(2\pi(kN+r))^{2m}\sum_{k=0}^{\infty}(2\pi(kN+r))^{-2m}\\
&\left.+4\sum_{k=0}^{\infty}f_{2(kN+r)}^2(2\pi(kN+r))^{2m}\sum_{k=0}^{\infty}(2\pi(kN+r))^{-2m}\right)\times(2\pi r)^{2m}\\
\le\ & \left(\frac{8m}{2m-1}\sum_{k=0}^{\infty}(f_{2(kN+N-r)-1}^2+f_{2(kN+N-r)}^2)\gamma_{kN+N-r}(2\pi(N-r))^{-2m}\right.\\
&\left.+\frac{8m}{2m-1}\sum_{k=0}^{\infty}(f_{2(kN+r)-1}^2+f_{2(kN+r)}^2)\gamma_{kN+r}(2\pi r)^{-2m}\right)\times(2\pi r)^{2m},
\end{aligned}$$

which, together with the fact $N \geq 2r$ for $1 \leq r \leq n/2$, leads to that

$$\sum_{1\leq r\leq n/2} |C_{s-1,r} + D_{0,r}|^2 (2\pi r)^{2m} \leq c'_m. \tag{68}$$

Furthermore, it can be verified that for $1 \leq r \leq n/2$,

$$\begin{aligned}\frac{\lambda_{c,r}^2 - b_{0,r}}{(\lambda+\lambda_{c,r})}(2\pi r)^{-2m} &\leq \frac{((2\pi r)^{-2m} + (2\pi(n-r))^{-2m} + \bar{c}_m(2\pi n)^{-2m})^2 - (2\pi r)^{-4m}}{((2\pi r)^{-2m} + (2\pi(n-r))^{-2m})^2}(2\pi r)^{-2m} \\ &\leq c'_m n^{-2m}, \end{aligned} \tag{69}$$

which leads to that

$$\begin{aligned}\frac{(\lambda+\lambda_{c,r})^2 - b_{0,r}}{(\lambda+\lambda_{c,r})^2}(2\pi r)^{-2m} &= \frac{\lambda^2 + 2\lambda\lambda_{c,r}}{(\lambda+\lambda_{c,r})^2}(2\pi r)^{-2m} + \frac{\lambda_{c,r}^2 - b_{0,r}}{(\lambda+\lambda_{c,r})^2}(2\pi r)^{-2m} \\ &\leq 2\lambda + c'_m n^{-2m}. \end{aligned} \tag{70}$$

Then, using (63)–(65) and (66)–(70) one gets that

$$\begin{aligned} T_1 &\geq \frac{Ns}{2}\sum_{1\leq r\leq n/2} \frac{b_{0,r}|C_{s-1,r} + D_{0,r}|^2}{(\lambda+\lambda_{c,r})^2} \\ &= \frac{Ns}{2}\left[\sum_{1\leq r\leq n/2} |C_{s-1,r} + D_{0,r}|^2 - \sum_{1\leq r\leq n/2} \frac{(\lambda+\lambda_{c,r})^2 - b_{0,r}}{(\lambda+\lambda_{c,r})^2}|C_{s-1,r} + D_{0,r}|^2\right] \\ &\geq \frac{Ns}{2}\left(\frac{1}{2}\|f\|_2^2 - c'_m n^{-2m} - c'_m N^{-2m} - c'_m(2\lambda + c'_m n^{-2m})\right) \geq C' Ns\sigma_{N,\lambda}, \end{aligned} \tag{71}$$

where the last inequality follows by $\|f\|_2^2 \geq 4C'(\lambda + n^{-2m} + \sigma_{N,\lambda})$ for a large constant C' satisfying $2C' > 2c'_m + (c'_m)^2$. To achieve the desired power, we need to enlarge C' further. This will be described later. Combining (71) with (55) and (71) we get that

$$T_1 \gg s \ \text{ uniformly for } f \in \mathcal{B} \text{ with } \|f\|_2^2 \geq 4C' d_{N,\lambda}^2. \tag{72}$$

Terms T_2 and T_3 can be handled similarly. To handle T_2, note that $T_2 = \mathbf{f}^T\Delta\boldsymbol{\epsilon}$, where $\mathbf{f} = (\mathbf{f}_1^T, \ldots, \mathbf{f}_s^T)^T$, $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_1^T, \ldots, \boldsymbol{\epsilon}_s^T)^T$, and Δ is defined in the proof of Theorem 4.1. We need to establish $\Delta \leq sI_N$. Define an arbitrary $\mathbf{a} = (a_1^T, \ldots, a_s^T)^T \in \mathbb{R}^N$, where each a_j is an (real) n-vector. Let $\xi_j = M^* a_j$ and $\xi = (\xi_1^T, \ldots, \xi_s^T)^T$. For simplicity, put $\xi_j = (\xi_{j,0}, \ldots, \xi_{j,n-1})^T$ for $1 \leq j \leq s$. Then based on (52) and (51), we have

$$\begin{aligned}\mathbf{a}^T\Delta\mathbf{a} &= \xi^*[(\Lambda_c + \lambda I_n)^{-1}\Gamma_{j,l}(\Lambda_c + \lambda I_n)^{-1}]_{1\leq j,l\leq s}\xi \\ &= \sum_{r=0}^{n-1}\sum_{j,l=1}^{s} \overline{\xi_{j,r}}\xi_{l,r}\frac{\delta_{j,l,r}}{(\lambda+\lambda_{c,r})^2} \\ &\leq \sum_{r=1}^{n-1}\frac{s\left(\frac{1}{(2\pi(n-r))^{4m}} + \frac{1}{(2\pi r)^{4m}}\right)\sum_{j=1}^{s}|\xi_{j,r}|^2}{(\lambda+\lambda_{c,r})^2} \\ &\quad + \frac{2s\sum_{q=1}^{\infty}\frac{1}{(2\pi q n)^{4m}}\sum_{j=1}^{s}|\xi_{j,0}|^2}{(\lambda+\lambda_{c,0})^2} \\ &\leq s\sum_{r=0}^{n-1}\sum_{j=1}^{s}|\xi_{j,r}|^2 = s\xi^*\xi = s\mathbf{a}^T\mathbf{a}, \end{aligned}$$

therefore, $\Delta \leq sI_N$. This leads to that, uniformly for $f \in \mathcal{B}$ with $\|f\|_2^2 \geq 4C' d_{N,\lambda}^2$, $E_f\{T_2^2\} = \mathbf{f}^T\Delta^2\mathbf{f} \leq sT_1$. Together with (72), we get that

$$\sup_{\substack{f\in\mathcal{B}\\ \|f\|_2\geq 2\sqrt{C'}d_{N,\lambda}}} P_f\left(|T_2| \geq \varepsilon^{-1/2}T_1^{1/2}s^{1/2}\right) \leq \varepsilon. \tag{73}$$

Note that (73) also applies to T_3. By Theorem 4.1, $(T_4/(Ns) - \mu_{N,\lambda})/\sigma_{N,\lambda}$ is $O_P(1)$ uniformly for f. Therefore, we can choose $C'_\varepsilon > 0$ s.t. $P_f(|T_4/(Ns) - \mu_{N,\lambda}|/\sigma_{N,\lambda} \geq C'_\varepsilon) \leq \varepsilon$ as $N \to \infty$.

It then follows by (71), (72) and (73) that for suitable large C' (e.g., $C' \geq 2(C'_\varepsilon + z_{1-\alpha/2})$), uniformly for $f \in \mathcal{B}$ with $\|f\|_2 \geq 2\sqrt{C'}d_{N,\lambda}$,

$$P_f\left(|T_{N,\lambda} - \mu_{N,\lambda}|/\sigma_{N,\lambda} \geq z_{1-\alpha/2}\right) \leq 3\varepsilon, \ \text{as } N \to \infty.$$

Proof is completed. ■

Proof [Proof of Theorem 4.3]

Define $B_N = \lfloor N^{2/(4m+1)} \rfloor$, the integer part of $N^{2/(4m+1)}$. We prove the theorem in two cases: $\lim_N nh > 0$ and $nh = o(1)$.

Case I: $\lim_N nh > 0$**.**

In this case, it can be shown by $s \gg N^{(4m-1)/(4m+1)}$ (equivalently $n \ll B_N$, leading to $B_N h \gg nh$ hence $B_N h \to \infty$) that $n^{-6m}h^{-4m+1/2}N \ll (B_N/n)^{6m}$. Choose g to be an integer satisfying

$$n^{-6m}h^{-4m+1/2}N \ll g^{6m} \ll (B_N/n)^{6m}. \tag{74}$$

Construct an $f = \sum_{\nu=1}^{\infty} f_\nu\varphi_\nu$ with

$$f_\nu^2 = \begin{cases} \frac{C}{n-1}(2\pi(gn+r))^{-2m}, & \nu = 2(gn+r)-1,\ r = 1,2,\ldots,n-1, \\ 0, & \text{otherwise.}\end{cases} \tag{75}$$

It can be seen that

$$J(f) = \sum_{r=1}^{s-1} f_{2(gn+r)-1}^2(2\pi(gn+r))^{2m} = C, \tag{76}$$

and

$$\begin{aligned}\|f\|_2^2 &= \sum_{r=1}^{n-1} f_{2(gn+r)-1}^2 \\ &= \frac{C}{n-1}\sum_{r=1}^{n-1}(2\pi(gn+r))^{-2m} \\ &\geq C(2\pi(gn+n))^{-2m} = \beta_{N,\lambda}^2 N^{-4m/(4m+1)}, \end{aligned} \tag{77}$$

where $\beta_{N,\lambda}^2 = C[B_N/(2\pi(gn+n))]^{2m}$. Due to (74) and $n \ll B_N$, we have $gn+n \ll 2B_N$, which further implies $\beta_{N,\lambda} \to \infty$ as $N \to \infty$.

Using the trivial fact $b_{s-2-g,n} = b_{g,n}$ for $0 \leq g \leq s-2$, one can show that

$$\begin{aligned}\sum_{j,l=1}^{s} \delta_{j,l,0}\overline{d_{j,0}}d_{l,0} &= \frac{Ns}{2}\left(2\sum_{g'=0}^{s-1}|C_{g',0}|^2 b_{g',n} + \sum_{g'=0}^{s-2} C_{s-2-g',0}C_{g',0}b_{g',n} + C_{s-1,0}^2 b_{s-1,n}\right. \\ &\quad \left. + \sum_{g'=0}^{s-2} D_{s-2-g',n}D_{g',n}b_{g',n} + D_{s-1,n}^2 b_{s-1,n}\right) \\ &\leq 2Ns\sum_{g'=0}^{s-1}|C_{g',0}|^2 b_{g',n} = 0, \end{aligned} \tag{78}$$

where the last equality follows by a trivial observation $C_{g',0} = 0$. It follows by (78), (63) and (65) that

$$\begin{aligned} T_1 &= \frac{Ns}{2}\sum_{r=1}^{n-1}\frac{\sum_{g'=0}^{s-1} b_{g',r}|C_{s-1-g',r} + D_{g',r}|^2}{(\lambda+\lambda_{c,r})^2} \\ &\leq Ns\sum_{r=1}^{n-1}\frac{\sum_{g'=0}^{s-1} b_{g',r}|C_{s-1-g',r}|^2}{(\lambda+\lambda_{c,r})^2} + Ns\sum_{r=1}^{n-1}\frac{\sum_{g'=0}^{s-1} b_{g',r}|D_{g',r}|^2}{(\lambda+\lambda_{c,r})^2} \\ &= Ns\sum_{r=1}^{n-1}\frac{\sum_{g'=0}^{s-1} b_{s-1-g',n-r}|C_{g',n-r}|^2}{(\lambda+\lambda_{c,r})^2} + Ns\sum_{r=1}^{n-1}\frac{\sum_{g'=0}^{s-1} b_{g',r}|D_{g',r}|^2}{(\lambda+\lambda_{c,r})^2} \\ &= 2Ns\sum_{r=1}^{n-1}\frac{\sum_{g'=0}^{s-1} b_{g',r}|D_{g',r}|^2}{(\lambda+\lambda_{c,r})^2} = 2Ns\sum_{r=1}^{n-1}\frac{b_{g,r}f_{2(gn+r)-1}^2}{(\lambda+\lambda_{c,r})^2}, \end{aligned} \tag{79}$$

where the last equality follows from the design of f, i.e., (75). Now it follows from (79) and the fact $b_{g,r} \leq c'_m(2\pi(gn+r))^{-4m}$, for some constant c'_m depending on m only, that

$$\begin{aligned} T_1 &\leq 2Ns\sum_{r=0}^{n-1}\frac{c'_m(2\pi(gn+r))^{-4m}\frac{C}{n-1}(2\pi(gn+r))^{-2m}}{(\lambda+\lambda_{c,r})^2} \\ &= \frac{2Nsc'_mC}{n-1}\sum_{r=1}^{n-1}\frac{(2\pi(gn+r))^{-6m}}{(\lambda+\lambda_{c,r})^2} \\ &\leq 2c'_mC(2\pi)^{-6m}Ns(gn)^{-6m}h^{-4m} \ll sh^{-1/2} \asymp Ns\sigma_{N,\lambda}, \end{aligned} \tag{80}$$

where the last "$\asymp$" follows from (12).

By (73) we have that

$$|T_2+T_3| = T_1^{1/2}s^{1/2}O_{P_f}(1) = o_{P_f}(sh^{-1/4}) = o_{P_f}(Ns\sigma_{N,\lambda}).$$

Hence, by (60) and Theorem 4.1 we have

$$\begin{aligned}\frac{T_{N,\lambda}-\mu_{N,\lambda}}{\sigma_{N,\lambda}} &= \frac{T_1+T_2+T_3}{Ns\sigma_{N,\lambda}} + \frac{T_4/(Ns)-\mu_{N,\lambda}}{\sigma_{N,\lambda}} \\ &= \frac{T_4/(Ns)-\mu_{N,\lambda}}{\sigma_{N,\lambda}} + o_{P_f}(1) \xrightarrow{d} N(0,1). \end{aligned}$$

Consequently, as $N \to \infty$

$$\inf_{\substack{f^\star \in \mathcal{B} \\ \|f^\star\|_2 \geq \beta_{N,\lambda}N^{-2m/(4m+1)}}} P_{f^\star}(\phi_{N,\lambda}=1) \leq P_f(\phi_{N,\lambda}=1) \to \alpha.$$

This shows the desired result in Case I.

Case II: $nh = o(1)$**.**

The proof is similar to Case I although a bit technical difference needs to be emphasized. Since $n \ll B_N$, it can be shown that $Nn^{-2m-1/2} \ll (B_N/n)^{6m}$. Choose g to be an integer satisfying

$$Nn^{-2m-1/2} \ll g^{6m} \ll (B_N/n)^{6m}. \tag{81}$$

Let $f = \sum_{\nu=1}^{\infty} f_\nu\varphi_\nu$ with f_ν satisfying (75). Similar to (76) and (77) one can show that $J(f) = C$ and $\|f\|_2^2 \geq \beta_{N,\lambda}^2 N^{-4m/(4m+1)}$, where $\beta_{N,\lambda}^2 = C[B_N/(2\pi(gn+n))]^{2m}$. It is clear that $\beta_{N,\lambda} \to \infty$ as $N \to \infty$. Then similar to (78), (63), (65) and (80) one can show that

$$\begin{aligned} T_1 &\leq 2Ns\sum_{r=1}^{n-1}\frac{b_{g,r}f_{2(gn+r)-1}^2}{(\lambda+\lambda_{c,r})^2} \\ &\leq \frac{2Nsc'_mC}{n-1}\sum_{r=1}^{n-1}\frac{(2\pi(gn+r))^{-6m}}{(\lambda+\lambda_{c,r})^2} \\ &\leq 2c'_mC(2\pi)^{-2m}Nsg^{-6m}n^{-2m} \\ &\ll sn^{1/2} \asymp Ns\sigma_{N,\lambda}, \end{aligned}$$

where the last line follows by (81) and (12). Then the desired result follows by arguments in the rest of Case I. Proof is completed. ■

References

P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods.* Springer, New York, 1987.

T. T. Cai and M. Yuan. Optimal estimation of the mean function based on discretely sampled functional data: Phase transition. *Annals of Statistics*, 39:2330–2355, 2011.

P. P. B. Eggermont and V. N. LaRiccia. *Maximum Penalized Likelihood Estimation: Volume II.* Springer, Series in Statistics, 2009.

Yu I. Ingster. Asymptotically minimax hypothesis testing for nonparametric alternatives i–iii. *Mathematical Methods of Statistics*, 2;3;4:249–268; 85–114; 171–189, 1993.

M. Liu, Z. Shang, and G. Cheng. How many machines can we use in parallel computing? Preprint, 2017.

Z. Shang and G. Cheng. Local and global asymptotic inference in smoothing spline models. *Annals of Statistics*, 41:2608–2638, 2013.

Z. Shang and G. Cheng. A Bayesian Splitotic Theory for Nonparametric Models, 2015.

J. Shao. *Mathematical Statistics.* Springer Texts in Statistics. Springer, New York, 2nd edition, 2003.

G. Wahba. *Spline Models for Observational Data.* SIAM, Philidelphia, 1990.

G. Xu, Z. Shang, and G. Cheng. Optimal tuning for divide-and-conquer kernel ridge regression with massive data. arxiv. preprint, 2017.

Y. Zhang, J. C. Duchi, and M. J. Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16:3299–3340, 2015.

T. Zhao, G. Cheng, and H. Liu. A partially linear framework for massive heterogeneous data. *Annals of Statistics*, 44:1400–1437, 2016.

Journal of Machine Learning Research 18 (2017) 1-67 Submitted 12/15; Revised 12/16; Published 10/17

Hinge-Loss Markov Random Fields and Probabilistic Soft Logic

Stephen H. Bach BACH@CS.STANFORD.EDU
Computer Science Department
Stanford University
Stanford, CA 94305, USA

Matthias Broecheler MATTHIAS@DATASTAX.COM
DataStax

Bert Huang BHUANG@VT.EDU
Computer Science Department
Virginia Tech
Blacksburg, VA 24061, USA

Lise Getoor GETOOR@SOE.UCSC.EDU
Computer Science Department
University of California, Santa Cruz
Santa Cruz, CA 95064, USA

Editor: Luc De Raedt

Abstract

A fundamental challenge in developing high-impact machine learning technologies is balancing the need to model rich, structured domains with the ability to scale to big data. Many important problem areas are both richly structured and large scale, from social and biological networks, to knowledge graphs and the Web, to images, video, and natural language. In this paper, we introduce two new formalisms for modeling structured data, and show that they can both capture rich structure and scale to big data. The first, hinge-loss Markov random fields (HL-MRFs), is a new kind of probabilistic graphical model that generalizes different approaches to convex inference. We unite three approaches from the randomized algorithms, probabilistic graphical models, and fuzzy logic communities, showing that all three lead to the same inference objective. We then define HL-MRFs by generalizing this unified objective. The second new formalism, probabilistic soft logic (PSL), is a probabilistic programming language that makes HL-MRFs easy to define using a syntax based on first-order logic. We introduce an algorithm for inferring most-probable variable assignments (MAP inference) that is much more scalable than general-purpose convex optimization methods, because it uses message passing to take advantage of sparse dependency structures. We then show how to learn the parameters of HL-MRFs. The learned HL-MRFs are as accurate as analogous discrete models, but much more scalable. Together, these algorithms enable HL-MRFs and PSL to model rich, structured data at scales not previously possible.

Keywords: Probabilistic graphical models, statistical relational learning, structured prediction

©2017 Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/15-631.html.

1. Introduction

In many problems in machine learning, the domains are rich and structured, with many interdependent elements that are best modeled jointly. Examples include social networks, biological networks, the Web, natural language, computer vision, sensor networks, and so on. Machine learning subfields such as statistical relational learning (Getoor and Taskar, 2007), inductive logic programming (Muggleton and De Raedt, 1994), and structured prediction (Bakir et al., 2007) all seek to represent dependencies in data induced by relational structure. With the ever-increasing size of available data, there is a growing need for models that are highly scalable while still able to capture rich structure.

In this paper, we introduce *hinge-loss Markov random fields* (HL-MRFs), a new class of probabilistic graphical models designed to enable scalable modeling of rich, structured data. HL-MRFs are analogous to discrete MRFs, which are undirected probabilistic graphical models in which probability mass is log-proportional to a weighted sum of feature functions. Unlike discrete MRFs, however, HL-MRFs are defined over continuous variables in the $[0, 1]$ unit interval. To model dependencies among these continuous variables, we use linear and quadratic hinge functions, so that probability density is lost according to a weighted sum of hinge losses. As we will show, hinge-loss features capture many common modeling patterns for structured data.

When designing classes of models, there is generally a trade off between scalability and expressivity: the more complex the types and connectivity structure of the dependencies, the more computationally challenging inference and learning become. HL-MRFs address a crucial gap between the two extremes. By using hinge-loss functions to model the dependencies among the variables, which admit highly scalable inference without restrictions on their connectivity structure, HL-MRFs can capture a wide range of useful relationships. One reason they are so expressive is that hinge-loss dependencies are at the core of a number of scalable techniques for modeling both discrete and continuous structured data.

To motivate HL-MRFs, we unify three different approaches for scalable inference in structured models: (1) randomized algorithms for MAX SAT (Goemans and Williamson, 1994), (2) local consistency relaxation (Wainwright and Jordan, 2008) for discrete Markov random fields defined using Boolean logic, and (3) reasoning about continuous information with fuzzy logic. We show that all three approaches lead to the same convex programming objective. We then define HL-MRFs by generalizing this unified inference objective as a weighted sum of hinge-loss features and using them as the weighted features of graphical models. Since HL-MRFs generalize approaches that reason about relational data with weighted logical knowledge bases, they retain the same high level of expressivity. As we show in Section 6.4, they are effective for modeling both discrete and continuous data.

We also introduce *probabilistic soft logic* (PSL), a new probabilistic programming language that makes HL-MRFs easy to define and use for large, relational data sets.[1] This idea has been explored for other classes of models, such as Markov logic networks (Richardson and Domingos, 2006) for discrete MRFs, relational dependency networks (Neville and Jensen, 2007) for dependency networks, and probabilistic relational models (Getoor et al., 2002) for Bayesian networks. We build on these previous approaches, as well as the connection between hinge-loss potentials and logical clauses, to define PSL. In addition to

1. An open source implementation, tutorials, and data sets are available at http://psl.linqs.org.

probabilistic rules, PSL provides syntax that enables users to easily apply many common modeling techniques, such as domain and range constraints, blocking and canopy functions, and aggregate variables defined over other random variables.

Our next contribution is to introduce a number of inference and learning algorithms. First, we examine MAP inference, i.e., the problem of finding a most probable assignment to the unobserved random variables. MAP inference in HL-MRFs is always a convex optimization. Although any off-the-shelf optimization toolkit could be used, such methods typically do not leverage the sparse dependency structures common in graphical models. We introduce a consensus-optimization approach to MAP inference for HL-MRFs, showing how the problem can be decomposed using the alternating direction method of multipliers (ADMM) and how the resulting subproblems can be solved analytically for hinge-loss potentials. Our approach enables HL-MRFs to easily scale beyond the capabilities of off-the-shelf optimization software or sampling-based inference in discrete MRFs. We then show how to learn HL-MRFs from training data using a variety of methods: structured perceptron, maximum pseudolikelihood, and large margin estimation. Since structured perceptron and large margin estimation rely on inference as subroutines, and maximum pseudolikelihood estimation is efficient by design, all of these methods are highly scalable for HL-MRFs. We evaluate them on core relational learning and structured prediction tasks, such as collective classification and link prediction. We show that HL-MRFs offer predictive accuracy comparable to analogous discrete models while scaling much better to large data sets.

This paper brings together and expands work on scalable models for structured data that can be either discrete, continuous, or a mixture of both (Broecheler et al., 2010a; Bach et al., 2012, 2013, 2015b). The effectiveness of HL-MRFs and PSL has been demonstrated on many problems, including information extraction (Liu et al., 2016) and automatic knowledge base construction (Pujara et al., 2013), extracting and evaluating natural-language arguments on the Web (Samadi et al., 2016), high-level computer vision (London et al., 2013), drug discovery (Fakhraei et al., 2014) and predicting drug-drug interactions (Sridhar et al., 2016), natural language semantics (Beltagy et al., 2014; Sridhar et al., 2015; Deng and Wiebe, 2015; Ebrahimi et al., 2016), automobile-traffic modeling (Chen et al., 2014), recommender systems (Kouki et al., 2015), information retrieval (Alshukaili et al., 2016), and predicting attributes (Li et al., 2014) and trust (Huang et al., 2013; West et al., 2014) in social networks. The ability to easily incorporate latent variables into HL-MRFs and PSL (Bach et al., 2015a) has enabled further applications, including modeling latent topics in text (Foulds et al., 2015), and predicting student outcomes in massive open online courses (MOOCs) (Ramesh et al., 2014, 2015). Researchers have also studied how to make HL-MRFs and PSL even more scalable by developing distributed implementations (Miao et al., 2013; Magliacane et al., 2015). That they are already being widely applied indicates HL-MRFs and PSL address an open need in the machine learning community.

The paper is organized as follows. In Section 2, we first consider models for structured prediction that are defined using logical clauses. We unify three different approaches to scalable inference in such models, showing that they all optimize the same convex objective. We then generalize this objective in Section 3 to define HL-MRFs. In Section 4, we introduce PSL, specifying the language and giving many examples of common usage. Next we introduce a scalable message-passing algorithm for MAP inference in Section 5 and a

number of learning algorithms in Section 6, evaluating them on a range of tasks. Finally, in Section 7, we discuss related work.

2. Unifying Convex Inference for Logic-Based Graphical Models

In many structured domains, propositional and first-order logics are useful tools for describing the intricate dependencies that connect the unknown variables. However, these domains are usually noisy; dependencies among the variables do not always hold. To address this, logical semantics can be incorporated into probability distributions to create models that capture both the structure and the uncertainty in machine learning tasks. One common way to do this is to use logic to define feature functions in a probabilistic model. We focus on Markov random fields (MRFs), a popular class of probabilistic graphical models. Informally, an MRF is a distribution that assigns probability mass using a scoring function that is a weighted combination of feature functions called potentials. We will use logical clauses to define these potentials. We first define MRFs more formally to introduce necessary notation:

Definition 1 *Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be a vector of random variables and let $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_m)$ be a vector of potentials where each potential $\phi_j(\boldsymbol{x})$ assigns configurations of the variables a real-valued score. Also, let $\boldsymbol{w} = (w_1, \ldots, w_m)$ be a vector of real-valued weights. Then, a* **Markov random field** *is a probability distribution of the form*

$$P(\boldsymbol{x}) \propto \exp\left(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})\right) . \tag{1}$$

In an MRF, the potentials should capture how the domain behaves, assigning higher scores to more probable configurations of the variables. If a modeler does not know how the domain behaves, the potentials should capture how it might behave, so that a learning algorithm can find weights that lead to accurate predictions. Logic provides an excellent formalism for defining such potentials in structured and relational domains.

We now introduce some notation to make this logic-based approach more formal. Consider a set of logical clauses $\boldsymbol{C} = \{C_1, \ldots, C_m\}$, i.e., a knowledge base, where each clause $C_j \in \boldsymbol{C}$ is a disjunction of literals and each literal is a variable x or its negation $\neg x$ drawn from the variables $\boldsymbol{x}$ such that each variable $x_i \in \boldsymbol{x}$ appears at most once in C_j. Let I_j^+ (resp. I_j^-) $\subset \{1, \ldots, n\}$ be the set of indices of the variables that are not negated (resp. negated) in C_j. Then C_j can be written as

$$\left(\bigvee_{i \in I_j^+} x_i\right) \bigvee \left(\bigvee_{i \in I_j^-} \neg x_i\right) . \tag{2}$$

Logical clauses of this form are expressive because they can be viewed equivalently as implications from conditions to consequences:

$$\bigwedge_{i \in I_j^-} x_i \implies \bigvee_{i \in I_j^+} x_i . \tag{3}$$

This "if-then" reasoning is intuitive and can describe many dependencies in structured data.

Assuming we have a logical knowledge base $\boldsymbol{C}$ describing a structured domain, we can embed it in an MRF by defining each potential ϕ_j using a corresponding clause C_j. If an assignment to the variables $\boldsymbol{x}$ satisfies C_j, then we let $\phi_j(\boldsymbol{x})$ equal 1, and we let it equal 0 otherwise. For our subsequent analysis we assume $w_j \geq 0$ $(\forall j = 1, \ldots, m)$. The resulting MRF preserves the structured dependencies described in $\boldsymbol{C}$ but enables much more flexible modeling. Clauses no longer must always hold, and the model can express uncertainty over different possible worlds. The weights express how strongly the model expects each corresponding clause to hold; the higher the weight, the more probable that it is true according to the model.

This notion of embedding weighted, logical knowledge bases in MRFs is an appealing one. For example, Markov logic (Richardson and Domingos, 2006) is a popular formalism that induces MRFs from weighted first-order knowledge bases. Given a data set, the first-order clauses are grounded using the constants in the data to create the set of propositional clauses $\boldsymbol{C}$. Each propositional clause has the weight of the first-order clause from which it was grounded. In this way, a weighted, first-order knowledge base can compactly specify an entire family of MRFs for a structured machine-learning task.

Although we now have a method for easily defining rich, structured models for a wide range of problems, there is a new challenge: finding a most probable assignment to the variables, i.e., MAP inference, is NP-hard (Shimony, 1994; Garey et al., 1976). This means that (unless P=NP) our only hope for performing tractable inference is to perform it approximately. Observe that MAP inference for an MRF defined by $\boldsymbol{C}$ is the integer linear program

$$\begin{aligned} \operatorname*{arg\,max}_{\boldsymbol{x} \in \{0,1\}^n} P(\boldsymbol{x}) &\equiv \operatorname*{arg\,max}_{\boldsymbol{x} \in \{0,1\}^n} \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) \\ &\equiv \operatorname*{arg\,max}_{\boldsymbol{x} \in \{0,1\}^n} \sum_{C_j \in \boldsymbol{C}} w_j \min \left\{ \sum_{i \in I_j^+} x_i + \sum_{i \in I_j^-} (1 - x_i), 1 \right\} . \end{aligned} \tag{4}$$

While this program is intractable, it does admit convex programming relaxations.

In this section, we show how convex programming can be used to perform tractable inference in MRFs defined by weighted knowledge bases. We first discuss in Section 2.1 an approach developed by Goemans and Williamson (1994) that views MAP inference as an instance of the classic MAX SAT problem and relaxes it to a convex program from that perspective. This approach has the advantage of providing strong guarantees on the quality of the discrete solutions it obtains. However, it has the disadvantage that general-purpose convex programming toolkits do not scale well to relaxed MAP inference for large graphical models (Yanover et al., 2006). In Section 2.2 we then discuss a seemingly distinct approach, local consistency relaxation, with complementary advantages and disadvantages: it offers highly scalable message-passing algorithms but comes with no quality guarantees. We then unite these approaches by proving that they solve equivalent optimization problems with identical solutions. Then, in Section 2.3, we show that the unified inference objective is also equivalent to exact MAP inference if the knowledge base $\boldsymbol{C}$ is interpreted using Łukasiewicz logic, an infinite-valued logic for reasoning about naturally continuous quantities such as similarity, vague or fuzzy concepts, and real-valued data.

That these three interpretations all lead to the same inference objective—whether reasoning about discrete or continuous information—is useful. To the best of our knowledge, we are the first to show their equivalence. This equivalence indicates that the same modeling formalism, inference algorithms, and learning algorithms can be used to reason scalably and accurately about both discrete and continuous information in structured domains. We generalize the unified inference objective in Section 3.1 to define hinge-loss MRFs, and in the rest of the paper we develop a probabilistic programming language and algorithms that realize the goal of a scalable and accurate framework for structured data, both discrete and continuous.

2.1 MAX SAT Relaxation

One approach to approximating objective (4) is to use relaxation techniques developed in the randomized algorithms community for the MAX SAT problem. Formally, the MAX SAT problem is to find a Boolean assignment to a set of variables that maximizes the total weight of satisfied clauses in a knowledge base composed of disjunctive clauses annotated with nonnegative weights. In other words, objective (4) is an instance of MAX SAT. Randomized approximation algorithms can be constructed for MAX SAT by independently rounding each Boolean variable x_i to true with probability p_i. Then, the expected weighted satisfaction $\hat{w}_j$ of a clause C_j is

$$\hat{w}_j = w_j \left(1 - \prod_{i \in I_j^+} (1 - p_i) \prod_{i \in I_j^-} p_i \right) , \tag{5}$$

also known as a (weighted) noisy-or function, and the expected total score $\hat{W}$ is

$$\hat{W} = \sum_{C_j \in \boldsymbol{C}} w_j \left(1 - \prod_{i \in I_j^+} (1 - p_i) \prod_{i \in I_j^-} p_i \right) . \tag{6}$$

Optimizing $\hat{W}$ with respect to the rounding probabilities would give the exact MAX SAT solution, so this randomized approach has not made the problem any easier yet, but Goemans and Williamson (1994) showed how to bound $\hat{W}$ below with a tractable linear program.

To approximately optimize $\hat{W}$, associate with each Boolean variable x_i a corresponding continuous variable $\hat{y}_i$ with domain $[0, 1]$. Then let $\hat{\boldsymbol{y}}^\star$ be the optimum of the linear program

$$\operatorname*{arg\,max}_{\hat{\boldsymbol{y}} \in [0,1]^n} \sum_{C_j \in \boldsymbol{C}} w_j \min \left\{ \sum_{i \in I_j^+} \hat{y}_i + \sum_{i \in I_j^-} (1 - \hat{y}_i), 1 \right\} . \tag{7}$$

Observe that objectives (4) and (7) are of the same form, except that the variables are relaxed to the unit hypercube in objective (7). Goemans and Williamson (1994) proved that if p_i is set to $\hat{y}_i^\star$ for all i, then $\hat{W} \geq .632\ Z^\star$, where $Z^\star$ is the optimal total weight for the MAX SAT problem. If each p_i is set using any function in a special class, then this

lower bound improves to a .75 approximation. One simple example of such a function is

$$p_i = \frac{1}{2}\hat{y}_i^\star + \frac{1}{4} \ . \tag{8}$$

In this way, objective (7) leads to an expected .75 approximation of the MAX SAT solution.

The following method of conditional probabilities (Alon and Spencer, 2008) can find a single Boolean assignment that achieves at least the expected score from a set of rounding probabilities, and therefore at least .75 of the MAX SAT solution when objective (7) and function (8) are used to obtain them. Each variable x_i is greedily set to the value that maximizes the expected weight over the unassigned variables, conditioned on either possible value of x_i and the previously assigned variables. This greedy maximization can be applied quickly because, in many models, variables only participate in a small fraction of the clauses, making the change in expectation quick to compute for each variable. Specifically, referring to the definition of $\hat{W}$ (6), the assignment to x_i only needs to maximize over the clauses C_j in which x_i participates, i.e., $i \in I_j^+ \cup I_j^-$, which is usually a small set.

This approximation is powerful because it is a tractable linear program that comes with strong guarantees on solution quality. However, even though it is tractable, general-purpose convex optimization toolkits do not scale well to large MAP problems. In the following subsection, we unify this approximation with a complementary one developed in the probabilistic graphical models community.

2.2 Local Consistency Relaxation

Another approach to approximating objective (4) is to apply a relaxation developed for Markov random fields called local consistency relaxation (Wainwright and Jordan, 2008). This approach starts by viewing MAP inference as an equivalent optimization over marginal probabilities.[2] For each $\phi_j \in \boldsymbol{\phi}$, let $\boldsymbol{\theta}_j$ be a marginal distribution over joint assignments $\boldsymbol{x}_j$. For example, $\theta_j(\boldsymbol{x}_j)$ is the probability that the subset of variables associated with potential ϕ_j is in a particular joint state $\boldsymbol{x}_j$. Also, let $x_j(i)$ denote the setting of the variable with index i in the state $\boldsymbol{x}_j$.

With this variational formulation, inference can be relaxed to an optimization over the *first-order local polytope* $\mathbb{L}$. Let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$ be a vector of probability distributions, where $\mu_i(k)$ is the marginal probability that x_i is in state k. The first-order local polytope is

$$\mathbb{L} \triangleq \left\{ (\boldsymbol{\theta}, \boldsymbol{\mu}) \geq \mathbf{0} \;\middle|\; \begin{array}{ll} \sum_{\boldsymbol{x}_j | x_j(i)=k} \theta_j(\boldsymbol{x}_j) = \mu_i(k) & \forall i, j, k \\ \sum_{\boldsymbol{x}_j} \theta_j(\boldsymbol{x}_j) = 1 & \forall j \\ \sum_{k=0}^{K_i - 1} \mu_i(k) = 1 & \forall i \end{array} \right\}, \tag{9}$$

which constrains each marginal distribution $\boldsymbol{\theta}_j$ over joint states $\boldsymbol{x}_j$ to be consistent only with the marginal distributions $\boldsymbol{\mu}$ over individual variables that participate in the potential ϕ_j.

MAP inference can then be approximated with the *first-order local consistency relaxation*:

$$\underset{(\boldsymbol{\theta}, \boldsymbol{\mu}) \in \mathbb{L}}{\arg\max} \ \sum_{j=1}^{m} w_j \sum_{\boldsymbol{x}_j} \theta_j(\boldsymbol{x}_j) \ \phi_j(\boldsymbol{x}_j), \tag{10}$$

2. This treatment is for discrete MRFs. We have omitted a discussion of continuous MRFs for conciseness.

which is an upper bound on the true MAP objective. Much work has focused on solving the first-order local consistency relaxation for large-scale MRFs, which we discuss further in Section 7. These algorithms are appealing because they are well-suited to the sparse dependency structures common in MRFs, so they can scale to large problems. However, in general, the solutions can be fractional, and there are no guarantees on the approximation quality of a tractable discretization of these fractional solutions.

We show that for MRFs with potentials defined by $\boldsymbol{C}$ and nonnegative weights, local consistency relaxation is equivalent to MAX SAT relaxation.

Theorem 2 *For an MRF with potentials corresponding to disjunctive logical clauses and associated nonnegative weights, the first-order local consistency relaxation of MAP inference is equivalent to the MAX SAT relaxation of Goemans and Williamson (1994). Specifically, any partial optimum $\boldsymbol{\mu}^\star$ of objective (10) is an optimum $\hat{\boldsymbol{y}}^\star$ of objective (7), and vice versa.*

We prove Theorem 2 in Appendix A. Our proof analyzes the local consistency relaxation to derive an equivalent, more compact optimization over only the variable pseudomarginals $\boldsymbol{\mu}$ that is identical to the MAX SAT relaxation. Theorem 2 is significant because it shows that the rounding guarantees of MAX SAT relaxation also apply to local consistency relaxation, and the scalable message-passing algorithms developed for local consistency relaxation also apply to MAX SAT relaxation.

2.3 Łukasiewicz Logic

The previous two subsections showed that the same convex program can approximate MAP inference in discrete, logic-based models, whether viewed from the perspective of randomized algorithms or variational methods. In this subsection, we show that this convex program can also be used to reason about naturally continuous information, such as similarity, vague or fuzzy concepts, and real-valued data. Instead of interpreting the clauses $\boldsymbol{C}$ using Boolean logic, we can interpret them using Łukasiewicz logic (Klir and Yuan, 1995), which extends Boolean logic to infinite-valued logic in which the propositions $\boldsymbol{x}$ can take truth values in the continuous interval $[0, 1]$. Extending truth values to a continuous domain enables them to represent concepts that are vague, in the sense that they are often neither completely true nor completely false. For example, the propositions that a sensor value is high, two entities are similar, or a protein is highly expressed can all be captured in a more nuanced manner in Łukasiewicz logic. We can also use the now continuous valued $\boldsymbol{x}$ to represent quantities that are naturally continuous (scaled to [0,1]), such as actual sensor values, similarity scores, and protein expression levels. The ability to reason about continuous values is valuable, as many important applications are not entirely discrete.

The extension to continuous values requires a corresponding extended interpretation of the logical operators $\wedge$ (conjunction), $\vee$ (disjunction), and $\neg$ (negation). The Łukasiewicz t-norm and t-co-norm are $\wedge$ and $\vee$ operators that correspond to the Boolean logic operators for integer inputs (along with the negation operator $\neg$):

$$x_1 \wedge x_2 = \max\{x_1 + x_2 - 1, 0\} \tag{11}$$

$$x_1 \vee x_2 = \min\{x_1 + x_2, 1\} \tag{12}$$

$$\neg x = 1 - x \ . \tag{13}$$

The analogous MAX SAT problem for Łukasiewicz logic is therefore

$$\arg\max_{\boldsymbol{x}\in[0,1]^n} \sum_{C_j\in\boldsymbol{C}} w_j \min\left\{\sum_{i\in I_j^+} x_i + \sum_{i\in I_j^-}(1-x_i), 1\right\} , \tag{14}$$

which is identical in form to the relaxed MAX SAT objective (7). Therefore, if an MRF is defined over continuous variables with domain $[0,1]^n$ and the logical knowledge base $\boldsymbol{C}$ defining the potentials is interpreted using Łukasiewicz logic, then *exact* MAP inference is identical to finding the optimum using the unified, relaxed inference objective derived for Boolean logic in the previous two subsections. This result shows the equivalence of all three approaches: MAX SAT relaxation, local consistency relaxation, and MAX SAT using Łukasiewicz logic.

3. Hinge-Loss Markov Random Fields

We have shown that a specific family of convex programs can be used to reason scalably and accurately about both discrete and continuous information. In this section, we generalize this family to define *hinge-loss Markov random fields* (HL-MRFs), a new kind of probabilistic graphical model. HL-MRFs retain the convexity and expressivity of convex programs discussed in Section 2, and additionally support an even richer space of dependencies.

To begin, we define HL-MRFs as density functions over continuous variables $\boldsymbol{y} = (y_1, \ldots, y_n)$ with joint domain $[0,1]^n$. These variables have different possible interpretations depending on the application. Since we are generalizing the interpretations explored in Section 2, HL-MRF MAP states can be viewed as rounding probabilities or pseudo-marginals, or they can represent naturally continuous information. More generally, they can be viewed simply as degrees of belief, confidences, or rankings of possible states; and they can describe discrete, continuous, or mixed domains. The application domain typically determines which interpretation is most appropriate. The formalisms and algorithms described in the rest of this paper are general with respect to such interpretations.

3.1 Generalized Inference Objective

To define HL-MRFs, we will first generalize the unified inference objective of Section 2 in several ways, which we first restate in terms of the HL-MRF variables $\boldsymbol{y}$:

$$\arg\max_{\boldsymbol{y}\in[0,1]^n} \sum_{C_j\in\boldsymbol{C}} w_j \min\left\{\sum_{i\in I_j^+} y_i + \sum_{i\in I_j^-}(1-y_i), 1\right\} . \tag{15}$$

For now, we are still assuming that the objective terms are defined using a weighted knowledge base $\boldsymbol{C}$, but we will quickly drop this requirement. To do so, we examine one term in isolation. Observe that the maximum value of any unweighted term is 1, which is achieved when a linear function of the variables is at least 1. We say that the term is *satisfied* whenever this occurs. When a term is unsatisfied, we can refer to its *distance to satisfaction*, which is how far it is from achieving its maximum value. Also observe that we can rewrite

the optimization explicitly in terms of distances to satisfaction:

$$\arg\min_{\boldsymbol{y}\in[0,1]^n} \sum_{C_j\in\boldsymbol{C}} w_j \max\left\{1 - \sum_{i\in I_j^+} y_i - \sum_{i\in I_j^-}(1-y_i), 0\right\} , \tag{16}$$

so that the objective is equivalently to minimize the total weighted distance to satisfaction. Each unweighted objective term now measures how far the linear constraint

$$1 - \sum_{i\in I_j^+} y_i - \sum_{i\in I_j^-}(1-y_i) \leq 0 \tag{17}$$

is from being satisfied.

3.1.1 Relaxed Linear Constraints

With this view of each term as a relaxed linear constraint, we can easily generalize them to arbitrary linear constraints. We no longer require that the inference objective be defined using only logical clauses, and instead each term can be defined using any function $\ell_j(\boldsymbol{y})$ that is linear in $\boldsymbol{y}$. These functions can capture more general dependencies, such as beliefs about the range of values a variable can take and arithmetic relationships among variables.

The new inference objective is

$$\arg\min_{\boldsymbol{y}\in[0,1]^n} \sum_{j=1}^{m} w_j \max\left\{\ell_j(\boldsymbol{y}), 0\right\} . \tag{18}$$

In this form, each term represents the distance to satisfaction of a linear constraint $\ell_j(\boldsymbol{y}) \leq 0$. That constraint could be defined using logical clauses as discussed above, or it could be defined using other knowledge about the domain. The weight w_j indicates how important it is to satisfy a constraint relative to others by scaling the distance to satisfaction. The higher the weight, the more distance to satisfaction is penalized. Additionally, two relaxed inequality constraints, $\ell_j(\boldsymbol{y}) \leq 0$ and $-\ell_j(\boldsymbol{y}) \leq 0$, can be combined to represent a relaxed equality constraint $\ell_j(\boldsymbol{y}) = 0$.

3.1.2 Hard Linear Constraints

Now that our inference objective admits arbitrary relaxed linear constraints, it is natural to also allow hard constraints that must be satisfied at all times. Hard constraints are important modeling tools. They enable groups of variables to represent mutually exclusive possibilities, such as a multinomial or categorical variable, and functional or partial functional relationships. Hard constraints can also represent background knowledge about the domain, restricting the domain to regions that are feasible in the real world. Additionally, they can encode more complex model components such as defining a random variable as an aggregate over other unobserved variables, which we discuss further in Section 4.3.5.

We can think of including hard constraints as allowing a weight w_j to take an infinite value. Again, two inequality constraints can be combined to represent an equality constraint. However, when we introduce an inference algorithm for HL-MRFs in Section 5, it

will be useful to treat hard constraints separately from relaxed ones, and further, treat hard inequality constraints separately from hard equality constraints. Therefore, in the definition of HL-MRFs, we will define these three components separately.

3.1.3 Generalized Hinge-Loss Functions

The objective terms measuring each constraint's distance to satisfaction are hinge losses. There is a flat region, on which the distance to satisfaction is 0, and an angled region, on which the distance to satisfaction grows linearly away from the hyperplane $\ell_j(\boldsymbol{y}) = 0$. This loss function is useful—as we discuss in the previous section, it is a bound on the expected loss in the discrete setting, among other things—but it is not appropriate for all modeling situations.

A piecewise-linear loss function makes MAP inference "winner take all," in the sense that it is preferable to fully satisfy the most highly weighted objective terms completely before reducing the distance to satisfaction of terms with lower weights. For example, consider the following optimization problem:

$$\underset{y_1 \in [0,1]}{\arg\min} \; w_1 \max\{y_1, 0\} + w_2 \max\{1 - y_1, 0\} \ . \tag{19}$$

If $w_1 > w_2 \geq 0$, then the optimizer is $y_1 = 0$ because the term that prefers $y_1 = 0$ overrules the term that prefers $y_1 = 1$. The result does not indicate any ambiguity or uncertainty, but if the two objective terms are potentials in a probabilistic model, it is sometimes preferable that the result reflect the conflicting preferences. We can change the inference problem so that it smoothly trades off satisfying conflicting objective terms by squaring the hinge losses. Observe that in the modified problem

$$\underset{y_1 \in [0,1]}{\arg\min} \; w_1 \left(\max\{y_1, 0\}\right)^2 + w_2 \left(\max\{1 - y_1, 0\}\right)^2 \tag{20}$$

the optimizer is now $y_1 = \frac{w_2}{w_1 + w_2}$, reflecting the relative influence of the two loss functions.

Another advantage of squared hinge-loss functions is that they can behave more intuitively in the presence of hard constraints. Consider the problem

$$\begin{aligned} \underset{(y_1, y_2) \in [0,1]^2}{\arg\min} \quad & \max\{0.9 - y_1, 0\} + \max\{0.6 - y_2, 0\} \\ \text{such that} \quad & y_1 + y_2 \leq 1 \ . \end{aligned} \tag{21}$$

The first term prefers $y_1 \geq 0.9$, the second term prefers $y_2 \geq 0.6$, and the constraint requires that y_1 and y_2 are mutually exclusive. Such problems are very common and arise when conflicting evidence of different strengths support two mutually exclusive possibilities. The evidence values 0.9 and 0.6 could come from many sources, including base models trained to make independent predictions on individual random variables, domain-specialized similarity functions, or sensor readings. For this problem, any solution $y_1 \in [0.4, 0.9]$ and $y_2 = 1 - y_1$ is an optimizer. This solution set includes counterintuitive optimizers like $y_1 = 0.4$ and $y_2 = 0.6$, even though the evidence supporting y_1 is stronger. Again, squared hinge losses

ensure the optimizers better reflect the relative strength of evidence. For the problem

$$\begin{aligned} \underset{(y_1, y_2) \in [0,1]^2}{\arg\min} \quad & \left(\max\{0.9 - y_1, 0\}\right)^2 + \left(\max\{0.6 - y_2, 0\}\right)^2 \\ \text{such that} \quad & y_1 + y_2 \leq 1 \ , \end{aligned} \tag{22}$$

the only optimizer is $y_1 = 0.65$ and $y_2 = 0.35$, which is a more informative solution.

We therefore complete our generalized inference objective by allowing either hinge-loss or squared hinge-loss functions. Users of HL-MRFs have the choice of either one for each potential, depending on which is appropriate for their task.

3.2 Definition

We can now formally state the full definition of HL-MRFs. They are defined so that a MAP state is a solution to the generalized inference objective proposed in the previous subsection. We state the definition in a conditional form for later convenience, but this definition is fully general since the vector of conditioning variables may be empty.

Definition 3 *Let $\boldsymbol{y} = (y_1, \ldots, y_n)$ be a vector of n variables and $\boldsymbol{x} = (x_1, \ldots, x_{n'})$ a vector of n' variables with joint domain $\boldsymbol{D} = [0,1]^{n+n'}$. Let $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_m)$ be a vector of m continuous potentials of the form*

$$\phi_j(\boldsymbol{y}, \boldsymbol{x}) = \left(\max\{\ell_j(\boldsymbol{y}, \boldsymbol{x}), 0\}\right)^{p_j} \tag{23}$$

where ℓ_j is a linear function of $\boldsymbol{y}$ and $\boldsymbol{x}$ and $p_j \in \{1, 2\}$. Let $\boldsymbol{c} = (c_1, \ldots, c_r)$ be a vector of r linear constraint functions associated with index sets denoting equality constraints $\mathcal{E}$ and inequality constraints $\mathcal{I}$, which define the feasible set

$$\tilde{\boldsymbol{D}} = \left\{ (\boldsymbol{y}, \boldsymbol{x}) \in \boldsymbol{D} \;\middle|\; \begin{array}{l} c_k(\boldsymbol{y}, \boldsymbol{x}) = 0, \forall k \in \mathcal{E} \\ c_k(\boldsymbol{y}, \boldsymbol{x}) \leq 0, \forall k \in \mathcal{I} \end{array} \right\} \ . \tag{24}$$

For $(\boldsymbol{y}, \boldsymbol{x}) \in \boldsymbol{D}$, given a vector of m nonnegative free parameters, i.e., weights, $\boldsymbol{w} = (w_1, \ldots, w_m)$, a **constrained hinge-loss energy function** *$f_{\boldsymbol{w}}$ is defined as*

$$f_{\boldsymbol{w}}(\boldsymbol{y}, \boldsymbol{x}) = \sum_{j=1}^{m} w_j \phi_j(\boldsymbol{y}, \boldsymbol{x}) \ . \tag{25}$$

We now define HL-MRFs by placing a probability density over the inputs to a constrained hinge-loss energy function. Note that we negate the hinge-loss energy function so that states with lower energy are more probable, in contrast with Definition 1. This change is made for later notational convenience.

Definition 4 *A* **hinge-loss Markov random field** *P over random variables $\boldsymbol{y}$ and conditioned on random variables $\boldsymbol{x}$ is a probability density defined as follows: if $(\boldsymbol{y}, \boldsymbol{x}) \notin \tilde{\boldsymbol{D}}$, then $P(\boldsymbol{y}|\boldsymbol{x}) = 0$; if $(\boldsymbol{y}, \boldsymbol{x}) \in \tilde{\boldsymbol{D}}$, then*

$$P(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{w}, \boldsymbol{x})} \exp\left(-f_{\boldsymbol{w}}(\boldsymbol{y}, \boldsymbol{x})\right) \tag{26}$$

where

$$Z(\boldsymbol{w}, \boldsymbol{x}) = \int_{\boldsymbol{y}|(\boldsymbol{y},\boldsymbol{x})\in\tilde{\boldsymbol{D}}} \exp\left(-f_{\boldsymbol{w}}(\boldsymbol{y}, \boldsymbol{x})\right) \, d\boldsymbol{y} \ . \tag{27}$$

In the rest of this paper, we will explore how to use HL-MRFs to solve a wide range of structured machine learning problems. We first introduce a probabilistic programming language that makes HL-MRFs easy to define for large, rich domains.

4. Probabilistic Soft Logic

In this section we introduce a general-purpose probabilistic programming language, *probabilistic soft logic* (PSL). PSL allows HL-MRFs to be easily applied to a broad range of structured machine learning problems by defining *templates* for potentials and constraints. In models for structured data, there are very often repeated patterns of probabilistic dependencies. A few of the many examples include the strength of ties between similar people in social networks, the preference for triadic closure when predicting transitive relationships, and the "exactly one active" constraints on functional relationships. Often, to make graphical models both easy to define and able to generalize across different data sets, these repeated dependencies are defined using templates. Each template defines an abstract dependency, such as the form of a potential function or constraint, along with any necessary parameters, such as the weight of the potential, each of which has a single value across all dependencies defined by that template. Given input data, an undirected graphical model is constructed from a set of templates by first identifying the random variables in the data and then "grounding out" each template by introducing a potential or constraint into the graphical model for each subset of random variables to which the template applies.

A PSL program is written in a declarative, first-order syntax and defines a class of HL-MRFs that are parameterized by the input data. PSL provides a natural interface to represent hinge-loss potential templates using two types of rules: logical rules and arithmetic rules. Logical rules are based on the mapping from logical clauses to hinge-loss potentials introduced in Section 2. Arithmetic rules provide additional syntax for defining an even wider range of hinge-loss potentials and hard constraints.

4.1 Definition

In this subsection we define PSL. Our definition covers the essential functionality that should be supported by all implementations, but many extensions are possible. The PSL syntax we describe can capture a wide range of HL-MRFs, but new settings and scenarios could motivate the development of additional syntax to make the construction of different kinds of HL-MRFs more convenient.

4.1.1 Preliminaries

We begin with a high-level definition of PSL programs.

Definition 5 *A* **PSL program** *is a set of rules, each of which is a template for hinge-loss potentials or hard linear constraints. When grounded over a base of ground atoms, a PSL program induces a HL-MRF conditioned on any specified observations.*

In the PSL syntax, many components are named using *identifiers*, which are strings that begin with a letter (from the set $\{\texttt{A}, \ldots, \texttt{Z}, \texttt{a}, \ldots, \texttt{z}\}$), followed by zero or more letters, numeric digits, or underscores.

PSL programs are grounded out over data, so the universe over which to ground must be defined.

Definition 6 *A* **constant** *is a string that denotes an element in the universe over which a PSL program is grounded.*

Constants are the elements in a universe of discourse. They can be entities or attributes. For example, the constant `"person1"` can denote a person, the constant `"Adam"` can denote a person's name, and the constant `"30"` can denote a person's age. In PSL programs, constants are written as strings in double or single quotes. Constants use backslashes as escape characters, so they can be used to encode quotes within constants. It is assumed that constants are unambiguous, i.e., different constants refer to different entities and attributes.[3] Groups of constants can be represented using variables.

Definition 7 *A* **variable** *is an identifier for which constants can be substituted.*

Variables and constants are the arguments to logical predicates. Together, they are generically referred to as terms.

Definition 8 *A* **term** *is either a constant or a variable.*

Terms are connected by relationships called predicates.

Definition 9 *A* **predicate** *is a relation defined by a unique identifier and a positive integer called its arity, which denotes the number of terms it accepts as arguments. Every predicate in a PSL program must have a unique identifier as its name.*

We refer to a predicate using its identifier and arity appended with a slash. For example, the predicate `Friends/2` is a binary predicate, i.e., taking two arguments, which represents whether two constants are friends. As another example, the predicate `Name/2` can relate a person to the string that is that person's name. As a third example, the predicate `EnrolledInClass/3` can relate two entities, a student and professor, with an additional attribute, the subject of the class.

Predicates and terms are combined to create atoms.

Definition 10 *An* **atom** *is a predicate combined with a sequence of terms of length equal to the predicate's arity. This sequence is called the atom's arguments. An atom with only constants for arguments is called a ground atom.*

Ground atoms are the basic units of reasoning in PSL. Each represents an unknown or observation of interest and can take any value in $[0, 1]$. For example, the ground atom `Friends("person1", "person2")` represents whether `"person1"` and `"person2"` are friends. Atoms that are not ground are placeholders for sets of ground atoms. For example, the atom `Friends(X, Y)` stands for all ground atoms that can be obtained by substituting constants for variables `X` and `Y`.

3. Note that ambiguous references to underlying entities can be modeled by using different constants for different references and representing whether they refer to the same underlying entity as a predicate.

4.1.2 Inputs

As we have already stated, PSL defines templates for hinge-loss potentials and hard linear constraints that are grounded out over a data set to induce a HL-MRF. We now describe how that data set is represented and provided as the inputs to a PSL program. The first inputs are two sets of predicates: a set $\mathbb{C}$ of *closed* predicates, the atoms of which are completely observed, and a set $\mathbb{O}$ of *open* predicates, the atoms of which may be unobserved. The third input is the *base* $\mathcal{A}$, which is the set of all ground atoms under consideration. All atoms in $\mathcal{A}$ must have a predicate in either $\mathbb{C}$ or $\mathbb{O}$. These are the atoms that can be substituted into the rules and constraints of a PSL program, and each will later be associated with a HL-MRF random variable with domain $[0, 1]$. The final input is a function $\mathcal{O} : \mathcal{A} \to [0, 1] \cup \{\emptyset\}$ that maps the ground atoms in the base to either an observed value in $[0, 1]$ or a symbol $\emptyset$ indicating that it is unobserved. The function $\mathcal{O}$ is only valid if all atoms with a predicate in $\mathbb{C}$ are mapped to a $[0, 1]$ value. Note that this definition makes the sets $\mathbb{C}$ and $\mathbb{O}$ redundant in a sense, since they can be derived from $\mathcal{A}$ and $\mathcal{O}$, but it will be convenient later to have $\mathbb{C}$ and $\mathbb{O}$ explicitly defined.

Ultimately, the method for specifying PSL's inputs is implementation-specific, since different choices make it more or less convenient for different scenarios. In this paper, we will assume that $\mathbb{C}$, $\mathbb{O}$, $\mathcal{A}$, and $\mathcal{O}$ exist, and we remain agnostic about how they were specified. However, to make this aspect of using PSL more concrete, we will describe one possible method for defining them here.

Our example method for specifying PSL's inputs is text-based. The first section of the text input is a definition of the constants in the universe, which are grouped into types. An example universe definition follows.

```
Person = {"alexis", "bob", "claudia", "david"}
Professor = {"alexis", "bob"}
Student = {"claudia", "david"}
Subject = {"computer science", "statistics"}
```

This universe includes six constants, four with two types (`"alexis"`, `"bob"`, `"claudia"`, and `"david"`) and two with one type (`"computer science"` and `"statistics"`).

The next section of input is the definition of predicates. Each predicate includes the types of constants it takes as arguments and whether it is closed. For example, we can define predicates for an advisor-student relationship prediction task as follows:

```
Advises(Professor, Student)
Department(Person, Subject) (closed)
EnrolledInClass(Student, Subject, Professor) (closed)
```

In this case, there is one open predicate (`Advises`) and two closed predicates (`Department` and `EnrolledInClass`).

The final section of input is any associated observations. They can be specified in a list, for example:

```
Advises("alexis", "david") = 1
Department("alexis", "computer science") = 1
Department("bob", "computer science") = 1
Department("claudia", "statistics") = 1
Department("david", "statistics") = 1
```

In addition, values for atoms with the `EnrolledInClass` predicate could also be specified. If a ground atom does not have a specified value, it will have a default observed value of 0 if its predicate is closed or remain unobserved if its predicate is open.

We now describe how this text input is processed into the formal inputs $\mathbb{C}$, $\mathbb{O}$, $\mathcal{A}$, and $\mathcal{O}$. First, each predicate is added to either $\mathbb{C}$ or $\mathbb{O}$ based on whether it is annotated with the `(closed)` tag. Then, for each predicate in $\mathbb{C}$ or $\mathbb{O}$, ground atoms of that predicate are added to $\mathcal{A}$ with each sequence of constants as arguments that can be created by selecting a constant of each of the predicate's argument types. For example, assume that the input file contains a single predicate definition

```
Category(Document, Cat_Name)
```

where the universe is `Document` $= \{$`"d1"`, `"d2"`$\}$ and `Cat_Name` $= \{$`"politics"`,`"sports"`$\}$. Then,

$$\mathcal{A} = \left\{ \begin{array}{l} \texttt{Category("d1", "politics")}, \\ \texttt{Category("d1", "sports")}, \\ \texttt{Category("d2", "politics")}, \\ \texttt{Category("d2", "sports")} \end{array} \right\} . \tag{28}$$

Finally, we define the function $\mathcal{O}$. Any atom in the explicit list of observations is mapped to the given value. Then, any remaining atoms in $\mathcal{A}$ with a predicate in $\mathbb{C}$ are mapped to 0, and any with a predicate in $\mathbb{O}$ are mapped to $\emptyset$.

Before moving on, we also note that PSL implementations can support predicates and atoms that are defined functionally. Such predicates can be thought of as a type of closed predicate. Their observed values are defined as a function of their arguments. One of the most common examples is inequality, atoms of which can be represented with the shorthand infix operator `!=`. For example, the following atom has a value of 1 when two variables `A` and `B` are replaced with different constants and 0 when replaced with the same constant.

```
A != B
```

Such functionally defined predicates can be implemented without requiring their values over all arguments to be specified by the user.

4.1.3 Rules and Grounding

Before introducing the syntax and semantics of specific PSL rules, we define the grounding procedure that induces HL-MRFs in general. Given the inputs $\mathbb{C}$, $\mathbb{O}$, $\mathcal{A}$, and $\mathcal{O}$, PSL induces

a HL-MRF $P(\boldsymbol{y}|\boldsymbol{x})$ as follows. First, each ground atom $a \in \mathcal{A}$ is associated with a random variable with domain $[0, 1]$. If $\mathcal{O}(a) = \emptyset$, then the variable is included in the free variables $\boldsymbol{y}$, and otherwise it is included in the observations $\boldsymbol{x}$ with a value of $\mathcal{O}(a)$.

With the variables in the distribution defined, each rule in the PSL program is applied to the inputs and produces hinge-loss potentials or hard linear constraints, which are added to the HL-MRF. In the rest of this subsection, we describe two kinds of PSL rules: logical rules and arithmetic rules.

4.1.4 Logical Rules

The first kind of PSL rule is a logical rule, which is made up of literals.

Definition 11 *A* **literal** *is an atom or a negated atom.*

In PSL, the prefix operator `!` or `~` is used for negation. A negated atom has a value of one minus the value of the unmodified atom. For example, if `Friends("person1", "person2")` has a value of 0.7, then `!Friends("person1", "person2")` has a value of 0.3.

Definition 12 *A* **logical rule** *is a disjunctive clause of literals. Logical rules are either weighted or unweighted. If a logical rule is weighted, it is annotated with a nonnegative weight and optionally a power of two.*

Logical rules express logical dependencies in the model. As in Boolean logic, the negation, disjunction (written as `||` or `|`), and conjunction (written as `&&` or `&`) operators obey De Morgan's Laws. Also, an implication (written as `->` or `<-`) can be rewritten as the negation of the body disjuncted with the head. For example

```
   P1(A, B) && P2(A, B) -> P3(A, B) || P4(A, B)
≡  !(P1(A, B) && P2(A, B)) || P3(A, B) || P4(A, B)
≡  !P1(A, B) || !P2(A, B) || P3(A, B) || P4(A, B)
```

Therefore, any formula written as an implication with (1) a literal or conjunction of literals in the body and (2) a literal or disjunction of literals in the head is also a valid logical rule, because it is equivalent to a disjunctive clause.

There are two kinds of logical rules: weighted or unweighted. A weighted logical rule is a template for a hinge-loss potential that penalizes how far the rule is from being satisfied. A weighted logical rule begins with a nonnegative weight and optionally ends with an exponent of two (`^2`). For example, the weighted logical rule

```
1 :  Advisor(Prof, S) && Department(Prof, Sub) -> Department(S, Sub)
```

has a weight of 1 and induces potentials propagating department membership from advisors to advisees. An unweighted logical rule is a template for a hard linear constraint that requires that the rule always be satisfied. For example, the unweighted logical rule

```
Friends(X, Y) && Friends(Y, Z) -> Friends(X, Z) .
```

induces hard linear constraints enforcing the transitivity of the `Friends/2` predicate. Note the period (`.`) that is used to emphasize that this rule is always enforced and disambiguate it from weighted rules.

A logical rule is grounded out by performing all distinct substitutions from variables to constants such that the resulting ground atoms are in the base $\mathcal{A}$. This procedure produces a set of *ground rules*, which are rules containing only ground atoms. Each ground rule will then be interpreted as either a potential or hard constraint in the induced HL-MRF. For notational convenience, we assume without loss of generality that all the random variables are unobserved, i.e., $\mathcal{O}(a) = \emptyset, \forall a \in \mathcal{A}$. If the input data contain any observations, the following description still applies, except that some free variables will be replaced with observations from $\boldsymbol{x}$. The first step in interpreting a ground rule is to map its disjunctive clause to a linear constraint. This mapping is based on the unified inference objective derived in Section 2. Any ground PSL rule is a disjunction of literals, some of which are negated. Let I^+ be the set of indices of the variables that correspond to atoms that are not negated in the ground rule, when expressed as a disjunctive clause, and, likewise, let I^- be the indices of the variables corresponding to atoms that are negated. Then, the clause is mapped to the inequality

$$1 - \sum_{i \in I^+} y_i - \sum_{i \in I^-} (1 - y_i) \leq 0 \ . \tag{29}$$

If the logical rule that templated the ground rule is weighted with a weight of w and is *not* annotated with `^2`, then the potential

$$\phi(\boldsymbol{y}, \boldsymbol{x}) = \max \left\{ 1 - \sum_{i \in I^+} y_i - \sum_{i \in I^-} (1 - y_i), 0 \right\} \tag{30}$$

is added to the HL-MRF with a parameter of w. If the rule is weighted with a weight w and annotated with `^2`, then the potential

$$\phi(\boldsymbol{y}, \boldsymbol{x}) = \left(\max \left\{ 1 - \sum_{i \in I^+} y_i - \sum_{i \in I^-} (1 - y_i), 0 \right\} \right)^2 \tag{31}$$

is added to the HL-MRF with a parameter of w. If the rule is unweighted, then the function

$$c(\boldsymbol{y}, \boldsymbol{x}) = 1 - \sum_{i \in I^+} y_i - \sum_{i \in I^-} (1 - y_i) \tag{32}$$

is added to the set of constraint functions and its index is included in the set $\mathcal{I}$ to define a hard inequality constraint $c(\boldsymbol{y}, \boldsymbol{x}) \leq 0$.

As an example of the grounding process, consider the following logical rule. As part of a program for link prediction, it is often helpful to model the transitivity of a relationship.

```
3 :  Friends(A, B) && Friends(B, C) -> Friends(C, A) ^2
```

Imagine that the input data are $\mathbb{C} = \{\}$, $\mathbb{O} = \{\texttt{Friends/2}\}$,

$$\mathcal{A} = \left\{ \begin{array}{l} \texttt{Friends("p1", "p2")}, \\ \texttt{Friends("p1", "p3")}, \\ \texttt{Friends("p2", "p1")}, \\ \texttt{Friends("p2", "p3")}, \\ \texttt{Friends("p3", "p1")}, \\ \texttt{Friends("p3", "p2")} \end{array} \right\} , \tag{33}$$

and $\mathcal{O}(a) = \emptyset, \forall a \in \mathcal{A}$. Then, the rule will induce six ground rules. One such ground rule is

```
3 :  Friends("p1", "p2") && Friends("p2", "p3") -> Friends("p3", "p1") ^2
```

which is equivalent to the following.

```
3 :  !Friends("p1", "p2") || !Friends("p2", "p3") || Friends("p3", "p1") ^2
```

If the atoms `Friends("p1", "p2")`, `Friends("p2", "p3")`, and `Friends("p3", "p1")` correspond to the random variables y_1, y_2, and y_3, respectively, then this ground rule is interpreted as the weighted hinge-loss potential

$$3 \ (\max\{y_1 + y_2 - y_3 - 1, 0\})^2 \ . \tag{34}$$

Since the grounding process uses the mapping from Section 2, logical rules can be used to reason accurately and efficiently about both discrete and continuous information. They are a convenient method for constructing HL-MRFs with the unified inference objective for weighted logical knowledge bases as their MAP inference objective. They also allow the user to seamlessly incorporate some of the additional features of HL-MRFs, such as squared potentials and hard constraints. Next, we introduce an even more flexible class of PSL rules.

4.1.5 Arithmetic Rules

Arithmetic rules in PSL are more general templates for hinge-loss potentials and hard linear constraints. Like logical rules, they come in weighted and unweighted variants, but instead of using logical operators they use arithmetic operators. In general, an arithmetic rule relates two linear combinations of atoms with an inequality or an equality. A simple example enforces the mutual exclusivity of liberal and conservative ideologies.

```
Liberal(P) + Conservative(P) = 1 .
```

Just like logical rules, arithmetic rules are grounded out by performing all possible substitutions of constants for variables to make ground atoms in the base $\mathcal{A}$. In this example, each substitution for `Liberal(P)` and `Conservative(P)` is constrained to sum to 1. Since the rule is unweighted and arithmetic, it defines a hard constraint $c(\boldsymbol{y}, \boldsymbol{x})$ and its index will be included in $\mathcal{E}$ because it is an equality constraint.

To make arithmetic rules more flexible and easy to use, we define some additional syntax. The first is a generalized definition of atoms that can be substituted with sums of ground atoms, rather than just a single atom.

Definition 13 *A* **summation atom** *is an atom that takes terms and/or sum variables as arguments. A summation atom represents the summations of ground atoms that can be obtained by substituting individual constants for variables and summing over all possible constants for sum variables.*

A sum variable is represented by prepending a plus symbol (`+`) to a variable. For example, the summation atom

```
Friends(P, +F)
```


is a placeholder for the sum of all ground atoms with predicate `Friends/2` in $\mathcal{A}$ that share a first argument. Note that sum variables can be used at most once in a rule, i.e., each sum variable in a rule must have a unique identifier. Summation atoms are useful because they can describe dependencies without needing to specify the number of atoms that can participate. For example, the arithmetic rule

```
Label(X, +L) = 1 .
```

says that labels for each constant substituted for `X` should sum to one, without needing to specify how many possible labels there are.

The substitutions for sum variables can be restricted using logical clauses as filters.

Definition 14 *A* **filter clause** *is a logical clause defined for a sum variable in an arithmetic rule. The logical clause only contains atoms (1) with predicates that appear in $\mathbb{C}$ and (2) that only take as arguments (a) constants, (b) variables that appear in the arithmetic rule, and (c) the sum variable for which it is defined.*

Filter clauses restrict the substitutions for a sum variable in the corresponding arithmetic rule by only including substitutions for which the clause evaluates to true. The filters are evaluated using Boolean logic. Each ground atom a is treated as having a value of 0 if and only if $\mathcal{O}(a) = 0$. Otherwise, it is treated as having a value of 1. For example, imagine that we want to restrict the summation in the following arithmetic rule to only constants that satisfy a property `Property/1`.

```
Link(X, +Y) <= 1 .
```

Then, we can add the following filter clause.

```
{Y: Property(Y)}
```

Then, the hard linear constraints templated by the arithmetic rule will only sum over constants substituted for `Y` such that `Property(Y)` is non-zero.

In arithmetic rules, atoms can also be modified with coefficients. These coefficients can be hard-coded. As a simple example, in the rule

```
Susceptible(X) >= 0.5 Biomarker1(X) + 0.5 Biomarker2(X) .
```

the property `Susceptible/1`, which represents the degree to which a patient is susceptible to a particular disease, must be at least the average value of two biomarkers.

PSL also supports two forms of coefficient-defining syntax. The first form of coefficient syntax is a cardinality function that counts the number of terms substituted for a sum variable. Cardinality functions enable rules that depend on the number of substitutions in order to be scaled correctly, such as when averaging. Cardinality is denoted by enclosing a sum variable, without the `+`, in pipes. For example, the rule

```
1 / |Y| Friends(X, +Y) = Friendliness(X) .
```

defines the `Friendliness/1` property of a person `X` in a social network as the average strength of their outgoing friendship links. In cases in which `Friends/2` is not symmetric, we can extend this rule to sum over both outgoing and incoming links as follows.

```
1 / |Y1| |Y2| Friends(X, +Y1) + 1 / |Y1| |Y2| Friends(+Y2, X)
= Friendliness(X) .
```

The second form of coefficient syntax is built-in coefficient functions. The exact set of supported functions is implementation specific, but standard functions like maximum and minimum should be included. Coefficient functions are prepended with `@` and use square brackets instead of parentheses to distinguish them from predicates. Coefficient functions can take either scalars or cardinality functions as arguments. For example, the following rule for matching two sets of constants requires that the sum of the `Matched/2` atoms be the minimum of the sizes of the two sets.

```
Matched(+X, +Y) = @Min[|X|, |Y|] .
```

Note that PSL's coefficient syntax can also be used to define constants, as in this example.

So far we have focused on using arithmetic rules to define templates for linear constraints, but they can also be used to define hinge-loss potentials. For example, the following arithmetic rule prefers that the degree to which a person `X` is extroverted (represented with `Extroverted/1`) does not exceed the average extroversion of their friends:

```
2 :  Extroverted(X) <= 1 / |Y| Extroverted(+Y) ^2
{Y: Friends(X, Y) || Friends(Y, X)}
```

This rule is a template for weighted hinge-loss potentials of the form

$$2 \left(\max \left\{ y_{i'} - \frac{1}{|\mathcal{F}|} \sum_{i \in \mathcal{F}} y_i, 0 \right\} \right)^2 , \tag{35}$$

where $y_{i'}$ is the variable corresponding to a grounding of the atom `Extroverted(X)` and $\mathcal{F}$ is the set of the indices of the variables corresponding to `Extroverted(Y)` atoms of the friends `Y` that satisfy the rule's filter clause. Note that the weight of 2 is distinct from the coefficients in the linear constraint $\ell(\boldsymbol{y}, \boldsymbol{x}) \leq 0$ defining the hinge-loss potential. If the arithmetic rule were an equality instead of an inequality, each grounding would be two hinge-loss potentials, one using $\ell(\boldsymbol{y}, \boldsymbol{x}) \leq 0$ and one using $-\ell(\boldsymbol{y}, \boldsymbol{x}) \leq 0$. In this way, arithmetic rules can define general hinge-loss potentials.

For completeness, we state the full, formal definition of an arithmetic rule and define its grounding procedure.

Definition 15 *An* **arithmetic rule** *is an inequality or equality relating two linear combinations of summation atoms. Each sum variable in an arithmetic rule can be used once. An arithmetic rule can be annotated with filter clauses for a subset of its sum variables that restrict its groundings. Arithmetic rules are either weighted or unweighted. If an arithmetic rule is weighted, it is annotated with a nonnegative weight and optionally a power of two.*

An arithmetic rule is grounded out by performing all distinct substitutions from variables to constants such that the resulting ground atoms are in the base $\mathcal{A}$. In addition, summation atoms are replaced by the appropriate summations over ground atoms (possibly restricted by corresponding filter clauses) and the coefficient is distributed across the summands. This leads to a set of ground rules for each arithmetic rule given a set of inputs. If the arithmetic rule is an unweighted inequality, each ground rule can be algebraically manipulated to be of the form $c(\boldsymbol{y}, \boldsymbol{x}) \leq 0$. Then $c(\boldsymbol{y}, \boldsymbol{x})$ is added to the set of constraint functions and its index is added to $\mathcal{I}$. If instead the arithmetic rule is an unweighted equality, each ground rule is manipulated to $c(\boldsymbol{y}, \boldsymbol{x}) = 0$, $c(\boldsymbol{y}, \boldsymbol{x})$ is added to the set of constraint functions, and its index is added to $\mathcal{E}$. If the arithmetic rule is a weighted inequality with weight w, each ground rule is manipulated to $\ell(\boldsymbol{y}, \boldsymbol{x}) \leq 0$ and included as a potential of the form

$$\phi(\boldsymbol{y}, \boldsymbol{x}) = \max \{ \ell(\boldsymbol{y}, \boldsymbol{x}), 0 \} \tag{36}$$

with a weight of w. If the arithmetic rule is a weighted equality with weight w, each ground rule is again manipulated to $\ell(\boldsymbol{y}, \boldsymbol{x}) \leq 0$ and two potentials are included,

$$\phi_1(\boldsymbol{y}, \boldsymbol{x}) = \max \{ \ell(\boldsymbol{y}, \boldsymbol{x}), 0 \}, \quad \phi_2(\boldsymbol{y}, \boldsymbol{x}) = \max \{ -\ell(\boldsymbol{y}, \boldsymbol{x}), 0 \} , \tag{37}$$

each with a weight of w. In either case, if the weighted arithmetic rule is annotated with `^2`, then the induced potentials are squared.

4.2 Expressivity

An important question is the expressivity of PSL, which uses disjunctive clauses with positive weights for its logical rules. Other logic-based languages support different types of clauses, such as Markov logic networks (Richardson and Domingos, 2006), which support clauses with conjunctions and clauses with negative weights. As we discuss in this section, PSL's logical rules capture a general class of structural dependencies, capable of modeling arbitrary probabilistic relationships among Boolean variables, such as those defined by Markov logic networks. The advantage of PSL is that it defines HL-MRFs, which are much more scalable than discrete MRFs and often just as accurate, as we show in Section 6.4.

The expressivity of PSL is tied to the expressivity of the MAX SAT problem, since they both use the same class of weighted clauses. There are two conditions on the clauses: (1) they have nonnegative weights, and (2) they are disjunctive. We first consider the nonnegativity requirement and show that can actually be viewed as a restriction on the structure of a clause. To illustrate, consider a weighted disjunctive clause of the form

$$-w \quad : \quad \left(\bigvee_{i \in I_j^+} x_i \right) \bigvee \left(\bigvee_{i \in I_j^-} \neg x_i \right) . \tag{38}$$

If this clause were part of a generalized MAX SAT problem, in which there were no restrictions on weight sign or clause structure, but the goal were still to maximize the sum of the weights of the satisfied clauses, then this clause could be replaced with an equivalent one

without changing the optimizer:

$$w \quad : \quad \left(\bigwedge_{i \in I_j^+} \neg x_i \right) \bigwedge \left(\bigwedge_{i \in I_j^-} x_i \right) . \tag{39}$$

Note that the clause has been changed in three ways: (1) the sign of the weight has been changed, (2) the disjunctions have been replaced with conjunctions, and (3) the literals have all been negated. Due to this equivalence, the restriction on the sign of the weights is subsumed by the restriction on the structure of the clauses. In other words, any set of clauses can be converted to a set with nonnegative weights that has the same optimizer, but it might require including conjunctions in the clauses. It is also easy to verify that if Equation (38) is used to define a potential in a discrete MRF, replacing it with a potential defined by (39) leaves the distribution unchanged, due to the normalizing partition function.

We now consider the requirement that clauses be disjunctive and illustrate how conjunctive clauses can be replaced by an equivalent set of disjunctive clauses. The idea is to construct a set of disjunctive clauses such that all assignments to the variables are mapped to the same score, up to a constant. A simple example is replacing a conjunction

$$w \quad : \quad x_1 \wedge x_2 \tag{40}$$

with disjunctions

$$w \quad : \quad x_1 \vee x_2 \tag{41}$$
$$w \quad : \quad \neg x_1 \vee x_2 \tag{42}$$
$$w \quad : \quad x_1 \vee \neg x_2 \, . \tag{43}$$

Observe that the total score for all assignments to the variables remains the same, up to a constant.

This example generalizes to a procedure for encoding any Boolean MRF into a set of disjunctive clauses with nonnegative weights. Park (2002) showed that the MAP problem for any discrete Bayesian network can be represented as an instance of MAX SAT. For distributions of bounded factor size, the MAX SAT problem has size polynomial in the number of variables and factors of the distribution. We describe how any Boolean MRF can be represented with disjunctive clauses and nonnegative weights. Given a Boolean MRF with arbitrary potentials defined by mappings from joint states of subsets of the variables to scores, a new MRF is created as follows. For each potential in the original MRF, a new set of potentials defined by disjunctive clauses is created. A conjunctive clause is created corresponding to each entry in the potential's mapping with a weight equal to the score assigned by the weighted potential in the original MRF. Then, these clauses are converted to equivalent disjunctive clauses as in the example of Equations (38) and (39) by also flipping the sign of their weights and negating the literals. Once this is done for all entries of all potentials, what remains is an MRF defined by disjunctive clauses, some of which might have negative weights. We make all weights positive by adding a sufficiently large constant to all weights of all clauses, which leaves the distribution unchanged due to the normalizing partition function.

It is important to note two caveats when converting arbitrary Boolean MRFs to MRFs defined using only disjunctive clauses with nonnegative weights. First, the number of clauses required to represent a potential in the original MRF is exponential in the degree of the potential. In practice, this is rarely a significant limitation, since MRFs often contain low-degree potentials. The other important point is that the step of adding a constant to all the weights increases the total score of the MAP state. Since the bound of Goemans and Williamson (1994) is relative to this score, the bound is loosened for the original problem the larger the constant added to the weights is. This is to be expected, since even approximating MAP is NP-hard in general (Abdelbar and Hedetniemi, 1998).

We have described how general structural dependencies can be modeled with the logical rules of PSL. It is possible to represent arbitrary logical relationships with them. The process for converting general rules to PSL's logical rules can be done automatically and made transparent to the user. We have elected in this section to define PSL's logical rules without making this conversion automatic to make clear the underlying formalism.

4.3 Modeling Patterns

PSL is a flexible language, and there are some patterns of usage that come up in many applications. We illustrate some of them in this subsection with a number of examples.

4.3.1 Domain and Range Rules

In many problems, the number of relations that can be predicted among some constants is known. For binary predicates, this background knowledge can be viewed as constraints on the domain (first argument) or range (second argument) of the predicate. For example, it might be background knowledge that each entity, such as a document, has exactly one label. An arithmetic rule to express this follows.

```
Label(Document, +LabelName) = 1 .
```

The predicate `Label` is said to be *functional*.

Alternatively, sometimes it is the first argument that should be summed over. For example, imagine the task of predicting relationships among students and professors. Perhaps it is known that each student has exactly one advisor. This constraint can be written as follows.

```
Advisor(+Professor, Student) = 1 .
```

The predicate `Advisor` is said to be *inverse functional*.

Finally, imagine a scenario in which two social networks are being aligned. The goal is to predict whether each pair of people, one from each network, is the same person, which is represented with atoms of the `Same` predicate. Each person aligns with at most one person in the other network, but might not align with anyone. This can be expressed with the following two arithmetic rules.

```
Same(Person1, +Person2) <= 1 .
Same(+Person1, Person2) <= 1 .
```

The predicate `Same` is said to be both *partial functional* and *partial inverse functional*.

Many variations on these examples are possible. For example, they can be generalized to predicates with more than two arguments. Additional arguments can either be fixed or summed over in each rule. As another example, domain and range rules can incorporate multiple predicates, so that an entity can participate in a fixed number of relations counted among multiple predicates.

4.3.2 Similarity

Many problems require explicitly reasoning about similarity, rather than simply whether entities are the same or different. For example, reasoning with similarity has been explored using kernel methods, such as kFoil (Landwehr et al., 2010) that bases similarity computation on the relational structure of the data. The continuous variables of HL-MRFs make modeling similarity straightforward, and PSL's support for functionally defined predicates makes it even easier. For example, in an entity resolution task, the degree to which two entities are believed to be the same might depend on how similar their names are. A rule expressing this dependency is

```
1.0 :  Name(P1, N1) && Name(P2, N2) && Similar(N1, N2) -> Same(P1, P2)
```

This rule uses the `Similar` predicate to measure similarity. Since it is a functionally defined predicate, it can be implemented as one of many different, possibly domain specialized, string similarity functions. Any similarity function that can output values in the range $[0, 1]$ can be used.

4.3.3 Priors

If no potentials are defined over a particular atom, then it is equally probable that it has any value between zero and one. Often, however, it should be more probable that an atom has a value of zero, unless there is evidence that it has a nonzero value. Since atoms typically represent the existence of some entity, attribute, or relation, this bias promotes sparsity among the things inferred to exist. Further, if there is a potential that prefers that an atom should have a value that is at least some numeric constant, such as when reasoning with similarities as discussed in Section 4.3.2, it often should also be more probable that an atom is no higher in value than is necessary to satisfy that potential. To accomplish both these goals, simple priors can be used to state that atoms should have low values in the absence of evidence to overrules those priors. A prior in PSL can be a rule consisting of just a negative literal with a small weight. For example, in a link prediction task, imagine that this preference should apply to atoms of the `Link` predicate. A prior is then

```
0.1 :  !Link(A, B)
```

which acts as a regularizer on `Link` atoms.

4.3.4 Blocks and Canopies

In many tasks, the number of unknowns can quickly grow large, even for modest amounts of data. For example, in a link prediction task the goal is to predict relations among entities. The number of possible links grows quadratically with the number of entities (for binary

relations). If handled naively, this growth could make scaling to large data sets difficult, but this problem is often handled by constructing *blocks* (e.g., Newcombe and Kennedy, 1962) or *canopies* (McCallum et al., 2000) over the entities, so that a limited subset of all possible links are actually considered. Blocking partitions the entities so that only links among entities in the same partition element, i.e., block, are considered. Alternatively, for a finer grained pruning, a canopy is defined for each entity, which is the set of other entities to which it could possibly link. Blocks and canopies can be computed using specialized, domain-specific functions, and PSL can incorporate them by including them as atoms in the bodies of rules. Since blocks can be seen as a special case of canopies, we let the atom `InCanopy(A, B)` be 1 if `B` is in the canopy or block of `A`, and 0 if it is not. Including `InCanopy(A, B)` atoms as additional conditions in the bodies of logical rules will ensure that the dependencies only exist between the desired entities.

4.3.5 Aggregates

Another powerful feature of PSL is its ability to easily define *aggregates*, which are rules that define random variables to be deterministic functions of sets of other random variables. The advantage of aggregates is that they can be used to define dependencies that do not scale in magnitude with the number of groundings in the data. For example, consider a model for predicting interests in a social network. A fragment of a PSL program for this task follows.

```
1.0 :  Interest(P1, I) && Friends(P1, P2) -> Interest(P2, I)
1.0 :  Age(P, "20-29") && Lives(P, "California") -> Interest(P, "Surfing")
```

These two rules express the belief that interests are correlated along friendship links in the social network, and also that certain demographic information is predictive of specific interests. The question any domain expert or learning algorithm faces is how strongly each rule should be weighted relative to each other. The challenge of answering this question when using templates is that the number of groundings of the first rule varies from person to person based on the number of friends, while the groundings of the second remain constant (one per person). This inconsistent scaling of the two types of dependencies makes it difficult to find weights that accurately reflect the relative influence each type of dependency should have across people with different numbers of friends.

Using an aggregate can solve this problem of inconsistent scaling. Instead of using a separate ground rule to relate the interest of each friend, we can define a rule that is only grounded once for each person, relating an average interest across all friends to each person's own interests. A PSL fragment for this approach is

```
1.0 :  AverageFriendInterest(P, I) -> Interest(P, I)
AverageFriendInterest(P, I) = 1 / |F| Interest(+F, I) .
{F: Friends(P, F)}

/* Demographic dependencies are also included.  */
```

where the predicate `AverageFriendInterest/2` is an aggregate that is constrained to be the average amount of interest each friend of a person `P` has in an interest `I`. The weight

of the logical rule can now be scaled more appropriately relative to other types of features because there is only one grounding per person.

For a more complex example, consider the problem of determining whether two references in the data refer to the same underlying person. One useful feature to use is whether they have similar sets of friends in the social network. Again, a rule could be defined that is grounded out for each friendship pair, but this would suffer from the same scaling issues as the previous example. Instead, we can use an aggregate to directly express how similar the two references' sets of friends are. A function that measures the similarity of two sets A and B is *Jaccard similarity*:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} .$$

Jaccard similarity is a nonlinear function, meaning that it cannot be used directly without breaking the log-concavity of HL-MRFs, but we can approximate it with a linear function. We define `SameFriends/2` as an aggregate that approximates Jaccard similarity (where `SamePerson/2` is functional and inverse functional).

```
SameFriends(A, B) = 1 / @Max[|FA|, |FB|] SamePerson(+FA, +FB) .
{FA : Friends(A, FA)}
{FB : Friends(B, FB)}
SamePerson(+P1, P2) = 1 .
SamePerson(P1, +P2) = 1 .
```

The aggregate `SameFriends/2` uses the sum of the `SamePerson/2` atoms as the intersection of the two sets, and the maximum of the sizes of the two sets of friends as a lower bound on the size of their union.

5. MAP Inference

Having defined HL-MRFs and a language for creating them, PSL, we turn to algorithms for inference and learning. The first task we consider is maximum a posteriori (MAP) inference, the problem of finding a most probable assignment to the free variables $\boldsymbol{y}$ given observations $\boldsymbol{x}$. In HL-MRFs, the normalizing function $Z(\boldsymbol{w}, \boldsymbol{x})$ is constant over $\boldsymbol{y}$ and the exponential is maximized by minimizing its negated argument, so the MAP problem is

$$\begin{aligned} \arg\max_{\boldsymbol{y}} P(\boldsymbol{y}|\boldsymbol{x}) &\equiv \arg\min_{\boldsymbol{y}|\boldsymbol{y},\boldsymbol{x}\in\tilde{\boldsymbol{D}}} f_{\boldsymbol{w}}(\boldsymbol{y}, \boldsymbol{x}) \\ &\equiv \arg\min_{\boldsymbol{y}\in[0,1]^n} \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{y}, \boldsymbol{x}) \\ \text{such that} \quad & c_k(\boldsymbol{y}, \boldsymbol{x}) = 0, \quad \forall k \in \mathcal{E} \\ & c_k(\boldsymbol{y}, \boldsymbol{x}) \leq 0, \quad \forall k \in \mathcal{I} . \end{aligned} \tag{44}$$

MAP is a fundamental problem because (1) it is the method we will use to make predictions, and (2) weight learning often requires performing MAP inference many times with different weights (as we discuss in Section 6). Here, HL-MRFs have a distinct advantage over general

discrete models, since minimizing $f_{\boldsymbol{w}}$ is a convex optimization rather than a combinatorial one. There are many off-the-shelf solutions for convex optimization, the most popular of which are interior-point methods, which have worst-case polynomial time complexity in the number of variables, potentials, and constraints (Nesterov and Nemirovskii, 1994). Although in practice they perform better than their worst-case bounds (Wright, 2005), they do not scale well to large structured prediction problems (Yanover et al., 2006). We therefore introduce a new algorithm for exact MAP inference designed to scale to large HL-MRFs by leveraging the sparse connectivity structure of the potentials and hard constraints that are typical of models for real-world tasks.

5.1 Consensus Optimization Formulation

Our algorithm uses *consensus optimization*, a technique that divides an optimization problem into independent subproblems and then iterates to reach a consensus on the optimum (Boyd et al., 2011). Given a HL-MRF $P(\boldsymbol{y}|\boldsymbol{x})$, we first construct an equivalent MAP problem in which each potential and hard constraint is a function of different variables. The variables are then constrained to make the new and original MAP problems equivalent. We let $\boldsymbol{y}_{(L,j)}$ be a local copy of the variables in $\boldsymbol{y}$ that are used in the potential function ϕ_j, $j = 1, \ldots, m$ and $\boldsymbol{y}_{(L,k+m)}$ be a copy of those used in the constraint function c_k, $k = 1, \ldots, r$. We refer to the concatenation of all of these vectors as $\boldsymbol{y}_L$. We also introduce a characteristic function χ_k for each constraint function where $\chi_k \left[c_k(\boldsymbol{y}_{(L,k+m)}, \boldsymbol{x})\right] = 0$ if the constraint is satisfied and infinity if it is not. Likewise, let $\chi_{[0,1]}$ be a characteristic function that is 0 if the input is in the interval $[0, 1]$ and infinity if it is not. We drop the constraints on the domain of $\boldsymbol{y}$, letting them range in principle over $\mathbb{R}^n$ and instead use these characteristic functions to enforce the domain constraints. This formulation will make computation easier when the problem is later decomposed. Finally, let $\boldsymbol{y}_{(C,\hat{i})}$ be the variables in $\boldsymbol{y}$ that correspond to $\boldsymbol{y}_{(L,\hat{i})}$, $\hat{i} = 1, \ldots, m + r$. Operators between $\boldsymbol{y}_{(L,\hat{i})}$ and $\boldsymbol{y}_{(C,\hat{i})}$ are defined element-wise, pairing the corresponding copied variables. Consensus optimization solves the reformulated MAP problem

$$\begin{aligned} \arg\min_{(\boldsymbol{y}_L, \boldsymbol{y})} \quad & \sum_{j=1}^{m} w_j \phi_j \left(\boldsymbol{y}_{(L,j)}, \boldsymbol{x}\right) + \sum_{k=1}^{r} \chi_k \left[c_k \left(\boldsymbol{y}_{(L,k+m)}, \boldsymbol{x}\right)\right] + \sum_{i=1}^{n} \chi_{[0,1]} \left[y_i\right] \\ \text{such that} \quad & \boldsymbol{y}_{(L,\hat{i})} = \boldsymbol{y}_{(C,\hat{i})} \quad \forall \hat{i} = 1, \ldots, m + r . \end{aligned} \tag{45}$$

Inspection shows that problems (44) and (45) are equivalent.

This reformulation enables us to relax the equality constraints $\boldsymbol{y}_{(L,\hat{i})} = \boldsymbol{y}_{(C,\hat{i})}$ in order to divide problem (45) into independent subproblems that are easier to solve, using the alternating direction method of multipliers (ADMM) (Glowinski and Marrocco, 1975; Gabay and Mercier, 1976; Boyd et al., 2011). The first step is to form the *augmented Lagrangian* function for the problem. Let $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{m+r})$ be a concatenation of vectors of Lagrange

multipliers. Then the augmented Lagrangian is

$$\mathcal{L}(\boldsymbol{y}_L, \boldsymbol{\alpha}, \boldsymbol{y}) = \sum_{j=1}^{m} w_j \phi_j \left(\boldsymbol{y}_{(L,j)}, \boldsymbol{x}\right) + \sum_{k=1}^{r} \chi_k \left[c_k \left(\boldsymbol{y}_{(L,k+m)}, \boldsymbol{x}\right)\right] + \sum_{i=1}^{n} \chi_{[0,1]}\left[y_i\right] + \sum_{\hat{i}=1}^{m+r} \boldsymbol{\alpha}_{\hat{i}}^{\top} \left(\boldsymbol{y}_{(L,\hat{i})} - \boldsymbol{y}_{(C,\hat{i})}\right) + \frac{\rho}{2} \sum_{\hat{i}=1}^{m+r} \left\|\boldsymbol{y}_{(L,\hat{i})} - \boldsymbol{y}_{(C,\hat{i})}\right\|_2^2 \quad (46)$$

using a step-size parameter $\rho > 0$. ADMM finds a saddle point of $\mathcal{L}(\boldsymbol{y}_L, \boldsymbol{\alpha}, \boldsymbol{y})$ by updating the three blocks of variables at each iteration t:

$$\boldsymbol{\alpha}_{\hat{i}}^{t} \leftarrow \boldsymbol{\alpha}_{\hat{i}}^{t-1} + \rho \left(\boldsymbol{y}_{(L,\hat{i})}^{t-1} - \boldsymbol{y}_{(C,\hat{i})}^{t-1}\right) \qquad \forall \hat{i} = 1, \ldots, m+r \quad (47)$$

$$\boldsymbol{y}_L^{t} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{y}_L} \; \mathcal{L}\left(\boldsymbol{y}_L, \boldsymbol{\alpha}^t, \boldsymbol{y}^{t-1}\right) \quad (48)$$

$$\boldsymbol{y}^{t} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{y}} \; \mathcal{L}\left(\boldsymbol{y}_L^t, \boldsymbol{\alpha}^t, \boldsymbol{y}\right) \quad (49)$$

The ADMM updates ensure that $\boldsymbol{y}$ converges to the global optimum $\boldsymbol{y}^\star$, the MAP state of $P(\boldsymbol{y}|\boldsymbol{x})$, assuming that there exists a feasible assignment to $\boldsymbol{y}$. We check convergence using the criteria suggested by Boyd et al. (2011), measuring the primal and dual residuals at the end of iteration t, defined as

$$\|\bar{\boldsymbol{r}}^t\|_2 \triangleq \left(\sum_{\hat{i}=1}^{m+r} \|\boldsymbol{y}_{(L,\hat{i})}^t - \boldsymbol{y}_{(C,\hat{i})}^t\|_2^2\right)^{\frac{1}{2}} \qquad \|\bar{\boldsymbol{s}}^t\|_2 \triangleq \rho \left(\sum_{i=1}^{n} \mathcal{K}_i (y_i^t - y_i^{t-1})^2\right)^{\frac{1}{2}} \quad (50)$$

where $\mathcal{K}_i$ is the number of copies made of the variable y_i, i.e., the number of different potentials and constraints in which the variable participates. The updates are terminated when both of the following conditions are satisfied

$$\|\bar{\boldsymbol{r}}^t\|_2 \leq \epsilon^{\text{abs}} \sqrt{\sum_{i=1}^{n} \mathcal{K}_i} + \epsilon^{\text{rel}} \max\left\{\left(\sum_{\hat{i}=1}^{m+r} \|\boldsymbol{y}_{(L,\hat{i})}^t\|_2^2\right)^{\frac{1}{2}}, \left(\sum_{i=1}^{n} \mathcal{K}_i (y_i^t)^2\right)^{\frac{1}{2}}\right\} \quad (51)$$

$$\|\bar{\boldsymbol{s}}^t\|_2 \leq \epsilon^{\text{abs}} \sqrt{\sum_{i=1}^{n} \mathcal{K}_i} + \epsilon^{\text{rel}} \left(\sum_{\hat{i}=1}^{m+r} \|\boldsymbol{\alpha}_{\hat{i}}^t\|_2^2\right)^{\frac{1}{2}} \quad (52)$$

using convergence parameters ϵ^{abs} and ϵ^{rel}.

5.2 Block Updates

We now describe how to implement the ADMM block updates (47), (48), and (49). Updating the Lagrange multipliers $\boldsymbol{\alpha}$ is a simple step in the gradient direction (47). Updating the local copies $\boldsymbol{y}_L$ (48) decomposes over each potential and constraint in the HL-MRF. For the

variables $\boldsymbol{y}_{(L,j)}$ for each potential ϕ_j, this requires independently optimizing the weighted potential plus a squared norm:

$$\operatorname*{arg\,min}_{\boldsymbol{y}_{(L,j)}} \; w_j \left(\max\left\{\ell_j(\boldsymbol{y}_{(L,j)}, \boldsymbol{x}), 0\right\}\right)^{p_j} + \frac{\rho}{2} \left\|\boldsymbol{y}_{(L,j)} - \boldsymbol{y}_{(C,j)} + \frac{1}{\rho}\boldsymbol{\alpha}_j\right\|_2^2 . \quad (53)$$

Although this optimization problem is convex, the presence of the hinge function complicates it. It could be solved in principle with an iterative method, such as an interior-point method, but such methods would become very expensive over many ADMM updates. Fortunately, we can reduce the problem to checking several cases and find solutions much more quickly.

There are three cases for $\boldsymbol{y}_{(L,j)}^\star$, the optimizer of problem (53), which correspond to the three regions in which the solution could lie: (1) the region $\ell(\boldsymbol{y}_{(L,j)}, \boldsymbol{x}) < 0$, (2) the region $\ell(\boldsymbol{y}_{(L,j)}, \boldsymbol{x}) > 0$, and (3) the region $\ell(\boldsymbol{y}_{(L,j)}, \boldsymbol{x}) = 0$. We check each case by replacing the potential with its value on the corresponding region, optimizing, and checking if the optimizer is in the correct region. We check the first case by replacing the potential ϕ_j with zero. Then, the optimizer of the modified problem is $\boldsymbol{y}_{(C,j)} - \boldsymbol{\alpha}_j/\rho$. If $\ell_j(\boldsymbol{y}_{(C,j)} - \boldsymbol{\alpha}_j/\rho, \boldsymbol{x}) \leq 0$, then $\boldsymbol{y}_{(L,j)}^\star = \boldsymbol{y}_{(C,j)} - \boldsymbol{\alpha}_j/\rho$, because it optimizes both the potential and the squared norm independently. If instead $\ell_j(\boldsymbol{y}_{(C,j)} - \boldsymbol{\alpha}_j/\rho, \boldsymbol{x}) > 0$, then we can conclude that $\ell_j(\boldsymbol{y}_{(L,j)}^\star, \boldsymbol{x}) \geq 0$, leading to one of the next two cases.

In the second case, we replace the maximum term with the inner linear function. Then the optimizer of the modified problem is found by taking the gradient of the objective with respect to $\boldsymbol{y}_{(L,j)}$, setting the gradient equal to the zero vector, and solving for $\boldsymbol{y}_{(L,j)}$. In other words, the optimizer is the solution for $\boldsymbol{y}_{(L,j)}$ to the equation

$$\nabla_{\boldsymbol{y}_{(L,j)}} \left[w_j \left(\ell_j(\boldsymbol{y}_{(L,j)}, \boldsymbol{x})\right)^{p_j} + \frac{\rho}{2} \left\|\boldsymbol{y}_{(L,j)} - \boldsymbol{y}_{(C,j)} + \frac{1}{\rho}\boldsymbol{\alpha}_j\right\|_2^2\right] = \boldsymbol{0} . \quad (54)$$

This condition defines a simple system of linear equations. If $p_j = 1$, then the coefficient matrix is diagonal and trivial to solve. If $p_j = 2$, then the coefficient matrix is symmetric and positive definite, and the system can be solved via Cholesky decomposition. (Since the potentials of an HL-MRF often have shared structures, perhaps templated by a PSL program, the Cholesky decompositions can be cached and shared among potentials for improved performance.) Let $\boldsymbol{y}'_{(L,j)}$ be the optimizer of the modified problem, i.e., the solution to equation (54). If $\ell_j(\boldsymbol{y}'_{(L,j)}, \boldsymbol{x}) \geq 0$, then $\boldsymbol{y}^\star_{(L,j)} = \boldsymbol{y}'_{(L,j)}$ because we know the solution lies in the region $\ell_j(\boldsymbol{y}_{(L,j)}, \boldsymbol{x}) \geq 0$ and the objective of problem (53) and the modified objective are equal on that region. In fact, if $p_j = 2$, then $\ell_j(\boldsymbol{y}'_{(L,j)}, \boldsymbol{x}) \geq 0$ whenever $\ell_j(\boldsymbol{y}_{(C,j)} - \boldsymbol{\alpha}_j/\rho, \boldsymbol{x}) \geq 0$, because the modified term is symmetric about the line $\ell_j(\boldsymbol{y}_{(L,j)}, \boldsymbol{x}) = 0$. We therefore will only reach the following third case when $p_j = 1$. If $\ell_j(\boldsymbol{y}_{(C,j)} - \boldsymbol{\alpha}_j/\rho, \boldsymbol{x}) > 0$ and $\ell_j(\boldsymbol{y}'_{(L,j)}, \boldsymbol{x}) < 0$, then we can conclude that $\boldsymbol{y}^\star_{(L,j)}$ is the projection of $\boldsymbol{y}_{(C,j)} - \boldsymbol{\alpha}_j/\rho$ onto the hyperplane $c_k(\boldsymbol{y}_{(L,j)}, \boldsymbol{x}) = 0$. This constraint must be active because it is violated by the optimizers of both modified objectives (Martins et al., 2015, Lemma 17). Since the potential has a value of zero whenever the constraint is active, solving problem (53) reduces to the projection operation.

For the local copies $\boldsymbol{y}_{(L,k+m)}$ for each constraint c_k, the subproblem is easier:

$$\underset{\boldsymbol{y}_{(L,k+m)}}{\arg\min}\ \chi_k\left[c_k(\boldsymbol{y}_{(L,k+m)},\boldsymbol{x})\right] + \frac{\rho}{2}\left\|\boldsymbol{y}_{(L,k+m)} - \boldsymbol{y}_{(C,k+m)} + \frac{1}{\rho}\boldsymbol{\alpha}_{k+m}\right\|_2^2. \tag{55}$$

Whether c_k is an equality or inequality constraint, the solution is the projection of $\boldsymbol{y}_{(C,k+m)} - \boldsymbol{\alpha}_{k+m}/\rho$ to the feasible set defined by the constraint. If c_k is an equality constraint, i.e., $k \in \mathcal{E}$, then the optimizer $\boldsymbol{y}^\star_{(L,k+m)}$ is the projection of $\boldsymbol{y}_{(C,k+m)} - \boldsymbol{\alpha}_{k+m}/\rho$ onto $c_k(\boldsymbol{y}_{(L,k+m)},\boldsymbol{x}) = 0$. If, on the other hand, c_k is an inequality constraint, i.e., $k \in \mathcal{I}$, then there are two cases. First, if $c_k(\boldsymbol{y}_{(C,k+m)} - \boldsymbol{\alpha}_{k+m}/\rho, \boldsymbol{x}) \leq 0$, then the solution is simply $\boldsymbol{y}_{(C,k+m)} - \boldsymbol{\alpha}_{k+m}/\rho$. Otherwise, it is again the projection onto $c_k(\boldsymbol{y}_{(L,k+m)},\boldsymbol{x}) = 0$.

To update the variables $\boldsymbol{y}$ (49), we solve the optimization

$$\underset{\boldsymbol{y}}{\arg\min}\ \sum_{i=1}^{n}\chi_{[0,1]}\left[y_i\right] + \frac{\rho}{2}\sum_{\hat{i}=1}^{m+r}\left\|\boldsymbol{y}_{(L,\hat{i})} - \boldsymbol{y}_{(C,\hat{i})} + \frac{1}{\rho}\boldsymbol{\alpha}_{\hat{i}}\right\|_2^2. \tag{56}$$

The optimizer is the state in which y_i is set to the average of its corresponding local copies added with their corresponding Lagrange multipliers divided by the step size ρ, and then clipped to the $[0,1]$ interval. More formally, let $\texttt{copies}(y_i)$ be the set of local copies y_c of y_i, each with a corresponding Lagrange multiplier α_c. Then, we update each y_i using

$$y_i \leftarrow \frac{1}{|\texttt{copies}(y_i)|}\sum_{y_c \in \texttt{copies}(y_i)}\left(y_c + \frac{\alpha_c}{\rho}\right) \tag{57}$$

and clip the result to $[0,1]$. Specifically, if, after update (57), $y_i > 1$, then we set y_i to 1 and likewise set it to 0 if $y_i < 0$.

Algorithm 1 shows the complete pseudocode for MAP inference. The method starts by initializing local copies of the variables that appear in each potential and constraint, along with a corresponding Lagrange multiplier for each copy. Then, until convergence, it iteratively performs the updates (47), (48), and (49). In the pseudocode, we have interleaved updates (47) and (48), updating both the Lagrange multipliers $\boldsymbol{\alpha}_{\hat{i}}$ and the local copies $\boldsymbol{y}_{(L,\hat{i})}$ together for each subproblem, because they are local operations that do not depend on other variables once $\boldsymbol{y}$ is updated in the previous iteration. This independence reveals another advantage of our inference algorithm: it is very easy to parallelize. The updates (47) and (48) can be performed in parallel, the results gathered, update (49) performed, and the updated $\boldsymbol{y}$ broadcast back to the subproblems. Parallelization makes our MAP inference algorithm even faster and more scalable.

5.3 Lazy MAP Inference

One interesting and useful property of HL-MRFs is that it is not always necessary to completely materialize the distribution in order to find a MAP state. Consider a subset $\hat{\boldsymbol{\phi}}$ of the index set $\{1,\ldots,m\}$ of the potentials $\boldsymbol{\phi}$. Observe that if a feasible assignment to $\boldsymbol{y}$ minimizes

$$\sum_{j\in\hat{\boldsymbol{\phi}}} w_j\phi_j(\boldsymbol{y},\boldsymbol{x}) \tag{58}$$

Algorithm 1 MAP Inference for HL-MRFs

Input: HL-MRF $P(\boldsymbol{y}|\boldsymbol{x})$, $\rho > 0$

Initialize $\boldsymbol{y}_{(L,j)}$ as local copies of variables $\boldsymbol{y}_{(C,j)}$ that are in ϕ_j, $j = 1,\ldots,m$
Initialize $\boldsymbol{y}_{(L,k+m)}$ as local copies of variables $\boldsymbol{y}_{(C,k+m)}$ that are in c_k, $k = 1,\ldots,r$
Initialize Lagrange multipliers $\boldsymbol{\alpha}_{\hat{i}}$ corresponding to copies $\boldsymbol{y}_{(L,\hat{i})}$, $\hat{i} = 1,\ldots,m+r$

while not converged **do**

  for $j = 1,\ldots,m$ **do**
    $\boldsymbol{\alpha}_j \leftarrow \boldsymbol{\alpha}_j + \rho(\boldsymbol{y}_{(L,j)} - \boldsymbol{y}_{(C,j)})$
    $\boldsymbol{y}_{(L,j)} \leftarrow \boldsymbol{y}_{(C,j)} - \frac{1}{\rho}\boldsymbol{\alpha}_j$
    if $\ell_j(\boldsymbol{y}_{(L,j)},\boldsymbol{x}) > 0$ **then**
      $\boldsymbol{y}_{(L,j)} \leftarrow \arg\min_{\boldsymbol{y}_{(L,j)}} w_j\left(\ell_j(\boldsymbol{y}_{(L,j)},\boldsymbol{x})\right)^{p_j} + \frac{\rho}{2}\left\|\boldsymbol{y}_{(L,j)} - \boldsymbol{y}_{(C,j)} + \frac{1}{\rho}\boldsymbol{\alpha}_j\right\|_2^2$
      if $\ell_j(\boldsymbol{y}_{(L,j)},\boldsymbol{x}) < 0$ **then**
        $\boldsymbol{y}_{(L,j)} \leftarrow \mathrm{Proj}_{\ell_j=0}(\boldsymbol{y}_{(C,j)} - \frac{1}{\rho}\boldsymbol{\alpha}_j)$
      end if
    end if
  end for

  for $k = 1,\ldots,r$ **do**
    $\boldsymbol{\alpha}_{k+m} \leftarrow \boldsymbol{\alpha}_{k+m} + \rho(\boldsymbol{y}_{(L,k+m)} - \boldsymbol{y}_{(C,k+m)})$
    $\boldsymbol{y}_{(L,k+m)} \leftarrow \mathrm{Proj}_{c_k}(\boldsymbol{y}_{(C,k+m)} - \frac{1}{\rho}\boldsymbol{\alpha}_{k+m})$
  end for

  for $i = 1,\ldots,n$ **do**
    $y_i \leftarrow \frac{1}{|\texttt{copies}(y_i)|}\sum_{y_c\in\texttt{copies}(y_i)}\left(y_c + \frac{\alpha_c}{\rho}\right)$
    Clip y_i to [0,1]
  end for

end while

and $\phi_j(\boldsymbol{y}, \boldsymbol{x}) = 0, \forall j \notin \hat{\boldsymbol{\phi}}$, then that assignment must be a MAP state because 0 is the global minimum for any potential. Therefore, if we can identify a set of potentials that is small, such that all the other potentials are 0 in a MAP state, then we can perform MAP inference in a reduced amount of time. Of course, identifying this set is as hard as MAP inference itself, but we can iteratively grow the set by starting with an initial set, performing inference over the current set, adding any potentials that have nonzero values, and repeating.

Since the lazy inference procedure requires that the assignment be feasible, there are two ways to handle any constraints in the HL-MRF. One is to include all constraints in the inference problem from the beginning. This strategy ensures feasibility, but the idea of lazy grounding can also be extended to constraints to improve performance further. Just as we check if potentials are unsatisfied, i.e., nonzero, we can also check if constraints are unsatisfied, i.e., violated. So the algorithm now iteratively grows the set of active potentials and active constraints, adding any that are unsatisfied until the MAP state of the HL-MRF defined by the active potentials and constraints is also a feasible MAP state of the true HL-MRF.

The efficiency of lazy MAP inference can be improved heuristically by not adding all unsatisfied potentials and constraints, but instead only adding those that are unsatisfied by some threshold. This heuristic can decrease computational cost significantly, although the results are no longer guaranteed to be correct. Bounding the resulting error when possible is an important direction for future work.

5.4 Evaluation of MAP Inference

In this section we evaluate the empirical performance of our MAP inference algorithm.[4] We compare its running times against those of MOSEK,[5] a commercial convex optimization toolkit that uses interior-point methods (IPMs). We confirm the results of Yanover et al. (2006) that IPMs do not scale well to large structured-prediction problems, and we show that our MAP inference algorithm scales much better. In fact, we observe that our method scales linearly in practice with the number of potentials and constraints in the HL-MRF.

We evaluate scalability by generating social networks of varying sizes, constructing HL-MRFs over them, and measuring the running time required to find a MAP state. We compare our algorithm to MOSEK's IPM. The social networks we generate are designed to be representative of common social-network analysis tasks. We generate networks of users that are connected by different types of relationships, such as friendship and marriage, and our goal is to predict the political preferences, e.g., liberal or conservative, of each user. We also assume that we have local information about each user, representing features such as demographic information.

We generate the social networks using power-law distributions according to a procedure described by Broecheler et al. (2010b). For a target number of users N, in-degrees and out-degrees d for each edge type are sampled from the power-law distribution $D(k) \equiv \alpha k^{-\gamma}$. Incoming and outgoing edges of the same type are then matched randomly to create edges until no more matches are possible. The number of users is initially the target number

4. Code is available at `https://github.com/stephenbach/bach-jmlr17-code`.
5. http://www.mosek.com

plus the expected number of users with zero edges, and then users without any edges are removed. We use six edge types with various parameters to represent relationships in social networks with different combinations of abundance and exclusivity, choosing γ between 2 and 3, and α between 0 and 1, as suggested by Broecheler et al. We then annotate each vertex with a value in $[-1, 1]$ uniformly at random to represent local features indicating one political preference or the other.

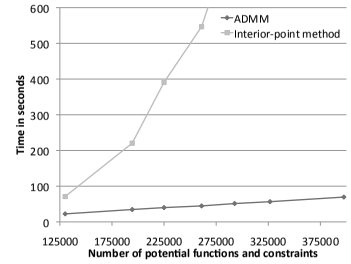
We generate social networks with between 22k and 66k vertices, which induce HL-MRFs with between 130k and 397k total potentials and constraints. In all the HL-MRFs, roughly 85% of those totals are potentials. For each social network, we create both a (log) piecewise-linear HL-MRF ($p_j = 1, \forall j = 1, \ldots, m$ in Definition 3) and a piecewise-quadratic one ($p_j = 2, \forall j = 1, \ldots, m$). We weight local features with a parameter of 0.5 and choose parameters in $[0, 1]$ for the relationship potentials representing a mix of more and less influential relationships.

We implement ADMM in Java and compare with the IPM in MOSEK (version 6) by encoding the entire MPE problem as a linear program or a second-order cone program as appropriate and passing the encoded problem via the Java native interface wrapper. All experiments are performed on a single machine with a 4-core 3.4 GHz Intel Core i7-3770 processor with 32GB of RAM. Each optimizer used a single thread, and all results are averaged over 3 runs.

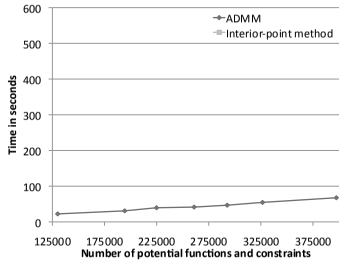
We first evaluate the scalability of ADMM when solving piecewise-linear MAP problems and compare with MOSEK's interior-point method. Figures 1a (normal scale) and 1c (log scale) show the results. The running time of the IPM quickly explodes as the problem size increases. The IPM's average running time on the largest problem is about 2,200 seconds (37 minutes). This result demonstrates the limited scalability of the interior-point method. In contrast, ADMM displays excellent scalability. The average running time on the largest problem is about 70 seconds. Further, the running time appears to grow linearly in the number of potential functions and constraints in the HL-MRF, i.e., the number of subproblems that must be solved at each iteration. The line of best fit for all runs on all sizes has a coefficient of determination $R^2 = 0.9972$. Combined with Figure 1a, this shows that ADMM scales linearly with increasing problem size in this experiment. We emphasize that the implementation of ADMM is research code written in Java and the IPM is a commercial package compiled to native machine code.

We then evaluate the scalability of ADMM when solving piecewise-quadratic MAP problem and again compare with MOSEK. Figures 1b (normal scale) and 1d (log scale) show the results. Again, the running time of the interior-point method quickly explodes. We can only test it on the three smallest problems, the largest of which took an average of about 21k seconds to solve (over 6 hours). ADMM again scales linearly to the problem ($R^2 = 0.9854$). It is just as fast for quadratic problems as linear ones, taking average of about 70 seconds on the largest problem.

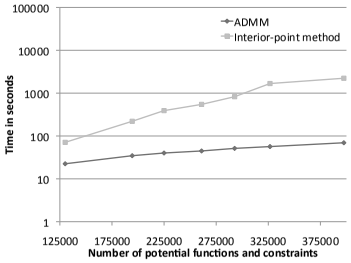
One of the advantages of IPMs is great numerical stability and accuracy. Consensus optimization, which treats both objective terms and constraints as subproblems, often returns solutions that are only optimal and feasible to moderate precision for non-trivially constrained problems (Boyd et al., 2011). Although this is often acceptable, we quantify the mix of infeasibility and suboptimality by repairing the infeasibility and measuring the resulting total suboptimality. We first project the solutions returned by consensus opti-



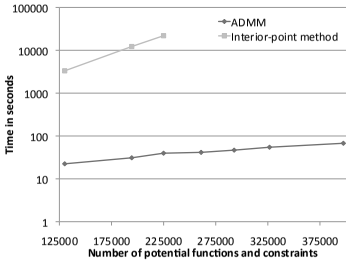
(a) Linear MAP problems



(b) Quadratic MAP problems



(c) Linear MAP problems (log scale)



(d) Quadratic MAP problems (log scale)

Figure 1: Average running times to find a MAP state for HL-MRFs.

mization onto the feasible region, which took a negligible amount of computational time. Let p_{ADMM} be the value of the objective in Problem (45) at such a point and let p_{IPM} be the value of the objective at the solution returned by the IPM. Then the relative error on that problem is $(p_{\text{ADMM}} - p_{\text{IPM}})/p_{\text{IPM}}$. The relative error was consistently small; it varied between 0.2% and 0.4%, and did not trend upward as the problem size increased. This shows that ADMM was accurate, in addition to being much more scalable.

6. Weight Learning

In this section we present three weight learning methods for HL-MRFs, each with a different objective function. The first method approximately maximizes the likelihood of the training data. The second method maximizes the pseudolikelihood. The third method finds a large-margin solution, preferring weights that discriminate the ground truth from other nearby states. Since weights are often shared among many potentials defined by a template, such as all the groundings of a PSL rule, we describe these learning algorithms in terms of templated HL-MRFs. We introduce some necessary notation for HL-MRF templates. Let $\mathcal{T} = (t_1, \ldots, t_s)$ denote a vector of templates with associated weights $\boldsymbol{W} = (W_1, \ldots, W_s)$. We partition the potentials by their associated templates and let t_q also denote the set of indices of the potentials defined by that template. So, $j \in t_q$ is a shorthand for saying that the potential $\phi_j(\boldsymbol{y}, \boldsymbol{x})$ was defined by template t_q. Then, we refer to the sum of the potentials defined by a template as

$$\Phi_q(\boldsymbol{y}, \boldsymbol{x}) = \sum_{j \in t_q} \phi_j(\boldsymbol{y}, \boldsymbol{x}) \ . \tag{59}$$

In the defined HL-MRF, the weight of the j-th hinge-loss potential is set to the weight of the template from which it was derived, i.e., $w_j = W_q$, for each $j \in t_q$. Equivalently, we can rewrite the hinge-loss energy function as

$$f_{\boldsymbol{w}}(\boldsymbol{y}, \boldsymbol{x}) = \boldsymbol{W}^\top \boldsymbol{\Phi}(\boldsymbol{y}, \boldsymbol{x}) \ , \tag{60}$$

where $\boldsymbol{\Phi}(\boldsymbol{y}, \boldsymbol{x}) = (\Phi_1(\boldsymbol{y}, \boldsymbol{x}), \ldots, \Phi_s(\boldsymbol{y}, \boldsymbol{x}))$.

6.1 Structured Perceptron and Approximate Maximum Likelihood Estimation

The canonical approach for learning parameters $\boldsymbol{W}$ is to maximize the log-likelihood of training data. The partial derivative of the log-likelihood with respect to a parameter W_q is

$$\frac{\partial \log P(\boldsymbol{y}|\boldsymbol{x})}{\partial W_q} = \mathbb{E}_{\boldsymbol{W}} \left[\Phi_q(\boldsymbol{y}, \boldsymbol{x})\right] - \Phi_q(\boldsymbol{y}, \boldsymbol{x}), \tag{61}$$

where $\mathbb{E}_{\boldsymbol{W}}$ is the expectation under the distribution defined by $\boldsymbol{W}$. For a smoother ascent, it is often helpful to divide the q-th component of the gradient by the number of groundings $|t_q|$ of the q-th template (Lowd and Domingos, 2007), which we do in our experiments. Computing the expectation is intractable, so we use a common approximation (e.g., Collins, 2002; Singla and Domingos, 2005; Poon and Domingos, 2011): the values of the potentials at the most probable setting of $\boldsymbol{y}$ with the current parameters, i.e., a MAP state. Using a MAP state makes this learning approach a structured variant of voted perceptron (Collins,

2002), and we expect it to do best when the space of explored distributions has relatively low entropy. Following voted perceptron, we take steps of fixed length in the direction of the gradient, then average the points after all steps. Any step that is outside the feasible region is projected back before continuing.

6.2 Maximum Pseudolikelihood Estimation

An alternative to structured perceptron is *maximum-pseudolikelihood estimation* (MPLE) (Besag, 1975), which maximizes the likelihood of each variable conditioned on all other variables, i.e.,

$$P^*(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^{n} P^*(y_i|\mathrm{MB}(y_i), \boldsymbol{x}) \tag{62}$$

$$= \prod_{i=1}^{n} \frac{1}{Z_i(\boldsymbol{W}, \boldsymbol{y}, \boldsymbol{x})} \exp\left[-f^i_{\boldsymbol{w}}(y_i, \boldsymbol{y}, \boldsymbol{x})\right]; \tag{63}$$

$$Z_i(\boldsymbol{W}, \boldsymbol{y}, \boldsymbol{x}) = \int_{y_i} \exp\left[-f^i_{\boldsymbol{w}}(y_i, \boldsymbol{y}, \boldsymbol{x})\right]; \tag{64}$$

$$f^i_{\boldsymbol{w}}(y_i, \boldsymbol{y}, \boldsymbol{x}) = \sum_{j: i \in \phi_j} w_j \phi_j\left(\{y_i \cup \boldsymbol{y}_{\setminus i}\}, \boldsymbol{x}\right). \tag{65}$$

Here, $i \in \phi_j$ means that y_i is involved in ϕ_j, and $\mathrm{MB}(y_i)$ denotes the *Markov blanket* of y_i—that is, the set of variables that co-occur with y_i in any potential function. The partial derivative of the log-pseudolikelihood with respect to W_q is

$$\frac{\partial \log P^*(\boldsymbol{y}|\boldsymbol{x})}{\partial W_q} = \sum_{i=1}^{n} \mathbb{E}_{y_i|\mathrm{MB}}\left[\sum_{j \in t_q : i \in \phi_j} \phi_j(\boldsymbol{y}, \boldsymbol{x})\right] - \Phi_q(\boldsymbol{y}, \boldsymbol{x}) \ . \tag{66}$$

Computing the pseudolikelihood gradient does not require joint inference and takes time linear in the size of $\boldsymbol{y}$. However, the integral in the above expectation does not readily admit a closed-form antiderivative, so we approximate the expectation. When a variable is unconstrained, the domain of integration is a one-dimensional interval on the real number line, so Monte Carlo integration quickly converges to an accurate estimate of the expectation.

We can also apply MPLE when the constraints are not too interdependent. For example, for linear equality constraints over disjoint groups of variables (e.g., variable sets that must sum to 1.0), we can block-sample the constrained variables by sampling uniformly from a simplex. These types of constraints are often used to represent categorical labels. We can compute accurate estimates quickly because these blocks are typically low-dimensional.

6.3 Large-Margin Estimation

A different approach to learning drops the probabilistic interpretation of the model and views HL-MRF inference as a prediction function. Large-margin estimation (LME) shifts the goal of learning from producing accurate probabilistic models to instead producing accurate MAP predictions. The learning task is then to find weights $\boldsymbol{W}$ that separate the ground truth from other nearby states by a large margin. We describe in this section

a large-margin method based on the cutting-plane approach for structural support vector machines (Joachims et al., 2009).

The intuition behind large-margin structured prediction is that the ground-truth state should have energy lower than any alternate state by a large margin. In our setting, the output space is continuous, so we parameterize this margin criterion with a continuous loss function. For any valid output state $\tilde{\boldsymbol{y}}$, a large-margin solution should satisfy

$$f_{\boldsymbol{w}}(\boldsymbol{y}, \boldsymbol{x}) \leq f_{\boldsymbol{w}}(\tilde{\boldsymbol{y}}, \boldsymbol{x}) - L(\boldsymbol{y}, \tilde{\boldsymbol{y}}), \quad \forall \tilde{\boldsymbol{y}}, \tag{67}$$

where the loss function $L(\boldsymbol{y}, \tilde{\boldsymbol{y}})$ measures the disagreement between a state $\tilde{\boldsymbol{y}}$ and the training label state $\boldsymbol{y}$. A common assumption is that the loss function decomposes over the prediction components, i.e., $L(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \sum_i L(y_i, \tilde{y}_i)$. In this work, we use the ℓ_1 distance as the loss function, so $L(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \sum_i \|y_i - \tilde{y}_i\|_1$. Since we do not expect all problems to be perfectly separable, we relax the large-margin constraint with a penalized slack ξ. We obtain a convex learning objective for a large-margin solution

$$\begin{aligned} \min_{\boldsymbol{W} \geq 0} \quad & \frac{1}{2}||\boldsymbol{W}||^2 + C\xi \\ \text{s.t.} \quad & \boldsymbol{W}^\top(\Phi(\boldsymbol{y}, \boldsymbol{x}) - \Phi(\tilde{\boldsymbol{y}}, \boldsymbol{x})) \leq -L(\boldsymbol{y}, \tilde{\boldsymbol{y}}) + \xi, \quad \forall \tilde{\boldsymbol{y}}, \end{aligned} \tag{68}$$

where $\Phi(\boldsymbol{y}, \boldsymbol{x}) = (\Phi_1(\boldsymbol{y}, \boldsymbol{x}), \ldots, \Phi_s(\boldsymbol{y}, \boldsymbol{x}))$ and $C > 0$ is a user-specified parameter. This formulation is analogous to the margin-rescaling approach by Joachims et al. (2009). Though such a structured objective is natural and intuitive, its number of constraints is the cardinality of the output space, which here is infinite. Following their approach, we optimize subject to the infinite constraint set using a *cutting-plane algorithm*: we greedily grow a set K of constraints by iteratively adding the worst-violated constraint given by a *separation oracle*, then updating $\boldsymbol{W}$ subject to the current constraints. The goal of the cutting-plane approach is to efficiently find the set of active constraints at the solution for the full objective, without having to enumerate the infinite inactive constraints. The worst-violated constraint is

$$\arg\min_{\tilde{\boldsymbol{y}}} \boldsymbol{W}^\top \Phi(\tilde{\boldsymbol{y}}, \boldsymbol{x}) - L(\boldsymbol{y}, \tilde{\boldsymbol{y}}). \tag{69}$$

The separation oracle performs loss-augmented inference by adding additional potentials to the HL-MRF. For ground truth in $\{0, 1\}$, these loss-augmenting potentials are also examples of hinge-losses, and thus adding them simply creates an augmented HL-MRF. The worst-violated constraint is then computed as standard inference on the loss-augmented HL-MRF. However, ground truth values in the interior $(0, 1)$ cause any distance-based loss to be concave, which require the separation oracle to solve a non-convex objective. In this case, we use the *difference of convex functions algorithm* (An and Tao, 2005) to find a local optimum. Since the concave portion of the loss-augmented inference objective pivots around the ground truth value, the subgradients are 1 or -1, depending on whether the current value is greater than the ground truth. We simply choose an initial direction for interior labels by rounding, and flip the direction of the subgradients for variables whose solution states are not in the interval corresponding to the subgradient direction until convergence.

Given a set K of constraints, we solve the SVM objective as in the primal form

$$\begin{aligned} \min_{\boldsymbol{W} \geq 0} \quad & \frac{1}{2}||\boldsymbol{W}||^2 + C\xi \\ \text{s.t.} \quad & K. \end{aligned} \tag{70}$$

We then iteratively invoke the separation oracle to find the worst-violated constraint. If this new constraint is not violated, or its violation is within numerical tolerance, we have found the max-margin solution. Otherwise, we add the new constraint to K, and repeat.

One fact of note is that the large-margin criterion always requires some slack for HL-MRFs with squared potentials. Since the squared hinge potential is quadratic and the loss is linear, there always exists a small enough distance from the ground truth such that an absolute (i.e., linear) distance is greater than the squared distance. In these cases, the slack parameter trades off between the peakedness of the learned quadratic energy function and the margin criterion.

6.4 Evaluation of Learning

To demonstrate the flexibility and effectiveness of learning with HL-MRFs, we test them on four diverse tasks: node labeling, link labeling, link prediction, and image completion.[6] Each of these experiments represents a problem domain that is best solved with structured-prediction approaches because their dependencies are highly structural. The experiments show that HL-MRFs perform as well as or better than canonical approaches.

For these diverse tasks, we compare against a number of competing methods. For node and link labeling, we compare HL-MRFs to discrete Markov random fields (MRFs). We construct them with Markov logic networks (MLNs) (Richardson and Domingos, 2006), which template discrete MRFs using logical rules similarly to PSL. We perform inference in discrete MRFs using Gibbs sampling, and we find approximate MAP states during learning using the search algorithm MaxWalkSat (Richardson and Domingos, 2006). For link prediction for preference prediction, a task that is inherently continuous and nontrivial to encode in discrete logic, we compare against Bayesian probabilistic matrix factorization (BPMF) (Salakhutdinov and Mnih, 2008). Finally, for image completion, we run the same experimental setup as Poon and Domingos (2011) and compare against the results they report, which include tests using sum product networks, deep belief networks (Hinton and Salakhutdinov, 2006), and deep Boltzmann machines (Salakhutdinov and Hinton, 2009).

We train HL-MRFs and discrete MRFs with all three learning methods: structured perceptron (SP), maximum pseudolikelihood estimation(MPLE), and large-margin estimation (LME). When appropriate, we evaluate statistical significance using a paired t-test with rejection threshold 0.01. We describe the HL-MRFs used for our experiments using the PSL rules that define them. To investigate the differences between linear and squared potentials we use both in our experiments. HL-MRF-L refers to a model with all linear potentials and HL-MRF-Q to one with all squared potentials. When training with SP and MPLE, we use 100 gradient steps and a step size of 1.0 (unless otherwise noted), and we average the iterates as in voted perceptron. For LME, we set $C = 0.1$. We experimented with various settings, but the scores of HL-MRFs and discrete MRFs were not sensitive to changes.

6. Code is available at `https://github.com/stephenbach/bach-jmlr17-code`.

6.4.1 Node Labeling

When classifying documents, links between those documents—such as hyperlinks, citations, or shared authorship—provide extra signal beyond the local features of individual documents. Collectively predicting document classes with these links tends to improve accuracy (Sen et al., 2008). We classify documents in citation networks using data from the Cora and Citeseer scientific paper repositories. The Cora data set contains 2,708 papers in seven categories, and 5,429 directed citation links. The Citeseer data set contains 3,312 papers in six categories, and 4,591 directed citation links. Let the predicate `Category/2` represent the category of each document and `Cites/2` represent a citation from one document to another.

The prediction task is, given a set of seed documents whose labels are observed, to infer the remaining document classes by propagating the seed information through the network. For each of 20 runs, we split the data sets 50/50 into training and testing partitions, and seed half of each set. To predict discrete categories with HL-MRFs we predict the category with the highest predicted value.

We compare HL-MRFs to discrete MRFs on this task. For prediction, we performed 2500 rounds of Gibbs sampling, 500 of which were discarded as burn-in. We construct both using the same logical rules, which simply encode the tendency for a class to propagate across citations. For each category `"C_i"`, we have the following two rules, one for each direction of citation.

```
Category(A, "C_i") && Cites(A, B) -> Category(B, "C_i")
Category(A, "C_i") && Cites(B, A) -> Category(B, "C_i")
```

We also constrain the atoms of the `Category/2` predicate to sum to 1.0 for a given document as follows.

```
Category(D, +C) = 1.0 .
```

Table 1 lists the results of this experiment. HL-MRFs are the most accurate predictors on both data sets. Both variants of HL-MRFs are also much faster than discrete MRFs. See Table 3 for average inference times over five folds.

6.4.2 Link Labeling

An emerging problem in the analysis of online social networks is the task of inferring the level of trust between individuals. Predicting the strength of trust relationships can provide useful information for viral marketing, recommendation engines, and internet security. HL-MRFs with linear potentials have been applied by Huang et al. (2013) to this task, showing superior results with models based on sociological theory. We reproduce their experimental setup using their sample of the signed Epinions trust network, orginally collected by Richardson et al. (2003), in which users indicate whether they trust or distrust other users. We perform eight-fold cross-validation. In each fold, the prediction algorithm observes the entire unsigned social network and all but 1/8 of the trust ratings. We measure prediction accuracy on the held-out 1/8. The sampled network contains 2,000 users, with 8,675 signed links. Of these links, 7,974 are positive and only 701 are negative, making it a sparse prediction task.

Table 1: Average accuracy of classification by HL-MRFs and discrete MRFs. Scores statistically equivalent to the best scoring method are typed in bold.

	Citeseer	Cora
HL-MRF-Q (SP)	**0.729**	**0.816**
HL-MRF-Q (MPLE)	**0.729**	**0.818**
HL-MRF-Q (LME)	0.683	0.789
HL-MRF-L (SP)	**0.724**	0.802
HL-MRF-L (MPLE)	**0.729**	**0.808**
HL-MRF-L (LME)	0.695	0.789
MRF (SP)	0.686	0.756
MRF (MPLE)	0.715	0.797
MRF (LME)	0.687	0.783

Table 2: Average area under ROC and precision-recall curves of social-trust prediction by HL-MRFs and discrete MRFs. Scores statistically equivalent to the best scoring method by metric are typed in bold.

	ROC	P-R (+)	P-R (-)
HL-MRF-Q (SP)	**0.822**	**0.978**	**0.452**
HL-MRF-Q (MPLE)	**0.832**	**0.979**	**0.482**
HL-MRF-Q (LME)	**0.814**	**0.976**	**0.462**
HL-MRF-L (SP)	0.765	0.965	0.357
HL-MRF-L (MPLE)	0.757	0.963	0.333
HL-MRF-L (LME)	0.783	0.967	**0.453**
MRF (SP)	0.655	0.942	0.270
MRF (MPLE)	0.725	0.963	0.298
MRF (LME)	0.795	**0.973**	**0.441**

Table 3: Average inference times (reported in seconds) of single-threaded HL-MRFs and discrete MRFs.

	Citeseer	Cora	Epinions
HL-MRF-Q	0.42	0.70	0.32
HL-MRF-L	0.46	0.50	0.28
MRF	110.96	184.32	212.36

We use a model based on the social theory of *structural balance*, which suggests that social structures are governed by a system that prefers triangles that are considered balanced. Balanced triangles have an odd number of positive trust relationships; thus, considering all possible directions of links that form a triad of users, there are sixteen logical implications of the following form.

```
Trusts(A,B) && Trusts(B,C) -> Trusts(A,C)
```

Huang et al. (2013) list all sixteen of these rules, a reciprocity rule, and a prior in their *Balance-Recip* model, which we omit to save space.

Since we expect these structural implications to vary in accuracy, learning weights for these rules provides better models. Again, we use these rules to define HL-MRFs and discrete MRFs, and we train them using various learning algorithms. For inference with discrete MRFs, we perform 5000 rounds of Gibbs sampling, of which the first 500 are burn-in. We compute three metrics: the area under the receiver operating characteristic (ROC) curve, and the areas under the precision-recall curves for positive trust and negative trust. On all three metrics, HL-MRFs with squared potentials score significantly higher. The differences among the learning methods for squared HL-MRFs are insignificant, but the differences among the models is statistically significant for the ROC metric. For area under the precision-recall curve for positive trust, discrete MRFs trained with LME are statistically tied with the best score, and both HL-MRF-L and discrete MRFs trained with LME are statistically tied with the best area under the precision-recall curve for negative trust. The results are listed in Table 2.

Though the random fold splits are not the same, using the same experimental setup, Huang et al. (2013) also scored the precision-recall area for negative trust of standard trust prediction algorithms EigenTrust (Kamvar et al., 2003) and TidalTrust (Golbeck, 2005), which scored 0.131 and 0.130, respectively. The logical models based on structural balance that we run here are significantly more accurate, and HL-MRFs more than discrete MRFs.

In addition to comparing favorably with regard to predictive accuracy, inference in HL-MRFs is also much faster than in discrete MRFs. Table 3 lists average inference times on five folds of three prediction tasks: Cora, Citeseer, and Epinions. This illustrates an important difference between performing structured prediction via convex inference versus sampling in a discrete prediction space: convex inference can be much faster.

6.4.3 Link Prediction

Preference prediction is the task of inferring user attitudes (often quantified by ratings) toward a set of items. This problem is naturally structured, since a user's preferences are often interdependent, as are an item's ratings. *Collaborative filtering* is the task of predicting unknown ratings using only a subset of observed ratings. Methods for this task range from simple nearest-neighbor classifiers to complex latent factor models. More generally, this problem is an instance of link prediction, since the goal is to predict links indicating preference between users and content. Since preferences are ordered rather than Boolean, it is natural to represent them with the continuous variables of HL-MRFs, with higher values indicating greater preference. To illustrate the versatility of HL-MRFs, we design a simple, interpretable collaborative filtering model for predicting humor preferences. We test this model on the Jester dataset, a repository of ratings from 24,983 users on a set of 100 jokes (Goldberg et al., 2001). Each joke is rated on a scale of $[-10, +10]$, which we normalize to $[0, 1]$. We sample a random 2,000 users from the set of those who rated all 100 jokes, which we then split into 1,000 train and 1,000 test users. From each train and test matrix, we sample a random 50% to use as the observed features $\boldsymbol{x}$; the remaining ratings are treated as the variables $\boldsymbol{y}$.

Our HL-MRF model uses an item-item similarity rule:

```
SimRating(J1, J2) && Likes(U, J1) -> Likes(U, J2)
```

where `J1` and `J2` are jokes and `U` is a user; the predicate `Likes/2` indicates the degree of preference (i.e., rating value); and `SimRating/2` is a closed predicate that measures the mean-adjusted cosine similarity between the observed ratings of two jokes. We also include the following rules to enforce that `Likes(U,J)` concentrates around the observed average rating of user `U` (represented with the predicate `AvgUserRating/1`) and item `J` (represented with the predicate `AvgJokeRating/1`), and the global average (represented with the predicate `AvgRating/1`).

```
AvgUserRating(U) -> Likes(U, J)
Likes(U, J) -> AvgUserRating(U)
AvgJokeRating(J) -> Likes(U, J)
Likes(U, J) -> AvgJokeRating(J)
AvgRating("constant") -> Likes(U, J)
Likes(U, J) -> AvgRating("constant")
```

The atom `AvgRating("constant")` takes a placeholder constant as an argument, since there is only one grounding of it for the entire HL-MRF. Again, all three of these predicates are closed and computed using averages of observed ratings. In all cases, the observed ratings are taken only from the training data for learning (to avoid leaking information about the test data) and only from the test data during testing.

We compare our HL-MRF model to a canonical latent factor model, *Bayesian probabilistic matrix factorization* (BPMF) (Salakhutdinov and Mnih, 2008). BPMF is a fully Bayesian treatment and is therefore considered "parameter-free;" the only parameter that must be specified is the rank of the decomposition. Based on settings used by Xiong et al.

Table 4: Normalized mean squared/absolute errors (NMSE/NMAE) for preference prediction using the Jester dataset. The lowest errors are typed in bold.

	NMSE	NMAE
HL-MRF-Q (SP)	0.0554	0.1974
HL-MRF-Q (MPLE)	0.0549	0.1953
HL-MRF-Q (LME)	0.0738	0.2297
HL-MRF-L (SP)	0.0578	0.2021
HL-MRF-L (MPLE)	0.0535	0.1885
HL-MRF-L (LME)	0.0544	**0.1875**
BPMF	**0.0501**	**0.1832**

Table 5: Mean squared errors per pixel for image completion. HL-MRFs produce the most accurate completions on the Caltech101 and the left-half Olivetti faces, and only sum-product networks produce better completions on Olivetti bottom-half faces. Scores for other methods are reported in Poon and Domingos (2011).

	HL-MRF-Q (SP)	SPN	DBM	DBN	PCA	NN
Caltech-Left	1741	1815	2998	4960	2851	2327
Caltech-Bottom	1910	1924	2656	3447	1944	2575
Olivetti-Left	927	942	1866	2386	1076	1527
Olivetti-Bottom	1226	918	2401	1931	1265	1793

(2010), we set the rank of the decomposition to 30 and use 100 iterations of burn in and 100 iterations of sampling. For our experiments, we use the code of Xiong et al. (2010). Since BPMF does not train a model, we allow BPMF to use all of the training matrix during the prediction phase.

Table 4 lists the normalized mean squared error (NMSE) and normalized mean absolute error (NMAE), averaged over 10 random splits. Though BPMF produces the best scores, the improvement over HL-MRF-L (LME) is not significant in NMAE.

6.4.4 Image Completion

Digital image completion requires models that understand how pixels relate to each other, such that when some pixels are unobserved, the model can infer their values from parts of the image that are observed. We construct pixel-grid HL-MRFs for image completion. We test these models using the experimental setup of Poon and Domingos (2011): we reconstruct images from the Olivetti face data set and the Caltech101 face category. The Olivetti data set contains 400 images, 64 pixels wide and tall, and the Caltech101 face category contains 435 examples of faces, which we crop to the center 64 by 64 patch, as was done by Poon

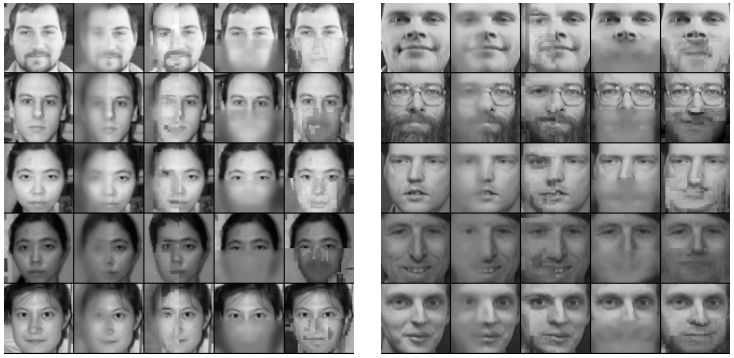


Figure 2: Example results on image completion of Caltech101 (left) and Olivetti (right) faces. From left to right in each column: (1) true face, left side predictions by (2) HL-MRFs and (3) SPNs, and bottom half predictions by (4) HL-MRFs and (5) SPNs. SPN completions are downloaded from Poon and Domingos (2011).

and Domingos (2011). Following their experimental setup, we hold out the last fifty images and predict either the left half of the image or the bottom half.

The HL-MRFs in this experiment are much more complex than the ones in our other experiments because we allow each pixel to have its own weight for the following rules, which encode agreement or disagreement between neighboring pixels:

```
Bright("P_ij", I) && North("P_ij", Q) -> Bright(Q, I)
Bright("P_ij", I) && North("P_ij", Q) -> !Bright(Q, I)
!Bright("P_ij", I) && North("P_ij", Q) -> Bright(Q, I)
!Bright("P_ij", I) && North("P_ij", Q) -> !Bright(Q, I)
```

where `Bright("P_ij", I)` is the normalized brightness of pixel `"P_ij"` in image `I`, and `North("P_ij", Q)` indicates that `Q` is the north neighbor of `"P_ij"`. We similarly include analogous rules for the south, east, and west neighbors, as well as the pixels mirrored across the horizontal and vertical axes. This setup results in up to 24 rules per pixel, (boundary pixels may not have north, south, east, or west neighbors) which, in a 64 by 64 image, produces 80,896 PSL rules.

We train these HL-MRFs using SP with a 5.0 step size on the first 200 images of each data set and test on the last fifty. For training, we maximize the data log-likelihood of uniformly random held-out pixels for each training image, allowing for generalization throughout the image. Table 5 lists our results and others reported by Poon and Domingos (2011) for sum-product networks (SPN), deep Boltzmann machines (DBM), deep belief networks (DBN), principal component analysis (PCA), and nearest neighbor (NN). HL-MRFs produce the best mean squared error on the left- and bottom-half settings for the Caltech101 set and the left-half setting in the Olivetti set. Only sum product networks produce lower error

on the Olivetti bottom-half faces. Some reconstructed faces are displayed in Figure 2, where the shallow, pixel-based HL-MRFs produce comparably convincing images to sum-product networks, especially in the left-half setting, where HL-MRFs can learn which pixels are likely to mimic their horizontal mirror. While neither method is particularly good at reconstructing the bottom half of faces, the qualitative difference between the deep SPN and the shallow HL-MRF completions is that SPNs seem to hallucinate different faces, often with some artifacts, while HL-MRFs predict blurry shapes roughly the same pixel intensity as the observed, top half of the face. The tendency to better match pixel intensity helps HL-MRFs score better quantitatively on the Caltech101 faces, where the lighting conditions are more varied than in Olivetti faces.

Training and predicting with these HL-MRFs takes little time. In our experiments, training each model takes about 45 minutes on a 12-core machine, while predicting takes under a second per image. While Poon and Domingos (2011) report faster training with SPNs, both HL-MRFs and SPNs clearly belong to a class of faster models when compared to DBNs and DBMs, which can take days to train on modern hardware.

7. Related Work

Researchers in artificial intelligence and machine learning have long been interested in predicting interdependent unknowns using structural dependencies. Some of the earliest work in this area is inductive logic programming (ILP) (Muggleton and De Raedt, 1994), in which structural dependencies are described with first-order logic. Using first-order logic has several advantages. First, it can capture many types of dependencies among variables, such as correlations, anti-correlations, and implications. Second, it can compactly specify dependencies that hold across many different sets of propositions by using variables as wild-cards that match entities in the data. These features enable the construction of intuitive, general-purpose models that are easily applicable or adapted to different domains. Inference for ILP finds the propositions that satisfy a query, consistent with a relational knowledge base. However, ILP is limited by its difficulty in coping with uncertainty. Standard ILP approaches only model dependencies which hold universally, and such dependencies are rare in real-world data.

Another broad area of research, probabilistic methods, directly models uncertainty over unknowns. Probabilistic graphical models (PGMs) (Koller and Friedman, 2009) are a family of formalisms for specifying joint distributions over interdependent unknowns through graphical structures. The graphical structure of a PGM generally represents conditional independence relationships among random variables. Explicitly representing conditional independence relationships allows a distribution to be more compactly parametrized. For example, in the worst case, a discrete distribution could be represented by an exponentially large table over joint assignments to the random variables. However, describing the distribution in smaller, conditionally independent pieces can be much more compact. Similar benefits apply to continuous distributions. Algorithms for probabilistic inference and learning can also operate over the conditionally independent pieces described by the graph structure. They are therefore straightforward to apply to a wide variety of distributions. Categories of PGMs include Markov random fields (MRFs), Bayesian networks (BNs), and

dependency networks (DNs). Constructing PGMs often requires careful design, and models are usually constructed for single tasks and data sets.

More recently, researchers have sought to combine the advantages of relational and probabilistic approaches, creating the field of *statistical relational learning* (SRL) (Getoor and Taskar, 2007). SRL techniques build probabilistic models of relational data, i.e., data composed of entities and relationships connecting them. Relational data is most often described using a relational calculus, but SRL techniques are also equally applicable to similar categories of data that go by other names, such as graph data or network data. Modeling relational data is inherently complicated by the large number of interconnected and overlapping structural dependencies that are typically present. This complication has motivated two directions of work. The first direction is algorithmic, seeking inference and learning methods that scale up to high dimensional models. The other direction is both user-oriented and—as a growing body of evidence shows—supported by learning theory, seeking formalisms for compactly specifying entire groups of dependencies in the model that share both form and parameters. Specifying these grouped dependencies, often in the form of templates via a domain-specific language, is convenient for users. Most often in relational data the structural dependencies hold without regard to the identities of entities, instead being induced by an entity's class (or classes) and the structure of its relationships with other entities. Therefore, many SRL models and languages give users the ability to specify dependencies in this abstract form and ground out models over specific data sets based on these definitions. In addition to convenience, recent work in learning theory says that repeated dependencies with tied parameters can be the key to generalizing from a few—or even one—large, structured training example(s) (London et al., 2016).

A related field to SRL is *structured prediction* (SP) (Bakir et al., 2007; Nowozin et al., 2016), which generalizes the tasks of classification and regression to the task of predicting structured objects. The loss function used during learning is generalized to a task-appropriate loss function that scores disagreement between predictions and the true structures. Often, models for structured prediction take the form of energy functions that are linear in their parameters. Therefore, prediction with such models is equivalent to MAP inference for MRFs. A distinct branch of SP is learn-to-search methods, in which the problem is decomposed into a series of one-dimension prediction problems. The challenge is to learn a good order in which to predict the components of the structure, so that each one-dimension prediction problem can be conditioned on the most useful information. Examples of learn-to-search methods include incremental structured perceptron (Collins and Roark, 2004), SEARN (Daumé III et al., 2009), DAgger (Ross et al., 2011), and AggreVaTe (Ross and Bagnell, 2014).

In this paper we focus on SP methods that perform joint prediction directly. Better understanding the differences and relative advantages of joint-prediction methods and learn-to-search methods is an important direction for future work. In the rest of this section we survey models and domain-specific languages for SP and SRL (Section 7.1), inference methods (Section 7.2), and learning methods (Section 7.3).

7.1 Models and Languages

SP and SRL encompass many approaches. One broad area of work—of which PSL is a part—uses first-order logic and other relational formalisms to specify templates for PGMs. Probabilistic relational models (Friedman et al., 1999) define templates for BNs in terms of a database schema, and they can be grounded out over instances of that schema to create BNs. Relational dependency networks (Neville and Jensen, 2007) template RNs using structured query language (SQL) queries over a relational schema. Markov logic networks (MLNs) (Richardson and Domingos, 2006) use first-order logic to define Boolean MRFs. Each logical clause in a first-order knowledge base is a template for a set of potentials when the MLN is grounded out over a set of propositions. Whether each proposition is true is a Boolean random variable, and the potential has a value of one when the corresponding ground clause is satisfied by the propositions and zero when it is not. (MLNs are formulated such that higher values of the energy function are more probable.) Clauses can either be weighted, in which case the potential has the weight of the clause that templated it, or unweighted, in which case in must hold universally, as in ILP. In these ways, MLNs are similar to PSL. Whereas MLNs are defined over Boolean variables, PSL is a templating language for HL-MRFs, which are defined over continuous variables. However, these continuous variables can be used to model discrete quantities. See Section 2 for more information on the relationships between HL-MRFs and discrete MRFs, and Section 6.4 for empirical comparisons between the two. As we show, HL-MRFs and PSL scale much better while retaining the rich expressivity and accuracy of their discrete counterparts. In addition, HL-MRFs and PSL can reason directly about continuous data.

PSL is part of a broad family of probabilistic programming languages (Gordon et al., 2014). The goals of probabilistic programming and SRL often overlap. Probabilistic programming seeks to make constructing probabilistic models easy for the end user, and separate model specification from the development of inference and learning algorithms. If algorithms can be developed for the entire space of models covered by a language, then it is easy for users to experiment with including and excluding different model components. It also makes it easy for existing models to benefit from improved algorithms. Separation of model specification and algorithms is also useful in SRL for the same reasons. In this paper we emphasize designing algorithms that are flexible enough to support the full class of HL-MRFs. Examples of probabilistic programming languages include IBAL (Pfeffer, 2001), BLOG (Milch et al., 2005), Markov logic (Richardson and Domingos, 2006), ProbLog (De Raedt et al., 2007), Church (Goodman et al., 2008), Figaro (Pfeffer, 2009), FACTORIE (McCallum et al., 2009), Anglican (Wood et al., 2014), and Edward (Tran et al., 2016).

Other formalisms have also been proposed for probabilistic reasoning over continuous domains and other domains equipped with semirings. Hybrid Markov logic networks (Wang and Domingos, 2008) mix discrete and continuous variables. In addition to the dependencies over discrete variables supported by MLNs, they support soft equality constraints between two variables of the same form as those defined by squared arithmetic rules in PSL, as well as linear potentials of the form $y_1 - y_2$ for a soft inequality constraint $y_1 > y_2$. Inference in hybrid MLNs is intractable. Wang and Domingos (2008) propose a random walk algorithm for approximate MAP inference. Another related formalism is aProbLog (Kimmig et al., 2011), which generalizes ProbLog to allow clauses to be annotated with elements from a

semiring, generalizing ProbLog's support for clauses annotated with probabilities. Many common inference tasks can be generalized from this perspective as algebraic model counting (Kimmig et al., 2016). The PITA system (Riguzzi and Swift, 2011) for probabilistic logic programming can also be viewe as implementing inference over various semirings.

7.2 Inference

Whether viewed as MAP inference for an MRF or SP without probabilistic semantics, searching over a structured space to find the optimal prediction is an important but difficult task. It is NP-hard in general (Shimony, 1994), so much work has focused on approximations and identifying classes of problems for which it is tractable. A well-studied approximation technique is local consistency relaxation (LCR) (Wainwright and Jordan, 2008). Inference is first viewed as an equivalent optimization over the realizable expected values of the potentials, called the marginal polytope. When the variables are discrete and each potential is an indicator that a subset of variables is in a certain state, this optimization becomes a linear program. Each variable in the program is the marginal probability that a variable is a particular state or the variables associated with a potential are in a particular joint state. The marginal polytope is then the set of marginal probabilities that are globally consistent. The number of linear constraints required to define the marginal polytope is exponential in the size of the problem, however, so the linear program has to be relaxed in order to be tractable. In a local consistency relaxation, the marginal polytope is relaxed to the local polytope, in which the marginals over variables and potential states are only locally consistent in the sense that each marginal over potential states sums to the marginal distributions over the associated variables.

A large body of work has focused on solving the LCR objective quickly. Typically, off-the-shelf convex optimization methods do not scale well for large graphical models and structured predictors (Yanover et al., 2006), so a large branch of research has investigated highly scalable message-passing algorithms. One approach is dual decomposition (DD) (Sontag et al., 2011), which solves a problem dual to the LCR objective. Many DD algorithms use coordinate descent, such as TRW-S (Kolmogorov, 2006), MSD (Werner, 2007), MPLP (Globerson and Jaakkola, 2007), and ADLP (Meshi and Globerson, 2011). Other DD algorithms use subgradient-based approaches (e.g., Jojic et al., 2010; Komodakis et al., 2011; Schwing et al., 2012).

Another approach to solving the LCR objective uses message-passing algorithms to solve the problem directly in its primal form. One well-known algorithm is that of Ravikumar et al. (2010a), which uses proximal optimization, a general approach that iteratively improves the solution by searching for nearby improvements. The authors also provide rounding guarantees for when the relaxed solution is integral, i.e., the relaxation is tight, allowing the algorithm to converge faster. Another message-passing algorithm that solves the primal objective is AD^3 (Martins et al., 2015), which uses the alternating direction method of multipliers (ADMM). AD^3 optimizes objective (10) for binary, pairwise MRFs and supports the addition of certain deterministic constraints on the variables. A third example of a primal message-passing algorithm is APLP (Meshi and Globerson, 2011), which is the primal analog of ADLP. Like AD^3, it uses ADMM to optimize the objective.

Other approaches to approximate inference include tighter linear programming relaxations (Sontag et al., 2008, 2012). These tighter relaxations enforce local consistency on variable subsets that are larger than individual variables, which makes them *higher-order local consistency relaxations*. Mezuman et al. (2013) developed techniques for special cases of higher-order relaxations, such as when the MRF contains cardinality potentials, in which the probability of a configuration depends on the number of variables in a particular state. Researchers have also explored nonlinear convex programming relaxations, e.g., Ravikumar and Lafferty (2006) and Kumar et al. (2006).

Previous analyses have identified particular subclasses whose local consistency relaxations are tight, i.e., the maximum of the relaxed program is exactly the maximum of the original problem. These special classes include graphical models with tree-structured dependencies, models with submodular potential functions, models encoding bipartite matching problems, and those with *nand* potentials and perfect graph structures (Wainwright and Jordan, 2008; Schrijver, 2003; Jebara, 2009; Foulds et al., 2011). Researchers have also studied performance guarantees of other subclasses of the first-order local consistency relaxation. Kleinberg and Tardos (2002) and Chekuri et al. (2005) considered the metric labeling problem. Feldman et al. (2005) used the local consistency relaxation to decode binary linear codes.

In this paper we examine the classic problem of MAX SAT—finding a joint Boolean assignment to a set of propositions that maximizes the sum of a set of weighted clauses that are satisfied—as an instance of SP. Researchers have also considered approaches to solving MAX SAT other than the one one we study, the randomized algorithm of Goemans and Williamson (1994). One line of work focusing on convex programming relaxations has obtained stronger rounding guarantees than Goemans and Williamson (1994) by using nonlinear programming, e.g., Asano and Williamson (2002) and references therein. Other work does not use the probabilistic method but instead searches for discrete solutions directly, e.g., Mills and Tsang (2000), Larrosa et al. (2008), and Choi et al. (2009). We note that one such approach, that of Wah and Shang (1997), is essentially a type of DD formulated for MAX SAT. A more recent approach blends convex programming and discrete search via mixed integer programming (Davies and Bacchus, 2013). Additionally, Huynh and Mooney (2009) introduced a linear programming relaxation for MLNs inspired by MAX SAT relaxations, but the relaxation of general Markov logic provides no known guarantees on the quality of solutions.

Finally, *lifted inference* takes advantage of symmetries in probability distributions to reduce the amount of work required for inference. Some of the earliest approaches identified repeated dependency structures in PGMs to avoid repeated computations (Koller and Pfeffer, 1997; Pfeffer et al., 1999). Lifted inference has been widely applied in SRL because the templates that are commonly used to define PGMs often induce symmetries. Various inference techniques for discrete MRFs have been extended to a lifted approach, including belief propagation (Jaimovich et al., 2007; Singla and Domingos, 2008; Kersting et al., 2009) and Gibbs sampling (Venugopal and Gogate, 2012). Approaches to lifted convex optimization (Mladenov et al., 2012) might be extended to HL-MRFs. See de Salvo Braz et al. (2007), Kersting (2012), and Kimmig et al. (2015) for more information on lifted inference.

7.3 Learning

Taskar et al. (2004) connected SP and PGMs by showing how to train MRFs with large-margin estimation, a generalization of the large-margin objective for binary classification used to train support vector machines (Vapnik, 2000). Large-margin learning is a well-studied approach to train structured predictors because it directly incorporates the structured loss function into a convex upper bound on the true objective: the regularized expected risk. The learning objective is to find the parameters with smallest norm such that a linear combination of feature functions assign a better score to the training data than all other possible predictions. The amount by which the score of the correct prediction must exceed the score of other predictions is scaled using the structured loss function. The objective is therefore encoded as a norm minimization problem subject to many linear constraints, one for each possible prediction in the structured space.

Structured SVMs (Tsochantaridis et al., 2005) extend large-margin estimation to a broad class of structured predictors and admit a tractable cutting-plane learning algorithm. This algorithm will terminate in a number of iterations linear in the size of the problem, and so the computational challenge of large-margin learning for structured prediction comes down to the task of finding the most violated constraint in the learning objective. This can be accomplished by optimizing the energy function plus the loss function. In other words, the task is to find the structure that is the best combination of being favored by the energy function but unfavored by the loss function. Often, the loss function decomposes over the components of the prediction space, so the combined energy function and loss function can often be viewed as simply the energy function of another structured predictor that is equally challenging or easy to optimize, such as when the space of structures is a set of discrete vectors and the loss function is the Hamming distance.

It is common during large-margin estimation that no setting of the parameters can predict all the training data without error. In this case, the training data is said to not be separable, again generalizing the notion of linear separability in the feature space from binary classification. The solution to this problem is to add slack variables to the constraints that require the training data to be assigned the best score. The magnitude of the slack variables are penalized in the learning objective, so estimation must trade off between the norm of the parameters and violating the constraints. Joachims et al. (2009) extend this formulation to a "one slack" formulation, in which a single slack variable is used for all the constraints across all training examples, which is more efficient. We use this framework for large-margin estimation for HL-MRFs in Section 6.3.

The repeated inferences required for large-margin learning, one to find the most-violated constraint at each iteration, can become computationally expensive. Therefore researchers have explored speeding up learning by interleaving the inference problem with the learning problem. In the cutting-plane formulation discussed above, the objective is equivalently a saddle-point problem, with the solution at the minimum with respect to the parameters and the maximum with respect to the inference variables. Taskar et al. (2005) proposed dualizing the inner inference problem to form a joint minimization. For SP problems with a tight duality gap, i.e., the dual problem has the same optimal value as the primal problem, this approach leads to an equivalent, convex optimization that can be solved for all variables simultaneously. In other words, the learning and most-violated constraint problems are

solved simultaneously, greatly reducing training time. For problems with non-tight duality gaps, e.g., MAP inference in general, discrete MRFs, Meshi et al. (2010) showed that the same principle can be applied by using approximate inference algorithms like dual decomposition to bound the primal objective.

A related problem to parameter learning is *structure learning*, i.e., identifying an accurate dependency structure for a model. A common SRL approach is searching over the space of templates for PGMs. For probabilistic relational models, Friedman et al. (1999) learned structures described in the vocabulary of relational schemas. For models that are templated with first-order-logic-like languages, such as PSL and MLNs, these approaches take the form of rule learning. Based on rule-learning techniques from inductive logic programming (e.g., Richards and Mooney, 1992; De Raedt and Dehaspe, 1996) a series of approaches have sought to learn MLN rules from relational data. Initially, Kok and Domingos (2005) learned rules by generating candidates and performing a beam search to identify rules that improved a weighted pseudolikelihood objective. Then, Mihalkova and Mooney (2007) observed that the previous approach generated candidate rules without regard to the data, so they introduced an approach that used the data to guide the proposal of rules via *relational pathfinding*. Kok and Domingos (2010) improved on that by first performing graph clustering to find common *motifs*, which are common subgraphs, to guide rule proposal. They observed that modifying a rule set one clause at a time often got stuck in poor local optima, and by using the motifs as refinement operators instead, they were able to converge to better optima. Other approaches to structure learning search directly over grounded PGMs, including ℓ_1-regularized pseudolikelihood maximization (Ravikumar et al., 2010b) and grafting (Perkins et al., 2003; Zhu et al., 2010). These methods can all be extended to HL-MRFs and PSL.

8. Conclusion

In this paper we introduced HL-MRFs, a new class of probabilistic graphical models that unite and generalize several approaches to modeling relational and structured data: Boolean logic, probabilistic graphical models, and fuzzy logic. HL-MRFs can capture relaxed, probabilistic inference with Boolean logic and exact, probabilistic inference with fuzzy logic, making them useful models for both discrete and continuous data. HL-MRFs also generalize these inference techniques with additional expressivity, allowing for even more flexibility. HL-MRFs are a significant addition to the the library of machine learning tools because they embody a useful point in the spectrum of models that trade off between scalability and expressivity. As we showed, they can be easily applied to a wide range of structured problems in machine learning and achieve high-quality predictive performance, competitive with or surpassing the performance of canonical approaches. However, these other models either do not scale as well, like discrete MRFs, or are not as versatile in their ability to capture a wide range of problems, like Bayesian probabilistic matrix factorization.

We also introduced PSL, a probabilistic programming language for HL-MRFs. PSL makes HL-MRFs easy to design, allowing users to encode their ideas for structural dependencies using an intuitive syntax based on first-order logic. PSL also helps accelerate a time-consuming aspect of the modeling process: refining a model. In contrast with other types of models that require specialized inference and learning algorithms depending on

which structural dependencies are included, HL-MRFs can encode many types of dependencies and scale well with the same inference and learning algorithms. PSL makes it easy to quickly add, remove, and modify dependencies in the model and rerun inference and learning, allowing users to quickly improve the quality of their models. Finally, because PSL uses a first-order syntax, each PSL program actually specifies an entire class of HL-MRFs, parameterized by the particular data set over which it is grounded. Therefore, a model or components of a model refined for one data set can easily be applied to others.

Next, we introduced inference and learning algorithms that scale to large problems. The MAP inference algorithm is far more scalable than standard tools for convex optimization because it leverages the sparsity that is so common to the dependencies in structured prediction. The supervised learning algorithms extend standard learning objectives to HL-MRFs. Together, this combination of an expressive formalism, a user-friendly probabilistic programming language, and highly scalable algorithms enables researchers and practitioners to easily build large-scale, accurate models of relational and structured data.[7]

This paper also lays the foundation for many lines of future work. Our analysis of local consistency relaxation (LCR) as a hierarchical optimization is a general proof technique, and it could be used to derive compact forms for other LCR objectives. As in the case of MRFs defined using logical clauses, such compact forms can simplify analysis and could lead to a greater understanding of LCR for other classes of MRFs. Another important line of work is understanding what guarantees apply to the MAP states of HL-MRFs. Can anything be said about their ability to approximate MAP inference in discrete models that go beyond the models already covered by the known rounding guarantees? Future directions also include developing new algorithms for HL-MRFs. One important direction is marginal inference for HL-MRFs and algorithms for sampling from them. Unlike marginal inference for discrete distributions, which computes the marginal probability that a variable is in a particular state, marginal inference for HL-MRFs requires finding the marginal probability that a variable is in a particular range. One option for doing so, as well as generating samples from HL-MRFs, is to extend the hit-and-run sampling scheme of Broecheler and Getoor (2010). This method was developed for continuous constrained MRFs with piecewise-linear potentials. There are also many new domains to which HL-MRFs and PSL can be applied. With these modeling tools, researchers can design and apply new solutions to structured prediction problems.

Acknowledgments

We acknowledge the many people who have contributed to the development of HL-MRFs and PSL. Contributors include Eriq Augustine, Shobeir Fakhraei, James Foulds, Angelika Kimmig, Stanley Kok, Ben London, Hui Miao, Lilyana Mihalkova, Dianne P. O'Leary, Jay Pujara, Arti Ramesh, Theodoros Rekatsinas, and V.S. Subrahmanian. This work was supported by NSF grants CCF0937094 and IIS1218488, and IARPA via DoI/NBC contract number D12PC00337. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

7. An open source implementation, tutorials, and data sets are available at `http://psl.linqs.org`.

Appendix A. Proof of Theorem 2

In this appendix, we prove the equivalence of objectives (7) and (10). Our proof analyzes the local consistency relaxation to derive an equivalent, more compact optimization over only the variable pseudomarginals $\boldsymbol{\mu}$ that is identical to the MAX SAT relaxation. Since the variables are Boolean, we refer to each pseudomarginal $\mu_i(1)$ as simply μ_i. Let $\boldsymbol{x}_j^F$ denote the unique setting such that $\phi_j(\boldsymbol{x}_j^F) = 0$. (I.e., $\boldsymbol{x}_j^F$ is the setting in which each literal in the clause C_j is false.)

We begin by reformulating the local consistency relaxation as a hierarchical optimization, first over the variable pseudomarginals $\boldsymbol{\mu}$ and then over the factor pseudomarginals $\boldsymbol{\theta}$. Due to the structure of local polytope $\mathbb{L}$, the pseudomarginals $\boldsymbol{\mu}$ parameterize inner linear programs that decompose over the structure of the MRF, such that—given fixed $\boldsymbol{\mu}$—there is an independent linear program $\hat{\phi}_j(\boldsymbol{\mu})$ over $\boldsymbol{\theta}_j$ for each clause C_j. We rewrite objective (10) as

$$\operatorname*{arg\,max}_{\boldsymbol{\mu} \in [0,1]^n} \sum_{C_j \in \boldsymbol{C}} \hat{\phi}_j(\boldsymbol{\mu}), \tag{71}$$

where

$$\hat{\phi}_j(\boldsymbol{\mu}) = \max_{\boldsymbol{\theta}_j} \; w_j \sum_{\boldsymbol{x}_j | \boldsymbol{x}_j \neq \boldsymbol{x}_j^F} \theta_j(\boldsymbol{x}_j) \tag{72}$$

$$\text{such that} \sum_{\boldsymbol{x}_j | x_j(i)=1} \theta_j(\boldsymbol{x}_j) = \mu_i \qquad \forall i \in I_j^+ \tag{73}$$

$$\sum_{\boldsymbol{x}_j | x_j(i)=0} \theta_j(\boldsymbol{x}_j) = 1 - \mu_i \qquad \forall i \in I_j^- \tag{74}$$

$$\sum_{\boldsymbol{x}_j} \theta_j(\boldsymbol{x}_j) = 1 \tag{75}$$

$$\theta_j(\boldsymbol{x}_j) \geq 0 \qquad \forall \boldsymbol{x}_j \; . \tag{76}$$

It is straightforward to verify that objectives (10) and (71) are equivalent for MRFs with disjunctive clauses for potentials. All constraints defining $\mathbb{L}$ can be derived from the constraint $\boldsymbol{\mu} \in [0,1]^n$ and the constraints in the definition of $\hat{\phi}_j(\boldsymbol{\mu})$. We have omitted redundant constraints to simplify analysis.

To make this optimization more compact, we replace each inner linear program $\hat{\phi}_j(\boldsymbol{\mu})$ with an expression that gives its optimal value for any setting of $\boldsymbol{\mu}$. Deriving this expression requires reasoning about any maximizer $\boldsymbol{\theta}_j^\star$ of $\hat{\phi}_j(\boldsymbol{\mu})$, which is guaranteed to exist because problem (72) is bounded and feasible[8] for any parameters $\boldsymbol{\mu} \in [0,1]^n$ and w_j.

We first derive a sufficient condition for the linear program to not be fully satisfiable, in the sense that it cannot achieve a value of w_j, the maximum value of the weighted potential

8. Setting $\theta_j(\boldsymbol{x}_j)$ to the probability defined by $\boldsymbol{\mu}$ under the assumption that the elements of $\boldsymbol{x}_j$ are independent, i.e., the product of the pseudomarginals, is always feasible.

$w_j \phi_j(\boldsymbol{x})$. Observe that, by the objective (72) and the simplex constraint (75), showing that $\hat{\phi}_j(\boldsymbol{\mu})$ is not fully satisfiable is equivalent to showing that $\theta^\star_j(\boldsymbol{x}^F_j) > 0$.

Lemma 16 *If*

$$\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) < 1 \ ,$$

then $\theta^\star_j(\boldsymbol{x}^F_j) > 0$.

Proof By the simplex constraint (75),

$$\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) < \sum_{\boldsymbol{x}_j} \theta^\star_j(\boldsymbol{x}_j) \ .$$

Also, by summing all the constraints (73) and (74),

$$\sum_{\boldsymbol{x}_j | \boldsymbol{x}_j \neq \boldsymbol{x}^F_j} \theta^\star_j(\boldsymbol{x}_j) \leq \sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) \ ,$$

because all the components of $\boldsymbol{\theta}^\star$ are nonnegative, and—except for $\theta^\star_j(\boldsymbol{x}^F_j)$—they all appear at least once in constraints (73) and (74). These bounds imply

$$\sum_{\boldsymbol{x}_j | \boldsymbol{x}_j \neq \boldsymbol{x}^F_j} \theta^\star_j(\boldsymbol{x}_j) < \sum_{\boldsymbol{x}_j} \theta^\star_j(\boldsymbol{x}_j) \ ,$$

which means $\theta^\star_j(\boldsymbol{x}^F_j) > 0$, completing the proof. ■

We next show that if $\hat{\phi}_j(\boldsymbol{\mu})$ is parameterized such that it is not fully satisfiable, as in Lemma 16, then its optimum always takes a particular value defined by $\boldsymbol{\mu}$.

Lemma 17 *If* $w_j > 0$ *and* $\theta^\star_j(\boldsymbol{x}^F_j) > 0$*, then*

$$\sum_{\boldsymbol{x}_j | \boldsymbol{x}_j \neq \boldsymbol{x}^F_j} \theta^\star_j(\boldsymbol{x}_j) = \sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) \ .$$

Proof We prove the lemma via the Karush-Kuhn-Tucker (KKT) conditions (Karush, 1939; Kuhn and Tucker, 1951). Since problem (72) is a maximization of a linear function subject to linear constraints, the KKT conditions are necessary and sufficient for any optimum $\boldsymbol{\theta}^\star_j$.

Before writing the relevant KKT conditions, we introduce some necessary notation. For a state $\boldsymbol{x}_j$, we need to reason about the variables that disagree with the unsatisfied state $\boldsymbol{x}^F_j$. Let

$$d(\boldsymbol{x}_j) \triangleq \left\{ i \in I_j^+ \cup I_j^- | \boldsymbol{x}_j(i) \neq \boldsymbol{x}^F_j(i) \right\}$$

be the set of indices for the variables that do not have the same value in the two states $\boldsymbol{x}_j$ and $\boldsymbol{x}^F_j$.

We now write the relevant KKT conditions for $\boldsymbol{\theta}^\star_j$. Let $\boldsymbol{\lambda}, \boldsymbol{\alpha}$ be real-valued vectors where $|\boldsymbol{\lambda}| = |I_j^+| + |I_j^-| + 1$ and $|\boldsymbol{\alpha}| = |\boldsymbol{\theta}_j|$. Let each λ_i correspond to a constraint (73) or (74)

for $i \in I_j^+ \cup I_j^-$, and let λ_Δ correspond to the simplex constraint (75). Also, let each $\alpha_{\boldsymbol{x}_j}$ correspond to a constraint (76) for each $\boldsymbol{x}_j$. Then, the following KKT conditions hold:

$$\alpha_{\boldsymbol{x}_j} \geq 0 \qquad \forall \boldsymbol{x}_j \tag{77}$$

$$\alpha_{\boldsymbol{x}_j} \theta^\star_j(\boldsymbol{x}_j) = 0 \qquad \forall \boldsymbol{x}_j \tag{78}$$

$$\lambda_\Delta + \alpha_{\boldsymbol{x}^F_j} = 0 \tag{79}$$

$$w_j + \sum_{i \in d(\boldsymbol{x}_j)} \lambda_i + \lambda_\Delta + \alpha_{\boldsymbol{x}_j} = 0 \qquad \forall \boldsymbol{x}_j \neq \boldsymbol{x}^F_j \ . \tag{80}$$

Since $\theta^\star_j(\boldsymbol{x}^F_j) > 0$, by condition (78), $\alpha_{\boldsymbol{x}^F_j} = 0$. By condition (79), then $\lambda_\Delta = 0$. From here we can bound the other elements of $\boldsymbol{\lambda}$. Observe that for every $i \in I_j^+ \cup I_j^-$, there exists a state $\boldsymbol{x}_j$ such that $d(\boldsymbol{x}_j) = \{i\}$. Then, it follows from condition (80) that there exists $\boldsymbol{x}_j$ such that, for every $i \in I_j^+ \cup I_j^-$,

$$w_j + \lambda_i + \lambda_\Delta + \alpha_{\boldsymbol{x}_j} = 0 \ .$$

Since $\alpha_{\boldsymbol{x}_j} \geq 0$ by condition (77) and $\lambda_\Delta = 0$, it follows that $\lambda_i \leq -w_j$. With these bounds, we show that, for any state $\boldsymbol{x}_j$, if $|d(\boldsymbol{x}_j)| \geq 2$, then $\theta^\star_j(\boldsymbol{x}_j) = 0$. Assume that for some state $\boldsymbol{x}_j$, $|d(\boldsymbol{x}_j)| \geq 2$. By condition (80) and the derived constraints on $\boldsymbol{\lambda}$,

$$\alpha_{\boldsymbol{x}_j} \geq (|d(\boldsymbol{x}_j)| - 1) w_j > 0 \ .$$

With condition (78), $\theta^\star_j(\boldsymbol{x}_j) = 0$. Next, observe that for all $i \in I_j^+$ (resp. $i \in I_j^-$) and for any state $\boldsymbol{x}_j$, if $d(\boldsymbol{x}_j) = \{i\}$, then $x_j(i) = 1$ (resp. $x_j(i) = 0$), and for any other state $\boldsymbol{x}'_j$ such that $x'_j(i) = 1$ (resp. $x'_j(i) = 0$), $d(\boldsymbol{x}'_j) \geq 2$. By constraint (73) (resp. constraint (74)), $\theta^\star(\boldsymbol{x}_j) = \mu_i$ (resp. $\theta^\star(\boldsymbol{x}_j) = 1 - \mu_i$).

We have shown that if $\theta^\star_j(\boldsymbol{x}^F_j) > 0$, then for all states $\boldsymbol{x}_j$, if $d(\boldsymbol{x}_j) = \{i\}$ and $i \in I_j^+$ (resp. $i \in I_j^-$), then $\theta^\star_j(\boldsymbol{x}_j) = \mu_i$ (resp. $\theta^\star_j(\boldsymbol{x}_j) = 1 - \mu_i$), and if $|d(\boldsymbol{x}_j)| \geq 2$, then $\theta^\star_j(\boldsymbol{x}_j) = 0$. This completes the proof. ■

Lemma 16 says if $\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) < 1$, then $\hat{\phi}_j(\boldsymbol{\mu})$ is not fully satisfiable, and Lemma 17 provides its optimal value. We now reason about the other case, when $\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) \geq 1$, and we show that this condition is sufficient to ensure that $\hat{\phi}_j(\boldsymbol{\mu})$ is fully satisfiable.

Lemma 18 *If* $w_j > 0$ *and*

$$\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) \geq 1 \ ,$$

then $\theta^\star_j(\boldsymbol{x}^F_j) = 0$.

Proof We prove the lemma by contradiction. Assume that $w_j > 0$, $\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) \geq 1$, and that the lemma is false, $\theta^\star_j(\boldsymbol{x}^F_j) > 0$. Then, by Lemma 17,

$$\sum_{\boldsymbol{x}_j | \boldsymbol{x}_j \neq \boldsymbol{x}^F_j} \theta^\star_j(\boldsymbol{x}_j) \geq 1 \ .$$

The assumption that $\theta_j^\star(\boldsymbol{x}_j^F) > 0$ implies

$$\sum_{\boldsymbol{x}_j} \theta_j^\star(\boldsymbol{x}_j) > 1,$$

which is a contradiction, since it violates the simplex constraint (75). The possibility that $\theta_j^\star(\boldsymbol{x}_j^F) < 0$ is excluded by the nonnegativity constraints (76). ■

For completeness and later convenience, we also state the value of $\hat{\phi}_j(\boldsymbol{\mu})$ when it is fully satisfiable.

Lemma 19 *If $\theta_j^\star(\boldsymbol{x}_j^F) = 0$, then*

$$\sum_{\boldsymbol{x}_j | \boldsymbol{x}_j \neq \boldsymbol{x}_j^F} \theta_j^\star(\boldsymbol{x}_j) = 1 \ .$$

Proof The lemma follows from the simplex constraint (75). ■

We can now combine the previous lemmas into a single expression for the value of $\hat{\phi}_j(\boldsymbol{\mu})$.

Lemma 20 *For any feasible setting of $\boldsymbol{\mu}$,*

$$\hat{\phi}_j(\boldsymbol{\mu}) = w_j \min\left\{\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i), 1\right\} \ .$$

Proof The lemma is trivially true if $w_j = 0$ since any assignment will yield zero value. If $w_j > 0$, then we consider two cases. In the first case, if $\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) < 1$, then, by Lemmas 16 and 17,

$$\hat{\phi}_j(\boldsymbol{\mu}) = w_j \left(\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i)\right) \ .$$

In the second case, if $\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i) \geq 1$, then, by Lemmas 18 and 19,

$$\hat{\phi}_j(\boldsymbol{\mu}) = w_j \ .$$

By factoring out w_j, we can rewrite this piecewise definition of $\hat{\phi}_j(\boldsymbol{\mu})$ as w_j multiplied by the minimum of $\sum_{i \in I_j^+} \mu_i + \sum_{i \in I_j^-} (1 - \mu_i)$ and 1, completing the proof. ■

This leads to our final equivalence result.

Theorem 2 *For an MRF with potentials corresponding to disjunctive logical clauses and associated nonnegative weights, the first-order local consistency relaxation of MAP inference is equivalent to the MAX SAT relaxation of Goemans and Williamson (1994). Specifically, any partial optimum $\boldsymbol{\mu}^\star$ of objective (10) is an optimum $\hat{\boldsymbol{y}}^\star$ of objective (7), and vice versa.*

Proof Substituting the solution of the inner optimization from Lemma 20 into the local consistency relaxation objective (71) gives a projected optimization over only $\boldsymbol{\mu}$ which is identical to the MAX SAT relaxation objective (7). ■

References

A. Abdelbar and S. Hedetniemi. Approximating MAPs for belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102(1):21–38, 1998.

N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley-Interscience, third edition, 2008.

D. Alshukaili, A. A. A. Fernandes, and N. W. Paton. Structuring linked data search results using probabilistic soft logic. In *International Semantic Web Conference (ISWC)*, 2016.

L. An and P. Tao. The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems. *Annals of Operations Research*, 133:23–46, 2005.

T. Asano and D. P. Williamson. Improved approximation algorithms for MAX SAT. *J. Algorithms*, 42(1):173–202, 2002.

S. H. Bach, M. Broecheler, L. Getoor, and D. P. O'Leary. Scaling MPE inference for constrained continuous Markov random fields. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

S. H. Bach, B. Huang, B. London, and L. Getoor. Hinge-loss Markov random fields: Convex inference for structured prediction. In *Uncertainty in Artificial Intelligence (UAI)*, 2013.

S. H. Bach, B. Huang, J. Boyd-Graber, and L. Getoor. Paired-dual learning for fast training of latent variable hinge-loss MRFs. In *International Conference on Machine Learning (ICML)*, 2015a.

S. H. Bach, B. Huang, and L. Getoor. Unifying local consistency and MAX SAT relaxations for scalable inference with rounding guarantees. In *Artificial Intelligence and Statistics (AISTATS)*, 2015b.

G. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan, editors. *Predicting Structured Data*. MIT Press, 2007.

I. Beltagy, K. Erk, and R. J. Mooney. Probabilistic soft logic for semantic textual similarity. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2014.

J. Besag. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society*, 24(3):179–195, 1975.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. *Distributed Optimization and Statistical Learning Via the Alternating Direction Method of Multipliers*. Now Publishers, 2011.

M. Broecheler and L. Getoor. Computing marginal distributions over continuous Markov networks for statistical relational learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

M. Broecheler, L. Mihalkova, and L. Getoor. Probabilistic similarity logic. In *Uncertainty in Artificial Intelligence (UAI)*, 2010a.

M. Broecheler, P. Shakarian, and V. S. Subrahmanian. A scalable framework for modeling competitive diffusion in social networks. In *Social Computing (SocialCom)*, 2010b.

C. Chekuri, S. Khanna, J. Naor, and L. Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM J. Discrete Math.*, 18 (3):608–625, 2005.

P. Chen, F. Chen, and Z. Qian. Road traffic congestion monitoring in social media with hinge-loss Markov random fields. In *IEEE International Conference on Data Mining (ICDM)*, 2014.

A. Choi, T. Standley, and A. Darwiche. Approximating weighted Max-SAT problems by compensating for relaxations. In *International Conference on Principles and Practice of Constraint Programming*, 2009.

M. Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

M. Collins and B. Roark. Incremental parsing with the perceptron algorithm. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

H. Daumé III, J. Langford, and D. Marcu. Search-based structured prediction. *Machine Learning*, 75(3):297–325, 2009.

J. Davies and F. Bacchus. Exploiting the power of MIP solvers in MAXSAT. In M. Järvisalo and A. Van Gelder, editors, *Theory and Applications of Satisfiability Testing – SAT 2013*, Lecture Notes in Computer Science, pages 166–181. Springer Berlin Heidelberg, 2013.

L. De Raedt and L. Dehaspe. Clausal discovery. *Machine Learning*, 26:1058–1063, 1996.

L. De Raedt, A. Kimmig, and H. Toivonen. ProbLog: A probabilistic Prolog and its application in link discovery. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

R. de Salvo Braz, E. Amir, and D. Roth. Lifted first-order probabilistic inference. In L. Getoor and B. Taskar, editors, *Introduction to statistical relational learning*, pages 433–451. MIT Press, 2007.

L. Deng and J. Wiebe. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

J. Ebrahimi, D. Dou, and D. Lowd. Weakly supervised tweet stance classification by relational bootstrapping. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

S. Fakhraei, B. Huang, L. Raschid, and L. Getoor. Network-based drug-target interaction prediction with probabilistic soft logic. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2014.

J. Feldman, M. J. Wainwright, and D. R. Karger. Using linear programming to decode binary linear codes. *Information Theory, IEEE Trans. on*, 51(3):954–972, 2005.

J. Foulds, N. Navaroli, P. Smyth, and A. Ihler. Revisiting MAP estimation, message passing and perfect graphs. In *AI & Statistics*, 2011.

J. Foulds, S. Kumar, and L. Getoor. Latent topic networks: A versatile probabilistic programming framework for topic models. In *International Conference on Machine Learning (ICML)*, 2015.

N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

L. Getoor and B. Taskar, editors. *Introduction to statistical relational learning*. MIT press, 2007.

L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of link structure. *Journal of Machine Learning Research (JMLR)*, 3:679–707, 2002.

A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

R. Glowinski and A. Marrocco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité, d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle*, 9(2):41–76, 1975.

M. X. Goemans and D. P. Williamson. New 3/4-approximation algorithms for the maximum satisfiability problem. *SIAM J. Discrete Math.*, 7(4):656–666, 1994.

J. Golbeck. *Computing and Applying Trust in Web-based Social Networks*. PhD thesis, University of Maryland, 2005.

K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

N. D. Goodman, V. K. Mansinghka, D. M. Roy, K. Bonawitz, and J. B. Tenenbaum. Church: A language for generative models. In *Uncertainty in Artificial Intelligence (UAI)*, 2008.

A. D. Gordon, T. A. Henzinger, A. V. Nori, and S. K. Rajamani. Probabilistic programming. In *International Conference on Software Engineering (ICSE, FOSE track)*, 2014.

G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

B. Huang, A. Kimmig, L. Getoor, and J. Golbeck. A flexible framework for probabilistic models of social trust. In *Conference on Social Computing, Behavioral-Cultural Modeling, & Prediction (SBP)*, 2013.

T. Huynh and R. Mooney. Max-margin weight learning for Markov logic networks. In *European Conference on Machine Learning (ECML)*, 2009.

A. Jaimovich, O. Meshi, and N. Friedman. Template based inference in symmetric relational Markov random fields. In *Uncertainty in Artificial Intelligence (UAI)*, 2007.

T. Jebara. MAP estimation, message passing, and perfect graphs. In *Uncertainty in Artificial Intelligence (UAI)*, 2009.

T. Joachims, T. Finley, and C. Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59, 2009.

V. Jojic, S. Gould, and D. Koller. Accelerated dual decomposition for MAP inference. In *International Conference on Machine Learning (ICML)*, 2010.

S. Kamvar, M. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in P2P networks. In *International Conference on the World Wide Web (WWW)*, 2003.

W. Karush. Minima of Functions of Several Variables with Inequalities as Side Constraints. Master's thesis, University of Chicago, 1939.

K. Kersting. Lifted probabilistic inference. In *European Conference on Artificial Intelligence (ECAI)*, 2012.

K. Kersting, B. Ahmadi, and S. Natarajan. Counting belief propagation. In *Uncertainty in Artificial Intelligence (UAI)*, 2009.

A. Kimmig, G. Van den Broeck, and L. De Raedt. An algebraic Prolog for reasoning about possible worlds. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2011.

A. Kimmig, L. Mihalkova, and L. Getoor. Lifted graphical models: A survey. *Machine Learning*, 99:1–45, 2015.

A. Kimmig, G. Van den Broeck, and L. De Raedt. Algebraic model counting. *Journal of Applied Logic*, 2016.

J. Kleinberg and É. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *J. ACM*, 49(5):616–639, 2002.

G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall, 1995.

S. Kok and P. Domingos. Learning the structure of Markov logic networks. In *International Conference on Machine Learning (ICML)*, 2005.

S. Kok and P. Domingos. Learning Markov logic networks using structural motifs. In *International Conference on Machine Learning (ICML)*, 2010.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

D. Koller and A. Pfeffer. Object-oriented Bayesian networks. In *Uncertainty in Artificial Intelligence (UAI)*, 1997.

V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 28(10):1568–1583, 2006.

N. Komodakis, N. Paragios, and G. Tziritas. MRF energy minimization and beyond via dual decomposition. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 33(3): 531–552, 2011.

P. Kouki, S. Fakhraei, J. Foulds, M. Eirinaki, and L. Getoor. HyPER: A flexible and extensible probabilistic framework for hybrid recommender systems. In *ACM Conference on Recommender Systems (RecSys)*, 2015.

H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Berkeley Symp. on Math. Statist. and Prob.*, 1951.

M. P. Kumar, P. H. S. Torr, and A. Zisserman. Solving Markov random fields using second order cone programming relaxations. In *Computer Vision and Pattern Recognition (CVPR)*, 2006.

N. Landwehr, A. Passerini, L. De Raedt, and P. Frasconi. Fast learning of relational kernels. *Machine Learning*, 78(3):305–342, 2010.

J. Larrosa, F. Heras, and S. de Givry. A logical approach to efficient Max-SAT solving. *Artificial Intelligence*, 172(2-3):204–233, 2008.

J. Li, A. Ritter, and D. Jurafsky. Inferring user preferences by probabilistic logical reasoning over social networks. *arXiv preprint arXiv:1411.2679*, 2014.

S. Liu, K. Liu, S. He, and J. Zhao. A probabilistic soft logic based approach to exploiting latent and global information in event classification. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

B. London, S. Khamis, S. H. Bach, B. Huang, L. Getoor, and L. Davis. Collective activity detection using hinge-loss Markov random fields. In *CVPR Workshop on Structured Prediction: Tractability, Learning and Inference*, 2013.

B. London, B. Huang, and L. Getoor. Stability and generalization in structured prediction. *Journal of Machine Learning Research (JMLR)*, 17(222):1–52, 2016.

D. Lowd and P. Domingos. Efficient weight learning for Markov logic networks. In *Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2007.

S. Magliacane, P. Stutz, P. Groth, and A. Bernstein. FoxPSL: An extended and scalable PSL implementation. In *AAAI Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches*, 2015.

A. F. T. Martins, M. A. T. Figueiredo, P. M. Q. Aguiar, N. A. Smith, and E. P. Xing. AD^3: Alternating Directions Dual Decomposition for MAP Inference in Graphical Models. *Journal of Machine Learning Research (JMLR)*, 16(Mar):495–545, 2015.

A. McCallum, K. Nigam, and L. H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2000.

A. McCallum, K. Schultz, and S. Singh. FACTORIE: Probabilistic programming via imperatively defined factor graphs. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

O. Meshi and A. Globerson. An alternating direction method for dual MAP LP relaxation. In *European Conference on Machine learning (ECML)*, 2011.

O. Meshi, D. Sontag, T. Jaakkola, and A. Globerson. Learning efficiently with approximate inference via dual losses. In *International Conference on Machine Learning (ICML)*, 2010.

E. Mezuman, D. Tarlow, A. Globerson, and Y. Weiss. Tighter linear program relaxations for high order graphical models. In *Uncertainty in Aritificial Intelligence (UAI)*, 2013.

H. Miao, X. Liu, B. Huang, and L. Getoor. A hypergraph-partitioned vertex programming approach for large-scale consensus optimization. In *IEEE International Conference on Big Data*, 2013.

L. Mihalkova and R. J. Mooney. Bottom-up learning of Markov logic network structure. In *International Conference on Machine Learning (ICML)*, 2007.

B. Milch, B. Marthi, S. Russell, D. Sontag, D. L. Ong, and A. Kolobov. BLOG: Probabilistic models with unknown objects. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

P. Mills and E. Tsang. Guided local search for solving SAT and weighted MAX-SAT problems. *J. Automated Reasoning*, 24(1-2):205–223, 2000.

M. Mladenov, B. Ahmadi, and K. Kersting. Lifted linear programming. In *Artificial Intelligence & Statistics (AISTATS)*, 2012.

S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19:629–679, 1994.

Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

J. Neville and D. Jensen. Relational dependency networks. *Journal of Machine Learning Research (JMLR)*, 8:653–692, 2007.

H. B. Newcombe and J. M. Kennedy. Record linkage: Making maximum use of the discriminating power of identifying information. *Communications of the ACM*, 5(11):563–566, 1962.

S. Nowozin, P. V. Gehler, J. Jancsary, and C. H. Lampert, editors. *Advanced Structured Prediction*. Neural Information Processing. MIT press, 2016.

J. D. Park. Using weighted MAX-SAT engines to solve MPE. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2002.

S. Perkins, K. Lacker, and J. Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research (JMLR)*, 3: 1333–1356, 2003.

A. Pfeffer. IBAL: A probabilistic rational programming language. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2001.

A. Pfeffer. Figaro: An object-oriented probabilistic programming language. Technical report, Charles River Analytics, 2009.

A. Pfeffer, D. Koller, B. Milch, and K. T. Takusagawa. SPOOK: A system for probabilistic object-oriented knowledge representation. In *Uncertainty in Artificial Intelligence (UAI)*, 1999.

H. Poon and P. Domingos. Sum-product networks: A new deep architecture. In *Uncertainty in Artificial Intelligence (UAI)*, 2011.

J. Pujara, H. Miao, L. Getoor, and W. Cohen. Knowledge graph identification. In *International Semantic Web Conference (ISWC)*, 2013.

A. Ramesh, D. Goldwasser, B. Huang, H. Daumé III, and L. Getoor. Learning latent engagement patterns of students in online courses. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

A. Ramesh, S. Kumar, J. Foulds, and L. Getoor. Weakly supervised models of aspect-sentiment for online course discussion forums. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

P. Ravikumar and J. Lafferty. Quadratic programming relaxations for metric labeling and Markov random field MAP estimation. In *International Conference on Machine Learning (ICML)*, 2006.

P. Ravikumar, A. Agarwal, and M. J. Wainwright. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *Journal of Machine Learning Research (JMLR)*, 11:1043–1080, 2010a.

P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010b.

B. L. Richards and R. J. Mooney. Learning relations by pathfinding. In *AAAI Conference on Artificial Intelligence (AAAI)*, 1992.

M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In D. Fensel, K. Sycara, and J. Mylopoulos, editors, *The Semantic Web - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 351–368. Springer Berlin / Heidelberg, 2003.

F. Riguzzi and T. Swift. The PITA system: Tabling and answer subsumption for reasoning under uncertainty. In *International Conference on Logic Programming (ICLP)*, 2011.

S. Ross and J. A. Bagnell. Reinforcement and Imitation Learning via Interactive No-Regret Learning, 2014.

S. Ross, G. J. Gordon, and J. A. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Artificial Intelligence & Statistics (AISTATS)*, 2011.

R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *Artificial Intelligence & Statistics (AISTATS)*, 2009.

R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *International Conference on Machine Learning (ICML)*, 2008.

M. Samadi, P. Talukdar, M. Veloso, and M. Blum. ClaimEval: Integrated and flexible framework for claim evaluation using credibility of sources. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag, 2003.

A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Globally convergent dual MAP LP relaxation solvers using Fenchel-Young margins. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

S. E. Shimony. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68(2): 399–410, 1994.

P. Singla and P. Domingos. Discriminative training of Markov logic networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2005.

P. Singla and P. Domingos. Lifted first-order belief propagation. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2008.

D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. In *Uncertainty in Aritificial Intelligence (UAI)*, 2008.

D. Sontag, A. Globerson, and T. Jaakkola. Introduction to dual decomposition for inference. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*, pages 219–254. MIT Press, 2011.

D. Sontag, D. K. Choe, and Y. Li. Efficiently searching for frustrated cycles in MAP inference. In *Uncertainty in Aritificial Intelligence (UAI)*, 2012.

D. Sridhar, J. Foulds, M. Walker, B. Huang, and L. Getoor. Joint models of disagreement and stance in online debate. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

D. Sridhar, S. Fakhraei, and L. Getoor. A probabilistic approach for collective similarity-based drug-drug interaction prediction. *Bioinformatics*, 32(20):3175–3182, 2016.

B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Neural Information Processing Systems (NIPS)*, 2004.

B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: A large margin approach. In *International Conference on Machine Learning (ICML)*, 2005.

D. Tran, A. Kucukelbir, A. B. Dieng, M. Rudolph, D. Liang, and D. M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research (JMLR)*, 6:1453–1484, 2005.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 2000.

D. Venugopal and V. Gogate. On lifting the Gibbs sampling algorithm. In *Neural Information Processing Systems (NIPS)*, 2012.

B. W. Wah and Y. Shang. Discrete Lagrangian-based search for solving MAX-SAT problems. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1997.

M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers, 2008.

J. Wang and P. Domingos. Hybrid Markov logic networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2008.

T. Werner. A linear programming approach to max-sum problem: A review. *Pattern Analysis and Machine Intelligence, IEEE Trans. on*, 29(7):1165–1179, 2007.

R. West, H. S. Paskov, J. Leskovec, and C. Potts. Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics (TACL)*, 2:297–310, 2014.

F. Wood, J. W. van de Meent, and V. Mansinghka. A new approach to probabilistic programming inference. In *Artificial Intelligence & Statistics (AISTATS)*, 2014.

M. Wright. The interior-point revolution in optimization: History, recent developments, and lasting consequences. *Bulletin of the American Mathematical Society*, 42(1):39–56, 2005.

L. Xiong, X. Chen, T. Huang, J. Schneider, and J. Carbonell. Temporal collaborative filtering with Bayesian probabilistic tensor factorization. In *SIAM International Conference on Data Mining*, 2010.

C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation – An empirical study. *Journal of Machine Learning Research (JMLR)*, 7:1887–1907, 2006.

J. Zhu, N. Lao, and E. P. Xing. Grafting-Light: Fast, Incremental Feature Selection and Structure Learning of Markov Random Fields. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2010.

Journal of Machine Learning Research 18 (2017) 1-49 Submitted 7/16; Revised 9/17; Published 11/17

Clustering with Hidden Markov Model on Variable Blocks

Lin Lin LLIN@PSU.EDU

Jia Li JIALI@PSU.EDU

Department of Statistics
Pennsylvanian State University
University Park, PA 16802, USA

Editor: Saharon Rosset

Abstract

Large-scale data containing multiple important rare clusters, even at moderately high dimensions, pose challenges for existing clustering methods. To address this issue, we propose a new mixture model called Hidden Markov Model on Variable Blocks (HMM-VB) and a new mode search algorithm called Modal Baum-Welch (MBW) for mode-association clustering. HMM-VB leverages prior information about chain-like dependence among groups of variables to achieve the effect of dimension reduction. In case such a dependence structure is unknown or assumed merely for the sake of parsimonious modeling, we develop a recursive search algorithm based on BIC to optimize the formation of ordered variable blocks. The MBW algorithm ensures the feasibility of clustering via mode association, achieving linear complexity in terms of the number of variable blocks despite the exponentially growing number of possible state sequences in HMM-VB. In addition, we provide theoretical investigations about the identifiability of HMM-VB as well as the consistency of our approach to search for the block partition of variables in a special case. Experiments on simulated and real data show that our proposed method outperforms other widely used methods.

Keywords: Gaussian mixture model, hidden Markov model, modal Baum-Welch algorithm, modal clustering

1. Introduction

Clustering is one of the most important topics in unsupervised learning, the goal of which is to discover structures from a collection of unlabeled data. Finite mixture modeling is a major statistical framework for clustering. Without attempting to review its expansive applications, we offer instead a few examples as a glimpse at the incredibly broad usage of mixture modeling (Escobar and West (1995); Allenby et al. (1998); McLachlan et al. (2002); Kasahara and Shimotsu (2009)).

One important advantage of the mixture model is that the goodness of fit to any data can be improved by increasing the number of mixture components. The simplest approach to clustering based on a mixture model is to assign each component to an individual cluster. However, there are several drawbacks to equate clusters with mixture components, among which is that the parametric distribution of a component is too restrictive for the potentially diverse shapes of clusters (see Li et al. (2007) for a thorough discussion). Various strategies have been proposed to merge multiple mixture components so that an individual cluster can be more properly modeled (Hennig, 2010; Li, 2005; Pyne et al., 2009; Finak et al., 2009; Chan et al., 2010; Aghaeepour et al., 2011; Lin et al., 2016; Melnykov, 2016). In the statistical learning literature, a prominent method for merging multiple mixture components into one cluster is based on the modes of the mixture density, the so-called modal clustering by Li et al. (2007). We discuss this method in more detail in Section 2.

Despite their wide applications, existing mixture modeling approaches are severely challenged by high dimensional data encountered in certain research areas, for example, cell subset identification using data generated by the high-throughput single-cell technologies (Perfetto et al., 2004; Bandura et al., 2009; Maecker et al., 2012; Chattopadhyay et al., 2014; Spitzer and Nolan, 2016). These data sets contain a large number of highly unbalanced clusters. Furthermore, the most interesting clusters for scientific investigation are often of remarkably low occurrence. Even when the data dimension is not impressively large by today's standard, say in the order of tens, existing methods have much difficulty for detecting clusters of very low probabilities. Low probability mixture components tend to be "concealed" by large background clusters in the data. For the usual mixture modeling approach, in order to capture the rare clusters, we must increase the number of components dramatically. On the other hand, the curse of dimensionality prevents the use of many components; the growing computational intensity is also a concern.

Our new method is motivated by the popular manual gating analysis of single-cell cytometry data (details in Section 5.2), in which the variables are divided into groups based on prior information and examined sequentially. The key idea here is to exploit the chain-like dependence among groups of variables in the construction of a mixture model. Lin et al. (2013) used this idea to build a relatively primitive model. In this approach, the variables are partitioned into two groups, and the mixture model is estimated using the hierarchical Dirichlet process prior. This two-block model is substantially more efficient and accurate in rare clusters identification than the conventional mixture models are. The existing method, however, is unable to move beyond two variable blocks in practice due to the exponential computational complexity. The Markov chain Monte Carlo (MCMC) simulation for the two-block model is already highly intensive and is coupled by the long-standing issue of label switching when MCMC is applied to estimate mixture models (e.g., Richardson and Green (1997); Celeux et al. (2000); Stephens (2000)).

The aim of this paper is to design an effective and computationally accessible statistical model that can fit data robustly in both high and low probability regions and can identify clusters of non-Gaussian shapes. We propose a new model to exploit sequential dependence among variable groups. We also develop an algorithm to search for such a dependence structure when it is unknown. Our experiments show that even if the sequential dependence is not backed up by domain knowledge, it can still be useful as a mathematical mechanism for parsimonious modeling.

Our major contributions include: (1) We develop a *hidden Markov model on variable blocks* (*HMM-VB*) to leverage the sequential dependence structure among variable groups. To the best of our knowledge, this work is the first to exploit sequential dependence among variable groups by HMM. (2) We derive the Baum-Welch estimation algorithm for HMM-VB and the new *Modal Baum-Welch (MBW)* algorithm for finding modes of a HMM-VB, based on which clusters are formed. MBW achieves linear computational complexity without compromising optimality. (3) We develop a search algorithm to determine the grouping and ordering of the variables for a HMM-VB when variable groups are not pre-specified. (4) We

©2017 Lin Lin and Jia Li.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-342.html.

develop theorems on the identifiability of HMM-VB given the variable block structure and the identifiability of the variable blocks under certain conditions. We prove the consistency of the BIC criterion for finding the variable blocks in a special case.

The rest of the paper is organized as follows. In Section 2, we introduce notations and overview existing techniques most relevant to our proposed methods. In Section 3, we present HMM-VB and efficient algorithms for model fitting and modal clustering. In Section 4, we provide theoretical results on identifiability, consistency, and the mode search algorithm for HMM-VB. Proofs of the theorems appear in Appendix B∼D. In Section 5, experimental results are reported for both simulated and real data including mass cytometry, single-cell genomics, and image data. Comparisons are made with some competing models and popular methods. We conclude with discussions in Section 6.

2. Preliminaries

Given a random vector $X = (X_1, X_2, ..., X_d)' \in \mathcal{R}^d$, let $\mathbf{x}_i = (x_{i1}, ..., x_{id})' \in \mathcal{R}^d$ be the i-th sample of X, where $i = 1, ..., n$. Denote by $\mathbb{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n) \in \mathcal{R}^{n\times d}$ the data matrix, and we let $\mathbb{X}_j$ be the j-th column of $\mathbb{X}$ which contains the values of X_j across all the sample points. For ease of notation, we let $\mathbf{x} \in \mathcal{R}^d$ be a realization of the random vector X.

The finite Gaussian mixture model (GMM) is commonly used for clustering. A GMM with M components has the density function: $f(\mathbf{x}|\theta) = \sum_{k=1}^{M} \pi_k \phi(\mathbf{x}|\theta_k)$, where π_k is the mixture component prior probability and $\phi(\cdot \mid \theta_k)$ is the multivariate normal density parameterized by $\theta_k = (\mu_k, \Sigma_k)$, μ_k being the d-dimensional mean vector and Σ_k the $d \times d$ covariance matrix. For model estimation, a latent indicator $Z \in \{1, 2, ..., M\}$ with $P(Z = k) = \pi_k$ is used. Specifically, conditioning on $Z = k$, X follows the k-th component distribution. Z is also called the component identity of X. To perform clustering, the usual approach is to compute the posterior probability $P(Z = k|X = \mathbf{x})$ and assign $\mathbf{x}$ to the cluster with the maximum posterior. However, this approach is inadequate to model clusters with arbitrary shapes and cannot ensure that the clusters are reasonably separated. One major idea explored in the literature is to merge multiple mixture components for a better and more flexible representation of an individual cluster. We refer to Melnykov and Maitra (2010) for a thorough review on clustering based on finite mixture models.

Banfield and Raftery (1993) and Celeux and Govaert (1995) propose to decompose the covariance matrix of the mixture components in a GMM into parts that control volume, shape, and orientation respectively and allow model constraints to be imposed in these aspects either individually or by combination. A model selection criterion is then used to choose a GMM in terms of not only the number of components but also the constraints on covariance matrices. A popular R package, namely `Mclust` (Fraley and Raftery, 2006), is implemented for this method, which we will use for comparison.

The Modal EM (MEM) algorithm developed by Li et al. (2007) performs efficient merging of mixture components. It resembles the expectation-maximization (EM) algorithm (Dempster et al., 1977), as reflected by the name "modal EM". However, the objective of MEM is to find an increasing path from any data point to a local maximum of a given density. Hence, the optimization objective of MEM is to find a local maximizer over $\mathbf{x}$ for $f(\mathbf{x}|\theta)$ under a given θ, while EM is to find local maxima over θ for $f(\mathbf{x}|\theta)$ given $\mathbf{x}$. Consider a general mixture density $f(\mathbf{x}) = \sum_{k=1}^{M} \pi_k f_k(\mathbf{x})$, where $f_k(\mathbf{x})$ is the density of the k-th mixture component.

Starting from any initial value denoted by $\mathbf{x}^{[0]}$, MEM solves a local maximum of the mixture density by the following two iterative steps: (1) At iteration r, let $p_k = \frac{\pi_k f_k(\mathbf{x}^{[r]})}{f(\mathbf{x}^{[r]})}$, $k = 1$, ..., M; (2) Update $\mathbf{x}^{[r+1]} = \text{argmax}_{\mathbf{x}} \sum_{k=1}^{M} p_k \log f_k(\mathbf{x})$. MEM stops when a pre-specified stopping criterion is met. Specifically for GMM with $f(\mathbf{x}) = \sum_{k=1}^{M} \pi_k \phi(\mathbf{x}|\mu_k, \Sigma_k)$, MEM becomes

1. E-step: Solve

$$p_k = \frac{\pi_k \phi_k(\mathbf{x}^{[r]} \mid \mu_k, \Sigma_k)}{f(\mathbf{x}^{[r]})}, \quad k = 1, \ ..., \ M. \tag{1}$$

2. M-step: Solve

$$\mathbf{x}^{[r+1]} = \left(\sum_{k=1}^{M} p_k \cdot \Sigma_k^{-1}\right)^{-1} \cdot \left(\sum_{k=1}^{M} p_k \cdot \Sigma_k^{-1} \mu_k\right). \tag{2}$$

The computational efficiency of MEM enabled the development of a new clustering approach by Li et al. (2007), referred to as *modal clustering*. In Li et al. (2007), a non-parametric Gaussian kernel density estimate is used, and MEM is applied to find the mode associated with every point. Data points associated with the same mode are assigned to the same cluster. In Lee and Li (2012), modal clustering based on the general finite GMM is studied. For computational efficiency, instead of applying MEM to every data point, it is applied to the means of the mixture components. Components with mean vectors associated with the same mode are merged into one cluster. Whether a point-wise mode association or a component-wise mode association is preferred depends on the nature of the application and the computational resources. In practice, the difference in the clustering results we have observed is quite small. We refer to the clustering method based on component-wise mode association as *modal GMM* and use it as a baseline for comparison with our new method.

Under the framework of modal clustering, the purpose of a mixture component is primarily for good density estimation. We no longer rely on a one-to-one correspondence between mixture components and clusters. Mode association also ensures that different clusters of data are well separated. See Li et al. (2007) for more detailed discussion on these advantages. The flexibility provided by modal clustering for fitting data is precisely what we need for the applications we consider. In the next section, we propose a HMM-type model which can be cast as a mixture model with an enormous number of components, even exceeding the data size. This complexity causes no difficulty in clustering via mode association.

3. Hidden Markov Model on Variable Blocks

As discussed in Section 1, in some specific application domains, a sequential dependence structure among groups of variables is available. This dependence prompts us to model a subset of variables conditioning on some other subset of variables. If we view the sequential ordering of the variable blocks as a "timeline", it seems natural to employ a HMM-type model, where each variable block follows a mixture distribution and the statistical dependence among the blocks is captured by the Markov process of the underlying states (that is, the

mixture component identities). The description of a conventional HMM is provided in Appendix A. We call our new model *Hidden Markov Model on Variable Blocks (HMM-VB)*.

Suppose the d-dimensional random vector X is partitioned into T blocks indexed by $t = 1, 2, ..., T$. Let the number of variables in block t be d_t, where $\sum_{t=1}^{T} d_t = d$. Assume that the d_1 variables in block 1 have indices before the d_2 variables in block 2, and so on. In general, obviously, such an ordering of variables may not hold. But this is only a matter of naming the variables and has no effect on our results. Let $X^{(t)}$ denote the t-th variable block. Without loss of generality, let $X^{(1)} = (X_1, X_2, ..., X_{d_1})'$ and $X^{(t)} = (X_{m_t+1}, X_{m_t+2}, ..., X_{m_t+d_t})'$, where $m_t = \sum_{\tau=1}^{t-1} d_\tau$, for $t = 2, ..., T$.

Denote the underlying state of $X^{(t)}$ as s_t, $t = 1, ..., T$. Let the index set of s_t be $\mathcal{S}_t = \{1, 2, ..., M_t\}$, where M_t is the number of mixture components for variable block $X^{(t)}$, $t = 1, ..., T$. Let the set of all possible sequences be $\hat{\mathcal{S}} = \mathcal{S}_1 \times \mathcal{S}_2 \cdots \times \mathcal{S}_T$. $|\hat{\mathcal{S}}| = \prod_{t=1}^{T} M_t$. We assume:

1. $\{s_1, s_2, ..., s_T\}$ follow a Markov chain. Let $\pi_k = P(s_1 = k)$, $k \in \mathcal{S}_1$. Let the transition probability matrix $A_t = (a_{k,l}^{(t)})$ between s_t and s_{t+1} be defined by $a_{k,l}^{(t)} = P(s_{t+1} = l | s_t = k)$, $k \in \mathcal{S}_t$, $l \in \mathcal{S}_{t+1}$.

2. Given s_t, $X^{(t)}$ is conditionally independent from other $s_{t'}$ and $X^{(t')}$, for all $t' \neq t$. We also assume that given $s_t = k$, the conditional density of $X^{(t)}$ is the Gaussian distribution $\phi(X^{(t)} \mid \mu_k^{(t)}, \Sigma_k^{(t)})$.

Let $\mathbf{s} = \{s_1, ..., s_T\}$. A realization of X is $\mathbf{x}$, and a realization of $X^{(t)}$ is $x^{(t)}$. To summarize, the density of HMM-VB is given by

$$f(\mathbf{x}) = \sum_{\mathbf{s} \in \hat{\mathcal{S}}} \left(\pi_{s_1} \prod_{t=1}^{T-1} a_{s_t, s_{t+1}}^{(t)} \right) \cdot \prod_{t=1}^{T} \phi(x^{(t)} | \mu_{s_t}^{(t)}, \Sigma_{s_t}^{(t)}). \tag{3}$$

Remark 1: Figure 1 illustrates two major differences between HMM-VB and the conventional HMM. (1) The variable blocks $X^{(t)}$'s are not from the same vector space. Hence, the parameters of the distribution of $X^{(t)}$ given $s_t = k$ depend not only on k but also on t; (2) The underlying Markov chain for $\{s_1, ..., s_T\}$ is not time invariant. In fact, the state space $\mathcal{S}_t$ varies with t.

Remark 2: Although the density function of HMM-VB in Eq. (3) indicates block diagonal covariance matrices, there are important differences from a typical GMM with the same constraint on the covariance. First, the Gaussian mean vectors in Eq. (3) reside on a lattice in the Cartesian product space $\mathcal{R}^{d_1} \times \mathcal{R}^{d_2} \cdots \times \mathcal{R}^{d_T}$. Secondly, the number of components grows exponentially with T. In fact, it is often larger than the sample size. The enormous number of components cannot be handled by a typical covariance constrained GMM from either estimation or computational feasibility perspectives.

Remark 3: Since HMMs can be represented as singly connected directed acyclic graphs, HMM-VB is a special case of graphical models (e.g., Bishop (2006); Koller and Friedman (2009)). We named our model HMM-VB because we derived the estimation method based on the popular Baum-Welch algorithm for HMM. Moreover, HMM itself is a widely known model in machine learning as well as statistics.

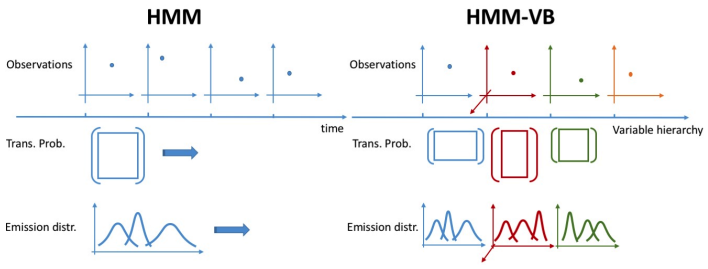


Figure 1: Comparison between HMM-VB and HMM models. The observations under HMM have to be of a fixed dimension, 2 in this illustration. Typically, only one transition probability matrix is applied through time, and the set of distributions conditioned on the states at any time spot is also fixed. HMM-VB models data across blocks of different variables, possibly of different dimensions. Both the transition probability matrix and the set of conditional distributions are defined individually at every "time" spot.

3.1 Maximum Likelihood Estimation

HMM is usually estimated by the EM algorithm. However, because the cardinality of $\bar{\mathcal{S}}$ grows exponentially with the sequence length, the computational complexity of a direct application of EM is of exponential complexity. This technical hurdle was overcome by the *Baum-Welch (BW) algorithm* which achieves complexity of linear order in the sequence length and quadratic in the number of states without compromising optimality. The BW algorithm, a special instance of EM, was developed in the 1960's before the general EM algorithm was developed in the 1970's. As a result, we still call the estimation algorithm Baum-Welch, following the convention of the literature on HMM. We present the BW algorithm for HMM model estimation in Appendix A. For a detailed exposure to HMM, we refer to Young et al. (1997).

We now present the corresponding BW algorithm for HMM-VB. It can be proved that the BW algorithm for HMM is an exact EM algorithm. The derivation of the BW algorithm for HMM-VB is similar to the derivation of BW for the usual HMM. We thus omit it here. Clearly, it is not meaningful to estimate HMM-VB using a single sequence, which is essentially a single data point in this case. HMM-VB is after all a model for $X \in \mathcal{R}^d$.

Denote by $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, ..., x_i^{(T)})'$ the ordered and grouped i-th sample according to the given variable blocks. We denote the original i-th sample before ordering by $\breve{\mathbf{x}}_i$. Consider estimation of HMM-VB based on a data set $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$. We define $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$ similarly as in Eq. (8) and Eq. (9) in Appendix A. For $i = 1, ..., n$,

$$L_k(\mathbf{x}_i, t) = P(s_{i,t} = k \mid \mathbf{x}_i), \ k \in \mathcal{S}_t, \tag{4}$$

$$H_{k,l}(\mathbf{x}_i, t) = P(s_{i,t} = k, s_{i,t+1} = l \mid \mathbf{x}_i), \ k \in \mathcal{S}_t, l \in \mathcal{S}_{t+1}. \tag{5}$$

The BW algorithm iterates the following two steps:

1. E-step: Under the current set of parameters, compute $L_k(\mathbf{x}_i, t)$, $i = 1, ..., n$, $k \in \mathcal{S}_t$, $t = 1, ..., T$, and $H_{k,l}(\mathbf{x}_i, t)$, $i = 1, ..., n$, $k \in \mathcal{S}_t$, $l \in \mathcal{S}_{t+1}$, $t = 1, ..., T-1$.

2. M-step: Update parameters by

$$\mu_k^{(t)} = \frac{\sum_{i=1}^n L_k(\mathbf{x}_i, t) x_i^{(t)}}{\sum_{i=1}^n L_k(\mathbf{x}_i, t)}, \quad k \in \mathcal{S}_t,\ t = 1, ..., T,$$

$$\Sigma_k^{(t)} = \frac{\sum_{i=1}^n L_k(\mathbf{x}_i, t) \left(x_i^{(t)} - \mu_k^{(t)}\right) \left(x_i^{(t)} - \mu_k^{(t)}\right)'}{\sum_{i=1}^n L_k(\mathbf{x}_i, t)}, \quad k \in \mathcal{S}_t,\ t = 1, ..., T,$$

$$a_{k,l}^{(t)} = \frac{\sum_{i=1}^n H_{k,l}(\mathbf{x}_i, t)}{\sum_{i=1}^n L_k(\mathbf{x}_i, t)}, \quad k \in \mathcal{S}_t,\ l \in \mathcal{S}_{t+1},\ t = 1, ..., T-1,$$

$$\pi_k \propto \sum_{i=1}^n L_k(\mathbf{x}_i, 1),\ k \in \mathcal{S}_1,\ s.t. \sum_{k \in \mathcal{S}_1} \pi_k = 1.$$

The above equations can be easily extended to the case of weighted sample points. It can occur in practice that each sample point is assigned with a weight. For instance, quantization is often used to reduce the data size significantly. Instead of using the original data, one may use the quantized points, each of which can represent a different number of original points and hence is assigned with a weight proportional to that number. Suppose weight w_i is assigned to sample $\mathbf{x}_i$. The E-step is not affected. In the M-step, we can simply multiply w_i in front of each summand appeared in the equations above.

The forward-backward algorithm for computing $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$ is essentially the same as the forward-backward algorithm for the usual HMM. The fact that the variable blocks are not from the same vector space and the state spaces vary with t does not cause any intrinsic difference. Suppress the sample index i and consider one sample point $\mathbf{x}$.

Define the forward probability $\alpha_k(\mathbf{x}, t)$ as the joint probability of observing the first t variable blocks $x^{(\tau)}$, $\tau = 1, ..., t$, and being in state k at time t:

$$\alpha_k(\mathbf{x}, t) = P(x^{(1)}, x^{(2)}, ..., x^{(t)}, s_t = k), \quad k \in \mathcal{S}_t.$$

This probability can be evaluated by the following recursive formula:

$$\begin{aligned} \alpha_k(\mathbf{x}, 1) &= \pi_k \phi(x^{(1)} \mid \mu_k^{(1)}, \Sigma_k^{(1)}), \quad k \in \mathcal{S}_1, \\ \alpha_k(\mathbf{x}, t) &= \phi(x^{(t)} \mid \mu_k^{(t)}, \Sigma_k^{(t)}) \sum_{l \in \mathcal{S}_{t-1}} \alpha_l(\mathbf{x}, t-1) a_{l,k}^{(t-1)}, \quad 1 < t \le T,\ k \in \mathcal{S}_t. \end{aligned}$$

Define the backward probability $\beta_k(\mathbf{x}, t)$ as the conditional probability of observing the variable blocks after time t, $x^{(\tau)}$, $\tau = t+1, ..., T$, given the state at block t is k:

$$\begin{aligned} \beta_k(\mathbf{x}, t) &= P(x^{(t+1)}, ..., x^{(T)} \mid s_t = k), \quad 1 \le t \le T-1,\ k \in \mathcal{S}_t, \\ &\quad \text{Set } \beta_k(\mathbf{x}, T) = 1, \quad \text{for all } k \in \mathcal{S}_T. \end{aligned}$$

The backward probability can be evaluated using the following recursion:

$$\begin{aligned} \beta_k(\mathbf{x}, T) &= 1, \quad k \in \mathcal{S}_T, \\ \beta_k(\mathbf{x}, t) &= \sum_{l \in \mathcal{S}_{t+1}} a_{k,l}^{(t)} \phi(x^{(t+1)} \mid \mu_l^{(t+1)}, \Sigma_l^{(t+1)}) \beta_l(\mathbf{x}, t+1), \quad 1 \le t < T, \quad k \in \mathcal{S}_t. \end{aligned}$$

The probabilities $L_k(\mathbf{x}, t)$ and $H_{k,l}(\mathbf{x}, t)$ are solved by

$$L_k(\mathbf{x}, t) = P(s_t = k \mid \mathbf{x}) = \frac{P(\mathbf{x}, s_t = k)}{P(\mathbf{x})} = \frac{\alpha_k(\mathbf{x}, t) \beta_k(\mathbf{x}, t)}{P(\mathbf{x})}, \quad k \in \mathcal{S}_t,$$

$$\begin{aligned} H_{k,l}(\mathbf{x}, t) &= P(s_t = k, s_{t+1} = l \mid \mathbf{x}) = \frac{P(\mathbf{x}, s_t = k, s_{t+1} = l)}{P(\mathbf{x})} \\ &= \frac{1}{P(\mathbf{x})} \alpha_k(\mathbf{x}, t) a_{k,l}^{(t)} \phi(x^{(t+1)} \mid \mu_l^{(t+1)}, \Sigma_l^{(t+1)}) \beta_l(\mathbf{x}, t+1), \quad k \in \mathcal{S}_t,\ l \in \mathcal{S}_{t+1}. \end{aligned}$$

The normalizing factor $P(\mathbf{x}) = \sum_{k \in \mathcal{S}_t} \alpha_k(\mathbf{x}, t) \beta_k(\mathbf{x}, t)$ holds for any t.

To initialize the model, we design several schemes. In our experiments, models from different initializations are estimated and the one with the maximum likelihood is chosen. In our baseline initialization scheme, k-means clustering is applied individually to each variable block using all the data instances. Based on the clustering result of k-means, we take every cluster as one mixture component and compute the sample mean and sample covariance matrix of data in that cluster. To reduce the sensitivity to the initial clustering result, we also compute the pooled common sample covariance matrix for the clusters. The initial covariance matrix of a component is then set as a convex combination of the cluster-specific sample covariance and the common sample covariance. The transition probabilities are always initialized to be uniform. Under the second initialization scheme, we randomly sample a subset from the whole data and apply the baseline initialization to the subset. Under the third initialization scheme, we randomly pick a subset from the data and treat points in this subset as the cluster centroids of the k-means. These centroids will induce a cluster partition of the whole data, based on which we initialize the component means and covariance matrices in the same way as the baseline method. Both the second and the third initialization schemes are repeated several times with different random starts.

3.2 Modal Baum-Welch Algorithm

HMM-VB can be viewed as a special case of a GMM where each component of the GMM corresponds to a particular sequence of states $\mathbf{s} = \{s_1, ..., s_T\}$, that is, a combination of states for all the variable blocks. We call this equivalent GMM the *GMM mapped from HMM-VB*. Each component is a Gaussian distribution with mean $\mu_{\mathbf{s}} = (\mu_{s_1}^{(1)}, \mu_{s_2}^{(2)}, ..., \mu_{s_T}^{(T)})$ (column-wise stack of vectors) and a covariance matrix, denoted by $\Sigma_{\mathbf{s}}$, that is block diagonal. The t-th diagonal block in $\Sigma_{\mathbf{s}}$ is $\Sigma_{s_t}^{(t)}$ with dimension $d_t \times d_t$. We can thus readily apply the modal clustering framework for GMM to data modeled by HMM-VB. However, the number of components in the mapped GMM is $M = \prod_{t=1}^T M_t$, which grows exponentially with T assuming similar M_t's. A direct application of the MEM algorithm (see Eq. (1) and (2)) is computationally infeasible. We discover that because of the block diagonal structure of the

covariance matrix of the GMM mapped from HMM-VB, we can in fact avoid computing the posterior of $\mathbf{x}$ belonging to each component (exponentially many of them!). Instead, we only need $L_k(\mathbf{x}, t)$ for all k and t when updating $\mathbf{x}$ in the M-step of MEM. Because the BW algorithm solves $L_k(\mathbf{x}, t)$ at a complexity linear in T, we can achieve linear complexity for solving the modes of HMM-VB as well. We call this new algorithm *Modal Baum-Welch (MBW)* Algorithm.

We denote by $x^{(t),r}$ the value of the t-th variable block at iteration r, and let $\mathbf{x}^{[r]} = (x^{(1),r}, x^{(2),r}, ..., x^{(T),r})$ be the concatenated and properly ordered full vector at iteration r. The equivalence of MBW and the Modal EM algorithm is ensured by Theorem 7 in Section 4.3, which is proved in Appendix D.

The MBW algorithm iterates the following two steps:

1. E-step: Compute $L_k(\mathbf{x}^{[r]}, t)$, for $k \in \mathcal{S}_t$, $t = 1, ..., T$.
2. M-step: For $t = 1, ..., T$,

$$x^{(t),r+1} = \left(\sum_{k \in \mathcal{S}_t} L_k(\mathbf{x}^{[r]}, t) \cdot \left(\Sigma_k^{(t)} \right)^{-1} \right)^{-1} \left(\sum_{k \in \mathcal{S}_t} L_k(\mathbf{x}^{[r]}, t) \cdot \left(\Sigma_k^{(t)} \right)^{-1} \cdot \mu_k^{(t)} \right).$$

The clustering method based on MBW is straightforward. We first find the state sequence $\mathbf{s}_i^{(*)}$ with maximum posterior given $\mathbf{x}_i$ by the Viterbi algorithm (Young et al., 1997):

$$\mathbf{s}_i^* = \arg\max_{\mathbf{s} \in \hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}_i),\ i = 1, ..., n.$$

Since different $\mathbf{x}_i$'s may yield the same sequence, we then identify the collection of distinct $\mathbf{s}_i^*$. For each distinct sequence, say $\mathbf{s}^*$, find $\mu_{\mathbf{s}^*} = (\mu_{s_1^*}^{(1)}, \mu_{s_2^*}^{(2)}, ..., \mu_{s_T^*}^{(T)})$. Use $\mu_{\mathbf{s}^*}$ as an initialization for MBW to find the mode associated with it. If $\mu_{\mathbf{s}_i^*}$ and $\mu_{\mathbf{s}_j^*}$ are brought to the same mode by MBW, the corresponding data vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ are put into the same cluster. When $|\hat{\mathcal{S}}|$ is very large, the number of different $\mathbf{s}_i^*$'s can become close to the data size. Hence, the amount of computation we can save by seeking modes starting from $\mu_{\mathbf{s}_i^*}$'s instead of the original data diminishes. As a result, in such cases, we recommend seeking modes directly from the original data.

3.3 Computational Complexity

For both the BW estimator of HMM-VB and the MBW algorithm, the vast majority of the computation is on obtaining $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$, $i = 1, ..., n$, $t = 1, ..., T$, $k \in \mathcal{S}_t$, $l \in \mathcal{S}_{t+1}$. For clarity of discussion, suppose the number of states in each block, $|\mathcal{S}_t|$, is a constant. Then the complexity for computing the two quantities is $O(nT|\mathcal{S}_t|^2)$.

The two quantities $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$ can be computed separately for each sample $\mathbf{x}_i$. Thus the E-step in both the BW and the MBW algorithms are easily parallelizable by simply dividing the data among multiple processors. For the MBW algorithm, since we search for the mode separately starting from each sample point, the M-step is also naturally parallelizable. For the BW estimation algorithm, in the M-step, some quantities need to be

transmitted to a central processor to update the parameters, which will then be broadcast to the distributed processors. However, an inspection of the M-step shows that we do not need to communicate $L_k(\mathbf{x}_i, t)$ and $H_{k,l}(\mathbf{x}_i, t)$ for every sample point $\mathbf{x}_i$ to the central processor. Suppose the first processor treats a data segment containing points 1, 2, ..., n_1. It only needs to send $\sum_{i=1}^{n_1} L_k(\mathbf{x}_i, t)$, $\sum_{i=1}^{n_1} L_k(\mathbf{x}_i, t) x_i^{(t)}$, $\sum_{i=1}^{n_1} L_k(\mathbf{x}_i, t) x_i^{(t)} x_i^{(t)'}$, $\sum_{i=1}^{n_1} H_{k,l}(\mathbf{x}_i, t)$, $k \in \mathcal{S}_t$, $l \in \mathcal{S}_{t+1}$, $t = 1, ..., T$, to the central processor. Thus the communication load from a processor treating one data segment to the central processor does not depend on the data size and is nearly as low as the number of parameters in the model (the negligible relative difference shrinks quickly when $|\mathcal{S}_t|$ grows). The central processor can then update the parameters precisely according to the formula in the M-step. Since the design of the parallel algorithm of BW is simple, we hereby skip the details for brevity.

3.4 Constructing Variable Dependence Structure

We refer to the grouping and ordering of the variable blocks, that is, the formation of $X^{(t)}$, $t = 1, ..., T$, as the dependence structure or variable block structure. We have so far assumed that the dependence structure of HMM-VB is given. To complete our new framework for clustering, we now address how to determine such a dependence structure when it is not pre-specified. We propose to seek a dependence structure that yields the HMM-VB with the minimum BIC for the data. We use the following definition of BIC:

$$BIC = -2\log(\hat{L}) + k\log(n),$$

where $\hat{L}$ is the maximum value of the likelihood function, n is the sample size, and k is the number of free parameters to be estimated. The challenge lies in the computational complexity of the combinatorial optimization problem. An exhaustive search is intractable even for moderate dimensions (the number of possible groupings and orderings is much larger than that of variable permutations). To achieve computational feasibility, we design a greedy local search scheme.

We first generate an ordering of the variables based on some prior knowledge or on a random permutation, which we call raw ordering hereafter. The raw ordering is not necessarily the final ordering of the variables after the grouping is decided, although the former indeed strongly influences the latter. We may generate several random raw orderings of the variables. Given any raw ordering, the variables are grouped by a step-wise optimization procedure. Based on the likelihoods of the corresponding HMM-VBs, the dependence structure will be chosen under each raw ordering. We then compare structures found from all the raw orderings and select the best from them.

Denote the raw ordering of the variables by Q, with $Q(j) \in \{1, ..., d\}$, for $j = 1, ..., d$. Note that $Q(j)$ is a bijection between $\{1, ..., d\}$ and itself. Denote by G the grouping for the original variable vector. For example, if $Q = (3, 2, 1)$, with $d = 3$, then the specified ordering of the input data for HMM-VB is $(X_{Q(1)}, X_{Q(2)}, X_{Q(3)}) = (X_3, X_2, X_1)$. Suppose that there are two variable blocks, where X_2 and X_3 form the first variable block and X_1 belongs to the second block. Then $G(Q(1)) = 1$, $G(Q(2)) = 1$ and $G(Q(3)) = 2$, or equivalently $G(1) = 2$, $G(2) = 1$, and $G(3) = 1$.

Next, we determine how the variables are grouped based on a step-wise selection process. The variables are treated one by one in the order given by Q. The algorithm ensures that

$G(Q(1))$, ..., $G(Q(j-1))$ have been determined when solving $G(Q(j))$. Specifically, suppose $G(Q(1)), ..., G(Q(j-1)) \in \{1, 2, ..., g\}$. That is, the first $j-1$ ordered variables have been put into g non-empty groups. Note that $g \leq j-1$. The possible value for $G(Q(j))$ is 1, ..., g, or $g+1$. If the $Q(j)$-th variable is put in any existing group, then $G(Q(j)) \leq g$; otherwise it forms by itself a new group with identity number $g+1$. In order to determine $G(Q(j))$, we compare exhaustively the structures with $G(Q(j)) = 1, 2, ..., g, g+1$, which means experimenting with putting the $Q(j)$-th variable in each existing group as well as forming a new group by the $Q(j)$-th variable alone. We use each structure to estimate a HMM-VB for the first j ordered variables: $X_{Q(1)}, X_{Q(2)}, ..., X_{Q(j)}$ (note that this is not the full dimensional data). These HMM-VBs are compared by BIC using the j-dimensional data, and the one with the optimal BIC is chosen, which in turn determines $G(Q(j))$. The process is repeated to sweep through all the variables until the full dimension $j = d$.

Note that we need to specify the number of components for each variable block. After extensive numerical experiments, we set $M_t = 10$ if $d_t \leq 5$, $M_t = 15$ if $d_t \in [6, 10]$, otherwise, $M_t = d_t + 10$ for t being any variable block. In addition, the greedy local search algorithm is quite robust to the change in the number of mixture components.

We denote the BIC of the estimated HMM-VB under raw ordering Q and grouping $G(Q(1))$, ..., $G(Q(j))$ for the partial data containing the first j ordered variables by $\mathcal{L}_{BIC}(\mathbb{X}_{Q(1),...,Q(j)}, G(Q(1)), ..., G(Q(j)), \theta)$, where θ denotes the parameters of HMM-VB. Recall that $\mathbb{X}_{Q(j)}$ denotes the $Q(j)$-th column of the data matrix $\mathbb{X}$. Our step-wise selection algorithm under a given raw ordering Q is as follows:

1. Input data matrix $\mathbb{X}$ and the ordering structure $\{Q(1), ..., Q(d)\}$.
2. Set $j = 1$, $g = 1$, $G(Q(1)) = 1$.
3. For $j = 2$, ..., d
   (a) For each $k = 1$, ..., g, $g+1$, obtain the maximum likelihood estimation of HMM-VB for partial data composed of $\mathbb{X}_{Q(1)}, \mathbb{X}_{Q(2)}, ..., \mathbb{X}_{Q(j)}$ under structure Q and $(G(Q(1)), ..., G(Q(j-1)), G(Q(j)) = k)$. Let the estimated parameter at k be $\theta^{*,k}$.
   (b) Compute
   $$G^*(Q(j)) = \underset{k \in \{1,...,g,g+1\}}{\operatorname{argmin}} \mathcal{L}_{BIC}(\mathbb{X}_{Q(1),...,Q(j)}, G(Q(1)), ..., G(Q(j)) = k, \theta^{*,k}).$$
   (c) Set $G(Q(j)) \longleftarrow G^*(Q(j))$.
   (d) If $G(Q(j)) = g+1$, set $g \longleftarrow g+1$.

Note that if the prior knowledge of the ordering structure is unknown, we randomly generate the raw ordering $\{Q(j)\}_{j=1}^d$ multiple times and adopt the one achieving the optimal BIC. We find that this approach performs well empirically as illustrated in the experiment section. It is also not difficult to see that this algorithm runs on the order of $O(d^2)$, which is efficient in practice.

4. Theoretical Properties

We study the identifiability of HMM-VB, prove the consistency of using BIC for model selection in a special case, and prove the optimality of MBW algorithm for HMM-VB.

4.1 Identifiability

Since HMM-VB can be viewed as a special case of a GMM, we need to ensure model identifiability, which is a necessary condition for estimating the parameters of a mixture model consistently. Specifically, we need to make sure that no two essentially different mixture parameters define the same distribution. We now introduce a type of GMM that includes HMM-VB as a special case and prove some results that help establish the identifiability of HMM-VB. A list of new definitions and notations is provided first.

The variable blocks $X^{(t)}$, $t = 1, ..., T$ are mutually exclusive and collectively exhaustive (that is, their union is the set of all the variables). For brevity, with a slight abuse of notation, using $X^{(t)}$ to mean both a subvector of X as well as a set of variables, but ensure that the specific meaning is clear from context. Denote a *variable partition* by $\mathcal{P} = \{X^{(1)}, X^{(2)}, ..., X^{(T)}\}$.

1. *Lattice GMM*: Denote the Gaussian parameter of a mixture component for variable block $X^{(t)}$ by $\theta_{i_t}^{(t)} = (\mu_{i_t}^{(t)}, \Sigma_{i_t}^{(t)})$, $i_t = 1, ..., M_t$. We define a *lattice GMM* on a variable partition $\mathcal{P}$ as a GMM that bears the form
$$f(\mathbf{x}) = \sum_{i_1=1}^{M_1} \sum_{i_2=1}^{M_2} \cdots \sum_{i_T=1}^{M_T} \pi(i_1, ..., i_T) \cdot \prod_{t=1}^{T} \phi(x^{(t)} \mid \theta_{i_t}^{(t)}). \tag{6}$$
   For a lattice GMM, a component is indexed by a T-tuple $(i_1, ..., i_T)$. Denote the number of components for each variable block collectively by $\mathbb{M} = \{M_1, ..., M_t\}$. We introduce latent state variables s_t for each block $X^{(t)}$, $s_t \in \{1, ..., M_t\}$, $t = 1, ..., T$. The joint pmf of s_t's is given by $\pi(i_1, ..., i_T)$. We use $\Pi(s_1, ..., s_T)$ to denote the joint pmf of $s_1, ..., s_T$ and $\Pi(s_{t_1}, s_{t_2}, ..., s_{t_k})$ to denote the marginal pmf of any subset of the latent states.
   $\Theta^{(t)} = \{\theta_{i_t}^{(t)}, i_t = 1, ..., M_t\}$ is the *grid* of parameters for variable block $X^{(t)}$. Clearly, a lattice-GMM is a mixture of components whose parameters are points from the Cartesian product of the grid of each variable block: $\Theta = \Theta^{(1)} \times \Theta^{(2)} \cdots \times \Theta^{(T)}$. We call Θ the *lattice* of parameters for the full dimensional vector X.
   We say $\theta_{i_t}^{(t)} \in \Theta^{(t)}$ *exists* if $P(s_t = i_t) > 0$, or equivalently there is at least one set of i_1, ..., i_{t-1}, i_{t+1}, ..., i_T such that $\pi(i_1, i_2, ..., i_T) > 0$. We say that the grid $\Theta^{(t)}$ is *distinct* if $\theta_{i_t}^{(t)} \neq \theta_{i'_t}^{(t)}$ for any $i_t \neq i'_t$. The grid $\Theta^{(t)}$ is *non-redundant* if it is distinct and every $\theta_{i_t}^{(t)} \in \Theta^{(t)}$, $i_t = 1, ..., M_t$, exists. If $\Theta^{(t)}$ is not non-redundant, then we can shrink the grid $\Theta^{(t)}$ by eliminating some θ_{i_t}'s and/or merging identical θ_{i_t}'s in the set. Lattice Θ is *non-redundant* if all $\Theta^{(t)}$, $t = 1, ..., T$ are non-redundant.
2. *Tight variable partition*: $\mathcal{P}$ is a *tight variable partition* (or simply tight partition) for a lattice GMM if by Eq. (6), the component prior $\pi(i_1, ..., i_T) > 0$ for all the T-tuples $(i_1, ..., i_T)$.

3. *Maximum variable partition*: A variable partition $\mathcal{P}_1$ is *nested* in partition $\mathcal{P}_2$, denoted by $\mathcal{P}_1 \succ \mathcal{P}_2$ or $\mathcal{P}_2 \prec \mathcal{P}_1$, if every variable block of $\mathcal{P}_1$ is a subset of a variable block of $\mathcal{P}_2$. That is, the partition $\mathcal{P}_1$ can be obtained from $\mathcal{P}_2$ by further dividing a variable block into smaller blocks. $\mathcal{P}$ is a *maximum variable partition* (or simply maximum partition) for a lattice GMM if $\mathcal{P}$ is a tight variable partition and there exists no other tight partition $\mathcal{P}'$ for the GMM such that $\mathcal{P}' \succ \mathcal{P}$.

We let $\mathcal{M}$ to denote the collection of parameters that specify a lattice GMM on the variable partition $\mathcal{P}$. $\mathcal{M} = \{\mathbb{M}, \Theta, \Pi(s_1, ..., s_T)\}$. We also use $\mathcal{M}_{X^{(t)}}$ to denote the marginal GMM for $X^{(t)}$ derived from $\mathcal{M}$. For instance, $\mathcal{M}_{X^{(t)}} = \{M_t, \Theta^{(t)}, \Pi(s_t)\}$. Similarly, we can have marginal GMM for multiple variable blocks, e.g., $\mathcal{M}_{X^{(t)}, X^{(t+1)}}$.

Suppose $\mathcal{O}$ is a permutation function from one index set to another and Γ is a set of indexed parameters. We denote the permuted parameters by $\mathcal{O}(\Gamma)$. For instance, if $\Gamma = \{\mu_1, \mu_2, \mu_3\}$ and $\mathcal{O}$ permutes $\{1, 2, 3\}$ to $\{3, 2, 1\}$, then $\mathcal{O}(\Gamma) = \{\mu_3, \mu_2, \mu_1\}$. Consider an index given by a T-tuple $(i_1, ..., i_T)$ and $\mathcal{O}_t$ is a permutation function on the tth position i_t. We use $\mathcal{O}_{1:T} = \mathcal{O}_1 \times \mathcal{O}_2 \cdots \times \mathcal{O}_T$ to denote the permutation on the T-tuple: $(i_1, i_2, ..., i_T) \to (\mathcal{O}_1(i_1), \mathcal{O}_2(i_2), ..., \mathcal{O}_T(i_T))$.

Lemma 1 *The identifiability of GMM gives that* $\sum_{k=1}^{M} \pi_k \phi(X|\mu_k, \Sigma_k) = \sum_{l=1}^{M^*} \pi_l^* \phi(X|\mu_l^*, \Sigma_l^*)$ *with distinct* (μ_k, Σ_k)*'s, distinct* (μ_l^*, Σ_l^*)*'s, and all the priors* $\pi_k > 0$ *and* $\pi_l^* > 0$, $k = 1, ..., M$, $l = 1, ..., M^*$ *implies* $M = M^*$ *and up to a permutation of mixture components,* $\pi_k = \pi_k^*$, $\mu_k = \mu_k^*$ *and* $\Sigma_k = \Sigma_k^*$.

See for instance Yakowitz and Spragins (1968); Titterington et al. (1985).

Theorem 2 *Let* $\mathcal{M}$ *and* $\mathcal{M}'$ *be two sets of parameters for a lattice GMM on the same variable partition* $\mathcal{P}$. *Assume that* $\mathcal{M}$ *and* $\mathcal{M}'$ *specify the same density function and their lattices* Θ *and* Θ' *are both non-redundant. Then the number of components for each variable block* $M_t = M_t'$, $t = 1, ..., T$. *There exists a unique permutation* $\mathcal{O}_t : i_t \to i_t'$ *for each variable block* $X^{(t)}$ *such that* $\mathcal{M}'_{X^{(t)}} = \mathcal{O}_t(\mathcal{M}_{X^{(t)}})$ *and* $\mathcal{M}' = \mathcal{O}_{1:T}(\mathcal{M})$.

Remark: $M_t = M_t'$ and $\mathcal{M}'_{X^{(t)}} = \mathcal{O}_t(\mathcal{M}_{X^{(t)}})$ are simple results of Lemma 1. The assumption that Θ and Θ' are non-redundant does not imply every prior in Eq. (6) is positive. Hence Lemma 1 cannot be directly applied to prove $\mathcal{M}$ and $\mathcal{M}'$ are identical up to permutation. We provide the proof for Theorem 2 in Appendix B.

The generic HMM-VB density in Eq. (3), given a pre-determined variable block structure can be re-written as

$$f(\mathbf{x}) = \sum_{i_1=1}^{M_1} \sum_{i_2=1}^{M_2} \cdots \sum_{i_T=1}^{M_T} \left(\pi_{i_1} a^{(1)}_{i_1, i_2} a^{(2)}_{i_2, i_3} \cdots a^{(T-1)}_{i_{T-1}, i_T} \right) \cdot \prod_{t=1}^{T} \phi(x^{(t)}|\mu^{(t)}_{i_t}, \Sigma^{(t)}_{i_t}). \tag{7}$$

It is clear that HMM-VB is a lattice GMM on partition $\mathcal{P}$. Specifically, the parameter set $\mathcal{M}$ for a HMM-VB is $\mathcal{M} = \{\mathbb{M}, \Theta, \pi_{i_1}, a^{(t-1)}_{i_{t-1}, i_t}, i_1 = 1, ..., M_1, i_t = 1, ..., M_t, t = 2, ..., T\}$.

We prove the following lemma in Appendix B which specifies the conditions under which a HMM-VB has non-redundant lattice.

Lemma 3 *The HMM-VB in Eq.* (7) *has non-redundant lattice* Θ *if and only if* Θ *is distinct and* $\pi_{i_1} > 0$, *for* $\forall i_1 \in \{1, ..., M_1\}$ *and for* $\forall t = 2, ..., T$ *and* $i_t \in \{1, ..., M_t\}$, *there exists at least one* $i_{t-1} \in \{1, ..., M_{t-1}\}$ *such that* $a^{(t-1)}_{i_{t-1}, i_t} > 0$.

Corollary 4 *For a given variable block structure, if two HMM-VBs,* $\mathcal{M}$ *and* $\mathcal{M}'$, *both have non-redundant lattices* Θ *and* Θ', *and define the same density function, then there exists a unique permutation* $\mathcal{O}_t$ *for the mixture components of every variable block* $X^{(t)}$, $t = 1, ..., T$, *such that* $\mathcal{M}' = \mathcal{O}_{1:T}(\mathcal{M})$.

Remark: Corollary 4 establishes the identifiability of HMM-VB under a given variable block structure. By Theorem 2, it is obvious that $\mathbb{M}' = \mathcal{O}_{1:T}(\mathbb{M})$, $\Theta' = \mathcal{O}_{1:T}(\Theta)$, and $\Pi'(s_1, ..., s_T) = \mathcal{O}_{1:T}(\Pi(s_1, ..., s_T))$. We only need to show that the last equation implies that the transition probabilities are identical up to permutation $\mathcal{O}_{1:T}$. We prove this in Appendix B.

We have so far assumed that the partition $\mathcal{P}$ for the lattice GMM is given. A natural question is whether $\mathcal{P}$ is identifiable. To answer the question, we assume $\mathcal{P}$ is a tight variable partition. Without this constraint, we can express any GMM with M components as a lattice GMM with every block containing a single variable and every block being assigned with M components. The number of components for the lattice GMM will be M^d with only M components assigned with non-zero priors. If we restrict to tight variable partitions, the maximum partition will not be trivial. Furthermore, we can prove that the the maximum partition always exists and is unique. Thus the identifiability of a GMM (according to Lemma 1) ensures that the maximum partition is identifiable. That is, the variable block structure, in its most refined partition, is identifiable.

Theorem 5 *The maximum variable partition of a GMM exists and is unique.*

The proof is given in Appendix B.

Suppose $\mathcal{M}$ is a HMM-VB on a tight partition $\mathcal{P}$. It is obvious that $\mathcal{P}$ is a tight partition for a HMM-VB if and only if $\pi_{i_1} > 0$, $\forall i_1 \in \{1, ..., M_1\}$, and all the transition probabilities are positive. If $\mathcal{P}$ is a maximum partition for the equivalent lattice GMM of $\mathcal{M}$, $\mathcal{P}$ is identifiable once $\mathcal{M}$ is given (by Theorem 5). However, because the latent states s_1, ..., s_T of the HMM-VB follow a Markov chain (an extra constraint), even if $\mathcal{P}$ cannot be further refined for the HMM-VB, it is not necessarily the maximum partition for the corresponding lattice GMM. In other words, a lattice GMM that is not a HMM-VB on its maximum partition can be an HMM-VB on a coarser partition. Another subtlety with HMM-VB is that even when $\mathcal{P}$ is identifiable, the order of the variable blocks is not for the simple reason that a reversed Markov chain is still Markov. The fact that the order of the variable blocks is not identifiable is also clear from the extreme case when the latent states s_1, ..., s_T are independent and therefore any order of them is valid. On the other hand, because our mode-based clustering method for HMM-VB searches for the modes of the joint density, the modes are not affected as long as the equivalent lattice GMM is the same. Hence regardless of the order of the variable blocks (e.g., forward chain or backward chain), if the HMM-VB is a correct model, the modes will not change, and neither will the clustering result.

4.2 Consistency of BIC for Model Selection

In this section, we prove a special case that the probability of selecting the true variable block structure by minimizing BIC approaches 1 as $n \to \infty$ under some assumptions and the conditions that the true variable structure is among the candidates and the number of components for each variable block is known.

Let the number of variable blocks be T and its true value be T^0. Denote the true number of components in each block by M_t^0, $t = 1, ..., T^0$. Also let $\mathbb{M}^{0^{T^0}} = \{M_1^0, ..., M_{T^0}^0\}$. Denote the variable index sets of the true ordered blocks by $\mathbf{C^0}^{T^0} = (C_1^0, C_2^0, ..., C_{T^0}^0)$. For example, C_1^0 contains the indices of the variables in the first block. For a particular sequence of variable blocks $\mathbf{C}^T = (C_1, C_2, ..., C_T)$, we use the notation $X^{(C_t)}$ to denote the tth block of variables according to $\mathbf{C}^T$. Denote the parameters of a model collectively by $\gamma_{(\mathbf{C}^T, \mathbb{M}^T)} \in \Gamma_{(\mathbf{C}^T, \mathbb{M}^T)}$, where $\Gamma_{(\mathbf{C}^T, \mathbb{M}^T)}$ is the space of the parameters.

Denote by g the true density function of X. Let $D_{KL}(g||f) = \int g(x) \log(g(x)/f(x))dx$ be the Kullback-Leibler divergence from density f to g. We define the following two notations:

$$\begin{aligned}\gamma^*_{(\mathbf{C}^T, \mathbf{M}^T)} &= \underset{\gamma_{(\mathbf{C}^T, \mathbf{M}^T)} \in \Gamma_{(\mathbf{C}^T, \mathbf{M}^T)}}{\operatorname{argmin}} D_{KL}(g||f(.|\gamma_{(\mathbf{C}^T, \mathbf{M}^T)})) \\ &= \underset{\gamma_{(\mathbf{C}^T, \mathbf{M}^T)} \in \Gamma_{(\mathbf{C}^T, \mathbf{M}^T)}}{\operatorname{argmax}} E_X\{\log f(X|\gamma_{(\mathbf{C}^T, \mathbf{M}^T)})\}, \\ \hat{\gamma}_{(\mathbf{C}^T, \mathbf{M}^T)} &= \underset{\gamma_{(\mathbf{C}^T, \mathbf{M}^T)} \in \Gamma_{(\mathbf{C}^T, \mathbf{M}^T)}}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} \log\{f(\mathbf{x}_i|\gamma_{(\mathbf{C}^T, \mathbf{M}^T)})\}.\end{aligned}$$

Consider the case where the true model contains finite T^0 variable blocks and all the models in consideration are restricted to have the same number of variable blocks ($T = T^0$). To simplify the notation, all the dependence over T^0 and T is omitted below. We make two assumptions:

A1 There exists a unique $(\mathbf{C}^0, \mathbb{M}^0)$ such that $g = f(.|\gamma^*_{(\mathbf{C}^0, \mathbb{M}^0)})$ for some parameter value γ^*.

As discussed previously, when the order of the variable blocks is reversed, we obtain a HMM-VB that yields the same density function for X although parameterized differently. Hence, the order is not identifiable. Thus, the uniqueness in assumption **A1** is implicitly up to a reverse ordering. To further simplify the notation, the dependency over $\mathbb{M}^0$ is omitted below.

A2 $\gamma^*_{(\mathbf{C})}$ and $\hat{\gamma}_{(\mathbf{C})}$ are assumed to belong to a compact subspace $\Gamma'_{(\mathbf{C})}$:

$$\Gamma'_{(\mathbf{C})} = (\Lambda \times \mathcal{A} \times \mathcal{B}(\eta, |C_1|)^{M_1^0} \times \mathcal{D}_{|C_1|}^{M_1^0} \times \mathcal{B}(\eta, |C_2|)^{M_2^0} \times \mathcal{D}_{|C_2|}^{M_2^0} \times \cdots \times \mathcal{D}_{|C_T|}^{M_T^0} \times \mathcal{B}(\eta, |C_T|)^{M_T^0}) \cap \Gamma_{(\mathbf{C})},$$

where

[1] $\Lambda = \{(\pi_1, ..., \pi_{M_1^0}) \in [0,1]^{M_1^0}; \sum_{k=1}^{M_1^0} \pi_k = 1\}$ denotes the set of possible proportions,

[2] $$\mathcal{A} = \left\{ \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,M_{t+1}^0} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,M_{t+1}^0} \\ \vdots & \vdots & \cdots & \vdots \\ a_{M_t^0,1} & a_{M_t^0,2} & \cdots & a_{M_t^0,M_{t+1}^0} \end{pmatrix} \in [0,1]^{M_t^0 \times M_{t+1}^0}; \forall j \in 1 : M_t^0, \textstyle\sum_{k=1}^{M_{t+1}^0} a_{j,k} = 1 \right\}_{t=1}^{T-1},$$

[3] $\mathcal{B}(\eta, d) = \{\mathbf{x} \in \mathcal{R}^d, ||\mathbf{x}|| \leq \eta\}$, where $\forall \mathbf{x} \in \mathcal{R}^d$, $||\mathbf{x}|| = \sqrt{\sum_{i=1}^d x_i^2}$,

[4] $|C|$ denotes the cardinality of the set C,

[5] $\mathcal{D}_d$ is the set of $d \times d$ positive definite matrices with eigenvalues in $[a, b]$ with $0 < a < b$.

Theorem 6 *Under the special case that $T = T^0 < \infty$, and assumptions* **A1**, **A2**, *the ordered variable block structure $\hat{\mathbf{C}}$ that minimizes BIC under a given $\mathbb{M}^0$ is consistent in the sense that $P(\hat{\mathbf{C}} = \mathbf{C}^0) \to 1$ as $n \to \infty$.*

The proof is given in Appendix C.

Remark: We point out that the proved consistency in the asymptotic setting as stated above holds for marginal likelihood approaches as well. We observe empirically, however, that BIC performs better for model selection when the sample size is only modestly large. This is expected as BIC addresses overfitting, which has been discussed extensively in literature (e.g., Burnham and Anderson (2003)). Our simulation study in Section 5.1.1 further confirms the advantage of BIC when the sample size is modestly large.

4.3 Modal Baum-Welch Algorithm and Its Optimality

Recall the definition for probability $L_k(\mathbf{x}, t) = P(s_t = k \mid \mathbf{x})$, $k \in \mathcal{S}_t$, $t = 1, ..., T$.

Theorem 7 *For a HMM-VB, suppose the solution of the M-step in the MEM algorithm provided by Eq. (2) is divided into blocks $\mathbf{x}^{[r+1]} = (x^{(1),r+1}, ..., x^{(T),r+1})$. Then*

$$x^{(t),r+1} = \left(\sum_{k \in \mathcal{S}_t} L_k(\mathbf{x}^{[r]}, t) \cdot \left(\Sigma_k^{(t)}\right)^{-1} \right)^{-1} \left(\sum_{k \in \mathcal{S}_t} L_k(\mathbf{x}^{[r]}, t) \cdot \left(\Sigma_k^{(t)}\right)^{-1} \cdot \mu_k^{(t)} \right), \ t = 1, ..., T.$$

This theorem ensures that the MBW algorithm is the exact special case of the MEM algorithm when the GMM is a HMM-VB. The proof is provided in Appendix D.

5. Experiments

In this section, we present experimental results on several simulated data sets (Section 5.1), one mass cytometry data (Section 5.2), and two data sets with very high dimensions (Section 5.3). For each data set, the BW algorithm was run repeatedly starting from multiple initial models. In Section 3.1, the different ways of initialization are described. Among the final models, we choose the one yielding the maximum likelihood.

5.1 Simulation with Various Types of Variable Block Structures

We first conduct simulation studies to examine the effectiveness of HMM-VB at capturing small clusters, the overall clustering performance, and its robustness against different parameter settings.

5.1.1 Two variable blocks

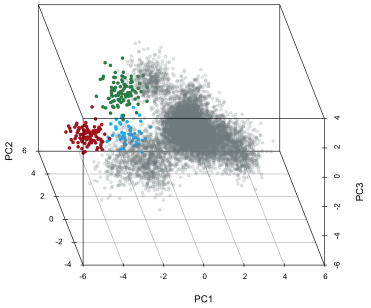
Using a similar set-up as in Lin et al. (2013), a sample of size $10,000$ with dimension $d = 8$ is drawn from a hierarchical mixture model. There are two variable blocks. Following the notations in the previous section, $\mathbf{x}_i$ is divided into two variable blocks, also called subvectors, $x_i^{(1)}$ and $x_i^{(2)}$. The first subvector contains the first 5 dimensions: $x_i^{(1)} = (x_{i,1}, ..., x_{i,5})$, with $d_1 = 5$. The second subvector contains the last 3 dimensions: $x_i^{(2)} = (x_{i,6}, x_{i,7}, x_{i,8})$, with $d_2 = 3$. In particular, $x_i^{(1)}$'s are generated from a mixture of 7 normal distributions such that the last two normal distributions (assigned with component priors 0.01 and 0.02 respectively) have high mean values for the second and third dimensions. The other normal components have very different proportions and mean vectors. The second subvector, $x_i^{(2)}$'s, are generated from a mixture of 10 normal distributions, where only two of them have high mean values across all three dimensions. The component proportions of $x_i^{(2)}$ vary according to which normal component $x_i^{(1)}$ was generated from, as in the assumption of HMM-VB. A detailed description of the data generation mechanism is in Appendix E. The data is designed to have at least one distinct cluster after standardization (subtract mean and divided by the standard error). In particular, the standardized data have a well-separated region that the five dimensions x_2, x_3, x_6, x_7, x_8 are of high positive values, and the rest are negative. The particular designed data region contains 100 data points, which account for only 1% of the data.

We also run the MBW algorithm to find modes of the true density (see Appendix E). The result serves two purposes: 1) to validate the effectiveness of MBW for finding small clusters when there is no model estimation error; 2) to be used as a ground truth for comparison with our HMM-VB method. MBW identifies perfectly that particular cluster of size 100. In addition, it finds in total 16 modes (clusters). Among them, there are two additional small clusters of size 54 and 98 respectively. Figure 5.1.1 (Left) shows the three smallest clusters.

To fit HMM-VB given the dependence structure among the 8 variables, we only need to specify M_1 and M_2, the numbers of mixture components for the two variable blocks. If casted as a GMM, the HMM-VB has $M_1 \times M_2$ components for the full dimensional data. Model selection by BIC is conducted to select the optimal M_1 and M_2. Summaries on various model specifications are listed in Figure 5.1.1 (Right). The model with $M_1 = 7$ and $M_2 = 10$ has the lowest BIC and is thus chosen.

Recall that we search modes by MBW starting from the mean of every component that has been chosen by a data vector according to the maximum a posteriori rule. We call the components that have not been chosen by any data point the empty mixture components, while the others are nonempty. Figure 5.1.1 (Right) shows that even though the total number of nonempty mixture components increases with M_1 and M_2, the number of clusters after modal clustering remains relatively stable.

In Table 1, we compare clustering results of HMM-VB with four baseline methods: K-means, agglomerative hierarchical clustering on a set of dissimilarities, Mclust, and modal GMM. We also tried other methods, such as spectral clustering using R `kernlab` package, pdfCluster (Azzalini and Menardi, 2014) using R `pdfCluster` package, and the mean-shift algorithm (Fukunaga and Hostetler, 1975; Cheng, 1995) using R `MeanShift` package. R `pdfCluster` program was aborted by the computer system when applied to the data. R `kernlab` and `MeanShift` packages failed to obtain results after several hours, while the



(M_1, M_2)	BIC	# nonempty components	# clusters
$(3, 3)$	$183,687.4$	8	2
$(3, 5)$	$176,134.6$	9	7
$(7, 10)$	$163,486.1$	35	16
$(8, 10)$	$163,724.0$	37	16
$(10, 10)$	$164,245.1$	45	16
$(10, 15)$	$164,822.9$	70	17
$(15, 10)$	$165,338.6$	56	16
$(15, 15)$	$166,313.4$	85	18
$(20, 15)$	$167,761.7$	121	17
$(20, 20)$	$206,478.6$	158	19

Figure 2: (Left): Visualization of the simulated data in Section 5.1.1 based on the first three principal components (PC). The smallest three clusters are plotted in blue (54 observations), green (98), red (100), and the rest are in grey. (Right): Comparisons of BIC, total number of nonempty components and clusters under various model specifications (M_1 and M_2).

methods in Table 1 take less than one minute. For K-means, hierarchical clustering, and Mclust, we used the three R functions `kmeans` (with 20 starting points), `hclust` and `Mclust`. In this analysis, we treat the 16 clusters found by MBW applied to the true mixture density as the ground truth. Hence, for K-means and hierarchical clustering, we manually set 16 as the number of clusters. The number of normal mixture components and the covariance structure are determined by BIC in the `Mclust` package (the package version is 5.2.3). More specifically, we let `Mclust` search a maximum of 20 mixture components and then used the BIC to choose among all the default covariance matrix models, as computational errors were generated when searching over all the models. For this data set, the optimal number of clusters chosen by Mclust is 15. The modal GMM analysis is performed using the same C codes for fitting the HMM-VB model. We simply specified a single variable block containing all the 8 variables. The HMM-VB model is then reduced to a usual GMM. In addition, we let the total number of mixture components be $7 \times 10 = 70$.

The first column of Table 1 shows the 16 true clusters and their cluster sizes. Let us call the true clusters E_1^*, ..., E_{16}^*. Results about each true cluster occupy one row in the table, while results obtained from every clustering method are reported in one column. Take the first row for cluster E_1^* as an example. E_1^* has $5,098$ data points. A cluster generated by any method is described by a pair of numbers, the one outside the parenthesis being the size of the cluster and the one inside being the number of points shared with the true cluster. For instance, HMM-VB generates a cluster containing $5,096$ points, among which $5,088$ are shared with E_1^*. This cluster is the best match with E_1^*. In comparison, modal GMM yields a cluster of size $4,695$ overlapping E_1^* by $4,467$. By K-means, hierarchical clustering or Mclust, E_1^* is split roughly into three clusters. Thus three pairs of cluster sizes are recorded in the first row under each method. For clarity of presentation, if the overlap between a

cluster and any E_i^* is too small, the cluster is not listed in the ith row. Reversely, a cluster generated by a certain method may also cover a substantial portion of points from multiple true clusters. For instance, hierarchical clustering yields one small cluster of size 257, which contains both part of E_1^* and E_2^*. A cluster of size 154 generated by K-means contains E_5^* and E_8^*. Therefore, a single cluster generated by a certain method may be reported in multiple rows. To indicate clearly such a case, the same color (except for black) is used to mark out a single cluster in any column. If a cluster is reported only once in any column, the result is shown in black (in other words, black is not a color code).

True cluster	HMM-VB	K-means	HC	Mclust	Modal GMM
$1:5,098$	$5,096(5,088)$	$1,700(1,693)$ $1,452(1,447)$ $1,477(1,474)$	$1,192(1,140)$ $3,852(3,757)$ $257(116)$	$2,240(2,239)$ $2,815(2,815)$ $211(28)$	$4,695(4,467)$
$2:184$	$187(184)$	$638(173)$	$257(132)$	$211(182)$	$4,695(140)$
$3:483$	$483(483)$	$485(481)$	$400(394)$ $81(81)$	$482(482)$	$299(297)$ $179(179)$
$4:258$	$263(254)$	$254(249)$	$257(247)$	$260(256)$	$214(18)$
$5:100$	$100(100)$	$154(100)$	$154(100)$	$154(100)$	$152(100)$
$6:375$	$375(373)$	$380(371)$	$1,242(362)$	$378(375)$	$374(362)$
$7:345$	$346(345)$	$510(338)$	$500(338)$	$513(345)$	$377(338)$
$8:54$	$54(54)$	$154(54)$	$154(54)$	$154(54)$	$152(52)$
$9:161$	$161(161)$	$510(156)$	$500(155)$	$513(161)$	$145(143)$
$10:502$	$505(502)$	$499(499)$	$552(499)$	$506(502)$	$348(348)$ $155(153)$
$11:201$	$195(191)$	$201(193)$	$198(188)$	$198(196)$	$214(196)$
$12:98$	$98(98)$	$98(98)$	$98(98)$	$98(98)$	$98(98)$
$13:384$	$384(384)$	$384(383)$	$383(382)$	$384(384)$	$383(383)$
$14:915$	$914(908)$	$929(907)$	$1,242(841)$ $2(1)$	$919(913)$	$852(848)$ $4,695(60)$
$15:707$	$704(704)$	$704(704)$	$697(697)$ $2(1)$	$707(707)$	$556(556)$ $149(149)$
$16:135$	$135(135)$	$135(135)$	$135(135)$	$135(135)$	$299(1)$
Time	4.087s	1.106s	4.245s	1.457s	48.169s

Table 1: Comparisons of clustering performance among HMM-VB, K-means, hierarchical clustering (HC), Mclust, and modal GMM for the simulation data in Section 5.1.1.

Table 1 demonstrates that HMM-VB outperforms clearly the other methods in terms of matching the true clusters, especially for the rare clusters. K-means, hierarchical clustering, and Mclust tend to yield similar-sized clusters. All of them split the largest true cluster E_1^* into 3 smaller ones. Except for HMM-VB, all the other four methods fail to correctly identify all three smallest clusters, E_5^*, E_8^* and E_{12}^*. Specifically, according to the four methods, E_5^* and E_8^* are in the same cluster. The CPU time per model fitting (based on the best model specification) on an iMac with Intel Core i7 3.0GHz/8GB memory is recorded, given in the last row of Table 1. The unit of time is second. Under the same specification of

the number of clusters, K-means is faster than hierarchical clustering. Mclust is performed on 15 mixture components, while both HMM-VB and modal GMM are fitted using 70 mixture components. The fact that HMM-VB is much faster than modal GMM shows the computational advantage to regularize a general GMM by a variable block structure.

In addition, we use this simulation data to test the search algorithm for the variable block structure, which is described in Section 3.4. We run the algorithm with 6 different initial random permutations. The algorithm successfully selects the correct variable blocks according to the true distribution. Furthermore, to empirically demonstrate the efficiency of BIC for model selection, we also incorporate the comparison with (log) marginal likelihood approach in Table 2. Both approaches select the same model under the larger sample size.

Random permutation	Log marginal likelihood	BIC
	$n = 10,000$	
$3,1,8,5,7,2,6,4$	$(3,5,2,4)*(1,8,7,6):-80887.22$	$(3,1,8,5,7,2,6,4):167105.9$
$6,4,7,3,5,2,8,1$	$(6,7,8)*(4,3,5)*(2)*(1):-80330.64^{\dagger}$	$(6,4,7,3,5,2,8,1):167113.3$
$4,6,8,5,7,2,3,1$	$(4)*(6,8,5,7)*(2,3)*(1):-82102.33$	$(4,6,8,5,7)*(2,3,1):166259$
$4,6,5,8,1,3,7,2$	$(4,5,3,2)*(6,8,1,7):-80924.62$	$(4,6,5,8,1,3,7,2):167593.6$
$5,3,2,6,8,7,1,4$	$(5,1,4)*(3,2)*(6,7)*(8):-80693.58$	$(5,3,2,1,4)*(6,8,7):164213.2^{\star}$
$7,4,5,3,1,6,8,2$	$(7,6,8)*(4,3,1)*(5)*(2):-80343.08$	$(7,4,5,3,1,6,8,2):167468.1$
	$n = 50,000$	
$3,1,8,5,7,2,6,4$	$(3,8,7,2,6,4)*(1,5):-406073.13$	$(3,1,2,4)*(8,5,6)*(7):817840.6$
$6,4,7,3,5,2,8,1$	$(6,4,7,3,2,8,1)*(5):-403308.14$	$(6,4,7,3,2,8,1)*(5):814125.2$
$4,6,8,5,7,2,3,1$	$(4,6,8,7)*(5,2,3)*(1):-402800.94$	$(4,6,8,7)*(5,2,3,1):809706.6$
$4,6,5,8,1,3,7,2$	$(4,6,8,7)*(5,1)*(3,2):-402785.41$	$(4,6,8,7)*(5,1)*(3,2):810212.5$
$5,3,2,6,8,7,1,4$	$(5,3,2,1,4)*(6,8,7):-402450.86^{\dagger}$	$(5,3,2,1,4)*(6,8,7):809110.6^{\star}$
$7,4,5,3,1,6,8,2$	$(7,6,8)*(4,3,1)*(5,2):-402624.38$	$(7,4,5,6,8,2)*(3,1):822285.2$

Table 2: Comparison of BIC and log marginal likelihood for model selection using 6 initial random permutations under two different sample sizes. We mark the model selected by BIC with * and the model selected by log marginal likelihood with †. We use parentheses to indicate a particular variable block, and $*$ to separate the different variable blocks.

5.1.2 No Information on Variable Blocks

We now study the clustering performance of HMM-VB when there are no clear-cut variable block structures. A sample of size $10,000$ with $d = 5$ dimensions is drawn from a $10-$component GMM. We ensure that all the 10 components correspond one-to-one with 10 distinct clusters by setting mean vectors far apart. Hence, each cluster can be described by a single Gaussian component. The detailed description of the data generation mechanism is in Appendix E. Figure 3 shows the scatter plots of the generated data. Colors indicate the cluster membership. Several clusters overlap tremendously within two-dimensional space. As in the previous experiment (Section 5.1.1), we run MBW on the true model density, which identifies in total 10 clusters with cluster memberships matching precisely the true latent component identities.

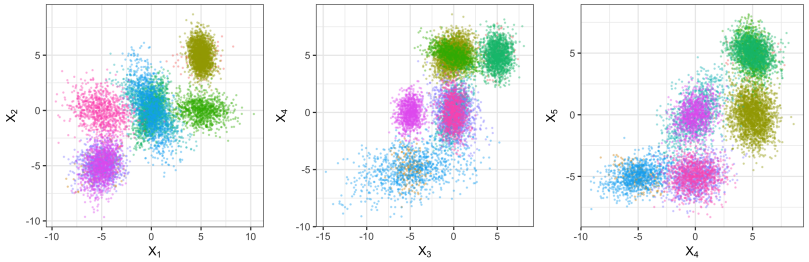


Figure 3: Pairwise scatter plots of simulated data from Section 5.1.2. Colors indicate the cluster membership.

True cluster 1 : 96	HMM-VB $\mathcal{M}_{1,2}$;$\mathcal{M}_3$ 94(94); 96(96)	K-means 1582(71) 2327(24)	HC 766(95)	Mclust 96(96)	Modal GMM 96(96)
2 : 190	190(190); 182(180)	191(190)	538(190)	190(190)	190(190)
3 : 2304	2302(2302); 2300(2300)	2327(2303)	2133(2133) 817(171)	2304(2304)	2304(2304)
4 : 993	991(987); 985(980)	1010(992)	817(321) 766(671)	993(993)	993(993)
5 : 1509	1512(1508); 1507(1506)	1582(1508)	1517(1509)	1509(1509)	1509(1509)
6 : 1012	1002(983); 997(976) 936(22); 966(28)	908(903) 942(29)	817(325) 593(591) 1967(89)	1009(1008)	1010(1009)
7 : 982	971(965); 911(906) 46(46)	517(517) 528(463)	538(348) 1967(28) 521(520) 84(84)	986(982)	985(982)
8 : 972	979(962); 942(938) 966(20)	971(955)	1967(919) 1064(53)	971(971)	972(971)
9 : 1011	1015(1009); 1017(1008)	1024(1009)	1064(1011)	1011(1011)	1011(1011)
10 : 931	936(903); 966(910)	942(909) 971(16)	1967(931)	931(930)	930(929)
Time	32.823s	0.298s	2.636s	0.902s	1.108s

Table 3: Comparison of clustering performance among three HMM-VB models, K-means, hierarchical clustering (HC), Mclust, and modal GMM for simulation data in Section 5.1.2. The first two HMM-VB models give the same results for major clusters.

To demonstrate the robustness of HMM-VB under different variable block structures, we fit the data using different model specifications. We let model $\mathcal{M}_1$ be a HMM-VB with 5 variable blocks each containing one variable and ordered by the given indices of the

variables. The number of components for each block chosen by BIC is $M_t = 7$, $t = 1, ..., 5$. Model $\mathcal{M}_2$ is defined similarly as $\mathcal{M}_1$, but we reverse the ordering of the variables such that $x = (x_5, x_4, ..., x_1)$. Lastly, we let $\mathcal{M}_3$ be a HMM-VB with 5 variable blocks ordered by a random permutation $x = (x_1, x_3, x_4, x_2, x_5)$. Table 3 shows the clustering results for the three HMM-VB models and the other four baseline methods. The format of the results in this table is the same as that for Table 1 in the previous section. Again we pre-set the number of clusters for K-means, hierarchical clustering, and modal GMM as the true value. The model specification for Mclust is similar to the previous experiment, but we allow `Mclust` to search for up to 12 Gaussian components. For clarity of presentation, clusters that share fewer than 5 points with any true cluster are not reported in Table 3. As expected $\mathcal{M}_1$ and $\mathcal{M}_2$ yield very similar results. As discussed at the end of Section 4.1, when the ordering of the variable blocks is reversed, the GMM casted from the HMM-VB is essentially the same. Small differences may arise due to some random factors caused by initialization. We simply report results from one of the two models. Table 3 suggests that HMM-VB is quite robust to the ordering of the variables. Both K-means and hierarchical clustering fail to extract the cluster structures. As expected, both Mclust and modal GMM can accurately recover all the clusters because the data is generated according to the generic GMM model.

The search scheme for selecting the optimal variable block structure determines that only one variable block is needed (that is, the usual GMM). This is consistent with the true model. However, Table 3 shows that although the HMM-VB with five blocks is not the right model, it can still perform reasonably well for clustering. This finding suggests that HMM-VB can be a new strategy to regularize a GMM, the complexity of which is normally only controlled by the number of components or constraints on the covariance matrices.

5.1.3 Large Data Size

Since HMM-VB is motivated by single-cell data analysis, where the sample size can be up to several millions, we now study the performance of HMM-VB for large data sets with moderately high dimensions. We let dimension $d = 40$, and sample size n ranges from $100,000$, $1,000,000$ and $5,000,000$. The first 10 dimensions are generated from a 3-component GMM. The remaining 30 dimensions are generated from a 5-component GMM, where the mixture component proportions vary according to which normal component the first 10 dimensions are generated from. In addition, the covariance matrices of the last 30 dimensions are block diagonal, containing two blocks of sizes 10×10 and 20×20. Furthermore, by specifically designing the mean vectors and the transition probability matrix from the first 10 dimensions to the rest, the data generated contain 5 distinct normal components which correspond to 5 distinct clusters. The detailed description for the data generation is in Appendix E. The relative proportions for the 5 clusters are about 0.005, 0.045, 0.07, 0.18, and 0.7.

In this particular study, Mclust encounters some numerical errors when performing model selection. Because K-means and hierarchical clustering perform poorly in the previous two studies, we hereby focus on the comparison between HMM-VB and modal GMM. To fit HMM-VB, we divide the 40 dimensions into 3 blocks: $x_i^{(1)} = (x_{i,1}, ..., x_{i,10})$, $x_i^{(2)} = (x_{i,11}, ..., x_{i,20})$, $x_i^{(3)} = (x_{i,21}, ..., x_{i,40})$. According to the model specification, we let $M_1 = 3$, $M_2 = 5$, and $M_3 = 5$. To fit modal GMM, we let the total number of mixture components to be $3\times5 = 15$.

True cluster	HMM-VB	Modal GMM
	$n = 100,000$	
$1:510$	$510(510)$	$4,940(510)$
$2:4,430$	$4,430(4,430)$	$4,940(4,430)$
$3:7,006$	$7,006(7,006)$	$7,006(7,006)$
$4:17,995$	$17,995(17,995)$	$17,995(17,995)$
$5:70,059$	$70,059(70,059)$	$70,059(70,059)$
Time	4.1 mins	9.9 mins
	$n = 1,000,000$	
$1:5,045$	$5,045(5,045)$	$50,279(5,045)$
$2:45,234$	$45,234(45,234)$	$50,279(45,234)$
$3:69,684$	$69,684(69,684)$	$69,684(69,684)$
$4:180,789$	$180,789(180,789)$	$180,789(180,789)$
$5:699,248$	$699,248(699,248)$	$699,248(699,248)$
Time	40.59 mins	64.55 mins
	$n = 5,000,000$	
$1:25,119$	$25,119(25,119)$	$250,142(25,119)$
$2:225,023$	$225,023(225,023)$	$250,142(225,023)$
$3:351,202$	$351,202(351,202)$	$351,202(351,202)$
$4:899,498$	$899,498(899,498)$	$899,498(899,498)$
$5:3,499,158$	$3,499,158(3,499,158)$	$3,499,158(3,499,158)$
Time	56.56 mins	322.25 mins

Table 4: Comparison of clustering performance between HMM-VB and modal GMM for simulation study in Section 5.1.3.

Table 4 shows the clustering results for the two models for simulated data of three different sample sizes. HMM-VB correctly identifies all the clusters for the three data sets. However, modal GMM cannot separate the two smallest clusters. In addition, HMM-VB requires much shorter computation time than modal GMM, even though the equivalent GMM of the HMM-VB has $3 \times 5 \times 5 = 75$ components. The difference in computation time is even more dramatic when the sample size is large. In order to detect more accurately the rare clusters by modal GMM, we could increase the number of mixture components. However, the increase may have to be large causing much slower computation.

5.1.4 Clustering Variation Study

To investigate the variation of clustering results for the above three simulations, we in addition randomly generate 100 replicates from the two variable blocks model in Section 5.1.1, 100 replicates from the 10-component GMMs with randomly generated covariance matrices described in Section 5.1.2, and only 10 replicates for the model in Section 5.1.3 due to the extensive computational time caused by large data size. We use the adjusted Rand index (ARI) (Rand, 1971) to assess the similarity between the clustering result by each method and the true cluster assignments. Summaries based on the ARI for the first two simulation studies are shown in Figure 4, and that for the last simulation is provided in Table 5.

For the simulation in Section 5.1.1, the variables have a two block structure. The average ARI achieved by HMM-VB is substantially better than the other methods, and the variation of the HMM-VB results is much smaller than those by Mclust or modal GMM. For the simulation in Section 5.1.2, Model 1 is used for HMM-VB. The variable structure assumed

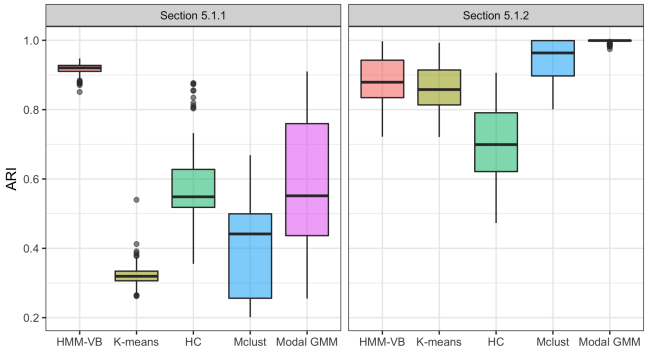


Figure 4: Boxplots of ARI based on 100 replicates generated by simulations in Section 5.1.1 and 5.1.2.

by HMM-VB is not right, while Mclust and modal GMM are based on the correct model. Nevertheless, the boxed range of ARIs by HMM-VB overlaps with that by Mclust although the average by HMM-VB is lower. The best result is obtained by modal GMM. For the third simulation in Section 5.1.3, although both HMM-VB and modal GMM yield high ARIs, the average number of clusters obtained by HMM-VB is always the same or nearly the same as the true value, while that by modal GMM is consistently lower by one. Finally, we note that ARI measures overall similarity between clustering results. Thus small clusters matter less than large clusters for ARI. If finding rare clusters is important, we should not rely only on ARI to evaluate the methods. Results shown in Table 1, 3, and 4 can be more pertinent.

Sample size	mean ARI	median ARI	sd ARI	$\bar{C}$
		HMM-VB		
$100,000$	1	1	0	5
$1,000,000$	1	1	0	5
$5,000,000$	0.9999	1	2.88×10^{-4}	4.9
		Modal GMM		
$100,000$	0.9991	0.9991	5.74×10^{-5}	4
$1,000,000$	0.9992	0.9991	2.88×10^{-4}	4.1
$5,000,000$	0.9513	0.9991	0.15	4.1

Table 5: Summaries of ARI and the mean number of clusters found ($\bar{C}$) based on 10 replicates generated by the simulation in Section 5.1.3.

5.2 Study of CyTOF Data

As a motivating example, current high-throughput flow cytometry experiments routinely measure $10 \sim 20$ parameters (cell markers/variables) on a large number of single cells. The current mass cytometry (CyTOF) can measure up to 50 parameters at a single cell level. A key first step to analyze this wealth of data is to partition the data (cells) from a blood or tissue sample into clusters based on the measured cell markers. The identified clusters are usually referred to as (cell) subsets.

Many studies adopt clustering based on mixture models to identify cell subsets objectively and automatically for single-cell data analysis (e.g., Boedigheimer and Ferbas (2008); Lo et al. (2008); Chan et al. (2008); Pyne et al. (2009); Aghaeepour et al. (2013); Lin et al. (2016)). One challenge encountered is that cell subsets of interest are typically of low frequencies (e.g., $\sim 0.01\%$ of total cells), while it is important to detect cell heterogeneity, especially very low frequency cell subsets for subsequent analysis, e.g., to understand the association between cellular heterogeneity and disease progression (Darrah et al., 2007; Lin et al., 2015a; Seshadri et al., 2015; Corey et al., 2015).

One important structure of the CyTOF data is a natural chain-like dependence among groups of variables (e.g., Roederer et al. (2004); Perfetto et al. (2004)). Biologists utilize such inherent property of the data to manually identify cell subsets by sequentially visualizing data on the groups of variables, known as manual gating, which has been extensively used in real data analysis. Specifically, the manual gating is a manual sequential process that visually demarcates cells in bounded regions (called gates) on histogram or 2-D scatter plot projections. Interestingly, the modal clustering technique has been applied to assist manual gating for cytometric data by Ray and Pyne (2012). Figure 5 provides a simplified illustration of the manual gating analysis on one flow cytometry data. Red lines are the gates, and cells within the region defined by the gates are identified as a specific cell subset. For example, to discriminate CD4+ T cells, which is one major cell subset, a sequence of subsetting procedures is performed. Two physical markers, Forward (FSC-A) and side (SSC-A) light scatter, are first used to distinguish lymphocytes from all the live cells. Lymphocytes can then be further partitioned based on 3 fluorescence parameters: CD3, CD4, and CD8 cell-surface markers. CD4+ T cells are the subclass of lymphocytes having high values of CD3 and CD4 but low value of CD8. Within CD4+ T-cell populations, additional functional markers such as intracellular makers (IL2 and IFNg) can further distinguish many functionally different CD4+ T-cell subsets. The sequence of groups of markers to use is called *gating hierarchy*, which is determined by expert knowledge. The shape and location of the gates are manually drawn. The gates are typically used to dichotomize the continuous marker expressions into binary value: positive and negative.

In manual gating analysis, the variables are grouped and ordered based on expert knowledge. By using the well-established gating hierarchy that projects cells on lower dimensions, each subplot provides a finer resolution of the cellular heterogeneity. In the sequential visualization process, any move one step further means a new group of variables are examined. Despite its wide usage, the manual gating analysis is highly subjective, time consuming, and hard to reproduce.

In this section, we study the performance of HMM-VB on a data set obtained from CyTOF experiment (Becher et al., 2014). The particular data set that we analyze here is

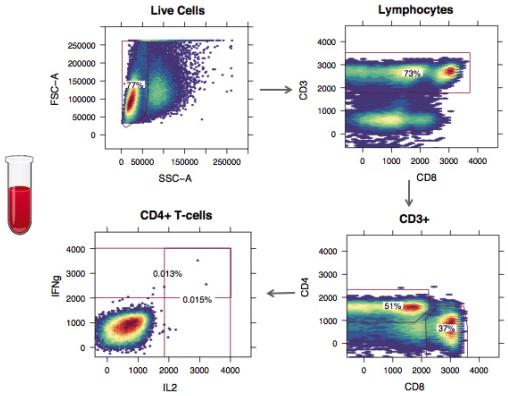


Figure 5: A simplified example of cell subsets identification by manual gating analysis. The flow cytometry measurements on single cells from a blood sample are shown using 4 heat maps of 2-D scatter plots projected on different dimensions (markers). The red lines on each subplots are called gates. Cells within the red lines are the subset of interest, which is subsetted and projected on the next subplot in the sequence. The percentages are the frequencies of the identified cell subsets relative to the total number of cells.

from mouse lung sample obtained from three C57BI6 wild-type mice and three Csf2rb$^{-/-}$ mice, which in total contains $46,204$ single cells with 39 measured cell markers. According to the gating hierarchy provided in Becher et al. (2014), it defines roughly 11 variable blocks, with maximum block size 8 and minimum block size 1. Becher et al. (2014) performed automated clustering on a projected (latent) $2-$dimensional space by first using nonlinear dimension reduction technique on the original $39-$dimensional space. However, it has been studied, e.g. Lin et al. (2015b), that dimension reduction generates a "cluttered" display, which can prevent density estimation from accurately representing low probability regions.

We first standardized the data to compensate for the nearly singular covariance matrix when a single Gaussian is fit despite the moderate dimension. This factor prevents the direct use of GMM. We note that HMM-VB encounters no difficulty in fitting the original data because of the smaller dimensions of the individual variable blocks. In order to compare modal GMM and HMM-VB, we work on the standardized data through the rest of the section.

We first fit the data using HMM-VB. There are 11 variable blocks. For the ith variable blocks with dimension lower than 4, we set the corresponding number of mixture components $M_i = 5$. For the jth variable blocks with dimension between 5 and 7, we set the number of components $M_j = 10$. For the other variable blocks, we set the number of components to 15. After modal clustering, this model results in 194 clusters of size greater than 5. The average

CPU time per model fitting is 11.4 min. We then fit the data using GMM with $M = 500$. On the same iMac, the model fitting takes 284.2 min, 25 times longer than does HMM-VB. Modal clustering results in 232 clusters of size greater than 5.

We compare the performance of the two models for clustering a relatively low probability region, which is shown in Figure 6. Specifically, Figure 6 compares the finer cellular compositions of one well separated region, which is visualized on two latent dimensions obtained from a dimension reduction technique applied to the original 39 dimensions, as provided by the result of Becher et al. (2014). HMM-VB (right subplot) uses 18 clusters to define this particular region. GMM (left subplot) uses 40 clusters to define the same region. However, some of the clusters include cells that are visually far away from the particular region. This result indicates that GMM has difficulty in estimating the structure of this moderately high-dimensional data.

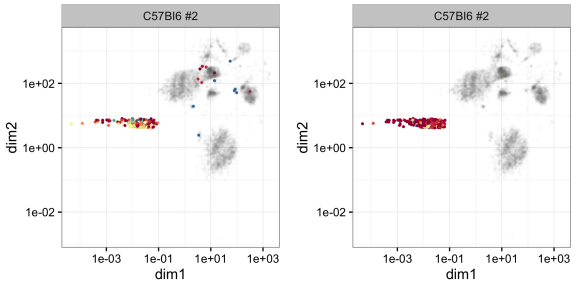


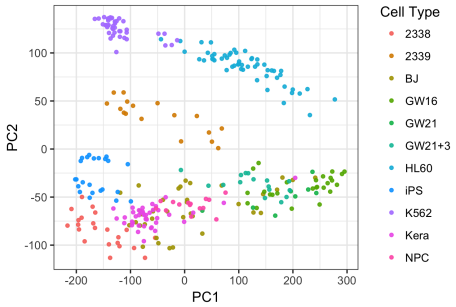
Figure 6: Left: GMM clustering analysis for a particular data region of one selected mouse. Right: The corresponding HMM-VB clustering analysis. The data is shown in grey. Clustering results are in different colors with one color defines one cluster.

5.3 Examples of Very High Dimensional Data

We apply HMM-VB to two data sets with dimension $d \gg n$.

5.3.1 Single-Cell Genomics Data

Current single-cell genomics technologies can routinely generate transcriptomics data at the single-cell level. Similar to the cytometry data, a key first analysis step is to identify cellular heterogeneity among single cells. The challenge is to cluster high dimensional data with substantially more variables (genes) than objects (cells). Existing clustering methods fall roughly into two categories. The first category is based largely on pairwise similarity measures among single cells (e.g., Guo et al. (2015); Wang et al. (2017)). The second category applies clustering, e.g., K-means and mixture models, in a reduced dimension space (typically 2 or 3 dimensions) obtained by principal component analysis or other nonlinear dimension reduction techniques (e.g., Satija et al. (2015); Žurauskienė and Yau (2016); Kiselev et al.



Cell type: size	HMM-VB
2338: 22	22(22)
2339: 17	17(17)
HL60: 54	54(54)
iPS: 23	23(23)
K562: 42	42(42)
NPC: 15	15(15)
BJ: 37	29(29); 9(8)
GW16: 26	28(15); 22(11)
GW21: 8	28(5); 22(3)
GW21+3: 16	28(8); 22(8)
Kera: 40	39(39); 9(1)

Figure 7: (Left): Visualization of the single cell RNA-seq data based on the first two principal components. Colors indicate the 11 cell types. (Right): Clustering performance for HMM-VB compared with the 11 cell types.

(2017)). Here, we study the performance of HMM-VB on a single-cell RNA-seq data set for 300 cells whose transcriptional measurements were taken across 8,686 genes. The data set is obtained from the Github repository: `https://github.com/JustinaZ/pcaReduce` provided by Žurauskienė and Yau (2016). The data were derived from 11 cell types: K562 – myeloid (chronic leukemia); HL60 – myeloid (acute leukemia); CRL-2339 – lymphoblastoid; iPS – pluripotent; CRL-2338 – epithelial; BJ – fibroblast (from human foreskin); Kera – foreskin keratinocyte; NPC – neural progenitor cells; GW(16, 21, 21+3) – fetal cortex at gestational week (16,21, 21+3 weeks). Figure 7 (Left) shows the data projected on the first two principal components for visualization. It is clear that the 11 cell types in the two-dimensional principal component space cannot be separated into distinct groups. Žurauskienė and Yau (2016) perform clustering analysis in the subspace spanned by the first 30 principal components, and demonstrate the superiority of pcaReduce in terms of clustering accuracy over some competing methods. However, pcaReduce can only correctly identify 4 cell types which are HL60, CRL-2339, CRL-2338, and NPC.

We first identified the top 500 highly variable genes based on their sample variances after applying log transformation. We then fit this reduced data of size 300×500 using HMM-VB. The genes are divided into 5 variable blocks of equal dimension, and each block is set to have 40 mixture components. We also tried other model specifications, and they yielded similar clustering results. HMM-VB identifies 12 cell clusters. More importantly, HMM-VB not only correctly identifies the 4 cell types that were also discovered by pcaReduce, it also finds correctly 2 other cell types: iPS and K562. Moreover, pcaReduce groups the three cell types GW(16, 21, 21+3) into one cluster. HMM-VB partitions these three cell types into two clusters, which could be visually supported by Figure 7 (Left). Figure 7 (Right) provides the details of the clustering performance.

Because the dimension is higher than the data size for this data set, we constrained the covariance matrix for each variable block to be diagonal. This constraint can be offset by having more components for a good fit of data. We found that when the dimension

of the variable block is at similar order of the data size, without any constraint, the estimated covariance matrix is sometimes singular or ill-conditioned even if a single Gaussian distribution is assumed. We also found that the number of modes in a HMM-VB can become quite large if the number of variable blocks is large, while the number of modes is much more stable with the increase of components in each variable block. In summary, as practical guidance on the usage of HMM-VB for very high dimensional data ($d \gg n$), we recommend the use of diagonal covariance and the avoidance of too many variable blocks.

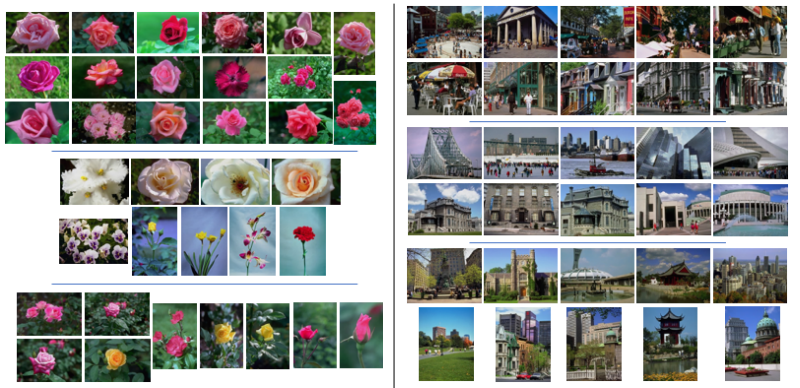
5.3.2 Image Data Analysis

We take a collection of general-purpose photograph images and represent each image by a high dimensional vector. First, the Red, Blue, and Green values of each pixel in an image are converted to the LUV color space. Second, each image is divided into 50×50 blocks and the average LUV values of all the pixels in each block are used to characterize this block. The whole image is represented by a stacked vector of the LUV values of the 2500 blocks. Thus the dimension of the data is $d = 7500$. We put the LUV values of every image block into one variable block. There are thus 2500 variable blocks each of dimension 3. These variable blocks are ordered according to a zig-zag row by row scan, that is, to scan from left to right in the first row, then right to left in the second, so on so forth. Clearly, because each variable block corresponds to a particular location in the image plane, the clustering result based on HMM-VB will be sensitive to the position of objects in an image. For many applications in multimedia retrieval or computer vision, such sensitivity is undesirable. However, our purpose here is not to promote our experiment as a standard for clustering images, but to show the viability of our method when $d \gg n$. We acknowledge that for any particular application, it may be necessary to perform certain pre-processing, such as extracting sub-images based on segmentation, etc. An in-depth evaluation from an image analysis perspective is beyond the scope of this paper.

We estimated the HMM-VB on the aforementioned variable block structure, assuming 5 mixture components for each block. Since the dimension of the variable blocks is small, we did not constrain the covariance matrices to be diagonal in this analysis. We estimated a model for a set of 200 photos of flowers, a set of 200 photos of city scenes, and a set of 2400 photos of various themes. On a Mac with Intel Core i5 3.5GHz/16GB Memory, the time for training is respectively 263, 326, 2118 seconds for the three data sets, and the time for solving the modes for all the images in a data set is respectively 118, 141, 1930 seconds. We found that due to the very high dimension, if we apply the same stringent criterion of identical modes, then nearly every image becomes a single cluster. We obtain larger clusters when the criterion is relaxed. We show in Figure 8 example clusters generated.

6. Discussion

We have developed a novel mixture model, namely, HMM-VB, with the goal of automated clustering of large-scale and high-dimensional data that contain rare clusters. One key feature of HMM-VB is its ability to leverage sequential dependence among groups of variables for more effective clustering, especially in identifying rare clusters that are almost undetectable by existing mixture modeling approaches. Technically, our clustering method integrates two new algorithms, one for estimating HMM-VB and the other for performing modal



(a) Flower (b) City scenes

Figure 8: Example images in the clusters generated for two data sets. The clusters are separated by the horizontal bars.

clustering, both necessary for making the new model a practical tool in large-scale data analysis. Because the number of mixture components grows exponentially with the number of variable groups, existing EM and MEM algorithms for estimation and mode identification have intractable exponential complexity. We have derived and implemented algorithms with linear complexity in the number of variable blocks for both tasks.

We have so far declared clusters based on association with identical modes. Due to the huge number of mixture components in the GMM casted from HMM-VB, many tiny clusters can be generated. The total number of points in those clusters can be a small fraction of the entire data. In this situation, more sophisticated methods that measure the separability of these tiny clusters and other larger ones can be used for clustering (e.g., Lee and Li (2012)). As another direction of future work, we can test the capability of the variable block search algorithm to reveal hidden dependence structure among the variables or to help validate existing gating hierarchy.

Acknowledgments

We would like to acknowledge support for this work from the National Science Foundation under grant ECCS-1462230 and DMS-1521092. We also thank the reviewers and the AE wholeheartedly for many detailed and constructive suggestions.

Appendix A

We present here the conventional HMM and the Baum-Welch (BW) algorithm for estimation. Consider sequential data $\mathbf{x} = \{x_1, x_2, ..., x_t, ..., x_T\}$, with $x_t \in \mathcal{R}^d$. Same as in the mixture model, assume there is an underlying state s_t associated with every x_t, for $t = 1, ..., T$. The underlying state is the counterpart of the mixture component identity in GMM. The state $s_t \in \mathcal{S} = \{1, 2, ..., M\}$, where M is the number of states. Let the set of all possible state sequences be $\bar{\mathcal{S}}$, that is, the set of T-tuples on $\mathcal{S}$. $|\bar{\mathcal{S}}| = M^T$. Denote $\mathbf{s} = \{s_1, ..., s_T\} \in \bar{\mathcal{S}}$. The basic assumptions of a HMM are:

1. The underlying states $\{s_1, s_2, ..., s_T\}$ follow a Markov chain. The Markov chain is usually time invariant with transition probability matrix $A = (a_{k,l})$, where $a_{k,l} = P(s_{t+1} = l \mid s_t = k)$, $k, l \in \mathcal{S}$. Denote the initial probabilities of states by $\pi_k = P(s_1 = k)$, $k \in \mathcal{S}$.

2. Given the hidden state s_t, the observation x_t is conditionally independent from $s_{t'}$ and $x_{t'}$ for any $t' \neq t$, and the distribution of x_t given s_t depends on s_t, but not on t. Denote the conditional density of $P(x_t|s_t = k)$ by $\varphi_k(x_t)$. In particular, for Gaussian HMM, $\varphi_k(x_t) = \phi(x_t|\mu_k, \Sigma_k)$.

In summary,

$$P(\mathbf{x}, \mathbf{s}) = P(\mathbf{s})P(\mathbf{x} \mid \mathbf{s}) = \pi_{s_1}\varphi_{s_1}(x_1)a_{s_1,s_2}\varphi_{s_2}(x_2)\cdots a_{s_{T-1},s_T}\varphi_{s_T}(x_T),$$
$$P(\mathbf{x}) = \sum_{\mathbf{s}\in\bar{\mathcal{S}}} P(\mathbf{s})P(\mathbf{x} \mid \mathbf{s}) = \sum_{\mathbf{s}\in\bar{\mathcal{S}}} \pi_{s_1}\varphi_{s_1}(x_1)a_{s_1,s_2}\varphi_{s_2}(x_2)\cdots a_{s_{T-1},s_T}\varphi_{s_T}(x_T).$$

The parameters to be estimated in a HMM are the transition probabilities: $a_{k,l}$, $k, l = 1, ..., M$, the initial probabilities: π_k, $k = 1, ..., M$, and μ_k, Σ_k for each state $k = 1, ..., M$.

Under a set of parameters, let $L_k(t)$ be the conditional probability of being in state k at position t given the entire observed sequence $\mathbf{x} = \{x_1, x_2, ..., x_T\}$. Let $I(\cdot)$ be the indicator function that equals 1 when the argument is true and 0 otherwise. Then

$$L_k(t) = P(s_t = k|\mathbf{x}) = \sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{x})I(s_t = k), \quad k \in \mathcal{S}. \tag{8}$$

Let $H_{k,l}(t)$ be the conditional probability of being in state k at position t and being in state l at position $t+1$, i.e., seeing a transition from k to l at t, given the entire observed sequence $\mathbf{x}$. Then

$$\begin{aligned} H_{k,l}(t) &= P(s_t = k, s_{t+1} = l|\mathbf{x}) \\ &= \sum_{\mathbf{s}} P(\mathbf{s} \mid \mathbf{x})I(s_t = k)I(s_{t+1} = l), \quad k, l \in \mathcal{S}. \end{aligned} \tag{9}$$

Note that $L_k(t) = \sum_{l=1}^M H_{k,l}(t)$, $\sum_{k=1}^M L_k(t) = 1$. Since $\mathbf{s} \in \bar{\mathcal{S}}$ and the $|\bar{\mathcal{S}}| = M^T$, it is infeasible to compute $L_k(t)$ and $H_{k,l}(t)$ by the above equations directly. As part of the BW algorithm, the *forward-backward* algorithm is used to compute $L_k(t)$ and $H_{k,l}(t)$ efficiently. The amount of computation needed is at the order of M^2T; and the memory required is at the order of MT.

Define the forward probability $\alpha_k(t)$ as the joint probability of observing the first t vectors x_τ, $\tau = 1, ..., t$, and being in state k at time t:

$$\alpha_k(t) = P(x_1, x_2, ..., x_t, s_t = k).$$

This probability can be evaluated by the following recursive formula:

$$\begin{aligned} \alpha_k(1) &= \pi_k\varphi_k(x_1), \quad 1 \le k \le M, \\ \alpha_k(t) &= \varphi_k(x_t)\sum_{l=1}^M \alpha_l(t-1)a_{l,k}, \quad 1 < t \le T,\ 1 \le k \le M. \end{aligned}$$

Define the backward probability $\beta_k(t)$ as the conditional probability of observing the vectors after time t, x_τ, $\tau = t+1, ..., T$, given the state at time t is k.

$$\begin{aligned} \beta_k(t) &= P(x_{t+1}, ..., x_T \mid s_t = k), \quad 1 \le t \le T-1, \\ \beta_k(T) &= 1, \quad \text{for all } k. \end{aligned}$$

As with the forward probability, the backward probability can be evaluated using the following recursion:

$$\begin{aligned} \beta_k(T) &= 1, \\ \beta_k(t) &= \sum_{l=1}^M a_{k,l}\varphi_l(x_{t+1})\beta_l(t+1), \quad 1 \le t < T. \end{aligned}$$

The probabilities $L_k(t)$ and $H_{k,l}(t)$ are solved by

$$L_k(t) = P(s_t = k \mid \mathbf{x}) = \frac{P(\mathbf{x}, s_t = k)}{P(\mathbf{x})} = \frac{1}{P(\mathbf{x})}\alpha_k(t)\beta_k(t),$$

$$\begin{aligned} H_{k,l}(t) &= P(s_t = k, s_{t+1} = l \mid \mathbf{x}) = \frac{P(\mathbf{x}, s_t = k, s_{t+1} = l)}{P(\mathbf{x})} \\ &= \frac{1}{P(\mathbf{x})}\alpha_k(t)a_{k,l}\varphi_l(x_{t+1})\beta_l(t+1), \end{aligned}$$

where $P(\mathbf{x}) = \sum_{k=1}^M \alpha_k(t)\beta_k(t)$.

For notational brevity, we assume all the sequences are of length T. The extension to sequences of different lengths is trivial. Denote the ith sequence by $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, ..., x_{i,T}\}$, $i = 1, ..., n$. In each iteration, we compute the forward and backward probabilities for each sequence separately as previously described. We also compute $L_k(t)$ and $H_{k,l}(t)$ separately for each sequence. As a general pattern of notations, we put a superscript (i) to indicate the quantities for the ith sequence.

1. Compute the forward and backward probabilities $\alpha_k^{(i)}(t)$, $\beta_k^{(i)}(t)$, $k = 1, ..., M$, $t = 1, ..., T$, $i = 1, ..., n$, under the current set of parameters.

$$\begin{aligned}
\alpha_k^{(i)}(1) &= \pi_k \varphi_k(x_{i,1}), \quad 1 \le k \le M,\ 1 \le i \le n, \\
\alpha_k^{(i)}(t) &= \varphi_k(x_{i,t}) \sum_{l=1}^{M} \alpha_l^{(i)}(t-1) a_{l,k}, \quad 1 < t \le T,\ 1 \le k \le M,\ 1 \le i \le n, \\
\beta_k^{(i)}(T) &= 1, \quad 1 \le k \le M,\ 1 \le i \le n, \\
\beta_k^{(i)}(t) &= \sum_{l=1}^{M} a_{k,l} \varphi_l(x_{i,t+1}) \beta_l^{(i)}(t+1), \quad 1 \le t < T,\ 1 \le k \le M,\ 1 \le i \le n.
\end{aligned}$$

2. Compute $L_k^{(i)}(t)$, $H_{k,l}^{(i)}(t)$ using $\alpha_k^{(i)}(t)$, $\beta_k^{(i)}(t)$. Let $P(\mathbf{x}_i) = \sum_{k=1}^{M} \alpha_k^{(i)}(1) \beta_k^{(i)}(1)$.

$$\begin{aligned}
L_k^{(i)}(t) &= \frac{1}{P(\mathbf{x}_i)} \alpha_k^{(i)}(t) \beta_k^{(i)}(t), \\
H_{k,l}^{(i)}(t) &= \frac{1}{P(\mathbf{x}_i)} \alpha_k^{(i)}(t) a_{k,l} \varphi_l(x_{i,t+1}) \beta_l^{(i)}(t+1).
\end{aligned}$$

3. Update the parameters using $L_k^{(i)}(t)$, $H_{k,l}^{(i)}(t)$.

$$\begin{aligned}
\mu_k &= \frac{\sum_{i=1}^{n} \sum_{t=1}^{T} L_k^{(i)}(t) x_{i,t}}{\sum_{i=1}^{n} \sum_{t=1}^{T} L_k^{(i)}(t)}, \\
\Sigma_k &= \frac{\sum_{i=1}^{n} \sum_{t=1}^{T} L_k^{(i)}(t) (x_{i,t} - \mu_k)(x_{i,t} - \mu_k)'}{\sum_{i=1}^{n} \sum_{t=1}^{T} L_k^{(i)}(t)}, \\
a_{k,l} &= \frac{\sum_{i=1}^{n} \sum_{t=1}^{T-1} H_{k,l}^{(i)}(t)}{\sum_{i=1}^{n} \sum_{t=1}^{T-1} L_k^{(i)}(t)}.
\end{aligned}$$

Appendix B

We prove the results in Section 4.1.

The proof for Theorem 2

Proof Because $\mathcal{M}$ and $\mathcal{M}'$ define the same density function, their marginal densities on any block $X^{(t)}$ are also identical. Since $\Theta^{(t)}$ is non-redundant for any t, by Lemma 1, we have $\mathcal{M}'_{X^{(t)}} = \mathcal{O}_t(\mathcal{M}_{X^{(t)}})$, where $\mathcal{O}_t$ is the permutation on the mixture components of $X^{(t)}$, for $t = 1, ..., T$. The permutation $\mathcal{O}_t$ is unique because $\Theta^{(t)}$ is distinct.

Again by Lemma 1, if we remove all the components in $\mathcal{M}$ and $\mathcal{M}'$ with zero priors, that is, $P(s_1 = i_1, s_2 = i_2, ..., s_t = i_T) = 0$ and $P(s'_1 = i'_1, s'_2 = i'_2, ..., s'_T = i'_T) = 0$, the two models have the same number of components, and the Gaussian component parameters as well as the positive priors in $\mathcal{M}$ and $\mathcal{M}'$ are identical up to permutation. Let the permutation

that matches the two models be $\mathcal{O}^*$. Note that $\mathcal{O}^*$ is only defined on T-tuples $(i_1, ..., i_T)$ such that $P(s_1 = i_1, s_2 = i_2, ..., s_t = i_T) > 0$.

Let $\mathcal{O}_{1:T} = \mathcal{O}_1 \times \mathcal{O}_2 \cdots \mathcal{O}_T$. It is obvious that $\Theta' = \mathcal{O}_{1:T}(\Theta)$. Consider a T-tuple with $P(s_1 = i_1, ..., s_T = i_T) > 0$. Let $(i_1^*, ..., i_T^*) = \mathcal{O}^*(i_1, ..., i_T)$. By Lemma 1,

$$\mu_{i_t}^{(t)} = \mu'^{(t)}_{i_t^*}, \quad \Sigma_{i_t}^{(t)} = \Sigma'^{(t)}_{i_t^*},$$

$$\mu_{i_t}^{(t)} = \mu'^{(t)}_{\mathcal{O}_t(i_t)}, \quad \Sigma_{i_t}^{(t)} = \Sigma'^{(t)}_{\mathcal{O}_t(i_t)}.$$

Because $\Theta'^{(t)}$ is distinct, $i_t^* = \mathcal{O}_t(i_t)$, for $t = 1, ..., T$. Thus we have shown that if $P(s_1 = i_1, ..., s_T = i_T) > 0$, $\mathcal{O}^*(i_1, ..., i_T) = \mathcal{O}_{1:T}(i_1, ..., i_T)$. Because $\mathcal{M}$ and $\mathcal{M}'$ have the same number of components with positive priors, if $P(i_1, ..., i_T) = 0$, we can extend the definition of $\mathcal{O}^*$ to these T-tuples by simply setting it to $\mathcal{O}_{1:T}$. Thus we have shown $\mathcal{M}' = \mathcal{O}_{1:T}(\mathcal{M})$. ■

The proof for Lemma 3

Proof Because $\Theta^{(t)}$ is non-redundant for $t = 1, ..., T$, and $\pi_{i_1} = P(s_1 = i_1)$, we have $\pi_{i_1} > 0$, $\forall i_1 \in \{1, ..., M_1\}$. We also have $P(s_t = i_t) > 0$, $\forall i_t \in \{1, ..., M_t\}$, $t = 2, ..., T$. Since $P(s_t = i_t) = \sum_{i_{t-1}=1}^{M_{t-1}} P(s_{t-1} = i_{t-1}) a_{i_{t-1}, i_t}^{(t-1)}$ and $P(s_t = i_t) > 0$, there exist at least one i_{t-1} such that $a_{i_{t-1}, i_t}^{(t-1)} > 0$.

The reverse can be proved by induction. Because $\pi_{i_1} = P(s_1 = i_1)$ and $\pi_{i_1} > 0$, we have $P(s_1 = i_1) > 0$, $\forall i_1 \in \{1, ..., M_1\}$. Assume $P(s_{t-1} = i_{t-1}) > 0$, $\forall i_{t-1} \in \{1, ..., M_{t-1}\}$, then $P(s_t = i_t) = \sum_{i_{t-1}=1}^{M_{t-1}} P(s_{t-1} = i_{t-1}) a_{i_{t-1}, i_t}^{(t-1)} > 0$ because at least for one i_{t-1}, the summand $P(s_{t-1} = i_{t-1}) a_{i_{t-1}, i_t}^{(t-1)} > 0$. ■

The proof for Corollary 4

Proof Because HMM-VB on partition $\mathcal{P}$ is a lattice GMM on the same partition, by Theorem 2, there exists a unique permutation $\mathcal{O}_t$ for each variable block $X^{(t)}$, $t = 1, ..., T$ such that $\mathbb{M}' = \mathcal{O}_{1:T}(\mathbb{M})$ and $\Theta' = \mathcal{O}_{1:T}(\Theta)$.

We only need to show that under permutation $\mathcal{O}_{1:T}$, π_{i_1}, $a_{i_{t-1}, i_t}^{(t-1)}$, $i_1 = 1, ..., M_1$, $i_t = 1, ..., M_t$, $t = 2, ..., T$ are also identical. By Theorem 2, $\Pi(s_1, ..., s_T)$ and $\Pi'(s'_1, ..., s'_T)$ are identical up to permutation $\mathcal{O}_{1:T}$. The same is true for any marginal $\Pi(s_t)$ or $\Pi(s_{t-1}, s_t)$. That is

$$P(s_t = i_t) = P(s'_t = \mathcal{O}_t(i_t)), \ \ \forall t = 1, ..., T, \ \forall i_t = 1, ..., M_t.$$

Since π_{i_1}, $i_1 = 1, ..., M_1$, is the marginal $\Pi(s_1)$, it is identical to π'_{i_1} up to permutation $\mathcal{O}_1$.

Under permutation $\mathcal{O}_{t-1:t}$, we have

$$P(s'_{t-1} = \mathcal{O}_{t-1}(i_{t-1}), s'_t = \mathcal{O}_t(i_t)) = P(s_{t-1} = i_{t-1}, s_t = i_t).$$

Also,

$$P(s'_{t-1} = \mathcal{O}_{t-1}(i_{t-1}), s'_t = \mathcal{O}_t(i_t)) = P(s'_{t-1} = \mathcal{O}_{t-1}(i_{t-1})) a'^{(t-1)}_{\mathcal{O}_{t-1}(i_{t-1}), \mathcal{O}_t(i_t)},$$

$$P(s_{t-1}=i_{t-1}, s_t=i_t) = P(s_{t-1}=i_{t-1}) a^{(t-1)}_{i_{t-1},i_t}.$$

Because Θ and Θ' are non-redundant, $P(s_{t-1}=i_{t-1})>0$, $\forall i_{t-1}$, and similarly $P(s'_{t-1}=i'_{t-1})>0$, $\forall i'_{t-1}$, we thus have

$$a'^{(t-1)}_{\mathcal{O}_{t-1}(i_{t-1}),\mathcal{O}_t(i_t)} = a^{(t-1)}(i_{t-1}, i_t),$$
$$\forall t = 2, ..., T,\ \forall i_{t-1} = 1, ..., M_{t-1},\ \forall i_t = 1, ..., M_t.$$

■

The proof for Theorem 5

Proof If all the variables are put in one block, we can remove any component with zero prior. Hence, the partition containing only one block is always a tight partition. Thus a tight partition always exists. Because there are only finitely many partitions, the maximum partition must exist. We have proved the existence of a maximum partition for any GMM.

We now prove the uniqueness. Suppose $\mathcal{P}_1$ and $\mathcal{P}_2$ are both maximum partitions and $\mathcal{P}_1 \neq \mathcal{P}_2$. Let $\mathcal{P}_i = \{X^{(i,1)}, X^{(i,2)}, ..., X^{(i,T_i)}\}$, $i = 1, 2$. Let the kth Gaussian component parameter for block $X^{(i,j)}$ in partition $\mathcal{P}_i$ be $\theta_k^{(i,j)}$. For any sub-vector of $X^{(i,j)}$ that contains some of the variables in $X^{(i,j)}$, which for instance is denoted by $X^{(i,j,l)}$, we denote the projection of $\theta_k^{(i,j)}$ on the sub-vector by $\theta_k^{(i,j)}(X^{(i,j,l)})$. Specifically, the mean vector projected on $X^{(i,j,l)}$ is a sub-vector of the mean vector $\mu_k^{(i,j)}$ keeping the corresponding dimensions in $X^{(i,j,l)}$, and the covariance matrix projected on $X^{(i,j,l)}$ is a sub-matrix of the covariance $\Sigma_k^{(i,j)}$ keeping the entries for the covariances between the dimensions in $X^{(i,j,l)}$.

By definition of maximum partition, $\mathcal{P}_1 \not\succ \mathcal{P}_2$ and vice versa $\mathcal{P}_1 \not\prec \mathcal{P}_2$. Therefore there exists a $X^{(1,j)}$ such that the variables in this block do not belong to a single block in $\mathcal{P}_2$. Without loss of generality and for brevity of notation, assume $X^{(1,1)}$ is such a block and its variables fall into K blocks in $\mathcal{P}_2$. We divide block $X^{(1,1)}$ into K sub-blocks such that each sub-block belongs to a distinct block in $\mathcal{P}_2$. Denote the K sub-blocks by $X^{(1,1,k)}$. $X^{(1,1)} = \{X^{(1,1,1)}, X^{(1,1,2)}, ..., X^{(1,1,K)}\}$. Again without loss of generality assume sub-block $X^{(1,1,k)} \subset X^{(2,k)}$, $k = 1, ..., K$.

Suppose the number of components for each block in $\mathcal{P}_1$ is M_1, ..., M_{T_1}, and the number of components for each block in $\mathcal{P}_2$ is M'_1, ..., M'_{T_2}. Without loss of generality, we can assume the lattice Θ and Θ' corresponding to $\mathcal{P}_1$ and $\mathcal{P}_2$ are non-redundant. The fact all the component parameters in Θ (or Θ') exist is guaranteed already by $\mathcal{P}_1$ (or $\mathcal{P}_2$) being a tight partition. If any $\Theta^{(t)}$ is not distinct (containing identical components), we can always combine those components without affecting the partition $\mathcal{P}_1$. Therefore we can assume Θ and Θ' are non-redundant without the loss of generality.

Let the underlying component identity be s_j, $j = 1, ..., T_1$ for $\mathcal{P}_1$ and s'_j, $j = 1, ..., T_2$ for $\mathcal{P}_2$. Because the GMM is a lattice-GMM on both $\mathcal{P}_1$ and $\mathcal{P}_2$, the marginal density of variable block $X^{(1,1)}$ according to partition $\mathcal{P}_1$ and $\mathcal{P}_2$ respectively is given by

$$f_{X^{(1,1)}}(x^{(1,1)}) = \sum_{i_1=1}^{M_1} P(s_1 = i_1) \phi(x^{(1,1)} \mid \theta_{i_1}^{(1,1)}), \tag{10}$$

$$f_{X^{(1,1)}}(x^{(1,1)}) = \sum_{i'_1=1}^{M'_1} \cdots \sum_{i'_K=1}^{M'_K} P(s'_1 = i'_1, ..., s'_K = i'_K) \prod_{j=1}^{K} \phi(x^{(1,1,j)} \mid \theta_{i'_j}^{(2,j)}(X^{(1,1,j)})). \tag{11}$$

Because both $\mathcal{P}_1$ and $\mathcal{P}_2$ are tight partitions, the priors in Eqs. (10) and (11) are all positive. The components in mixture (10) are distinct by the non-redundancy of Θ, while the components in (11) may not because $X^{(1,1,j)}$'s are sub-vectors of the blocks in $\mathcal{P}_2$. However, if they are not distinct, we can combine components without losing the positive priors on the components. To be strict, suppose we combined components and mixture model (11) becomes the following model containing distinct components:

$$f_{X^{(1,1)}}(x^{(1,1)}) = \sum_{i''_1=1}^{M''_1} \cdots \sum_{i''_K=1}^{M''_K} P(s''_1 = i''_1, ..., s''_K = i''_K) \prod_{j=1}^{K} \phi(x^{(1,1,j)} \mid \theta_{i''_j}^{(2,j)}(X^{(1,1,j)})). \tag{12}$$

By the identifiability of GMM (Lemma 1), the number of terms in Eq. (12), $M''_1 \times M''_2 \cdots \times M''_K = M_1$ and for each i_1, there is a unique K-tuple $(i''_1, i''_2, ..., i''_K)$ such that

$$\phi(x^{(1,1)} \mid \theta_{i_1}^{(1,1)}) = \prod_{j=1}^{K} \phi(x^{(1,1,j)} \mid \theta_{i''_j}^{(2,j)}(X^{(1,1,j)})). \tag{13}$$

The density of the full-dimensional X according to $\mathcal{P}_1$ is

$$f_{X^{(1,1)}}(x^{(1,1)}) = \sum_{i_1=1}^{M_1} \cdots \sum_{i_T=1}^{M_T} \pi(i_1, i_2, ..., i_T) \prod_{j=1}^{T} \phi(x^{(1,j)} \mid \theta_{i_j}^{(1,j)}).$$

Because the map from i_1 to $(i''_1, i''_2, ..., i''_K)$ is bijective, we can define a set of prior

$$\tilde{\pi}(i''_1, i''_2, ..., i''_K, i_2, i_3, ..., i_T) = \pi(i_1, i_2, ..., i_T).$$

By Eq. (13), we can write

$$\begin{aligned} f_X(x) \quad = \quad & \sum_{i''_1=1}^{M''_1} \cdots \sum_{i''_K=1}^{M''_K} \sum_{i_2=1}^{M_2} \cdots \sum_{i_T=1}^{M_T} \tilde{\pi}(i''_1, ..., i''_K, i_2, ..., i_T) \cdot \\ & \prod_{l=1}^{K} \phi(x^{(1,1,l)} \mid \theta_{i''_l}^{(2,l)}(X^{(1,1,l)})) \prod_{j=2}^{T} \phi(x^{(1,j)} \mid \theta_{i_j}^{(1,j)}). \end{aligned} \tag{14}$$

Eq. (14) shows that the GMM is a lattice-GMM on the partition

$$\mathcal{P}_3 = \{X^{(1,1,1)}, X^{(1,1,2)}, \cdots, X^{(1,1,K)}, X^{(1,2)}, ..., X^{(1,T)}\},$$

and $\mathcal{P}_3$ is a tight partition. Clearly, $\mathcal{P}_3 \succ \mathcal{P}_1$ and $\mathcal{P}_3 \neq \mathcal{P}_1$. This contradicts the assumption that $\mathcal{P}_1$ is a maximum partition. We have thus proved the maximum partition is unique.

■

Appendix C

In this section, we prove Theorem 6.

Proof We follow the idea of the proof of Theorem 2 in Maugis et al. (2009). According to the definition, $\hat{\mathbf{C}} = \text{argmin}_{\mathbf{C}\in\mathcal{C}} BIC(\mathbf{C})$, where $\mathcal{C}$ is the set of all possible partitions under $T = T^0$,

$$BIC(\mathbf{C}) = -2\sum_{i=1}^{n} \log(f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})) + \lambda(\mathbf{C})\log(n),$$

where $\lambda(\mathbf{C})$ is the number of model parameters for variable blocks $\mathbf{C}$. Thus

$$\begin{aligned} P(\hat{\mathbf{C}} = \mathbf{C}^0) &= P(BIC(\mathbf{C}^0) \leq BIC(\mathbf{C}), \forall \mathbf{C} \in \mathcal{C}) \\ &= P(BIC(\mathbf{C}) - BIC(\mathbf{C}^0) \geq 0, \forall \mathbf{C} \in \mathcal{C}). \end{aligned}$$

Denote by $\nu(\mathbf{C}) = \lambda(\mathbf{C}) - \lambda(\mathbf{C}^0)$ and $\Delta BIC(\mathbf{C}) = BIC(\mathbf{C}) - BIC(\mathbf{C}^0)$, then

$$\Delta BIC(\mathbf{C}) = 2n\{\frac{1}{n}\sum_{i=1}^{n} \log[\frac{f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C}^0)})}{g(\mathbf{x}_i)}] - \frac{1}{n}\sum_{i=1}^{n}[\frac{f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})}{g(\mathbf{x}_i)}]\} + \nu(\mathbf{C})\log(n). \tag{15}$$

All the possible partition $\mathcal{C}$ can be decomposed as follows

$$\mathcal{C} = \mathbf{C}^0 \cup \{\mathbf{C} \in \mathcal{C}; D_{KL}(g||f(.|\gamma^*_{(\mathbf{C})})) \neq 0\}.$$

Let $\mathcal{C}_1 = \{\mathbf{C} \in \mathcal{C}; D_{KL}(g||f(.|\gamma^*_{\mathbf{C}})) \neq 0\}$. We only need to prove that

$$\forall \mathbf{C} \in \mathcal{C}_1,\ P(\Delta BIC(\mathbf{C}) < 0) \xrightarrow[n\to\infty]{} 0.$$

Let $\mathcal{Q}_n(\mathbf{C}) = \frac{1}{n}\sum_{i=1}^{n} \log[\frac{f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})}{g(\mathbf{x}_i)}]$, and $\mathcal{Q}(\mathbf{C}) = -D_{KL}(g||f(.|\gamma^*_{(\mathbf{C})}))$,
Following Eq. (15), we then have

$$\begin{aligned} &P(\Delta BIC(\mathbf{C}) < 0) = P(2n[\mathcal{Q}_n(\mathbf{C}^0) - \mathcal{Q}_n(\mathbf{C})] + \nu(\mathbf{C})\log(n) < 0) \\ =&P(\mathcal{Q}_n(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}^0) + \mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) + \mathcal{Q}(\mathbf{C}) - \mathcal{Q}_n(\mathbf{C}) + \frac{\nu(\mathbf{C})\log(n)}{2n} < 0). \end{aligned}$$

Thus, for all $\epsilon > 0$, and according to the Lemma 8 below,

$$\begin{aligned} P(\Delta BIC(\mathbf{C}) < 0) \leq& P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}_n(\mathbf{C}^0) > \epsilon) + P(\mathcal{Q}_n(\mathbf{C}) - \mathcal{Q}(\mathbf{C}) > \epsilon) + \\ & P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) + \frac{\nu(\mathbf{C})\log(n)}{2n} < 2\epsilon). \end{aligned}$$

From Proposition 1 below, $\forall \mathbf{C}, \mathcal{Q}_n(\mathcal{C}) \xrightarrow[n\to\infty]{P} \mathcal{Q}(\mathcal{C})$. Thus,

$$\forall \epsilon > 0, P(\mathcal{Q}_n(\mathbf{C}) - \mathcal{Q}(\mathbf{C}) > \epsilon) \leq P(|\mathcal{Q}_n(\mathbf{C}) - \mathcal{Q}(\mathbf{C})| > \epsilon) \xrightarrow[n\to\infty]{} 0.$$

Then,

$$P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) + \frac{\nu(\mathbf{C})\log(n)}{2n} < 2\epsilon) \leq P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) - 2\epsilon < |\frac{\nu(\mathbf{C})\log(n)}{2n}|).$$

We know that $\frac{\nu(\mathbf{C})\log(n)}{2n} \xrightarrow[n\to\infty]{} 0$ and $\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) > 0$ because $\mathbf{C} \in \mathcal{C}_1$. Taking $\epsilon = \frac{\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C})}{4} > 0$, we get

$$P(\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C}) + \frac{\nu(\mathbf{C})\log(n)}{2n} < 2\epsilon) \leq P(\frac{\mathcal{Q}(\mathbf{C}^0) - \mathcal{Q}(\mathbf{C})}{2} < |\frac{\nu(\mathbf{C})\log(n)}{2n}|) \xrightarrow[n\to\infty]{} 0.$$

Finally, $P(\Delta BIC(\mathbf{C}) < 0) \xrightarrow[n\to\infty]{} 0$. ■

Lemma 8 *Let A and B be two real random variables,*

$$\forall \epsilon \in \mathcal{R}, P(A + B \leq 0) \leq P(A \leq \epsilon) + P(-B > \epsilon).$$

Proposition 1 *Under Assumptions* **A1** *and* **A2**, $\forall \mathbf{C} \in \mathcal{C}$,

$$\frac{1}{n}\sum_{i=1}^{n} \log\{\frac{g(\mathbf{x}_i)}{f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})}\} \xrightarrow[n\to\infty]{P} D_{KL}(g||f(.|\gamma^*_{(\mathbf{C})})).$$

Proof By the law of large numbers, if $E[|\log(g(X))|] < \infty$,

$$\frac{1}{n}\sum_{i=1}^{n} \log[g(\mathbf{x}_i)] \xrightarrow[n\to\infty]{P} E_X[\log(g(X))]. \tag{16}$$

If we can show that for the family

$$\mathcal{G}_{(\mathbf{C})} := \{\log[f(.|\gamma)]; \gamma \in \Gamma'_{(\mathbf{C})}\},$$

the following holds

$$\frac{1}{n}\sum_{i=1}^{n} \log[f(\mathbf{x}_i|\hat{\gamma}_{(\mathbf{C})})] \xrightarrow[n\to\infty]{P} E_X[\log f(X|\gamma^*{}_{(\mathbf{C})})], \tag{17}$$

then Proposition 1 is proved by combining Eqs. (16) and (17).

Now we only need to prove that $E_X[\log f(X)] < \infty$. First, by the Assumption **A1**, $\Gamma'_{(\mathbf{C})}$ is a compact metric space. And for all $\mathbf{x} \in \mathcal{R}^p$, $\gamma_{(\mathbf{C})} \in \Gamma'_{(\mathbf{C})} \to \log[f(\mathbf{x}|\gamma_{(\mathbf{C})})]$ is continuous. We just need to verify that there is an envelope function G of $\mathcal{G}_{(\mathbf{C})}$ being g-integrable.

Denote by $\mathcal{M} = [M_2^0] \times [M_3^0] \times \cdots \times [M_T^0] = \{1, 2, ..., M_2^0\} \times \{1, 2, ..., M_3^0\} \times \cdots \times \{1, 2, ..., M_T^0\}$. We first write out $\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})]$ explicitly:

$$\begin{aligned}
&\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})] = \log\Big[\sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}\Pi_{j=1}^T N(x^{(C_j)}|\mu_{k_j}^{(j)},\Sigma_{k_j}^{(j)})\Big]\\
&=\log\Big[\sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}\\
&\quad\Pi_{j=1}^T(2\pi)^{-|C_j|/2}|\Sigma_{k_j}^{(j)}|^{-1/2}\exp\{-\frac{(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)})}{2}\}\\
&\le\ \log\Big[\sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}(2\pi a)^{-\frac{d}{2}}\Big]\\
&\le\ -\frac{d}{2}\log[2\pi a],
\end{aligned}$$

where $x^{(C_j)}$ being the C_j-th variable block. The inequality is obtained according to Lemma 9 and $(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)}) \le 0$, for $j = 1, 2, ..., T$, hence is exponential function bounded by 1.

Now we need to obtain a lower bound, and we use the concavity of the logarithm function:

$$\begin{aligned}
&\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})] \ge \sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}\\
&\qquad\Pi_{j=1}^T(2\pi)^{-|C_j|/2}|\Sigma_{k_j}^{(j)}|^{-1/2}\exp\{-\frac{(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)})}{2}\}\\
&=-\frac{d}{2}\log(2\pi)-\frac{1}{2}\sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}\Big\{\sum_{j=1}^T\log[|\Sigma_{k_j}^{(j)}|]\\
&\quad+\sum_{j=1}^T(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)})\Big\}.
\end{aligned}$$

According to Lemma 9 and $\forall k_j$, $\mu_{k_j}^{(j)} \in \mathcal{B}(\eta, |C_j|)$

$$\begin{aligned}
(x^{(C_j)}-\mu_{k_j}^{(j)})'{\Sigma_{k_j}^{(j)}}^{-1}(x^{(C_j)}-\mu_{k_j}^{(j)}) &\le \frac{||x^{(C_j)}-\mu_{k_j}^{(j)}||^2}{a}\\
&\le \frac{2(||x^{(C_j)}||^2+||\mu_{k_j}^{(j)}||^2)}{a}\\
&\le \frac{2(||x^{(C_j)}||^2+\eta^2)}{a}.
\end{aligned}$$

Hence,

$$\begin{aligned}
\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})] &\ge -\frac{d}{2}\log(2\pi)-\frac{1}{2}\sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}\\
&\quad\Big\{\log(b^d)+\sum_{j=1}^T\frac{2}{a}(||x^{(C_j)}||^2+\eta^2)\Big\}\\
&=-\frac{d}{2}\log(2\pi b)-\frac{||\mathbf{x}||^2+2\eta}{a}.
\end{aligned}$$

Hence, each function of the family $\mathcal{G}_{(\mathbf{C})}$ is bounded by

$$-\frac{d}{2}\log(2\pi b)-\frac{||\mathbf{x}||^2+2\eta}{a} \le \log[f(\mathbf{x}|\gamma_{(\mathbf{C})})] \le \frac{d}{2}\log[2\pi a].$$

Therefore, for all $\gamma_{(\mathbf{C})} \in \Gamma'_{(\mathbf{C})}$ and all $\mathbf{x} \in \mathcal{R}^p$,

$$|\log[f(\mathbf{x}|\gamma_{(\mathbf{C})})]| \le Z_1(a,b,\eta)+Z_2(\eta,a)||\mathbf{x}||^2$$

defining the envelope function G, where $Z_1(a,b,\eta)$ and $Z_2(\eta,a)$ are two positive constants.

In order to verify that G is g-integrable, we need to show that $\int||\mathbf{x}||^2 g(\mathbf{x})d\mathbf{x} < \infty$.

$$\begin{aligned}
&\int||\mathbf{x}||^2 g(\mathbf{x})d\mathbf{x} = \int||\mathbf{x}||^2 f(\mathbf{x}|\gamma^*_{(\mathbf{C}^0)})d\mathbf{x}\\
&=\sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}\int||\mathbf{x}||^2\phi(\mathbf{x}|\mu_{(k_1,k_2,...,k_T)},\Sigma_{(k_1,k_2,...,k_T)})d\mathbf{x}\\
&\le\sum_{k_1=1}^{M_1^0}\pi_{k_1}\sum_{(k_2,k_3,...,k_T)\in\mathcal{M}} a_{k_1,k_2}a_{k_2,k_3}\cdots a_{k_{T-1},k_T}(2\sum_{j=1}^T||\mu_{k_j}^{(j)}||^2+2\sum_{j=1}^T tr(\Sigma_{k_j}^{(j)})),
\end{aligned}$$

where $\mu_{(k_1,k_2,...,k_T)} = (\mu_{k_1}^{(1)}, \mu_{k_2}^{(2)}, ..., \mu_{k_T}^{(T)})$ and $\Sigma_{(k_1,k_2,...,k_T)}$ is block diagonal with the t-th diagonal block being $\Sigma_{k_t}^{(t)}$, for $t = 1, 2, ..., T$. Because $\int||\mathbf{x}||^2\phi(\mathbf{x}|0,\Sigma)d\mathbf{x} = tr(\Sigma)$. Then using the triangle inequality, we get $\int||\mathbf{x}||^2\phi(\mathbf{x}|\mu,\Sigma)d\mathbf{x} \le 2(||\mu||^2+tr(\Sigma))$. Further, from Lemma 9,

$$\int||\mathbf{x}||^2 g(\mathbf{x})d\mathbf{x} \le 4\eta^2+2bp.$$

Hence, G is g-integrable. Since $\log(g) \in \mathcal{G}_{(\mathbf{C}^0)}$, it implies that $E[|\log g(X)|] \le E[G(X)] < \infty$. We then prove the theorem. ■

Lemma 9 *Let $\Sigma \in \mathcal{D}_d$, where $\mathcal{D}_d$ is defined in* **A**1. *Then*

1. $a^d \le |\Sigma| \le b^d$ *and* $tr(\Sigma) \le bd$
2. $\forall \mathbf{x} \in \mathcal{R}^d$, $\mathbf{x}'\mathbf{x}/b \le \mathbf{x}'\Sigma^{-1}\mathbf{x} \le \mathbf{x}'\mathbf{x}/a$

Proof See for instance Maugis et al. (2007). ■

Appendix D

We prove Theorem 7 in Section 4.3. A configuration of a set of states is a particular tuple of values for these states. Use notation $\Psi(\cdot)$ for the set of all possible configurations of any group of states.

Proof For a HMM-VB, the configuration of the whole state sequence $\mathbf{s} = \{s_1, s_2, ..., s_T\}$ is treated as the index for one mixture component in the mapped GMM. The set of all possible configurations is $\Psi(\{s_1, ..., s_T\})$. Let $\hat{\mathcal{S}} = \Psi(\{s_1, ..., s_T\})$ and $\mathcal{S}_t = \Psi(s_t)$. Each component is a Gaussian distribution with mean $\mu_{\mathbf{s}} = (\mu_{s_1}^{(1)}, \mu_{s_2}^{(2)}, ..., \mu_{s_T}^{(T)})$ (column-wise stack of vectors) and a covariance matrix, denoted by $\Sigma_{\mathbf{s}}$, that is block diagonal. The tth diagonal block in $\Sigma_{\mathbf{s}}$ is $\Sigma_{s_t}^{(t)}$ with dimension $d_t \times d_t$

$$\Sigma_{\mathbf{s}} = \begin{pmatrix} \Sigma_{s_1}^{(1)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \Sigma_{s_2}^{(2)} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \Sigma_{s_T}^{(T)} \end{pmatrix}.$$

If we apply MEM directly to HMM-VB and keep in mind that $\mathbf{s}$ is the index for the mixture component, we need to compute the posterior $P(\mathbf{s} \mid \mathbf{x})$ in the E-step and

$$\left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x})\Sigma_{\mathbf{s}}^{-1}\right)^{-1} \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x})\Sigma_{\mathbf{s}}^{-1}\mu_{\mathbf{s}}\right)$$

in the M-step. The computational hurdle is that the number of possible sequences $\mathbf{s}$, that is, $|\hat{\mathcal{S}}|$, grows exponentially with T (assuming similar $|\mathcal{S}_t|$).

Because $\Sigma_{\mathbf{s}}$ is block diagonal, we have

$$\begin{aligned} &\left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x})\Sigma_{\mathbf{s}}^{-1}\right)^{-1} \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x})\Sigma_{\mathbf{s}}^{-1}\mu_{\mathbf{s}}\right) \\ &= \begin{pmatrix} \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_1}^{(1)}\right)^{-1}\right)^{-1} \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_1}^{(1)}\right)^{-1} \mu_{s_1}^{(1)}\right) \\ \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_2}^{(2)}\right)^{-1}\right)^{-1} \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_2}^{(2)}\right)^{-1} \mu_{s_2}^{(2)}\right) \\ \vdots \\ \vdots \\ \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_T}^{(T)}\right)^{-1}\right)^{-1} \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_T}^{(T)}\right)^{-1} \mu_{s_T}^{(T)}\right) \end{pmatrix}. \end{aligned}$$

Hence the tth variable block of $\mathbf{x}$ is given by

$$x^{(t)} = \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_t}^{(t)}\right)^{-1}\right)^{-1} \left(\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_t}^{(t)}\right)^{-1} \mu_{s_t}^{(t)}\right), \quad t = 1, 2, ..., T.$$

Let $I(\cdot)$ be the indicator function that equals 1 when the argument is true. Note that

$$\begin{aligned} \sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_t}^{(t)}\right)^{-1} &= \sum_{k\in\mathcal{S}_t} \left[\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) I(s_t = k)\right] \cdot \left(\Sigma_k^{(t)}\right)^{-1} \\ &= \sum_{k\in\mathcal{S}_t} L_k(\mathbf{x}, t) \cdot \left(\Sigma_k^{(t)}\right)^{-1} \end{aligned}$$

according to the definition of $L_k(\mathbf{x}, t)$ in Eq. (4). By the same technique, we can show that

$$\sum_{\mathbf{s}\in\hat{\mathcal{S}}} P(\mathbf{s} \mid \mathbf{x}) \left(\Sigma_{s_t}^{(t)}\right)^{-1} \mu_{s_t}^{(t)} = \sum_{k\in\mathcal{S}_t} L_k(\mathbf{x}, t) \cdot \left(\Sigma_k^{(t)}\right)^{-1} \mu_k^{(t)}.$$

■

It is interesting to note that Theorem 7 extends easily to a model more general than HMM-VB. For HMM-VB, s_1, ..., s_T follow a finite state Markov chain. *Bayesian network (BN)* is a generalization of Markov chain on a *directed acyclic graph (DAG)*, which is defined by all the conditional distributions of every state given any configuration of its parent states. If there is no parent, the marginal distribution of a state is specified. See Jensen (1996) for an introduction to BN. If we assume that the latent states s_1, ..., s_T are governed by a BN, and the observed $X^{(t)}$ conditioned on s_t follows a parametric distribution and is conditionally independent of any other $X^{(t')}$ and $s_{t'}$, then we obtain a special lattice MM more broadly defined than HMM-VB. We call this model *Mixture with Latent Bayesian Network* (*MLBN*). Define the following probability for MLBN:

$$L_k(\mathbf{x}, t) = P(s_t = k \mid \mathbf{x}), \quad k \in \mathcal{S}_t,\ t = 1, ..., T. \tag{18}$$

Clearly, the above definition is the same as Eq. (4) in the special case of HMM-VB. The MBW algorithm in Section 3.2 for HMM-VB applies to MLBN in general. We only need to use the definition in Eq. (18) for $L_k(\mathbf{x}^{[r]}, t)$. Theorem 7 can be proved for MLBN in the same way.

Appendix E

In this section, we provide the details for the data generation schemes of the simulation studies. We follow the notations in Eq. (3).

Simulation study in Section 5.1.1

It is a two-block design. The first variable block has 5 variables and is generated from a normal mixture of 7 components. We set $\pi = (0.51, 0.09, 0.20, 0.07, 0.10, 0.01, 0.02)$,

$$\mu^{(1)} = \begin{pmatrix} \mu_1^{(1)} \\ \mu_2^{(1)} \\ \vdots \\ \mu_7^{(1)} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4.5 & 0 & 0 \\ 4.5 & -2 & 0 & 0 & 0 \\ 0 & 0 & 4 & -1 & 4 \\ 0 & 3 & 0 & 5 & 0 \\ 0 & 7 & 7 & 0 & 0 \\ 0 & 7.7 & 8 & 0 & 0 \end{pmatrix},$$

$\Sigma_{1:5}^{(1)} = 1.5I_5$, $\Sigma_6^{(1)} = I_5$, $\Sigma_7^{(1)} = 0.5I_5$.

The second variable block has 3 variables, and is generated from a mixture of 10 Gaussian components. We set the transition probability matrix as

$$a^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.22 & 0.5 & 0.28 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.19 & 0 & 0 & 0 & 0 & 0.46 & 0.35 \\ 0 & 0 & 0 & 0.50 & 0.2 & 0.3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0.11 & 0.31 & 0.38 & 0 & 0 \\ 0 & 0 & 0.14 & 0.66 & 0 & 0.20 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.15 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0.65 \end{pmatrix},$$

$$\mu^{(2)} = \begin{pmatrix} \mu_1^{(2)} \\ \mu_2^{(2)} \\ \vdots \\ \mu_{10}^{(2)} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ -4 & -4 & -4 \\ 6.5 & 6.5 & 6.5 \\ -1 & 5 & 0 \\ -1.5 & 0 & 5 \\ 6 & 7 & 6.5 \\ -4.0 & 2 & 4.5 \\ 5 & -5 & -5 \\ -4 & 0 & 0 \\ 5 & -4.5 & -6 \end{pmatrix},$$

$\Sigma_{1,4,5}^{(2)} = 2I_3$, $\Sigma_2^{(2)} = \begin{pmatrix} 2.0 & 0.2 & 0.2 \\ 0.2 & 2.0 & 0.2 \\ 0.2 & 0.2 & 2.0 \end{pmatrix}$, $\Sigma_3^{(2)} = \begin{pmatrix} 2.0 & 0.9 & 0.9 \\ 0.9 & 2.0 & 0.9 \\ 0.9 & 0.9 & 1.5 \end{pmatrix}$, $\Sigma_6^{(2)} = \begin{pmatrix} 2.0 & 0.9 & 0.9 \\ 0.9 & 1.5 & 0.9 \\ 0.9 & 0.9 & 2.0 \end{pmatrix}$,

$\Sigma_7^{(2)} = \begin{pmatrix} 2.0 & -0.6 & -0.6 \\ -0.6 & 2.0 & 0.6 \\ -0.6 & 0.6 & 2.0 \end{pmatrix}$, $\Sigma_8^{(2)} = \begin{pmatrix} 2.0 & -0.6 & -0.6 \\ -0.6 & 5/3 & 0.6 \\ -0.6 & 0.6 & 5/3 \end{pmatrix}$, $\Sigma_{9,10}^{(2)} = \begin{pmatrix} 5/3 & -0.6 & -0.6 \\ -0.6 & 5/3 & 0.6 \\ -0.6 & 0.6 & 5/3 \end{pmatrix}$.

Simulation study in Section 5.1.2

The data is generated from a GMM with 10 components. Hence all the variables form one single variable block. We set

$$\pi = (0.01, 0.02, 0.23, 0.1, 0.15, 0.098, 0.098, 0.098, 0.098, 0.098),$$

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_{10} \end{pmatrix} = \begin{pmatrix} 5 & 5 & 5 & 5 & 5 \\ -5 & -5 & -5 & -5 & -5 \\ 5 & 5 & 0 & 5 & 0 \\ 5 & 0 & 0 & 5 & 5 \\ 0 & 0 & 5 & 5 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5 & -5 & -5 \\ -5 & -5 & 0 & 0 & -5 \\ -5 & -5 & -5 & 0 & 0 \\ -5 & 0 & 0 & 0 & -5 \end{pmatrix}.$$

All the variance matrices are independently generated from an inverse Wishart distribution with 10 degrees of freedom and diagonal scale matrix $7I_5$.

Simulation study in Section 5.1.3

The first 10 dimensions of the data are generated from a 3-component GMM. We set $\pi = (0.05, 0.25, 0.70)$,

$$\mu^{(1)} = \begin{pmatrix} \mu_1^{(1)} \\ \mu_2^{(1)} \\ \mu_3^{(1)} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 \\ -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 & -5 \end{pmatrix},$$

and $\Sigma_{1:3}^{(1)}$ are independently generated from an inverse Wishart distribution with 15 degrees of freedom and diagonal scale matrix $7I_{10}$.

The rest 30 dimensions are conditionally generated from a 5-component GMM. We set the transition probability matrix as

$$a^{(1)} = \begin{pmatrix} 0.1 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0.28 & 0.72 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\mu^{(2)} = \begin{pmatrix} \mu_1^{(2)} \\ \mu_2^{(2)} \\ \cdots \\ \mu_5^{(2)} \end{pmatrix} = \begin{pmatrix} \vec{0}_{30} \\ \vec{5}_{30} \\ -\vec{5}_{30} \\ -\vec{5}_{15}, \vec{5}_{15} \\ \vec{5}_{15}, -\vec{5}_{15} \end{pmatrix},$$

where $\vec{v}_l$ denotes a vector of the form $(v, v, ..., v)$ with l elements of the same value v. $\Sigma_{1:5}^{(2)}$ are block diagonals, which have the form of

$$\Sigma_k^{(2)} = \begin{pmatrix} \mathbf{A}_k^{(2)} & 0 \\ 0 & \mathbf{B}_k^{(2)} \end{pmatrix},$$

where $k = 1, ..., 5$, $\mathbf{A}_k^{(2)}$ is of size 10×10, which is independently generated from an inverse Wishart distribution with 15 degrees of freedom and diagonal scale matrix $7\mathbf{I}_{10}$. $\mathbf{B}_k^{(2)}$ is of size 20×20, and is independently generated from an inverse Wishart distribution with 25 degrees of freedom and diagonal scale matrix $7\mathbf{I}_{10}$.

References

Nima Aghaeepour, Radina Nikolic, Holger H. Hoos, and Ryan R. Brinkman. Rapid cell population identification in flow cytometry data. *Cytometry Part A*, 79A(1):6–13, 2011.

Nima Aghaeepour, Greg Finak, Holger Hoos, Tim R. Mosmann, Ryan Brinkman, Raphael Gottardo, and Richard H. Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nature Methods*, 10(3):228–238, Mar 2013.

Greg M. Allenby, Neeraj Arora, and James L. Ginter. On the heterogeneity of demand. *Journal of Marketing Research*, 35(3):384–389, 1998.

Adelchi Azzalini and Giovanna Menardi. Clustering via nonparametric density estimation: The r package pdfcluster. *Journal of Statistical Software, Articles*, 57(11):1–26, 2014.

Dmitry R. Bandura, Vladimir I. Baranov, Olga I. Ornatsky, Alexei Antonov, Robert Kinach, Xudong Lou, Serguei Pavlov, Sergey Vorobiev, John E. Dick, and Scott D. Tanner. Mass cytometry: Technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry. *Analytical Chemistry*, 81 (16):6813–6822, 2009.

Jeffrey D Banfield and Adrian E Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, pages 803–821, 1993.

Burkhard Becher, Andreas Schlitzer, Jinmiao Chen, Florian Mair, Hermi R. Sumatoh, Karen Wei Weng Teng, Donovan Low, Christiane Ruedl, Paola Riccardi-Castagnoli, Michael Poidinger, Melanie Greter, Florent Ginhoux, and Evan W. Newell. High-dimensional analysis of the murine myeloid cell system. *Nature Immunology*, 15(12):1181–1189, Dec 2014. Resource.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

Michael J. Boedigheimer and John Ferbas. Mixture modeling approach to flow cytometry data. *Cytometry Part A*, 73A(5):421–429, 2008.

Kenneth P Burnham and David R Anderson. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer Science & Business Media, 2003.

Gilles Celeux and Grard Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.

Gilles Celeux, Merrilee Hurn, and Christian P. Robert. Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, 95(451):957–970, 2000.

Cliburn Chan, Feng Feng, Janet Ottinger, David Foster, Mike West, and Thomas B. Kepler. Statistical mixture modeling for cell subtype identification in flow cytometry. *Cytometry Part A*, 73A(8):693–701, 2008.

Cliburn Chan, Lin Lin, Jacob Frelinger, Valerie Hebert, Dominic Gagnon, Claire Landry, Rafick-Pierre Sékaly, Jennifer Enzor, Janet Staats, Kent J. Weinhold, Maria Jaimes, and Mike West. Optimization of a highly standardized carboxyfluorescein succinimidyl ester flow cytometry panel and gating strategy design using discriminative information measure evaluation. *Cytometry Part A*, 77A(12):1126–1136, 2010.

Pratip K. Chattopadhyay, Todd M. Gierahn, Mario Roederer, and J. Christopher Love. Single-cell technologies for monitoring immune systems. *Nature Immunology*, 15(2): 128–135, Feb 2014. Review.

Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

Lawrence Corey, Peter B. Gilbert, Georgia D. Tomaras, Barton F. Haynes, Giuseppe Pantaleo, and Anthony S. Fauci. Immune correlates of vaccine protection against hiv-1 acquisition. *Science Translational Medicine*, 7(310):310rv7, 2015.

Patricia A. Darrah, Dipti T. Patel, Paula M. De Luca, Ross W. B. Lindsay, Dylan F. Davey, Barbara J. Flynn, Soren T. Hoff, Peter Andersen, Steven G. Reed, Sheldon L. Morris, Mario Roederer, and Robert A. Seder. Multifunctional th1 cells define a correlate of vaccine-mediated protection against leishmania major. *Nature Methods*, 13(7):843–850, Jul 2007.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B(Statistical Methodology)*, 39(1):1–38, 1977.

Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

Greg Finak, Ali Bashashati, Ryan Brinkman, and Raphael Gottardo. Merging mixture components for cell population identification in flow cytometry. *Advances in Bioinformatics*, Article ID 247646, 2009.

Chris Fraley and Adrian E Raftery. Mclust version 3: an r package for normal mixture modeling and model-based clustering. Technical report, DTIC Document, 2006.

Keinosuke Fukunaga and Larry D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

Minzhe Guo, Hui Wang, S Steven Potter, Jeffrey A Whitsett, and Yan Xu. Sincera: a pipeline for single-cell rna-seq profiling analysis. *PLoS Computational Biology*, 11(11): e1004575, 2015.

Christian Hennig. Methods for merging gaussian mixture components. *Advances in Data Analysis and Classification*, 4(1):3–34, 2010.

Finn V. Jensen. *Introduction to Bayesian Networks*. Springer-Verlag New York, Inc., New York, NY, USA, 1996.

Hiroyuki Kasahara and Katsumi Shimotsu. Nonparametric identification of finite mixture models of dynamic discrete choices. *Econometrica*, 77(1):135–175, 2009.

Vladimir Yu Kiselev, Kristina Kirschner, Michael T. Schaub, Tallulah Andrews, Andrew Yiu, Tamir Chandra, Kedar N. Natarajan, Wolf Reik, Mauricio Barahona, Anthony R Green, and Martin Hemberg. Sc3: consensus clustering of single-cell rna-seq data. *Nature Methods*, 2017.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

Hyangmin Lee and Jia Li. Variable selection for clustering by separability based on ridgelines. *Journal of Computational and Graphical Statistics*, 21(2):315–337, February 2012.

Jia Li. Clustering based on a multi-layer mixture model. *Journal of Computational and Graphical Statistics*, 14(3):547 – 568, 2005.

Jia Li, Surajit Ray, and Bruce G. Lindsay. A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8(8):1687–1723, August 2007.

Lin Lin, Cliburn Chan, Sine R. Hadrup, Thomas M. Froesig, Quanli Wang, and Mike West. Hierarchical bayesian mixture modelling for antigen-specific t-cell subtyping in combinatorially encoded flow cytometry studies. *Statistical Applications in Genetics and Molecular Biology*, 12(3):309–331, June 2013.

Lin Lin, Greg Finak, Kevin Ushey, Chetan Seshadri, Thomas R Hawn, Nicole Frahm, Thomas J. Scriba, Hassan Mahomed, Willem Hanekom, Pierre-Alexandre Bart, Giuseppe Pantaleo, Georgia D. Tomaras, Supachai Rerks-Ngarm, Jaranit Kaewkungwal, Sorachai Nitayaphan, Punnee Pitisuttithum, Nelson L. Michael, Jerome H. Kim, Merlin L. Robb, Robert J. O'Connell, Nicos Karasavvas, Peter Gilbert, Stephen C. De Rosa, M. Juliana McElrath, and Raphael Gottardo. COMPASS identifies t-cell subsets correlated with clinical outcomes. *Nature Biotechnology*, 33(6):610–616, June 2015a.

Lin Lin, Jacob Frelinger, Wenxin Jiang, Greg Finak, Chetan Seshadri, Pierre-Alexandre Bart, Giuseppe Pantaleo, Julie McElrath, Steve DeRosa, and Raphael Gottardo. Identification and visualization of multidimensional antigen-specific t-cell populations in polychromatic cytometry data. *Cytometry Part A*, 87(7):675–682, 2015b.

Lin Lin, Cliburn Chan, and Mike West. Discriminative variable subsets in bayesian classification with mixture models, with application in flow cytometry studies. *Biostatistics*, 17 (1):40–53, 2016.

Kenneth Lo, Ryan Remy Brinkman, and Raphael Gottardo. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A*, 73A(4):321–332, 2008.

Holden T. Maecker, J. Philip McCoy, and Robert Nussenblatt. Standardizing immunophenotyping for the human immunology project. *Nature Reviews Immunology*, 12(3):191–200, Mar 2012.

Cathy Maugis, Gilles Celeux, and Marie-Laure Martin-Magniette. *Variable selection for clustering with Gaussian mixture models*. PhD thesis, INRIA, 2007.

Cathy Maugis, Gilles Celeux, and Marie-Laure Martin-Magniette. Variable selection for clustering with gaussian mixture models. *Biometrics*, 65(3):701–709, 2009.

Geoff J. McLachlan, Richard W. Bean, and David Peel. A mixture model-based approach to the clustering of microarray expression data. *Bioinformatics*, 18(3):413–422, 2002.

Volodymyr Melnykov. Merging mixture components for clustering through pairwise overlap. *Journal of Computational and Graphical Statistics*, 25(1):66–90, 2016.

Volodymyr Melnykov and Ranjan Maitra. Finite mixture models and model-based clustering. *Statistics Surveys*, 4:80–116, 2010.

Stephen P. Perfetto, Pratip K. Chattopadhyay, and Mario Roederer. Seventeen-colour flow cytometry: unravelling the immune system. *Nature Reviews Immunology*, 4(8):648–655, Aug 2004.

Saumyadipta Pyne, Xinli Hu, Kui Wang, Elizabeth Rossin, Tsung-I Lin, Lisa M. Maier, Clare Baecher-Allan, Geoffrey J. McLachlan, Pablo Tamayo, David A. Hafler, Philip L. De Jager, and Jill P. Mesirov. Automated high-dimensional flow cytometric data analysis. *Proceedings of the National Academy of Sciences*, 106(21):8519–8524, 2009.

William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

Surajit Ray and Saumyadipta Pyne. A computational framework to emulate the human perspective in flow cytometric data analysis. *PloS ONE*, 7(5):e35693, 2012.

Sylvia Richardson and Peter J. Green. On bayesian analysis of mixtures with an unknown number of components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4):731–792, 1997.

Mario Roederer, Jason M. Brenchley, Michael R. Betts, and Stephen C. De Rosa. Flow cytometric analysis of vaccine responses: how many colors are enough? *Clinical Immunology*, 110(3):199 – 205, 2004.

Rahul Satija, Jeffrey A. Farrell, David Gennert, Alexander F. Schier, and Aviv Regev. Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology*, 33(5): 495–502, 2015.

Chetan Seshadri, Lin Lin, Thomas J. Scriba, Glenna Peterson, David Freidrich, Nicole Frahm, Stephen C. DeRosa, D. Branch Moody, Jacques Prandi, Martine Gilleron, Hassan Mahomed, Wenxin Jiang, Greg Finak, Willem A. Hanekom, Raphael Gottardo, M. Juliana McElrath, and Thomas R. Hawn. T cell responses against mycobacterial lipids and proteins are poorly correlated in south african adolescents. *The Journal of Immunology*, 195(10):4595–4603, 2015.

Matthew H. Spitzer and Garry P. Nolan. Mass cytometry: Single cells, many features. *Cell*, 165(4):780–791, May 2016.

Matthew Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):795–809, 2000.

Michael Titterington, Adrian F.M. Smith, and Ehud Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley, 1985.

Bo Wang, Junjie Zhu, Emma Pierson, Daniele Ramazzotti, and Serafim Batzoglou. Visualization and analysis of single-cell rna-seq data by kernel-based similarity learning. *Nature Methods*, 14(4):414–416, 2017.

Sidney J. Yakowitz and John D. Spragins. On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, 39(1):209–214, 1968.

Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. Cambridge University Press, 1997.

Justina Žurauskienė and Christopher Yau. pcareduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, 17(1):140, Mar 2016.

Journal of Machine Learning Research 18 (2017) 1-55 Submitted 4/16; Revised 5/17; Published 11/17

Approximation Vector Machines for Large-scale Online Learning

Trung Le trung.l@deakin.edu.au
Tu Dinh Nguyen tu.nguyen@deakin.edu.au
Vu Nguyen v.nguyen@deakin.edu.au
Dinh Phung dinh.phung@deakin.edu.au
Centre for Pattern Recognition and Data Analytics,
School of Information Technology,
Deakin University, Australia, Waurn Ponds Campus

Editor: Koby Crammer

Abstract

One of the most challenging problems in kernel online learning is to bound the model size and to promote model sparsity. Sparse models not only improve computation and memory usage, but also enhance the generalization capacity – a principle that concurs with the law of parsimony. However, inappropriate sparsity modeling may also significantly degrade the performance. In this paper, we propose Approximation Vector Machine (AVM), a model that can simultaneously encourage sparsity and safeguard its risk in compromising the performance. In an online setting context, when an incoming instance arrives, we approximate this instance by one of its neighbors whose distance to it is less than a predefined threshold. Our key intuition is that since the newly seen instance is expressed by its nearby neighbor the optimal performance can be analytically formulated and maintained. We develop theoretical foundations to support this intuition and further establish an analysis for the common loss functions including Hinge, smooth Hinge, and Logistic (i.e., for the classification task) and ℓ_1, ℓ_2, and ε-insensitive (i.e., for the regression task) to characterize the gap between the approximation and optimal solutions. This gap crucially depends on two key factors including the frequency of approximation (i.e., how frequent the approximation operation takes place) and the predefined threshold. We conducted extensive experiments for classification and regression tasks in batch and online modes using several benchmark datasets. The quantitative results show that our proposed AVM obtained comparable predictive performances with current state-of-the-art methods while simultaneously achieving significant computational speed-up due to the ability of the proposed AVM in maintaining the model size.

Keywords: kernel, online learning, large-scale machine learning, sparsity, big data, core set, stochastic gradient descent, convergence analysis

1. Introduction

In modern machine learning systems, data usually arrive continuously in stream. To enable efficient computation and to effectively handle memory resource, the system should be able to adapt according to incoming data. Online learning represents a family of efficient and scalable learning algorithms for building a predictive model incrementally from a sequence of data examples (Rosenblatt, 1958; Zinkevich, 2003). In contrast to the conventional learning algorithms (Joachims, 1999; Chang and Lin, 2011), which usually require a costly procedure to retrain the entire dataset when a new instance arrives, online learning aims to utilize the new incoming instances to improve the model given the knowledge of the correct answers to previous processed data (and possibly additional available information), making them suitable for large-scale online applications wherein data usually arrive sequentially and evolve rapidly.

The seminal line of work in online learning, referred to as *linear online learning* (Rosenblatt, 1958; Crammer et al., 2006; Dredze et al., 2008), aims at learning a linear predictor in the input space. The crucial limitation of this approach lies in its over-simplified linear modeling choice and consequently may fail to capture non-linearity commonly seen in many real-world applications. This motivated the works in *kernel-based online learning* (Freund and Schapire, 1999; Kivinen et al., 2004) in which a linear model in the feature space corresponding with a nonlinear model in the input space, hence allows one to cope with a variety of data distributions.

One common issue with *kernel-based online learning approach*, also known as the *curse of kernelization*, is that the model size (i.e., the number of vectors with non-zero coefficients) may grow linearly with the data size accumulated over time, hence causing computational problem and potential memory overflow (Steinwart, 2003; Wang et al., 2012). Therefore in practice, one might prefer kernel-based online learning methods with guaranty on a limited and bounded model size. In addition, enhancing model sparsity is also of great interest to practitioners since this allows the generalization capacity to be improved; and in many cases leading to a faster computation. However, encouraging sparsity needs to be done with care since an inappropriate sparsity-encouraging mechanism may compromise the performance. To address the curse of kernelization, budgeted approaches (Crammer et al., 2004; Dekel et al., 2005; Cavallanti et al., 2007; Wang and Vucetic, 2010; Wang et al., 2012; Le et al., 2016a,c) limits the model size to a predefined budget B. Specifically, when the current model size exceeds this budget, a budget maintenance strategy (e.g., removal, projection, or merging) is triggered to recover the model size back to the budget B. In these approaches, determining a suitable value for the predefined budget in a principled way is important, but challenging, since setting a small budget makes the learning faster but may suffer from underfitting, whereas a large budget makes the model fit better to data but may dramatically slow down the training process. An alternative way to address the curse of kernelization is to use random features (Rahimi and Recht, 2007) to approximate a kernel function (Ming et al., 2014; Lu et al., 2015; Le et al., 2016b). For example, Lu et al. (2015) proposed to transform data from the input space to the random-feature space, and then performed SGD in the feature space. However, in order for this approach to achieve good kernel approximation, excessive number of random features is required which could lead to a serious computational issue. To reduce the impact number of random features, Le et al. (2016b) proposed to distribute the model in dual space including the original feature space and the random feature space that approximates the first space.

1. In fact, we used a subset of the dataset a9a which has 123 features. We then project all data points onto 3D using t-SNE. We note that the t-SNE does not do clustering, it only reduces the dimensionality into 3D for visualization while trying to preserve the local properties of the data.

©2017 Trung Le, Tu Dinh Nguyen, Vu Nguyen, and Dinh Phung.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-191.html.

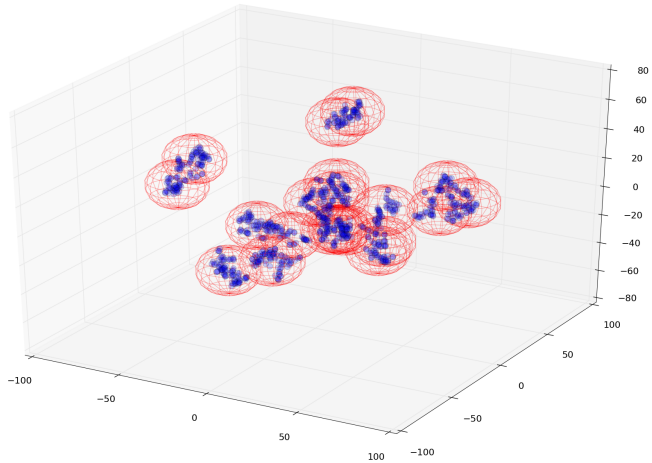


Figure 1: An illustration of the hypersphere coverage for $1,000$ data samples which locate in 3D space. We cover this dataset using hyperspheres with the diameter $\delta = 7.0$, resulting in 20 hypersphere cells as shown in the figure (cf. Sections (6.3,9)). All data samples in a same cell are approximated by a core point in this cell. The model size is therefore significantly reduced from 1,000 to 20.[1]

In this paper, we propose *Approximation Vector Machine* (AVM) to simultaneously encourage model sparsity[2] while preserving the model performance. Our model size is theoretically proven to be bounded regardless of the data distribution and data arrival order. To promote sparsity, we introduce the notion of δ-*coverage* which partitions the data space into overlapped cells whose diameters are defined by δ (cf. Figure 1). This coverage can be constructed in advance or on the fly. Our experiment on the real datasets shows that the coverage can impressively boost sparsity; for example with dataset *KDDCup99* of $4,408,589$ instances, our model size is 115 with $\delta = 3$ (i.e., only 115 cells are required); with dataset *airlines* of $5,336,471$ instances, our model size is 388 with $\delta = 1$.

In an online setting context, when an incoming instance arrives, it can be approximated with the corresponding core point in the cell that contains it. Our intuitive reason is that when an instance is approximated by an its nearby core point, the performance would be largely preserved. We further developed rigorous theory to support this intuitive reason. In particular, our convergence analysis (covers six popular loss functions, namely Hinge, smooth Hinge, and Logistic for classification task and ℓ_2, ℓ_1, and ε-insensitive for regression task) explicitly characterizes the gap between the approximate and optimal solutions. The analysis shows that this gap crucially depends on two key factors including the cell diameter

2. Model sparsity can be computed as the ratio of the model size and the number of vectors received so far.

δ and the approximation process. In addition, the cell parameter δ can be used to efficiently control the trade-off between sparsity level and the model performance. We conducted extensive experiments to validate the proposed method on a variety of learning tasks, including classification in batch mode, classification and regression in online mode on several benchmark large-scale datasets. The experimental results demonstrate that our proposed method maintains a comparable predictive performance while simultaneously achieving an order of magnitude speed-up in computation comparing with the baselines due to its capacity in maintaining model size. We would like to emphasize at the outset that unlike budgeted algorithms (e.g., (Crammer et al., 2004; Dekel et al., 2005; Cavallanti et al., 2007; Wang and Vucetic, 2010; Wang et al., 2012; Le et al., 2016a,c, 2017)), our proposed method is nonparametric in the sense that the number of core sets grow with data on demand, hence care should be exercised in practical implementation.

The rest of this paper is organized as follows. In Section 2, we review works mostly related to ours. In Section 3, we present the primal and dual forms of Support Vector Machine (SVM) as they are important background for our work. Section 4 formulates the proposed problem. In Section 5, we discuss the standard SGD for kernel online learning with an emphasis on the *curse of kernelization*. Section 6 presents our proposed AVM with full technical details. Section 7 devotes to study the suitability of loss functions followed by Section 8 where we extend the framework to multi-class setting. Finally, in Section 9, we conduct extensive experiments on several benchmark datasets and then discuss experimental results as well as their implications. In addition, all supporting proof is provided in the appendix sections.

2. Related Work

One common goal of online kernel learning is to bound the model size and to encourage sparsity. Generally, research in this direction can be broadly reviewed into the following themes.

Budgeted Online Learning. This approach limits the model size to a predefined budget B. When the model size exceeds the budget, a budget maintenance strategy is triggered to decrement the model size by one. Three popular budget maintenance strategies are *removal*, *projection*, and *merging*. In the removal strategy, the most redundant support vector is simply eliminated. In the projection strategy, the information of the most redundant support vector is conserved through its projection onto the linear span of the remaining support vectors. The merging strategy first selects two vectors, and then merges them into one before discarding them. Forgetron (Dekel et al., 2005) is the first budgeted online learning method that employs the removal strategy for the budget maintenance. At each iteration, if the classifier makes a mistake, it conducts a three-step update: (i) running the standard Perceptron (Rosenblatt, 1958) update; (ii) shrinking the coefficients of support vectors with a scaling factor; and (iii) removing the support vector with the smallest coefficient. Randomized Budget Perceptron (RBP) (Cavallanti et al., 2007) randomly removes a support vector when the model size overflows the budget. Budget Perceptron (Crammer et al., 2004) and Budgeted Passive Aggressive (BPA-S)(Wang and Vucetic, 2010) attempt to discard the most redundant support vector (SV). Orabona et al. (2009) used the projection to automatically discover the model size. The new vector is added to the support set if its

projection onto the linear span of others in the feature space exceeds a predefined threshold, or otherwise its information is kept through the projection. Other works involving the projection strategy include Budgeted Passive Aggressive Nearest Neighbor (BPA-NN) (Wang and Vucetic, 2010; Wang et al., 2012). The merging strategy was used in some works (Wang and Vucetic, 2009; Wang et al., 2012).

Random Features. The idea of random features was proposed in (Rahimi and Recht, 2007). Its aim is to approximate a shift-invariant kernel using the harmonic functions. In the context of online kernel learning, the problem of model size vanishes since we can store the model directly in the random features. However, the arising question is to determine the appropriate number of random features D to sufficiently approximate the real kernel while keeping this dimension as small as possible for an efficient computation. Ming et al. (2014) investigated the number of random features in the online kernel learning context. Recently, Lu et al. (2015) proposed to run stochastic gradient descent (SGD) in the random feature space rather than that in the real feature space. The theory accompanied with this work shows that with a high confidence level, SGD in the random feature space can sufficiently approximate that in the real kernel space. Nonetheless, in order to achieve good kernel approximation in this approach, excessive number of random features is required, possibly leading to a serious computational issue. To reduce the impact of the number of random features to learning performance, Le et al. (2016b) proposed to store core vectors in the original feature space, whilst storing remaining vectors in the random feature space that sufficiently approximates the first space.

Core Set. This approach utilizes a core set to represent the model. This core set can be constructed on the fly or in advance. Notable works consist of the Core Vector Machine (CVM) (Tsang et al., 2005) and its simplified version, the Ball Vector Machine (BVM) (Tsang et al., 2007). The CVM was based on the achievement in computational geometry (Badoiu and Clarkson, 2002) to reformulate a variation of ℓ_2-SVM as a problem of finding minimal enclosing ball (MEB) and the core set includes the points lying furthest away the current centre of the current MEB. Our work can be categorized into this line of thinking. However, our work is completely different to (Tsang et al., 2005, 2007) in the mechanism to determine the core set and update the model. In addition, the works of (Tsang et al., 2005, 2007) are not applicable for the online learning.

3. Primal and Dual Forms of Support Vector Machine

Support Vector Machine (SVM) Cortes and Vapnik (1995) represents one of the state-of-the-art methods for classification. Given a training set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$, the data instances are mapped to a feature space using the transformation $\Phi(.)$, and then SVM aims to learn an optimal hyperplane in the feature space such that the margin, the distance from the closest data instance to the hyperplane, is maximized. The optimization problem

of SVM can be formulated as follows

$$\begin{aligned} &\min_{\mathbf{w},b} \left(\frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{N} \sum_{i=1}^{N} \xi_i \right) \\ &\text{s.t.} : y_i \left(\mathbf{w}^\mathsf{T} \Phi(x_i) + b \right) \geq 1 - \xi_i,\ i = 1, ..., N \\ &\xi_i \geq 0,\ i = 1, ..., N \end{aligned} \tag{1}$$

where $\lambda > 0$ is the regularization parameter, $\Phi(.)$ is the transformation from the input space to the feature space, and $\boldsymbol{\xi} = [\xi_i]_{i=1}^N$ is the vector of slack variables.

Using Karush-Kuhn-Tucker theorem, the above optimization problem is transformed to the *dual form* as follows

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \left(\frac{1}{2} \boldsymbol{\alpha}^\mathsf{T} Q \boldsymbol{\alpha} - \boldsymbol{e}^\mathsf{T} \boldsymbol{\alpha} \right) \\ &\text{s.t.} : \boldsymbol{y}^\mathsf{T} \boldsymbol{\alpha} = 0 \\ &0 \leq \alpha_i \leq \frac{1}{\lambda N},\ i = 1, ..., N \end{aligned}$$

where $Q = [y_i y_j K(x_i, x_j)]_{i,j=1}^N$ is the Gram matrix, $K(x, x') = \Phi(x)^\mathsf{T} \Phi(x')$ is a kernel function, $\boldsymbol{e} = [1]_{N \times 1}$ is the vector of all 1, and $\boldsymbol{y} = [y_i]_{i=1,..,N}^\mathsf{T}$.

The dual optimization problem can be solved using the solvers (Joachims, 1999; Chang and Lin, 2011). However, the computational complexity of the solvers is over-quadratic (Shalev-Shwartz and Srebro, 2008) and the dual form does not appeal to the online learning setting. To scale up SVM and make it appealing to the online learning, we rewrite the constrained optimization problem in Eq. (1) in the *primal form* as follows

$$\min_{\mathbf{w}} \left(\frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{N} \sum_{i=1}^{N} l(\mathbf{w}; x_i, y_i) \right) \tag{2}$$

where $l(\mathbf{w}; x, y) = \max\left(0, 1 - y \mathbf{w}^\mathsf{T} \Phi(x)\right)^3$ is Hinge loss.

In our current interest, the advantages of formulating the optimization problem of SVM in the primal form as in Eq. (2) are at least two-fold. First, it encourages the application of SGD-based method to propose a solution for the online learning context. Second, it allows us to extend Hinge loss to any appropriate loss functions (cf. Section 7) to enrich a wider class of problems that can be addressed.

3. We can eliminate the bias b by simply adjusting the kernel.

4. Problem Setting

We consider two following optimization problems for batch and online settings respectively in Eqs. (3) and (4)

$$\min_{\mathbf{w}} f(\mathbf{w}) \triangleq \frac{\lambda}{2}\|\mathbf{w}\|^2 + \mathbb{E}_{(x,y)\sim\mathbb{P}_N}\left[l(\mathbf{w};x,y)\right] \triangleq \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^{N} l(\mathbf{w};x_i,y_i) \quad (3)$$

$$\min_{\mathbf{w}} f(\mathbf{w}) \triangleq \frac{\lambda}{2}\|\mathbf{w}\|^2 + \mathbb{E}_{(x,y)\sim\mathbb{P}_{\mathcal{X},\mathcal{Y}}}\left[l(\mathbf{w};x,y)\right] \quad (4)$$

where $l(\mathbf{w};x,y)$ is a *convex* loss function, $\mathbb{P}_{\mathcal{X},\mathcal{Y}}$ is the joint distribution of (x,y) over $\mathcal{X}\times\mathcal{Y}$ with the data domain $\mathcal{X}$ and the label domain $\mathcal{Y}$, and $\mathbb{P}_N$ specifies the empirical distribution over the training set $\mathcal{D} = \{(x_1,y_1),\ldots,(x_N,y_N)\}$. Furthermore, we assume that the convex loss function $l(\mathbf{w};x,y)$ satisfies the following property: there exists two positive numbers A and B such that $\left\|l'(\mathbf{w};x,y)\right\| \leq A\|\mathbf{w}\|^{1/2} + B, \forall \mathbf{w},x,y$. As demonstrated in Section 7, this condition is valid for all common loss functions. Hereafter, for given any function $g(\mathbf{w})$, we use the notation $g'(\mathbf{w}_0)$ to denote the gradient (or any sub-gradient) of $g(.)$ w.r.t $\mathbf{w}$ evaluated at $\mathbf{w}_0$.

It is clear that given a fixed $\mathbf{w}$, there exists a random variable g such that $\mathbb{E}[g \mid \mathbf{w}] = f'(\mathbf{w})$. In fact, we can specify $g = \lambda\mathbf{w} + l'(\mathbf{w};x_t,y_t)$ where $(x_t,y_t) \sim \mathbb{P}_{\mathcal{X},\mathcal{Y}}$ or $\mathbb{P}_N$. We assume that a *positive semi-definite* (p.s.d.) and *isotropic* (iso.) kernel Rasmussen and Williams (2005) is used, i.e., $K\left(x,x'\right) = k\left(\left\|x-x'\right\|^2\right)$, where $k: \mathcal{X}\to\mathbb{R}$ is an appropriate function. Let $\Phi(.)$ be the feature map corresponding the kernel (i.e., $K\left(x,x'\right) = \Phi(x)^{\mathsf{T}}\Phi\left(x'\right)$). To simplify the convergence analysis, without loss of generality we further assume that $\|\Phi(x)\|^2 = K(x,x) = 1, \forall x\in\mathcal{X}$. Finally, we denote the optimal solution of optimization problem in Eq. (3) or (4) by $\mathbf{w}^*$, that is, $\mathbf{w}^* = \text{argmin}_{\mathbf{w}} f(\mathbf{w})$.

5. Stochastic Gradient Descent Method

We introduce the standard kernel stochastic gradient descent (SGD) in Algorithm 1 wherein the standard learning rate $\eta_t = \frac{1}{\lambda t}$ is used (Shalev-Shwartz et al., 2007, 2011). Let α_t be a scalar such that $l'(\mathbf{w}_t;x_t,y_t) = \alpha_t\Phi(x_t)$ (we note that this scalar exists for all common loss functions as presented in Section 7). It is apparent that at the iteration t the model $\mathbf{w}_t$ has the form of $\mathbf{w}_t = \sum_{i=1}^{t}\alpha_i^{(t)}\Phi(x_i)$. The vector x_i $(1\leq i\leq t)$ is said to be a *support vector* if its coefficient $\alpha_i^{(t)}$ is nonzero. The model is represented through the support vectors, and hence we can define the model size to be $\left\|\boldsymbol{\alpha}^{(t)}\right\|_0$ and model sparsity as the ratio between the current model size and t (i.e., $\left\|\boldsymbol{\alpha}^{(t)}\right\|_0/t$). Since it is likely that α_t is nonzero (e.g., with Hinge loss, it happens if x_t lies in the margins of the current hyperplane), the standard kernel SGD algorithm is vulnerable to the curse of kernelization, that is, the model size, is almost linearly grown with the data size accumulated over time Steinwart (2003). Consequently, the computation gradually becomes slower or even infeasible when the data size grows rapidly.

Algorithm 1 Stochastic Gradient Descent algorithm.

Input: λ, p.s.d. kernel $K(.,.) = \Phi(\cdot)^{\mathsf{T}}\Phi(\cdot)$

$\mathbf{w}_1 = \mathbf{0}$
for $t = 1,2,\ldots T$ **do**
  Receive (x_t,y_t) //$(x_t,y_t)\sim\mathbb{P}_{\mathcal{X},\mathcal{Y}}$ *or* $\mathbb{P}_N$
  $\eta_t = \frac{1}{\lambda t}$
  $g_t = \lambda\mathbf{w}_t + l'(\mathbf{w}_t;x_t,y_t) = \lambda\mathbf{w}_t + \alpha_t\Phi(x_t)$
  $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t g_t = \frac{t-1}{t}\mathbf{w}_t - \eta_t\alpha_t\Phi(x_t)$
end for

Output: $\overline{\mathbf{w}}_T = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t$ or $\mathbf{w}_{T+1}$

6. Approximation Vector Machines for Large-scale Online Learning

In this section, we introduce our proposed Approximation Vector Machine (AVM) for online learning. The main idea is that we employ an overlapping partition of sufficiently small cells to cover the data domain, i.e., $\mathcal{X}$ or $\Phi(\mathcal{X})$; when an instance arrives, we approximate this instance by a corresponding core point in the cell that contains this instance. Our intuition behind this approximation procedure is that since the instance is approximated by its neighbor, the performance would not be significantly compromised while gaining significant speedup. We start this section with the definition of δ-coverage, its properties and connection with the feature space. We then present AVM and the convergence analysis.

6.1 δ-coverage over a domain

To facilitate our technical development in sequel, we introduce the notion of δ-coverage in this subsection. We first start with the usual definition of a diameter for a set.

Definition 1. (*diameter*) Given a set A, the diameter of this set is defined as $D(A) = \sup_{x,x'\in A}||x-x'||$. This is the maximal pairwise distance between any two points in A.

Next, given a domain $\mathcal{X}$ (e.g., the data domain, input space) we introduce the concept of δ-coverage for $\mathcal{X}$ using a collection of sets.

Definition 2. (*δ-coverage*) The collection of sets $\mathcal{P} = (P_i)_{i\in I}$ is said to be an δ-coverage of the domain $\mathcal{X}$ iff $\mathcal{X}\subset\cup_{i\in I}P_i$ and $D(P_i)\leq\delta, \forall i\in I$ where I is the index set (not necessarily discrete) and each element $P_i\in\mathcal{P}$ is further referred to as a *cell*. Furthermore if the index set I is finite, the collection $\mathcal{P}$ is called a *finite* δ-coverage.

Definition 3. (*core set, core point*) Given an δ-coverage $\mathcal{P} = (P_i)_{i\in I}$ over a given domain $\mathcal{X}$, for each $i\in I$, we select an arbitrary point c_i from the cell P_i, then the collection of all c_i (s) is called the *core set* $\mathcal{C}$ of the δ-coverage $\mathcal{P}$. Each point $c_i\in\mathcal{C}$ is further referred to as a *core point*.

We show that these definitions can be also extended to the feature space with the mapping Φ and kernel K via the following theorem.

Theorem 4. *Assume that the p.s.d. and isotropic kernel* $K(x,x') = k\left(||x-x'||^2\right)$, *where* $k(.)$ *is a monotonically continuous decreasing function with* $k(0) = 1$, *is examined and* $\Phi(.)$ *is its induced feature map. If* $\mathcal{P} = (P_i)_{i\in I}$ *is an* δ*-coverage of the domain* $\mathcal{X}$ *then* $\Phi(\mathcal{P}) = (\Phi(\mathcal{P}_i))_{i\in I}$ *is* **also** *an* δ_Φ*-coverage of the domain* $\Phi(\mathcal{X})$, *where* $\delta_\Phi = \sqrt{2(1-k(\delta^2))}$ *is a monotonically increasing function and* $\lim_{\delta\to 0}\delta_\Phi = 0$.

In particular, the Gaussian kernel given by $K(x,x') = \exp(-\gamma\left\|x-x'\right\|^2)$ is a p.s.d. and iso. kernel and $\delta_\Phi = \sqrt{2(1-\exp(-\gamma\delta^2))}$. Theorem 4 further reveals that the image of an δ-coverage in the input space is an δ_Φ-coverage in the feature space and when the diameter δ approaches 0, so does the induced diameter δ_Φ. For readability, the proof of this theorem is provided in Appendix 11.

We have further developed methods and algorithms to efficiently construct δ-coverage, however to maintain the readability, we defer this construction to Section 6.3.

6.2 Approximation Vector Machines

We now present our proposed Approximation Vector Machine (AVM) for online learning. In an online setting, instances arise on the fly and we need an efficient approach to incorporate incoming instances into the learner. Different from the existing works (cf. Section 2), our approach is to construct an δ-coverage $\mathcal{P} = (P_i)_{i\in I}$ over the input domain $\mathcal{X}$, and for each incoming instance x we find the cell P_i that contains this instance and approximate this instance by a core point $c_i \in P_i$. The coverage $\mathcal{P}$ and core set $\mathcal{C}$ can either be constructed in advance or on the fly as presented in Section 6.3.

In Algorithm 2, when receiving an incoming instance (x_t, y_t), we compute the scalar α_t such that $\alpha_t\Phi(x_t) = l'(\mathbf{w}_t; x_t, y_t)$ (cf. Section 7) in Step 5. Furthermore at Step 7 we introduce a Bernoulli random variable Z_t to govern the approximation procedure. This random variable could be either statistically independent or dependent with the incoming instances and the current model. In Section 9.2, we report on different settings for Z_t and how they influence the model size and learning performance. Our findings at the outset is that, the naive setting with $\mathbb{P}(Z_t = 1) = 1, \forall t$ (i.e., always performing approximation) returns the sparsest model while obtaining comparable learning performance comparing with the other settings. Moreover, as shown in Steps 9 and 11, we only approximate the incoming data instance by the corresponding core point (i.e., c_{i_t}) if $Z_t = 1$. In addition, if $Z_t = 1$, we find a cell that contains this instance in Step 8. It is worth noting that the δ-coverage and the cells are constructed on the fly along with the data arrival (cf. Algorithms 3 and 4). In other words, the incoming data instance might belong to an existing cell or a new cell that has the incoming instance as its core point is created.

Furthermore to ensure that $\|\mathbf{w}_t\|$ is bounded for all $t \geq 1$ in the case of ℓ_2 loss, if $\lambda \leq 1$ then we project $\mathbf{w}_t - \eta_t h_t$ onto the hypersphere with centre origin and radius $y_{\max}\lambda^{-1/2}$, i.e., $\mathcal{B}\left(\mathbf{0}, y_{\max}\lambda^{-1/2}\right)$. Since it can be shown that with ℓ_2 loss the optimal solution $\mathbf{w}^*$ lies in $\mathcal{B}\left(\mathbf{0}, y_{\max}\lambda^{-1/2}\right)$ (cf. Theorem 23 in Appendix 12), this operation could possibly result in a faster convergence. In addition, by reusing the previous information, this operation can be efficiently implemented. Finally, we note that with ℓ_2 loss and $\lambda > 1$, we do not need to perform a projection to bound $\|\mathbf{w}_t\|$ since according to Theorem 25 in Appendix 12, $\|\mathbf{w}_t\|$ is bounded by $\frac{y_{\max}}{\lambda-1}$. Here it is worth noting that we have defined $y_{\max} = \max_{y\in\mathcal{Y}}|y|$ and this notation is only used in the analysis for the regression task with the ℓ_2 loss.

Algorithm 2 Approximation Vector Machine.

Input: λ, p.s.d. & iso. $K(.,.) = \Phi(\cdot)^\mathsf{T}\Phi(\cdot)$, δ-coverage $\mathcal{P} = (P_i)_{i\in I}$

1: $\mathbf{w}_1 = 0$
2: **for** $t = 1, \ldots, T$ **do**
3: Receive (x_t, y_t) //$(x_t, y_t) \sim \mathbb{P}_{\mathcal{X},\mathcal{Y}}$ *or* $\mathbb{P}_N$
4: $\eta_t = \frac{1}{\lambda t}$
5: $l'(\mathbf{w}_t; x_t, y_t) = \alpha_t\Phi(x_t)$ //cf. Section 7
6: Sample a Bernoulli random variable Z_t
7: **if** $Z_t = 1$ **then**
8: Find $i_t \in I$ such that $x_t \in P_{i_t}$
9: $h_t = \lambda\mathbf{w}_t + \alpha_t\Phi(c_{i_t})$ //*do approximation*
10: **else**
11: $h_t = \lambda\mathbf{w}_t + \alpha_t\Phi(x_t)$
12: **end if**
13: **if** ℓ_2 loss is used **and** $\lambda \leq 1$ **then**
14: $\mathbf{w}_{t+1} = \prod_{\mathcal{B}\left(\mathbf{0}, y_{\max}\lambda^{-1/2}\right)}(\mathbf{w}_t - \eta_t h_t)$
15: **else**
16: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t h_t$
17: **end if**
18: **end for**

Output: $\overline{\mathbf{w}}_T = \frac{\sum_{t=1}^T \mathbf{w}_t}{T}$ or $\mathbf{w}_{T+1}$

In what follows, we present the theoretical results for our proposed AVM including the convergence analysis for a general convex or smooth loss function and the upper bound of the model size under the assumption that the incoming instances are drawn from an arbitrary distribution and arrive in a random order.

6.2.1 Analysis for Generic Convex Loss Function

We start with the theoretical analysis for Algorithm 2. The *decision of approximation* (i.e., the random variable Z_t) could be statistically independent or dependent with the current model parameter $\mathbf{w}_t$ and the incoming instance (x_t, y_t). For example, one can propose an algorithm in which the *decision of approximation* is performed iff the confidence level of the incoming instance w.r.t the current model is greater than 1, i.e., $y_t\mathbf{w}_t^\mathsf{T}\Phi(x_t) \geq 1$. We shall develop our theory to take into account all possible cases.

Theorem 5 below establishes an upper bound on the regret under the possible assumptions of the statistical relationship among the decision of approximation, the data distribution, and the current model. Based on Theorem 5, in Theorem 8 we further establish an inequality for the error incurred by a single-point output with a high confidence level.

Theorem 5. *Consider the running of Algorithm 2 where* (x_t, y_t) *is uniformly sampled from the training set* $\mathcal{D}$ *or the joint distribution* $\mathbb{P}_{\mathcal{X},\mathcal{Y}}$, *the following statements hold*

i) If Z_t and $\mathbf{w}_t$ are independent for all t (i.e., the decision of approximation only depends on the data distribution) then

$$\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right) - f\left(\mathbf{w}^*\right)\right] \leq \frac{H\left(\log\left(T\right)+1\right)}{2\lambda T} + \frac{\delta_\Phi M^{1/2} W^{1/2}}{T}\sum_{t=1}^{T}\mathbb{P}\left(Z_t = 1\right)^{1/2}$$

where H, M, W are positive constants.

ii) If Z_t is independent with both (x_t, y_t) and $\mathbf{w}_t$ for all t (i.e., the decision of approximation is independent with the current hyperplane and the data distribution) then

$$\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right) - f\left(\mathbf{w}^*\right)\right] \leq \frac{H\left(\log\left(T\right)+1\right)}{2\lambda T} + \frac{\delta_\Phi M^{1/2} W^{1/2}}{T}\sum_{t=1}^{T}\mathbb{P}\left(Z_t = 1\right)$$

iii) In general, we always have

$$\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right) - f\left(\mathbf{w}^*\right)\right] \leq \frac{H\left(\log\left(T\right)+1\right)}{2\lambda T} + \delta_\Phi M^{1/2} W^{1/2}$$

Remark 6. Theorem 5 consists of the standard convergence analysis. In particular, if the approximation procedure is never performed, i.e., $\mathbb{P}\left(Z_t = 1\right) = 0$, $\forall t$, we have the regret bound $\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right) - f\left(\mathbf{w}^*\right)\right] \leq \frac{H(\log(T)+1)}{2\lambda T}$.

Remark 7. Theorem 5 further indicates that there exists an error gap between the optimal and the approximate solutions. When δ decreases to 0, this gap also decreases to 0. Specifically, when $\delta = 0$ (so does δ_Φ), any incoming instance is approximated by itself and consequently, the gap is exactly 0.

Theorem 8. *Let us define the gap by d_T, which is $\frac{\delta_\Phi M^{1/2} W^{1/2}}{T}\sum_{t=1}^{T}\mathbb{P}\left(Z_t = 1\right)^{1/2}$ (if Z_t is independent with $\mathbf{w}_t$), $\frac{\delta_\Phi M^{1/2} W^{1/2}}{T}\sum_{t=1}^{T}\mathbb{P}\left(Z_t = 1\right)$ (if Z_t is independent with (x_t, y_t) and $\mathbf{w}_t$), or $\delta_\Phi M^{1/2} W^{1/2}$. Let r be any number randomly picked from $\{1, 2, \ldots, T\}$. With the probability at least $1-\delta$, the following statement holds*

$$f\left(\mathbf{w}_r\right) - f\left(\mathbf{w}^*\right) \leq \frac{H\left(\log\left(T\right)+1\right)}{2\lambda T} + d_T + \Delta_T\sqrt{\frac{1}{2}\log\frac{1}{\delta}}$$

where $\Delta_T = \max\limits_{1\leq t\leq T}\left(f\left(\mathbf{w}_t\right) - f\left(\mathbf{w}^*\right)\right)$.

We now present the convergence analysis for the case when we output the α-suffix average result as proposed in Rakhlin et al. (2012). With $0 < \alpha < 1$, let us denote

$$\overline{\mathbf{w}}_T^\alpha = \frac{1}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T}\mathbf{w}_t$$

where we assume that the fractional indices are rounded to their ceiling values.

Theorem 9 establishes an upper bound on the regret for the α-suffix average case, followed by Theorem 10 which establishes an inequality for the error incurred by a α-suffix average output with a high confidence level.

Theorem 9. *Consider the running of Algorithm 2 where (x_t, y_t) is uniformly sampled from the training set $\mathcal{D}$ or the joint distribution $\mathbb{P}_{\mathcal{X},\mathcal{Y}}$, the following statements hold*

i) If Z_t and $\mathbf{w}_t$ are independent for all t (i.e., the decision of approximation only depends on the data distribution) then

$$\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T^\alpha\right) - f\left(\mathbf{w}^*\right)\right] \leq \frac{\lambda\left(1-\alpha\right)}{2\alpha}W_T^\alpha + \frac{\delta_\Phi M^{1/2} W^{1/2}}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T}\mathbb{P}\left(Z_t = 1\right)^{1/2} + \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T}$$

where H, M, W are positive constants and $W_T^\alpha = \mathbb{E}\left[\left\|\mathbf{w}_{(1-\alpha)T+1} - \mathbf{w}^*\right\|^2\right]$.

ii) If Z_t is independent with both (x_t, y_t) and $\mathbf{w}_t$ for all t (i.e., the decision of approximation is independent with the current hyperplane and the data distribution) then

$$\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T^\alpha\right) - f\left(\mathbf{w}^*\right)\right] \leq \frac{\lambda\left(1-\alpha\right)}{2\alpha}W_T^\alpha + \frac{\delta_\Phi M^{1/2} W^{1/2}}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T}\mathbb{P}\left(Z_t = 1\right) + \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T}$$

iii) In general, we always have

$$\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T^\alpha\right) - f\left(\mathbf{w}^*\right)\right] \leq \frac{\lambda\left(1-\alpha\right)}{2\alpha}W_T^\alpha + \delta_\Phi M^{1/2} W^{1/2} + \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T}$$

Theorem 10. *Let us once again define the induced gap by d_T, which is respectively*

$$\frac{\lambda\left(1-\alpha\right)}{2\alpha}W_T^\alpha + \frac{\delta_\Phi M^{1/2} W^{1/2}}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T}\mathbb{P}\left(Z_t = 1\right)^{1/2} \text{ (if } Z_t \text{ is independent with } \mathbf{w}_t\text{)},$$

$$\frac{\lambda\left(1-\alpha\right)}{2\alpha}W_T^\alpha + \frac{\delta_\Phi M^{1/2} W^{1/2}}{\alpha T}\sum_{t=(1-\alpha)T+1}^{T}\mathbb{P}\left(Z_t = 1\right) \text{ (if } Z_t \text{ is independent with } (x_t, y_t) \text{ and } \mathbf{w}_t\text{)},$$

or $\frac{\lambda(1-\alpha)}{2\alpha}W_T^\alpha + \delta_\Phi M^{1/2} W^{1/2}$. Let r be any number randomly picked from $\{(1-\alpha)T+1, 2, \ldots, T\}$. With the probability at least $1-\delta$, the following statement holds

$$f\left(\mathbf{w}_r\right) - f\left(\mathbf{w}^*\right) \leq \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T} + d_T + \Delta_T^\alpha\sqrt{\frac{1}{2}\log\frac{1}{\delta}}$$

where $\Delta_T^\alpha = \max\limits_{(1-\alpha)T+1\leq t\leq T}\left(f\left(\mathbf{w}_t\right) - f\left(\mathbf{w}^*\right)\right)$.

Remark 11. Theorems 8 and 10 concern with the theoretical warranty if rendering any single-point output $\mathbf{w}_r$ rather than the average outputs. The upper bound gained in Theorem 10 is tighter than that gained in Theorem 8 in the sense that the quantity $\frac{H\log(1/(1-\alpha))}{2\lambda\alpha T} + \Delta_T^\alpha\sqrt{\frac{1}{2}\log\frac{1}{\delta}}$ decreases faster and may decrease to 0 when $T \to +\infty$ given a confidence level $1-\delta$.

6.2.2 Analysis for Smooth Loss Function

Definition 12. A loss function $l(\mathbf{w}; x, y)$ is said to be μ-strongly smooth w.r.t a norm $\|.\|$ iff for all $\boldsymbol{u}, \boldsymbol{v}$ and (x, y) the following condition satisfies

$$l(\boldsymbol{v}; x, y) \leq l(\boldsymbol{u}; x, y) + l'(\boldsymbol{u}; x, y)^{\mathsf{T}}(\boldsymbol{v} - \boldsymbol{u}) + \frac{\mu}{2}\|\boldsymbol{v} - \boldsymbol{u}\|^2$$

Another equivalent definition of μ-strongly smooth function is

$$\left\| l'(\boldsymbol{u}; x, y) - l'(\boldsymbol{v}; x, y) \right\|_* \leq \mu \|\boldsymbol{v} - \boldsymbol{u}\|$$

where $\|.\|_*$ is used to represent the dual norm of the norm $\|.\|$.
It is well-known that

- ℓ_2 loss is 1-strongly smooth w.r.t $\|.\|_2$.
- Logistic loss is 1-strongly smooth w.r.t $\|.\|_2$.
- τ-smooth Hinge loss (Shalev-Shwartz and Zhang, 2013) is $\frac{1}{\tau}$-strongly smooth w.r.t $\|.\|_2$.

Theorem 13. *Assume that ℓ_2, Logistic, or τ-smooth Hinge loss is used, let us denote $L = \frac{\lambda}{2} + 1$, $\frac{\lambda}{2} + 1$, or $\frac{\lambda}{2} + \tau^{-1}$ respectively. Let us define the gap by d_T as in Theorem 10. Let r be any number randomly picked from $\{(1-\alpha)T + 1, 2, \ldots, T\}$. With the probability at least $(1 - \delta)$, the following statement holds*

$$f(\mathbf{w}_r) - f(\mathbf{w}^*) \leq \frac{H \log(1/(1-\alpha))}{2\lambda\alpha T} + d_T + \frac{L M_T^\alpha}{2}\sqrt{\frac{1}{2}\log\frac{1}{\delta}}$$

where $M_T^\alpha = \max_{(1-\alpha)T+1 \leq t \leq T} \|\mathbf{w}_t - \mathbf{w}^*\|$.

Remark 14. Theorem 13 extends Theorem 10 for the case of smooth loss function. This allows the gap $\frac{H \log(1/(1-\alpha))}{2\lambda\alpha T} + \frac{L M_T^\alpha}{2}\sqrt{\frac{1}{2}\log\frac{1}{\delta}}$ to be quantified more precisely regarding the discrepancy in the model itself rather than that in the objective function. The gap $\frac{H \log(1/(1-\alpha))}{2\lambda\alpha T} + \frac{L M_T^\alpha}{2}\sqrt{\frac{1}{2}\log\frac{1}{\delta}}$ could possibly decrease rapidly when T approaches $+\infty$.

Algorithms	Regret	Budget
Forgetron (Dekel et al., 2005)	*NA*	*MB*
PA-I, II (Crammer et al., 2006)	*NA*	*NB*
Randomized Budget Perceptron (Cavallanti et al., 2007)	*NA*	*NB*
Projection (Orabona et al., 2009)	*NA*	*AB*
Kernelized Pegasos (Shalev-Shwartz et al., 2011)	$\mathrm{O}\left(\frac{\log(T)}{T}\right)$	*NB*
Budgeted SGD (Wang et al., 2012)	$\mathrm{O}\left(\frac{\log(T)}{T}\right)$	*MB*
Fourier OGD (Lu et al., 2015)	$\mathrm{O}\left(\frac{1}{\sqrt{T}}\right)$	*MB*
Nystrom OGD (Lu et al., 2015)	$\mathrm{O}\left(\frac{1}{\sqrt{T}}\right)$	*MB*
AVM (average output)	$\mathrm{O}\left(\frac{\log(T)}{T}\right)$	*AB*
AVM (α-suffix average output)	$\mathrm{O}\left(\frac{1}{T}\right)$	*AB*

Table 1: Comparison on the regret bounds and the budget sizes of the kernel online algorithms. On the column of budget size, NB stands for *Not Bound* (i.e., the model size is not bounded and learning method is vulnerable to the curse of kernelization), MB stands for *Manual Bound* (i.e., the model size is manually bounded by a predefined budget), and AB is an abbreviation of *Automatic Bound* (i.e., the model size is automatically bounded and this model size is automatically inferred).

To end this section, we present the regret bound and the obtained budget size for our AVM(s) together with those of algorithms listed in Table 1. We note that some early works on online kernel learning mainly focused on the mistake rate and did not present any theoretical results regarding the regret bounds.

6.2.3 Upper Bound of Model Size

In what follows, we present the theoretical results regarding the model size and sparsity level of our proposed AVM. Theorem 15 shows that AVM offers a high level of freedom to control the model size. Especially, if we use the always-on setting (i.e., $\mathbb{P}(Z_t = 1) = 1, \forall t$), the model size is bounded regardless of the data distribution and data arrival order.

Theorem 15. *Let us denote $\mathbb{P}(Z_t = 1) = p_t$, $\mathbb{P}(Z_t = 0) = q_t$, and the number of cells generated after the iteration t by M_t. If we define the model size, i.e., the size of support set, after the iteration t by S_t, the following statement holds*

$$\mathbb{E}[S_T] \leq \sum_{t=1}^{T} q_t + \sum_{t=1}^{T} p_t \mathbb{E}[M_t - M_{t-1}] \leq \sum_{t=1}^{T} q_t + \mathbb{E}[M_T]$$

Specially, if we use some specific settings for p_t, we can bound the model size $\mathbb{E}[S_t]$ accordingly as follows
i) If $p_t = 1, \forall t$ then $\mathbb{E}[S_T] \leq \mathbb{E}[M_T] \leq |\mathcal{P}|$, where $|\mathcal{P}|$ specifies the size of the partition $\mathcal{P}$, i.e., its number of cells.
ii) If $p_t = \max\left(0, 1 - \frac{\beta}{t}\right), \forall t$ then $\mathbb{E}[S_T] \leq \beta(\log(T) + 1) + \mathbb{E}[M_T]$.
iii) If $p_t = \max\left(0, 1 - \frac{\beta}{t^\rho}\right), \forall t$, where $0 < \rho < 1$, then $\mathbb{E}[S_T] \leq \frac{\beta T^{1-\rho}}{1-\rho} + \mathbb{E}[M_T]$.

iv) If $p_t = \max\left(0, 1 - \frac{\beta}{t^\rho}\right), \forall t$, *where* $\rho > 1$, *then* $\mathbb{E}[S_T] \leq \beta\zeta(\rho) + \mathbb{E}[M_T] \leq \beta\zeta(\rho) + |\mathcal{P}|$, *where* $\zeta(.)$ *is* ζ*- Riemann function defined by the integral* $\zeta(s) = \frac{1}{\Gamma(s)} \int_0^{+\infty} \frac{t^{s-1}}{e^s - 1} dt$.

Remark 16. We use two parameters β and ρ to flexibly control the rate of approximation p_t. It is evident that when β increases, the rate of approximation decreases and consequently the model size and accuracy increase. On the other hand, when ρ increases, the rate of approximation increases as well and it follows that the model size and accuracy decreases. We conducted experiment to investigate how the variation of these two parameters influence the model size and accuracy (cf. Section 9.2).

Remark 17. The items i) and iv) in Theorem 15 indicate that if $\mathbb{P}(Z_t = 1) = p_t = \max\left(0, 1 - \frac{\beta}{t^\rho}\right)$, where $\rho > 1$ or $\rho = +\infty$, then the model size is bounded by $\beta\zeta(\rho) + |\mathcal{P}|$ (by convention we define $\zeta(+\infty) = 0$). In fact, the tight upper bound is $\beta\zeta(\rho) + \mathbb{E}[M_T]$, where M_T is the number of unique cells used so far. It is empirically proven that M_T could be very small comparing with T and $|\mathcal{P}|$. In addition, since all support sets of $\mathbf{w}_t$ $(1 \leq t \leq T)$ are all lain in the core set, if we output the average $\overline{\mathbf{w}}_T = \frac{\sum_{t=1}^T \mathbf{w}_t}{T}$ or α-suffix average $\overline{\mathbf{w}}_T^\alpha = \frac{1}{\alpha T} \sum_{t=(1-\alpha)T+1}^T \mathbf{w}_T$, the model size is still bounded.

Remark 18. The items ii) and iii) in Theorem 15 indicate that if $\mathbb{P}(Z_t = 1) = p_t = \max\left(0, 1 - \frac{\beta}{t^\rho}\right)$, where $0 < \rho \leq 1$ then although the model size is not bounded, it would slowly increase comparing with T, i.e., $\log(T)$ or $T^{1-\rho}$ when ρ is around 1.

6.3 Construction of δ-Coverage

In this section, we return to the construction of δ-coverage defined in Section 6.1 and present two methods to construct a finite δ-coverage. The first method employs hypersphere cells (cf. Algorithm 3) whereas the second method utilizes the hyperrectangle cells (cf. Algorithm 4). In these two methods, the cells in coverage are constructed on the fly when the incoming instances arrive. Both are theoretically proven to be a finite coverage.

Algorithm 3 Constructing hypersphere δ-coverage.

$\mathcal{P} = \emptyset$
$n = 0$
for $t = 1, 2, \ldots$ **do**
  Receive (x_t, y_t)
  $i_t = \text{argmin}_{k \leq n} \|x_t - c_k\|$
  if $\|x_t - c_{i_t}\| \geq \delta/2$ **then**
    $n = n + 1$
    $c_n = x_t$
    $i_t = n$
    $\mathcal{P} = \mathcal{P} \cup [\mathcal{B}(c_n, \delta/2)]$
  end if
end for

Algorithm 4 Constructing hyperrectangle δ-coverage.

$\mathcal{P} = \emptyset$
$a = \delta/\sqrt{d}$
$n = 0$
for $t = 1, 2, \ldots$ **do**
  Receive (x_t, y_t)
  $i_t = 0$
  for $i = 1$ to n **do**
    if $\|x_t - c_i\|_\infty < a$ **then**
      $i_t = i$
      break
    end if
  end for
  if $i_t = 0$ **then**
    $n = n + 1$
    $c_n = x_t$
    $i_t = n$
    $\mathcal{P} = \mathcal{P} \cup [\mathcal{R}(c_n, a)]$
  end if
end for

Algorithm 3 employs a collection of open hypersphere cell $\mathcal{B}(c, R)$, which is defined as $\mathcal{B}(c, R) = \{x \in \mathbb{R}^d : \|x - c\| < R\}$, to cover the data domain. Similar to Algorithm 3, Algorithm 4 uses a collection of open hyperrectangle $\mathcal{R}(c, a)$, which is given by $\mathcal{R}(c, a) = \{x \in \mathbb{R}^d : \|x - c\|_\infty < a\}$, to cover the data domain.

Both Algorithms 3 and 4 are constructed in the common spirit: if the incoming instance (x_t, y_t) is outside all current cells, a new cell whose centre or vertex is this instance is generated. It is noteworthy that the variable i_t in these two algorithms specifies the cell that contains the new incoming instance and is the same as itself in Algorithm 2.

Theorem 19 establishes that regardless of the data distribution and data arrival order, Algorithms 3 and 4 always generate a finite δ-coverage which implies a bound on the model size of AVM. It is noteworthy at this point that in some scenarios of data arrival, Algorithms 3 and 4 might not generate a coverage for the entire space $\mathcal{X}$. However, since the generated sequence $\{x_t\}_t$ cannot be outside the set $\cup_i \mathcal{B}(c_i, \delta)$ and $\cup_i \mathcal{R}(c_i, \delta)$, without loss of generality we can restrict $\mathcal{X}$ to $\cup_i \mathcal{B}(c_i, \delta)$ or $\cup_i \mathcal{R}(c_i, \delta)$ by assuming that $\mathcal{X} = \cup_i \mathcal{B}(c_i, \delta)$ or $\mathcal{X} = \cup_i \mathcal{R}(c_i, \delta)$.

Theorem 19. *Let us consider the coverages formed by the running of Algorithms 3 and 4. If the data domain* $\mathcal{X}$ *is compact (i.e., close and bounded) then these coverages are all finite* δ*-coverages whose sizes are all dependent on the data domain* $\mathcal{X}$ *and independent with the sequence of incoming data instances* (x_t, y_t) *received.*

Remark 20. Theorem 19 also reveals that regardless of the data arrival order, the model size of AVM is always bounded (cf. Remark 17). Referring to the work of (Cucker and Smale, 2002), it is known that this model size cannot exceed $\left(\frac{4D(\mathcal{X})}{\delta}\right)^d$. However with

many possible data arrival orders, the number of active cells or the model size of AVM is significantly smaller than the aforementioned theoretical bound.

6.4 Complexity Analysis

We now present the computational complexity of our AVM(s) with the hypersphere δ-coverage at the iteration t. The cost to find the hypersphere cell in Step 5 of Algorithm 2 is $\mathrm{O}\left(d^2 M_t\right)$. The cost to calculate α_t in Step 6 of Algorithm 2 is $\mathrm{O}\left(S_t\right)$ if we consider the kernel operation as a unit operation. If ℓ_2 loss is used and $\lambda \leq 1$, we need to do a projection onto the hypersphere $\mathcal{B}\left(\mathbf{0}, y_{\max}\lambda^{-1/2}\right)$ which requires the evaluation of the length of the vector $\mathbf{w}_t - \eta_t h_t$ (i.e., $\|\mathbf{w}_t - \eta_t h_t\|$) which costs S_t unit operations using incremental implementation. Therefore, the computational operation at the iteration t of AVM(s) is either $\mathrm{O}\left(d^2 M_t + S_t\right) = \mathrm{O}\left(\left(d^2+1\right) S_t\right)$ or $\mathrm{O}\left(d^2 M_t + S_t + S_t\right) = \mathrm{O}\left(\left(d^2+2\right) S_t\right)$ (since $M_t \leq S_t$).

7. Suitability of Loss Functions

We introduce six types of loss functions that can be used in our proposed algorithm, namely Hinge, Logistic, ℓ_2, ℓ_1, $\varepsilon-$insensitive, and τ-smooth Hinge. We verify that these loss functions satisfying the necessary condition, that is, $\left\|l'\left(\mathbf{w}; x, y\right)\right\| \leq A\left\|\mathbf{w}\right\|^{1/2} + B$ for some appropriate positive numbers A, B (this is required for our problem formulation presented in Section 4).

For comprehensibility, without loss of generality, we assume that $\left\|\Phi\left(x\right)\right\| = K\left(x,x\right)^{1/2} = 1$, $\forall x \in \mathcal{X}$. At the outset of this section, it is noteworthy that for classification task (i.e., Hinge, Logistic, and τ-smooth Hinge cases), the label y is either -1 or 1 which instantly implies $|y| = y^2 = 1$.

- **Hinge loss**

$$l\left(\mathbf{w}; x, y\right) = \max\left\{0, 1 - y\mathbf{w}^{\mathsf{T}}\Phi\left(x\right)\right\}$$
$$l'\left(\mathbf{w}; x, y\right) = -\mathbb{I}_{\left\{y\mathbf{w}^{\mathsf{T}}\Phi(x) \leq 1\right\}} y\Phi\left(x\right)$$

  where $\mathbb{I}_S$ is the indicator function which renders 1 if the logical statement S is true and 0 otherwise.

  Therefore, by choosing $A = 0$, $B = 1$ we have

$$\left\|l'\left(\mathbf{w}; x, y\right)\right\| \leq \left\|\Phi\left(x\right)\right\| \leq 1 = A\left\|\mathbf{w}\right\|^{1/2} + B$$

- **ℓ_2 loss**

  In this case, at the outset we cannot verify that $\left\|l'\left(\mathbf{w}; x, y\right)\right\| \leq A\left\|\mathbf{w}\right\|^{1/2} + B$ for all $\mathbf{w}, x, y$. However, to support the proposed theory, we only need to check that $\left\|l'\left(\mathbf{w}_t; x, y\right)\right\| \leq A\left\|\mathbf{w}_t\right\|^{1/2} + B$ for all $t \geq 1$. We derive as follows

$$l\left(\mathbf{w}; x, y\right) = \frac{1}{2}\left(y - \mathbf{w}^{\mathsf{T}}\Phi\left(x\right)\right)^2$$
$$l'\left(\mathbf{w}; x, y\right) = \left(\mathbf{w}^{\mathsf{T}}\Phi\left(x\right) - y\right)\Phi\left(x\right)$$

$$\left\|l'\left(\mathbf{w}_t; x, y\right)\right\| = \left|\mathbf{w}_t^{\mathsf{T}}\Phi\left(x\right) + y\right|\left\|\Phi\left(x\right)\right\| \leq \left|\mathbf{w}_t^{\mathsf{T}}\Phi\left(x\right)\right| + y_{\max}$$
$$\leq \left\|\Phi\left(x\right)\right\|\left\|\mathbf{w}_t\right\| + y_{\max} \leq A\left\|\mathbf{w}_t\right\|^{1/2} + B$$

  where $B = y_{\max}$ and $A = \begin{cases} y_{\max}^{1/2}\lambda^{-1/4} & \text{if } \lambda \leq 1 \\ y_{\max}^{1/2}\left(\lambda - 1\right)^{-1/2} & \text{otherwise} \end{cases}$.

  Here we note that we make use of the fact that $\left\|\mathbf{w}_t\right\| \leq y_{\max}\left(\lambda - 1\right)^{-1}$ if $\lambda > 1$ (cf. Theorem 25 in Appendix 12) and $\left\|\mathbf{w}_t\right\| \leq y_{\max}\lambda^{-1/2}$ otherwise (cf. Line 12 in Algorithm 2).

- **ℓ_1 loss**

$$l\left(\mathbf{w}; x, y\right) = \left|y - \mathbf{w}^{\mathsf{T}}\Phi\left(x\right)\right|$$
$$l'\left(\mathbf{w}; x, y\right) = \operatorname{sign}\left(\mathbf{w}^{\mathsf{T}}\Phi\left(x\right) - y\right)\Phi\left(x\right)$$

  Therefore, by choosing $A = 0$, $B = 1$ we have

$$\left\|l'\left(\mathbf{w}; x, y\right)\right\| = \left\|\Phi\left(x\right)\right\| \leq 1 = A\left\|\mathbf{w}\right\|^{1/2} + B$$

- **Logistic loss**

$$l\left(\mathbf{w}; x, y\right) = \log\left(1 + \exp\left(-y\mathbf{w}^{\mathsf{T}}\Phi\left(x\right)\right)\right)$$
$$l'\left(\mathbf{w}; x, y\right) = \frac{-y\exp\left(-y\mathbf{w}^{\mathsf{T}}\Phi\left(x\right)\right)\Phi\left(x\right)}{\exp\left(-y\mathbf{w}^{\mathsf{T}}\Phi\left(x\right)\right) + 1}$$

  Therefore, by choosing $A = 0$, $B = 1$ we have

$$\left\|l'\left(\mathbf{w}; x, y\right)\right\| < \left\|\Phi\left(x\right)\right\| \leq 1 = A\left\|\mathbf{w}\right\|^{1/2} + B$$

- **ε-insensitive loss**

$$l\left(\mathbf{w}; x, y\right) = \max\left\{0, \left|y - \mathbf{w}^{\mathsf{T}}\Phi\left(x\right)\right| - \varepsilon\right\}$$
$$l'\left(\mathbf{w}; x, y\right) = \mathbb{I}_{\left\{\left|y - \mathbf{w}^{\mathsf{T}}\Phi(x)\right| > \varepsilon\right\}}\operatorname{sign}\left(\mathbf{w}^{\mathsf{T}}\Phi\left(x\right) - y\right)\Phi\left(x\right)$$

  Therefore, by choosing $A = 0$, $B = 1$ we have

$$\left\|l'\left(\mathbf{w}; x, y\right)\right\| \leq \left\|\Phi\left(x\right)\right\| \leq 1 = A\left\|\mathbf{w}\right\|^{1/2} + B$$

- **τ-smooth Hinge loss** Shalev-Shwartz and Zhang (2013)

$$l\left(\mathbf{w}; x, y\right) = \begin{cases} 0 & \text{if } y\mathbf{w}^{\mathsf{T}}\Phi\left(x\right) > 1 \\ 1 - y\mathbf{w}^{\mathsf{T}}\Phi\left(x\right) - \frac{\tau}{2} & \text{if } y\mathbf{w}^{\mathsf{T}}\Phi\left(x\right) < 1 - \tau \\ \frac{1}{2\tau}\left(1 - y\mathbf{w}^{\mathsf{T}}\Phi\left(x\right)\right)^2 & \text{otherwise} \end{cases}$$
$$l'\left(\mathbf{w}; x, y\right) = -\mathbb{I}_{\left\{y\mathbf{w}^{\mathsf{T}}\Phi(x) < 1 - \tau\right\}} y\Phi\left(x\right)$$
$$+ \tau^{-1}\mathbb{I}_{1 - \tau \leq y\mathbf{w}^{\mathsf{T}}\Phi(x) \leq 1}\left(y\mathbf{w}^{\mathsf{T}}\Phi\left(x\right) - 1\right) y\Phi\left(x\right)$$

Therefore, by choosing $A = 0$, $B = 2$, we have

$$\left\| l^{'}\left(\mathbf{w}; x, y\right)\right\| \leq |y| \left\|\Phi\left(x\right)\right\| + \tau^{-1} |y| \left\|\Phi\left(x\right)\right\| \tau \leq 2 = A \left\|\mathbf{w}\right\|^{1/2} + B$$

8. Multiclass Setting

In this section, we show that our proposed framework could also easily extend to the multi-class setting. We base on the work of Crammer and Singer (2002) for multiclass classification to formulate the optimization problem in multi-class setting as

$$\min_{W} \left(f\left(W\right) \triangleq \frac{\lambda}{2} \left\|W\right\|_{2,2}^{2} + \frac{1}{N} \sum_{i=1}^{N} l\left(\mathbf{w}_{y_i}^{\mathsf{T}} \Phi\left(x_i\right) - \mathbf{w}_{z_i}^{\mathsf{T}} \Phi\left(x_i\right)\right)\right)$$

where we have defined

$$z_i = \underset{j \neq y_i}{\operatorname{argmax}}\, \mathbf{w}_j^{\mathsf{T}} \Phi\left(x_i\right),$$

$$W = \left[\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m\right], \left\|W\right\|_{2,2}^{2} = \sum_{j=1}^{m} \left\|\mathbf{w}_j\right\|^2,$$

$$l\left(a\right) = \begin{cases} \max\left(0, 1 - a\right) & \text{Hinge loss} \\ \log\left(1 + e^{-a}\right) & \text{Logistic loss} \end{cases}$$

For the exact update, at the t-th iteration, we receive the instance (x_t, y_t) and modify W as follows

$$\mathbf{w}_j^{(t+1)} = \begin{cases} \frac{t-1}{t} \mathbf{w}_j^{(t)} - \eta_t l^{'}\left(a\right) \Phi\left(x_t\right) & \text{if } j = y_t \\ \frac{t-1}{t} \mathbf{w}_j^{(t)} + \eta_t l^{'}\left(a\right) \Phi\left(x_t\right) & \text{if } j = z_t \\ \frac{t-1}{t} \mathbf{w}_j^{(t)} & \text{otherwise} \end{cases}$$

where $a = \mathbf{w}_{y_t}^{\mathsf{T}} \Phi\left(x_t\right) - \mathbf{w}_{z_t}^{\mathsf{T}} \Phi\left(x_t\right)$ and $l^{'}(a) = -\mathbb{I}_{\{a<1\}}$ or $-1/\left(1 + e^{a}\right)$.

The algorithm for Approximation Vector Machine with multiclass setting proceeds as in Algorithm 5.

Algorithm 5 Multiclass Approximation Vector Machine.

Input: λ, p.s.d. & iso. kernel $K\left(.,.\right)$, δ-coverage $\mathcal{P} = \left(P_i\right)_{i \in I}$

$W_1 = 0$
for $t = 1, \ldots, T$ **do**
  Receive $\left(x_t, y_t\right)$ //$(x_t, y_t) \sim \mathbb{P}_{\mathcal{X},\mathcal{Y}}$ *or* $\mathbb{P}_N$
  $a = \mathbf{w}_{y_t}^{\mathsf{T}} \Phi\left(x_t\right) - \max_{j \neq y_t} \mathbf{w}_j^{\mathsf{T}} \Phi\left(x_t\right)$
  $W^{(t+1)} = \frac{t-1}{t} W^{(t)}$
  Sample a Bernoulli random variable Z_t
  if $Z_t = 1$ **then**
    Find $i_t \in I$ such that $x_t \in P_{i_t}$
    $\mathbf{w}_{y_t}^{(t+1)} = \mathbf{w}_{y_t}^{(t+1)} - \eta_t l^{'}\left(a\right) \Phi\left(c_{i_t}\right)$ //*do approximation*
    $\mathbf{w}_{z_t}^{(t+1)} = \mathbf{w}_{z_t}^{(t+1)} + \eta_t l^{'}\left(a\right) \Phi\left(c_{i_t}\right)$
  else
    $\mathbf{w}_{y_t}^{(t+1)} = \mathbf{w}_{y_t}^{(t+1)} - \eta_t l^{'}\left(a\right) \Phi\left(x_t\right)$
    $\mathbf{w}_{z_t}^{(t+1)} = \mathbf{w}_{z_t}^{(t+1)} + \eta_t l^{'}\left(a\right) \Phi\left(x_t\right)$
  end if
end for

Output $\overline{W}^{(T)} = \frac{\sum_{t=1}^{T} W^{(t)}}{T}$ or $W^{(t+1)}$

9. Experiments

In this section, we conduct comprehensive experiments to quantitatively evaluate the capacity and scalability of our proposed Approximation Vector Machine (AVM) on classification and regression tasks under three different settings:

- *Batch classification*[4]: the regular binary and multiclass classification tasks that follow a standard validation setup, wherein each dataset is partitioned into training set and testing set. The models are trained on the training part, and then their discriminative capabilities are verified on the testing part using classification accuracy measure. The computational costs are commonly measured based on the training time.

- *Online classification*: the binary and multiclass classification tasks that follow a purely online learning setup, wherein there is no division of training and testing sets as in batch setting. The algorithms sequentially receive and process a single data sample turn-by-turn. When an individual data point comes, the models perform prediction to compute the mistake rate first, then use the feature and label information of such data point to continue their learning procedures. Their predictive performances and computational costs are measured basing on the average of mistake rate and execution time, respectively, accumulated in the learning progress on the entire dataset.

- *Online regression*: the regression task that follows the same setting of online classification, except the predictive performances are measured based on the regression error rate accumulated in the learning progress on the entire dataset.

4. This setting is also known as offline classification.

Our main goal is to examine the scalability, classification and regression capabilities of AVMs by directly comparing with those of several recent state-of-the-art batch and online learning approaches using a number of datasets with a wide range of sizes. Our models are implemented in Python with Numpy package. The source code and experimental scripts are published for reproducibility[5]. In what follows, we present the data statistics, experimental setup, results and our observations.

9.1 Data Statistics and Experimental Setup

We use 11 datasets whose statistics are summarized in Table 2. The datasets are selected in a diverse array of sizes in order to clearly expose the differences among scalable capabilities of the models. Five of which (*year*, *covtype*, *poker, KDDCup99*, *airlines*) are large-scale datasets with hundreds of thousands and millions of data points, whilst the rest are ordinal-size databases. Except the *airlines*, all of the datasets can be downloaded from LIBSVM[6] and UCI[7] websites.

Dataset	#training	#testing	#features	#classes	Source
a9a	32,561	16,281	123	2	UCI
w8a	49,749	14,951	300	2	LIBSVM
cod-rna	59,535	271,617	8	2	LIBSVM
ijcnn1	49,990	91,701	22	2	LIBSVM
covtype	522,911	58,101	54	7	LIBSVM
poker	25,010	1,000,000	10	10	UCI
KDDCup99	4,408,589	489,842	41	23	UCI
airlines	5,336,471	592,942	8	2	ASA
Dataset	**#training**	**#testing**	**#features**	**value**	**Source**
casp	45,730	–	9	[0, 1]	UCI
slice	53,500	–	384	[0, 1]	UCI
year	515,345	–	90	[0, 1]	UCI
airlines	5,929,413	–	8	$\mathbb{R}^+$	ASA

Table 2: Data statistics. #training: number of training samples; #testing: number of testing samples.

The airlines dataset is provided by American Statistical Association (ASA[8]). The dataset contains information of all commercial flights in the US from October 1987 to April 2008. The aim is to predict whether a flight will be delayed or not and how long in minutes the flight will be delayed in terms of departure time. The departure delay time is provided in the flight database. A flight is considered *delayed* if its delay time is above 15 minutes, and *non-delayed* otherwise. The average delay of a flight in 2008 was of 56.3 minutes. Following the procedure of (Hensman et al., 2013), we further process the data in two steps. First, we join the data with the information of individual planes basing on their tail numbers in order to obtain the manufacture year. This additional information is provided as a supplemental data source on ASA website. We then extract 8 features of many available fields: the age of the aircraft (computed based on the manufacture year), journey distance, airtime, scheduled departure time, scheduled arrival time, month, day of week and month. All features are normalized into the range $[0, 1]$.

5. https://github.com/tund/avm.
6. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.
7. https://archive.ics.uci.edu/ml/datasets.html.
8. The data can be downloaded from http://stat-computing.org/dataexpo/2009/.

In batch classification experiments, we follow the original divisions of training and testing sets in LIBSVM and UCI sites wherever available. For *KDDCup99*, *covtype* and *airlines* datasets, we split the data into 90% for training and 10% for testing. In online classification and regression tasks, we either use the entire datasets or concatenate training and testing parts into one. The online learning algorithms are then trained in a single pass through the data. In both batch and online settings, for each dataset, the models perform 10 runs on different random permutations of the training data samples. Their prediction results and time costs are then reported by taking the average with the standard deviation of the results over these runs.

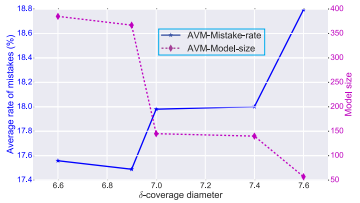
For comparison, we employ some baseline methods that will be described in the following sections. Their C++ implementations with Matlab interfaces are published as a part of LIBSVM, BudgetedSVM[9] and LSOKL[10] toolboxes. Throughout the experiments, we utilize RBF kernel, i.e., $K\left(x, x'\right) = \exp\left(-\gamma \left\|x - x'\right\|^2\right)$ for all algorithms including ours. We use hypersphere strategy to construct the δ-coverage (cf. Section 6.3), due to its better performance than that of hyperrectangle approach during model evaluation. All experiments are conducted using a Windows machine with 3.46GHz Xeon processor and 96GB RAM.

9.2 Model Evaluation on The Effect of Hyperparameters

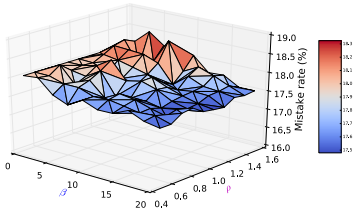
In the first experiment, we investigate the effect of hyperparameters, i.e., δ-coverage diameter, sampling parameters β and ρ (cf. Section 6.2.3) on the performance of AVMs. Particularly, we conduct an initial analysis to quantitatively evaluate the sensitivity of these hyperparameters and their impact on the predictive accuracy and model size. This analysis provides a heuristic approach to find the best setting of hyperparameters. Here the AVM with Hinge loss is trained following the online classification scheme using two datasets *a9a* and *cod-rna*.

To find the plausible range of coverage diameter, we use a heuristic approach as follows. First we compute the mean and standard deviation of pairwise Euclidean distances between any two data samples. Treating the mean as the radius, the coverage diameter is then varied around twice of this mean bounded by twice of the standard deviation. Fig. 2a and Fig. 3a report the average mistake rates and model sizes of AVMs with respect to (w.r.t) these values for datasets *a9a* and *cod-rna*, respectively. Here we set $\beta = 0$ and $\rho = 1.0$. There is a consistent pattern in both figures: the classification errors increase for larger δ whilst the model sizes decrease. This represents the trade-off between model performance and model size via the model coverage. To balance the performance and model size, in these cases, we can choose $\delta = 7.0$ for *a9a* data and $\delta = 1.0$ for *cod-rna* data.

9. http://www.dabi.temple.edu/budgetedsvm/index.html.
10. http://lsokl.stevenhoi.com/.

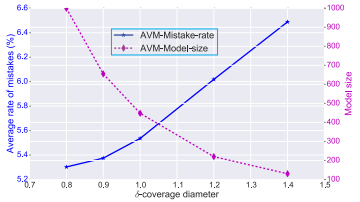


(a) The effect of δ-coverage diameter on the mistake rate and model size.

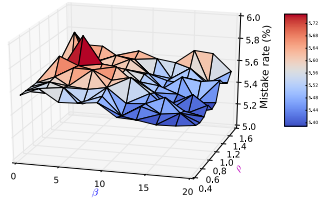


(b) The effect of β and ρ on the classification mistake rate. $\beta = 0$ means always approximating.

Figure 2: Performance evaluation of AVM with Hinge loss trained using *a9a* dataset with different values of hyperparameters.



(a) The effect of δ-coverage diameter on the mistake rate and model size.



(b) The effect of β and ρ on the classification mistake rate. $\beta = 0$ means always approximating.

Figure 3: Performance evaluation of AVM with Hinge loss trained using *cod-rna* dataset with different values of hyperparameters.

Fixing the coverage diameters, we vary β and ρ in 10 values monotonically increasing from 0 to 10 and from 0.5 to 1.5, respectively, to evaluate the classification performance. The smaller β and larger ρ indicate that the machine approximates the new incoming data more frequently, resulting in less powerful prediction capability. This can be observed in Fig. 2b and Fig. 3b, which depict the average mistake rates in 3D as a function of these values for dataset *a9a* and *cod-rna*. Here $\beta = 0$ means that the model always performs approximation without respect to the value of ρ. From these visualizations, we found that the AVM with always-on approximation mode still can achieve fairly comparable classification results. Thus we set $\beta = 0$ for all following experiments.

9.3 Batch Classification

We now examine the performances of AVMs in classification task following batch mode. We use eight datasets: *a9a*, *w8a*, *cod-rna*, *KDDCup99*, *ijcnn1*, *covtype*, *poker* and *airlines*

(delayed and non-delayed labels). We create two versions of our approach: AVM with Hinge loss (AVM-Hinge) and AVM with Logistic loss (AVM-Logit). It is noteworthy that the Hinge loss is not a smooth function with undefined gradient at the point that the classification confidence $yf(x) = 1$. Following the sub-gradient definition, in our experiment, we compute the gradient given the condition that $yf(x) < 1$, and set it to 0 otherwise.

Baselines. For discriminative performance comparison, we recruit the following state-of-the-art baselines to train kernel SVMs for classification in batch mode:

- LIBSVM: one of the most widely-used and state-of-the-art implementations for batch kernel SVM solver (Chang and Lin, 2011). We use the one-vs-all approach as the default setting for the multiclass tasks;
- LLSVM: low-rank linearization SVM algorithm that approximates kernel SVM optimization by a linear SVM using low-rank decomposition of the kernel matrix (Zhang et al., 2012);
- BSGD-M: budgeted stochastic gradient descent algorithm which extends the Pegasos algorithm (Shalev-Shwartz et al., 2011) by introducing a merging strategy for support vector budget maintenance (Wang et al., 2012);
- BSGD-R: budgeted stochastic gradient descent algorithm which extends the Pegasos algorithm (Shalev-Shwartz et al., 2011) by introducing a removal strategy for support vector budget maintenance (Wang et al., 2012);
- FOGD: Fourier online gradient descent algorithm that applies the random Fourier features for approximating kernel functions (Lu et al., 2015);
- NOGD: Nystrom online gradient descent (NOGD) algorithm that applies the Nystrom method to approximate large kernel matrices (Lu et al., 2015).

Hyperparameters setting. There are a number of different hyperparameters for all methods. Each method requires a different set of hyperparameters, e.g., the regularization parameters (C in LIBSVM, λ in Pegasos and AVM), the learning rates (η in FOGD and NOGD), the coverage diameter (δ in AVM) and the RBF kernel width (γ in all methods). Thus, for a fair comparison, these hyperparameters are specified using cross-validation on training subset.

Particularly, we further partition the training set into 80% for learning and 20% for validation. For large-scale databases, we use only 1% of training set, so that the searching can finish within an acceptable time budget. The hyperparameters are varied in certain ranges and selected for the best performance on the validation set. The ranges are given as follows: $C \in \{2^{-5}, 2^{-3}, ..., 2^{15}\}$, $\lambda \in \{2^{-4}/N, 2^{-2}/N, ..., 2^{16}/N\}$, $\gamma \in \{2^{-8}, 2^{-4}, 2^{-2}, 2^{0}, 2^{2}, 2^{4}, 2^{8}\}$, $\eta \in \{16.0, 8.0, 4.0, 2.0, 0.2, 0.02, 0.002, 0.0002\}$ where N is the number of data points. The coverage diameter δ of AVM is selected following the approach described in Section 9.2. For the budget size B in NOGD and Pegasos algorithm, and the feature dimension D in FOGD for each dataset, we use identical values to those used in Section 7.1.1 of Lu et al. (2015).

Results. The classification results, training and testing time costs are reported in Table 3. Overall, the batch algorithms achieve the highest classification accuracies whilst those of online algorithms are lower but fairly competitive. The online learning models, however, are much sparser, resulting in a substantial speed-up, in which the training time costs and model sizes of AVMs are smallest with orders of magnitude lower than those of the standard batch methods. More specifically, the LIBSVM outperforms the other approaches in most of datasets, on which its training phase finishes within the time limit (i.e., two hours), except for the *ijcnn1* data wherein its testing score is less accurate but very close to that of BSGD-M. The LLSVM achieves good results which are slightly lower than those of the state-of-the-art batch kernel algorithm. The method, however, does not support multiclass classification. These two batch algorithms – LIBSVM and LLSVM could not be trained within the allowable amount of time on large-scale datasets (e.g., *airlines*), thus are not scalable.

Furthermore, six online algorithms in general have significant advantages against the batch methods in computational efficiency, especially when running on large-scale datasets. Among these algorithms, the BSGD-M (Pegasos+merging) obtains the highest classification scores, but suffers from a high computational cost. This can be seen in almost all datasets, especially for the airlines dataset on which its learning exceeds the time limit. The slow training of BSGD-M is caused by the merging step with computational complexity $\mathcal{O}\left(B^2\right)$ (B is the budget size). By contrast, the BSGD-R (Pegasos+removal) runs faster than the merging approach, but suffers from very high inaccurate results due to its naive budget maintenance strategy, that simply discards the most redundant support vector which may contain important information.

In terms of predictive performance, our proposed methods outperform the recent advanced online learning algorithms – FOGD and NOGD in most scenarios. The AVM-based models are able to achieve slightly less accurate but fairly comparable results compared with those of the state-of-the-art LIBSVM algorithm. In terms of sparsity and speed, the AVMs are the fastest ones in the training and testing phases in all cases thanks to their remarkable smaller model sizes. The difference between the training speed of our AVMs and that of two approaches varies across datasets. The gap is more significant for datasets with higher dimensional feature spaces. This is expected because the procedure to compute random features for each data point of FOGD involves *sin* and *cos* operators which are costly. These facts indicate that our proposed online kernel learning algorithms are both efficient and effective in solving large-scale kernel classification problems. Thus we believe that the AVM is the fast alternative to the existing SVM solvers for large-scale classification tasks.

Finally, comparing two versions of AVMs, it can be seen that the discriminative performances of AVM with Logistic loss are better than those of AVM with Hinge loss in most of datasets. This is because the Logistic function is smoother than the Hinge function, whilst the Hinge loss encourages sparsity of the model. The AVM-Logit, however, contains additional exponential operators, resulting in worse training time.

Table 3: Classification performance of our AVMs and the baselines in batch mode. The notation $[\delta \mid S \mid B \mid D]$, next to the dataset name, denotes the diameter δ, the model size S of AVM-based models, the budget size B of budgeted algorithms, and the number of random features D of FOGD, respectively. The accuracy is reported in percent (%), the training time and testing time are in second. The best performance is in **bold**. It is noteworthy that the LLSVM does not support multiclass classification and we terminate all runs exceeding the limit of two hours, therefore some results are unavailable.

Dataset $[\delta \mid S \mid B \mid D]$	***a9a*** [7.0 | 135 | 1,000 | 4,000]			***w8a*** [13.0 | 131 | 1,000 | 4,000]		
Algorithm	Train	Test	Accuracy	Train	Test	Accuracy
LIBSVM	84.57	22.23	**84.92**	50.96	2.95	**99.06**
LLSVM	50.73	8.73	83.00	92.19	10.41	98.64
BSGD-M	232.59	2.88	84.76±0.16	264.70	5.16	98.17±0.07
BSGD-R	90.48	2.72	80.26±3.38	253.30	4.98	97.10±0.04
FOGD	15.99	2.87	81.15±5.05	32.16	3.55	97.92±0.38
NOGD	82.40	0.60	82.33±2.18	374.87	0.65	98.06±0.18
AVM-Hinge	**4.96**	**0.25**	83.55±0.50	**11.84**	**0.52**	96.87±0.28
AVM-Logit	5.35	**0.25**	83.83±0.34	12.54	**0.52**	96.96±0.00
Dataset $[\delta \mid S \mid B \mid D]$	***cod-rna*** [1.0 | 436 | 400 | 1,600]			***ijcnn1*** [1.0 | 500 | 1,000 | 4,000]		
Algorithm	Train	Test	Accuracy	Train	Test	Accuracy
LIBSVM	114.90	85.34	**96.39**	38.63	11.17	97.35
LLSVM	20.17	19.38	94.16	40.62	54.22	96.99
BSGD-M	90.62	5.66	95.67±0.21	93.05	6.13	**97.69±0.11**
BSGD-R	19.31	5.48	66.83±0.11	41.70	7.07	90.90±0.18
FOGD	7.62	11.95	92.65±4.20	7.31	10.10	90.64±0.07
NOGD	9.81	3.24	91.83±3.35	21.58	3.68	90.43±1.22
AVM-Hinge	**6.52**	**2.69**	94.38±1.16	**6.47**	2.71	91.14±0.71
AVM-Logit	7.03	2.86	93.10±2.11	6.86	**2.67**	91.19±0.95
Dataset $[\delta \mid S \mid B \mid D]$	***covtype*** [3.0 | 59 | 400 | 1,600]			***poker*** [12.0 | 393 | 1,000 | 4,000]		
Algorithm	Train	Test	Accuracy	Train	Test	Accuracy
LIBSVM	–	–	–	40.03	932.58	**57.91**
LLSVM	–	–	–	–	–	–
BSGD-M	2,413.15	3.75	**72.26±0.16**	414.09	123.57	54.10±0.22
BSGD-R	418.68	3.02	61.09±1.69	35.76	102.84	52.14±1.05
FOGD	69.94	2.45	59.34±5.85	9.61	101.29	46.62±5.00
NOGD	679.50	0.76	68.20±2.96	118.54	36.84	54.65±0.27
AVM-Hinge	**60.27**	0.26	64.31±0.37	3.86	8.21	55.49±0.13
AVM-Logit	61.92	**0.22**	64.42±0.34	**3.36**	**7.54**	55.60±0.17
Dataset $[\delta \mid S \mid B \mid D]$	***KDDCup99*** [3.0 | 115 | 200 | 400]			***airlines*** [1.0 | 388 | 1,000 | 4,000]		
Algorithm	Train	Test	Accuracy	Train	Test	Accuracy
LIBSVM	4,380.58	661.04	**99.91**	–	–	–
LLSVM	–	–	–	–	–	–
BSGD-M	2,680.58	21.25	99.73±0.00	–	–	–
BSGD-R	1,644.25	14.33	39.81±2.26	4,741.68	29.98	80.27±0.06
FOGD	706.20	22.73	99.75±0.11	1,085.73	861.52	80.37±0.21
NOGD	3,726.21	3.11	99.80±0.02	3,112.08	18.53	74.83±0.20
AVM-Hinge	**554.42**	**2.75**	99.82±0.05	**586.90**	6.55	**80.72±0.00**
AVM-Logit	576.76	2.80	99.72±0.06	642.23	**6.10**	**80.72±0.00**

9.4 Online Classification

The next experiment investigates the performance of the AVMs in online classification task where individual data point continuously come turn-by-turn in a stream. Here we also use eight datasets and two versions of our approach: AVM with Hinge loss (AVM-Hinge) and AVM with Logistic loss (AVM-Logit) which are used in batch classification setting (cf. Section 9.3).

Baselines. We recruit the two widely-used algorithms – Perceptron and OGD for regular online kernel classification without budget maintenance and 8 state-of-the-art budget online kernel learning methods as follows:

- Perceptron: the kernelized variant without budget of Perceptron algorithm (Freund and Schapire, 1999);
- OGD: the kernelized variant without budget of online gradient descent (Kivinen et al., 2004);
- RBP: a budgeted Perceptron algorithm using random support vector removal strategy (Cavallanti et al., 2007);
- Forgetron: a kernel-based Perceptron maintaining a fixed budget by discarding oldest support vectors (Dekel et al., 2005);
- Projectron: a Projectron algorithm using the projection strategy (Orabona et al., 2009);
- Projectron++: the aggressive version of Projectron algorithm (Orabona et al., 2009);
- BPAS: a budgeted variant of Passive-Aggressive algorithm with simple SV removal strategy (Wang and Vucetic, 2010);
- BOGD: a budgeted variant of online gradient descent algorithm using simple SV removal strategy (Zhao et al., 2012);
- FOGD and NOGD: described in Section 9.3.

Hyperparameters setting. For each method learning on each dataset, we follow the same hyperparameter setting which is optimized in the batch classification task. For time efficiency, we only include the fast algorithms FOGD, NOGD and AVMs for the experiments on large-scale datasets. The other methods would exceed the time limit when running on such data.

Results. Fig. 4 and Fig. 5 shows the relative performance convergence w.r.t classification error and computation cost of the AVMs in comparison with those of the baselines. Combining these two figures, we compare the average mistake rate and running time in Fig. 6. Table 4 reports the final average results in detailed numbers after the methods see all data samples. It is worthy to note that for the four biggest datasets (*KDDCup99*, *covtype*, *poker*, *airlines*) that consist of millions data points, we exclude the non-budgeted online learning algorithm because of their substantially expensive time costs. From these results, we can draw some observations as follows.

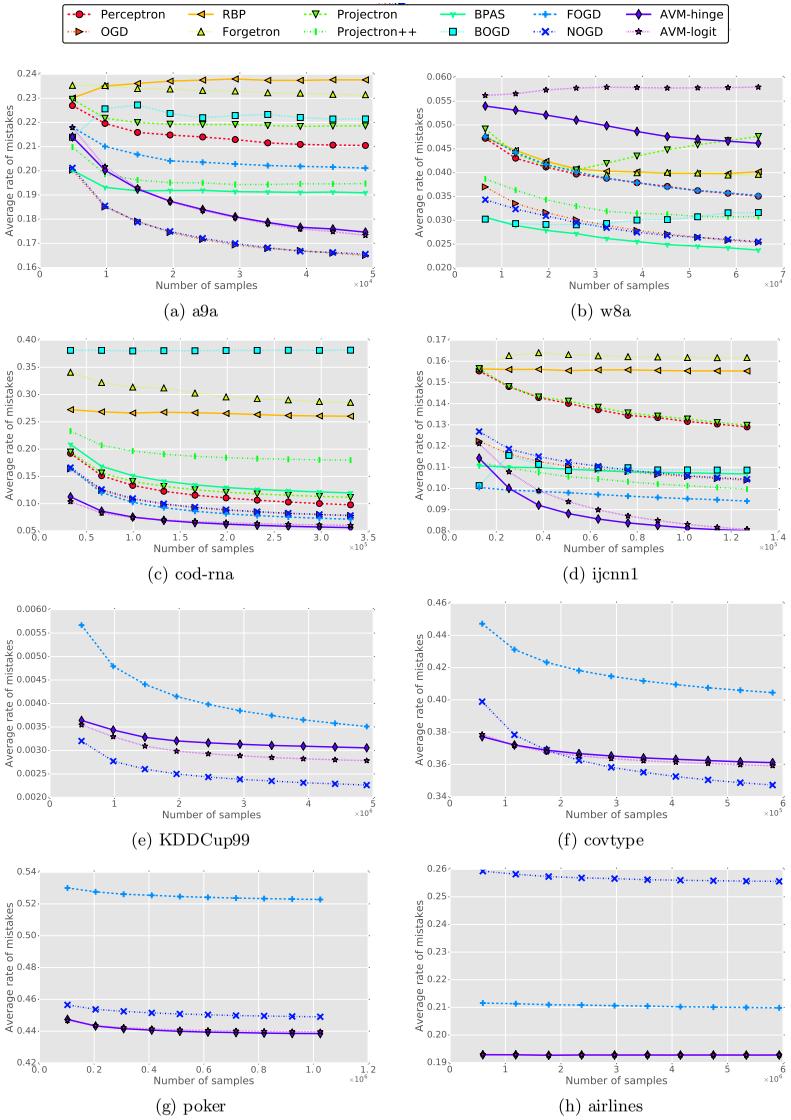
First of all, as can be seen from Fig. 4, there are three groups of algorithms that have different learning progresses in terms of classification mistake rate. The first group includes the BOGD, Projectron and Forgetron that have the error rates fluctuating at the beginning, but then being stable till the end. In the meantime, the rates of the models in the second group, including Perceptron, OGD, RBP, Projectron++ and BPAS, quickly saturate at a plateau after these methods see a few portions, i.e., one-tenth to two-tenth, of the data. By contrast, the last group includes the recent online learning approaches – FOGD, NOGD, and our proposed ones – AVM-Hinge, AVM-Logit, that regularly perform better as more data points come. Exceptionally, for the dataset *w8a*, the classification errors of the methods in the first group keep increasing after seeing four-tenth of the data, whilst those of the last group are unexpectedly worse.

Second, Fig. 6 plots average mistake rate against computational cost, which shows similar patterns as in the our first observation. In addition, it can be seen from Fig. 5 that all algorithms have normal learning pace in which the execution time is accumulated over the learning procedure. Only the Projectron++ is slow at the beginning but then performs faster after receiving more data.

According to final results summarized in Table 4, the budgeted online approaches show efficacies with substantially faster computation than the ones without budgets. This is more obvious for larger datasets wherein the execution time costs of our proposed models are several orders of magnitude lower than those of regular online algorithms. This is because the coverage scheme of AVMs impressively boost their model sparsities, e.g., using $\delta = 3$ resulting in 115 core points for dataset *KDDCup99* consisting of $4,408,589$ instances, and using $\delta = 1$ resulting in 388 core points for dataset *airlines* containing $5,336,471$ data samples.

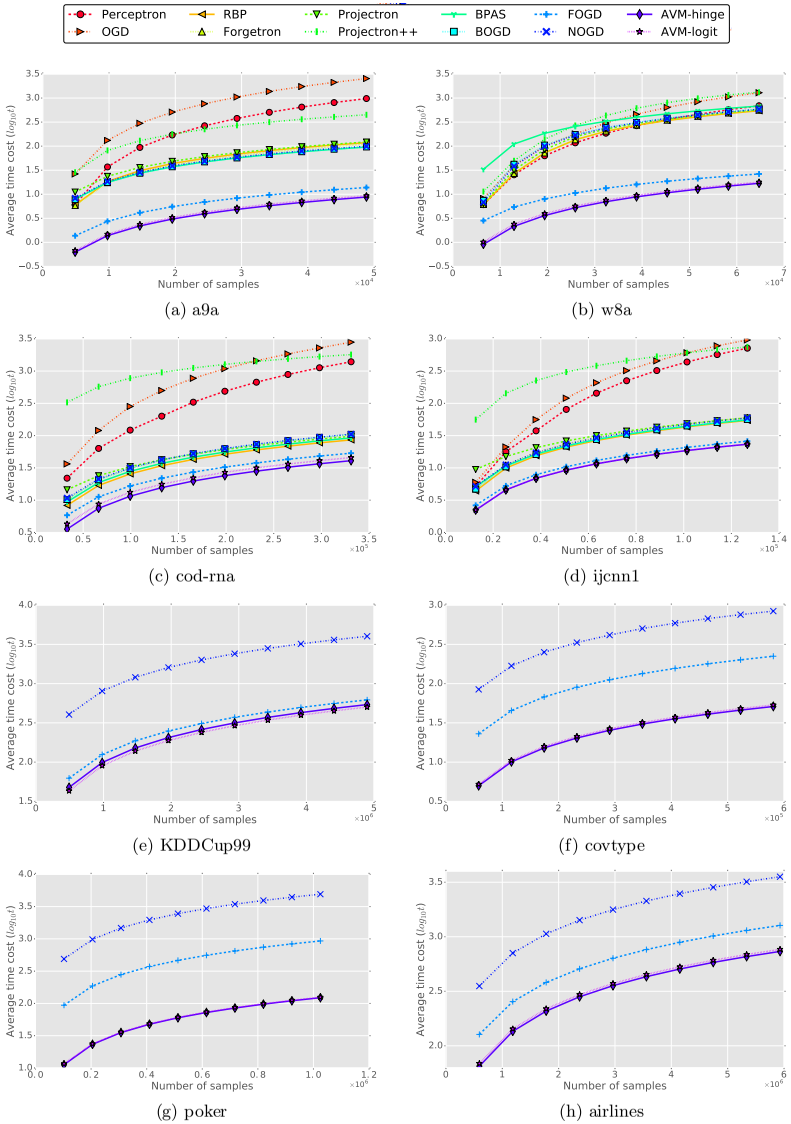
For classification capability, the non-budgeted methods only surpass the budgeted ones for the smallest dataset, that is, the OGD obtains the best performance for *a9a* data. This again demonstrates the importance of exploring budget online kernel learning algorithms. Between the two non-budgeted algorithms, the OGD achieves considerably better error rates than the Perceptron. The method, however, must perform much more expensive updates, resulting in a significantly larger number of support vectors and significantly higher computational time costs. This represents the trade-off between classification accuracy and computational complexity of the OGD.

Furthermore, comparing the performance of different existing budgeted online kernel learning algorithms, the AVM-Hinge and AVM-Logit outperform others in both discriminative performance and computation efficiency for almost all datasets. In particular, the AVM-based methods achieve the best mistake rates – 5.61 ± 0.17, 8.01 ± 0.18, 43.85 ± 0.09, 19.28 ± 0.00 for the *cod-rna*, *ijcnn1*, *poker* and *airlines* data, that are, respectively, 27.5%, 17.5%, 2.4%, 8.8% lower than the error rates of the second best models – two recent approaches FOGD and NOGD. On the other hand, the computation costs of the AVMs are significantly lower with large margins of hundreds of percents for large-scale databases *covtype*, *poker*, and *airlines* as shown in Table 4.



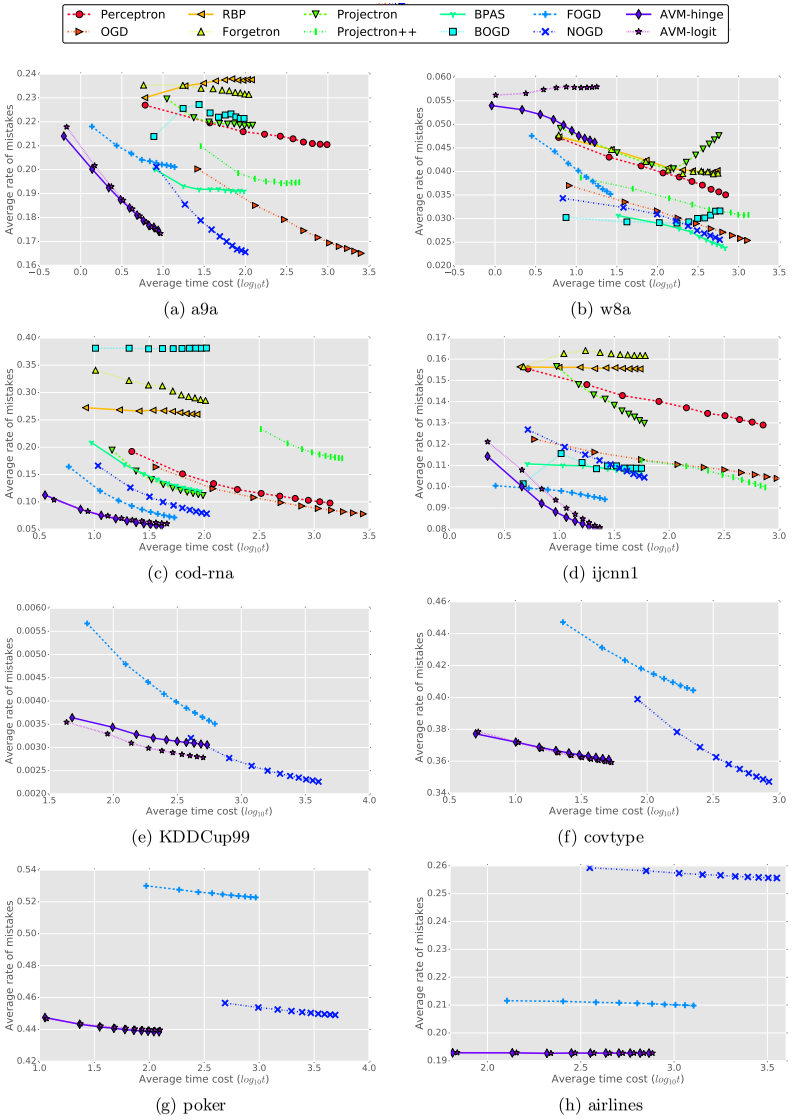
(a) a9a (b) w8a (c) cod-rna (d) ijcnn1 (e) KDDCup99 (f) covtype (g) poker (h) airlines

Figure 4: Convergence evaluation of online classification tasks: the average rate of mistakes as a function of the number of samples seen by the models. (Best viewed in colors).



(a) a9a (b) w8a (c) cod-rna (d) ijcnn1 (e) KDDCup99 (f) covtype (g) poker (h) airlines

Figure 5: Convergence evaluation of online classification task: the average time costs (seconds shown in the logarithm with base 10) as a function of the number of samples seen by the models. (Best viewed in colors).

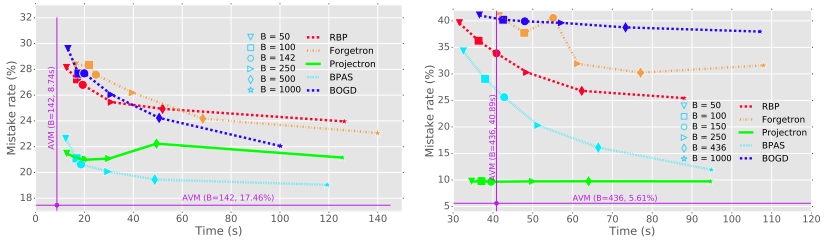


(a) a9a (b) w8a (c) cod-rna (d) ijcnn1 (e) KDDCup99 (f) covtype (g) poker (h) airlines

Figure 6: Average mistake rate vs. time cost for online classification. The average time (seconds) is shown in the logarithm with base 10. (Best viewed in colors).

Table 4: Classification performance of our proposed methods and the baselines in online mode. Note that δ, B and D are set to be the same as in batch classification tasks (cf., Section 9.3). The mistake rate is reported in percent (%) and the execution time is in second. The best performance is in **bold**.

Dataset [S]	***a9a*** [142]		***w8a*** [131]	
Algorithm	Mistake Rate	Time	Mistake Rate	Time
Perceptron	21.05±0.12	976.79	3.51±0.03	691.80
OGD	**16.50±0.06**	2,539.46	2.54±0.03	1,290.13
RBP	23.76±0.21	118.25	4.02±0.07	544.83
Forgetron	23.15±0.34	109.71	3.96±0.10	557.75
Projectron	21.86±1.73	122.08	4.76±1.13	572.20
Projectron++	19.47±2.22	449.20	3.08±0.63	1321.93
BPAS	19.09±0.17	95.81	**2.37±0.02**	681.46
BOGD	22.14±0.25	96.11	3.16±0.08	589.47
FOGD	20.11±0.10	13.79	3.52±0.05	26.40
NOGD	16.55±0.07	99.54	2.55±0.05	585.23
AVM-Hinge	17.46±0.12	**8.74**	4.62±0.78	**16.89**
AVM-Logit	17.33±0.16	9.31	5.80±0.02	17.86
Dataset [S]	***cod-rna*** [436]		***ijcnn1*** [500]	
Algorithm	Mistake Rate	Time	Mistake Rate	Time
Perceptron	9.79±0.04	1,393.56	12.85±0.09	727.90
OGD	7.81±0.03	2,804.01	10.39±0.06	960.44
RBP	26.02±0.39	85.84	15.54±0.21	54.29
Forgetron	28.56±2.22	102.64	16.17±0.26	60.54
Projectron	11.16±3.61	97.38	12.98±0.23	59.37
Projectron++	17.97±15.60	1,799.93	9.97±0.09	749.70
BPAS	11.97±0.09	92.08	10.68±0.05	55.44
BOGD	38.13±0.11	104.60	10.87±0.18	55.99
FOGD	7.15±0.03	53.45	9.41±0.03	25.93
NOGD	7.83±0.06	105.18	10.43±0.08	59.36
AVM-Hinge	**5.61±0.17**	**40.89**	**8.01±0.18**	**23.26**
AVM-Logit	6.01±0.20	45.67	8.07±0.20	23.36
Dataset [S]	***KDDCup99*** [115]		***covtype*** [59]	
Algorithm	Mistake rate	Time	Mistake rate	Time
FOGD	0.35±0.00	620.95	40.45±0.05	223.20
NOGD	**0.23±0.00**	4,009.03	**34.72±0.07**	838.47
AVM-Hinge	0.31±0.07	540.65	36.11±0.16	**51.12**
AVM-Logit	0.28±0.03	**503.34**	35.92±0.16	53.51
Dataset [S]	***poker*** [393]		***airlines*** [388]	
Algorithm	Mistake Rate	Time	Mistake Rate	Time
FOGD	52.28±0.04	928.89	20.98±0.01	1,270.75
NOGD	44.90±0.16	4,920.33	25.56±0.01	3,553.50
AVM-Hinge	**43.85±0.09**	**122.59**	**19.28±0.00**	**733.72**
AVM-Logit	43.97±0.07	124.86	**19.28±0.00**	766.19



(a) a9a (b) cod-rna

Figure 7: Predictive and wall-clock performance on two datasets: *a9a* and *cod-rna* of budgeted methods when the budget size B is varied. (Best viewed in colors).

Table 5: Online regression results of 6 baselines and 3 versions of our AVMs. The notation $[\delta; S; B; D]$ denotes the same meanings as those in Table 3. The regression loss is measured using root mean squared error (RMSE) and the execution time is reported in second. The best performance is in **bold**.

Dataset	***casp***		***slice***	
$[\delta \mid S \mid B \mid D]$	$[4.0 \mid 166 \mid 400 \mid 2,000]$		$[16.0 \mid 27 \mid 1,000 \mid 3,000]$	
Algorithm	RMSE	Time	RMSE	Time
RBP	0.3195±0.0012	7.15	0.1154±0.0006	810.14
Forgetron	0.3174±0.0008	10.14	0.1131±0.0004	1,069.15
Projectron	0.2688±0.0002	8.48	**0.0770±0.0002**	814.37
BOGD	0.2858±0.0002	6.20	0.1723±0.0001	816.16
FOGD	0.3775±0.0014	5.83	0.1440±0.0009	20.65
NOGD	**0.2512±0.0001**	6.99	0.0873±0.0002	812.69
AVM-ε	0.3165±0.0329	3.53	0.2013±0.0137	7.07
AVM-ℓ_1	0.3166±0.0330	3.44	0.2013±0.0138	7.13
AVM-ℓ_2	0.3274±0.0280	**3.31**	0.2590±0.0002	**6.88**

Dataset	***year***		***airlines***	
$[\delta \mid S \mid B \mid D]$	$[60.0 \mid 67 \mid 400 \mid 1,600]$		$[1.0 \mid 388 \mid 1,000 \mid 2,000]$	
Algorithm	RMSE	Time	RMSE	Time
RBP	0.1881±0.0002	605.42	36.5068±0.0010	3,418.89
Forgetron	0.1877±0.0004	904.09	36.5065±0.0003	5,774.47
Projectron	0.1390±0.0003	605.19	36.1365±0.0009	3,834.19
BOGD	0.2009±0.0000	596.10	35.7346±0.0010	3,058.96
FOGD	0.1581±0.0002	76.70	53.1638±0.0120	646.15
NOGD	0.1375±0.0005	607.37	**34.7421±0.0013**	3,324.38
AVM-ε	0.1286±0.0002	48.01	36.0901±0.0914	638.60
AVM-ℓ_1	**0.1232±0.0003**	47.29	36.3632±0.0192	**621.57**
AVM-ℓ_2	0.2420±0.0001	**46.63**	35.1128±0.0192	633.27

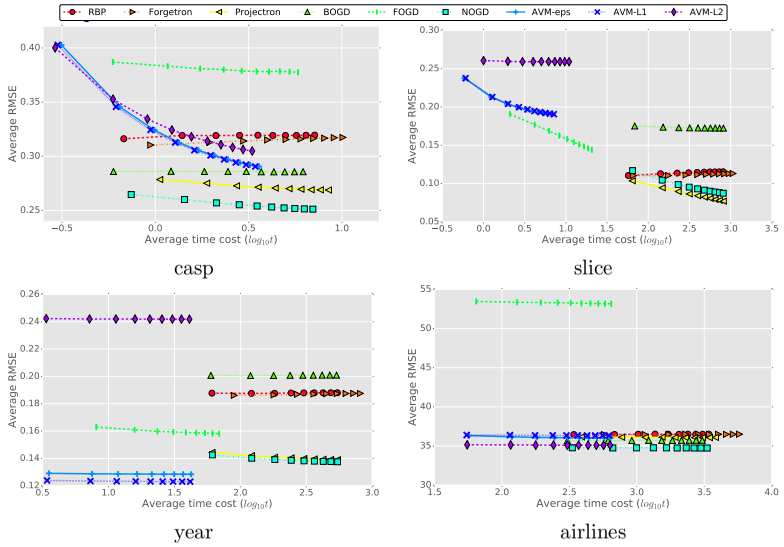


Figure 8: Average RMSE vs. time cost for online regression. The average time (seconds) is shown in the logarithm with base 10. (Best viewed in colors).

In all experiments, our proposed method produces the model sizes that are much smaller than the budget sizes of baseline methods. Thus we further investigate the performance of the budgeted baselines by varying the budget size B, and compare with our AVM with Hinge loss. Fig. 7 shows our analysis on two datasets *a9a* and *cod-rna*. It can be seen that the larger B helps model obtain better classification results, but hurts their running speed. For both datasets, the budgeted baselines with larger budget sizes still fail to beat the predictive performance of AVM. On the other hand, the baselines with smaller budget sizes run faster than the AVM on *cod-rna* dataset, but slower on *a9a* dataset.

Finally, two versions of AVMs demonstrate similar discriminative performances and computational complexities wherein the AVM-Logit is slightly slower due to the additional exponential operators as also seen in batch classification task. All aforementioned observations validate the effectiveness and efficiency of our proposed technique. Thus, we believe that our approximation machine is a promising technique for building scalable online kernel learning algorithms for large-scale classification tasks.

9.5 Online Regression

The last experiment addresses the online regression problem to evaluate the capabilities of our approach with three proposed loss functions – ℓ_1,ℓ_2 and ε-insensitive losses as described in Section 7. Incorporating these loss functions creates three versions: AVM-ε, AVM-ℓ_1 and AVM-ℓ_2. We use four datasets: *casp*, *slice*, *year* and *airlines* (delay minutes) with a wide

range of sizes for this task. We recruit six baselines: RBP, Forgetron, Projectron, BOGD, FOGD and NOGD (cf. more detailed description in Section 9.4).

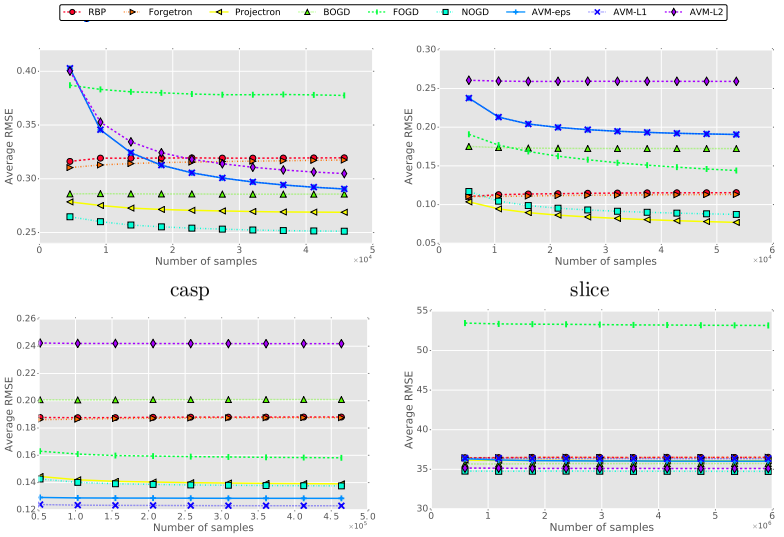
Hyperparameters setting. We adopt the same hyperparameter searching procedure for batch classification task as in Section 9.3. Furthermore, for the budget size B and the feature dimension D in FOGD, we follow the same strategy used in Section 7.1.1 of Lu et al. (2015). More specifically, these hyperparameters are separately set for different datasets as reported in Table 5. They are chosen such that they are roughly proportional to the number of support vectors produced by the batch SVM algorithm in LIBSVM running on a small subset. The aim is to achieve competitive accuracy using a relatively larger budget size for tackling more challenging regression tasks.

Results. Fig. 9a and Fig. 9b shows the relative performance convergence w.r.t regression error (root mean square root - RMSE) and computation cost (seconds) of the AVMs in comparison with those of the baselines. Combining these two figures, we compare the average error and running time in Fig. 8. Table 5 reports the final average results in detailed numbers after the methods traverse all data samples. From these results, we can draw some observations as follows.

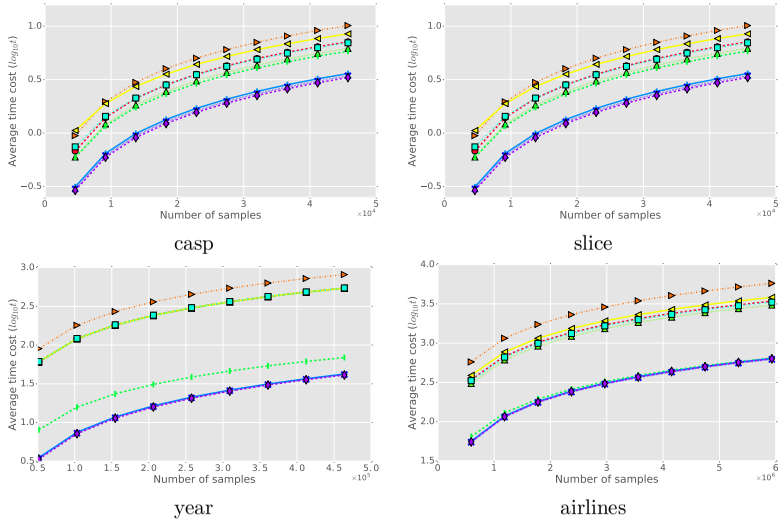
First of all, as can be seen from Fig. 9a, there are several different learning behaviors w.r.t regression loss, of the methods training on individual datasets. All algorithms, in general, reach their regression error plateaus very quickly as observed in the datasets *year* and *airlines* where they converge at certain points from the initiation of the learning. On the other hand, for *casp* and *slice* databases, the AVM-based models regularly obtain better performance, that is, their average RMSE scores keep reducing when receiving more data, except in *slice* data, the regression performance of AVM-ℓ_2 are almost unchanged during the learning. Note that, for these two datasets, the learning curve of AVM-ε coincides, thus is overplotted by that of AVM-ℓ_1, resulting in its no-show in the figure. Interestingly, the errors of RBP and Forgetron slightly increase throughout their online learning in these two cases.

Second, Fig. 8 plots average error against computational cost, which shows similar learning behaviors as in the our first observation. The computational cost progresses are simple and more obvious to comprehend than the regression progresses. As illustrated in Fig. 9b, all algorithms have nicely plausible execution time curves in which the time is accumulated over the learning procedure.

According to final results summarized in Table 5, our proposed models enjoy a significant advantage in computational efficacy whilst achieve better (for *year* dataset) or competitive regression results with other methods. The AVM, again, secures the best performance in terms of model sparsity. Among the baselines, the FOGD is the fastest, that is, its time costs can be considered to compare with those of our methods, but its regression performances are worse. The remaining algorithms usually obtain better results, but is traded off by the sacrifice of scalability. This, once again, verifies the effectiveness and efficiency of our proposed techniques. We believe that the AVM is a promising machine to perform online regression task for large-scale datasets.



(a) The average RMSE as a function of the number of samples seen by the models.



(b) The average time costs (seconds in logarithm of 10) as a function of the number of samples seen by the models.

Figure 9: Convergence evaluation of online regresion. (Best viewed in colors).

Finally, comparing the capability of three AVM's variants, all models demonstrate similar computational complexities wherein the AVM-ℓ_2 is slightly faster due to its simpler operator in computing the gradient as derived in Section 7. However, its regression errors are higher than two other methods – AVM-ε and AVM-ℓ_1.

10. Conclusion

In this paper, we have proposed Approximation Vector Machine (AVM) for large-scale online learning. The AVM is theoretically proven to have bounded and sparse model size while not hurting the predictive performance. We have validated our proposed method on several benchmark datasets. The experimental results show that the proposed AVM obtains a comparable predictive performance while simultaneously achieving an impressive model size and a computational speed-up compared with those of the baselines. Our future works are to apply AVM to the context of semi-supervised learning, anomaly detection, and support vector clustering.

Acknowledgment

We gratefully thank the editor and the anonymous reviewers for their valuable comments and thorough inspection of the article. This work was partially supported by the Australian Research Council (ARC) under the Discovery Project DP160109394.

11. Proofs Regarding δ-Coverage

Proof of Theorem 4

Assume that $\left\|x - x^{'}\right\| \leq \delta$, then we have

$$\begin{aligned}\left\|\Phi(x) - \Phi\left(x^{'}\right)\right\|^2 &= K(x,x) + K\left(x^{'},x^{'}\right) - 2K\left(x,x^{'}\right) = 2\left(1 - k\left(\left\|x - x^{'}\right\|^2\right)\right) \\ &\leq 2\left(1 - k\left(\delta^2\right)\right) = \delta_\Phi^2\end{aligned}$$

Furthermore, we have

$$\lim_{\delta\to 0}\delta_\Phi = 2^{1/2}\lim_{\delta\to 0}\left(1 - k\left(\delta^2\right)\right)^{1/2} = 2^{1/2}\left(1 - k(0)\right)^{1/2} = 0$$

Finally, since Gaussian kernel is a special radial kernel with $k(t) = \exp(-\gamma t)$, we obtain the final conclusion. ■

Proof of Theorem 19

Since the proof is similar for the hyperrectangle cell case, we present the proof for the hypersphere case. Let us consider the open coverage $\mathcal{U} = \left\{\mathcal{B}\left(z, \frac{\delta}{2}\right)\right\}_{z\in\mathcal{X}}$. From the compactness of the data domain $\mathcal{X}$, it apparent that from $\mathcal{U}$ we must be able to extract a finite subcoverage of size m, that is, $\mathcal{U}_m = \left\{\mathcal{B}\left(z_i, \frac{\delta}{2}\right)\right\}_{i=1}^m \subset \mathcal{U}$. From the construction of the coverage $\mathcal{P}$ in Algorithm 3, we know that

$$\|c_i - c_j\| > \delta/2 \text{ if } i \neq j$$

Hence, each open sphere in the finite coverage $\mathcal{U}_m$ is able to contain at most one core point of $\mathcal{P}$. It means that the cardinality of $\mathcal{P}$ must be less than or equal m, that is, $|\mathcal{P}| \leq m$. ■

Proofs Regarding Convergence Analysis

Given a finite δ-coverage $\mathcal{P} = (P_i)_{i\in I}$ with the core set $\mathcal{C} = (c_i)_{i\in I}$, when receiving an incoming instance (x_t, y_t) we approximate (x_t, y_t) by (c_{i_t}, y_t) with c_{i_t} is a core point whose cell contains x_t, that is, $x_t \in P_{i_t}$. We use a Bernoulli random variable Z_t to control if the approximation is performed or not, that is, $Z_t = 1$ indicates the approximation is performed.

Let us define $g_t = \lambda \mathbf{w}_t + l^{'}(\mathbf{w}_t; x_t, y_t) = \lambda \mathbf{w}_t + \alpha_t \Phi(x_t)$. We have the following

$$h_t = g_t + Z_t \Delta_t$$

where $\Delta_t = \alpha_t (\Phi(c_{i_t}) - \Phi(x_t))$.

The update rule becomes

$$\mathbf{w}_{t+1} = \prod_S (\mathbf{w}_t - \eta_t h_t)$$

where $S = \mathbb{R}^D$ (i.e., the feature space) or $\mathcal{B}\left(\mathbf{0}, y_{\max}\lambda^{-1/2}\right)$.

Lemma 21. *The following statements hold*

i) There exist two positive constants P and M such that $\mathbb{E}\left[\|\mathbf{w}_t\|^2\right] \le P^2$ and $\mathbb{E}\left[\alpha_t^2\right] \le M$ for all t.

ii) $\mathbb{E}\left[\left\|l^{'}(\mathbf{w}_t; x_t, y_t)\right\|^2\right] \le L = \left(A\sqrt{P} + B\right)^2$ for all t.

iii) $\mathbb{E}\left[\|g_t\|^2\right] \le G = \left(\lambda P + A\sqrt{P} + B\right)^2$ for all t.

iv) $\mathbb{E}\left[\|h_t\|^2\right] \le H = \left(\sqrt{G} + \delta_\Phi\left(A\sqrt{P} + B\right)\right)^2$ for all t.

Proof i) We prove by induction that $\mathbb{E}\left[\|\mathbf{w}_t\|^2\right] \le P^2$ where $P = \left(\frac{(\delta_\Phi+1)A + \sqrt{(\delta_\Phi+1)^2A^2 + 4B\lambda(\delta_\Phi+1)}}{2\lambda}\right)^2$ for all t. Assume that the claim is holding for t, using Minkowski inequality, we have

$$\begin{aligned}
\sqrt{\mathbb{E}\left[\|\mathbf{w}_{t+1}\|^2\right]} &\le \sqrt{\mathbb{E}\left[\left\|\prod_S (\mathbf{w}_t - \eta_t h_t)\right\|^2\right]} \le \sqrt{\mathbb{E}\left[\|\mathbf{w}_t - \eta_t h_t\|^2\right]} \\
&\le \frac{t-1}{t}\sqrt{\mathbb{E}\left[\|\mathbf{w}_t\|^2\right]} + \eta_t\sqrt{\mathbb{E}\left[\|l^{'}(\mathbf{w}_t; x_t, y_t)\|^2\right]} + \eta_t\sqrt{\mathbb{E}\left[\|\Delta_t\|^2\right]} \\
&\le \frac{t-1}{t}\sqrt{\mathbb{E}\left[\|\mathbf{w}_t\|^2\right]} + \frac{A\sqrt{\mathbb{E}\left[\|\mathbf{w}_t\|\right]} + B}{\lambda t} + \frac{\delta_\Phi\sqrt{\mathbb{E}\left[\alpha_t^2\right]}}{\lambda t} \\
&\le \frac{(t-1)P}{t} + \frac{A\sqrt{P} + B}{\lambda t} + \frac{\delta_\Phi\sqrt{\mathbb{E}\left[\|l^{'}(\mathbf{w}_t; x_t, y_t)\|^2\right]}}{\lambda t} \\
&\le \frac{(t-1)P}{t} + \frac{(\delta_\Phi+1)\left(A\sqrt{P} + B\right)}{\lambda t} = P
\end{aligned}$$

Note that we have used $\left\|l^{'}(\mathbf{w}_t; x_t, y_t)\right\|^2 = \alpha_t^2 K(x_t, x_t) = \alpha_t^2$, $\mathbb{E}\left[\|\mathbf{w}_t\|\right] \le \sqrt{\mathbb{E}\left[\|\mathbf{w}_t\|^2\right]} \le P$, and $u = P^{1/2} = \frac{(\delta_\Phi+1)A + \sqrt{(\delta_\Phi+1)^2A^2 + 4B\lambda(\delta_\Phi+1)}}{2\lambda}$ is the solution of the quadratic equation

$$u^2 - \frac{(\delta_\Phi+1)Au}{\lambda} - \frac{(\delta_\Phi+1)B}{\lambda} = 0$$

The proof of $\mathbb{E}\left[\alpha_t^2\right] \le M$ is trivial for the case of Hinge, ℓ_1, Logistic, $\varepsilon-$insensitive losses. In these cases, we simply choose $M = \max(y_{\max}, 1)^2$. We only need to consider the ℓ_2-loss case. In particular, we have

$$\begin{aligned}
\alpha_t^2 &= \left(\mathbf{w}_t^\mathsf{T}\Phi(x_t) - y_t\right)^2 \le 2\left(\left(\mathbf{w}_t^\mathsf{T}\Phi(x_t)\right)^2 + y_{\max}^2\right) \\
&\le 2\left(\|\mathbf{w}_t\|^2\|\Phi(x_t)\|^2 + y_{\max}^2\right) \le 2\left(\|\mathbf{w}_t\|^2 + y_{\max}^2\right)
\end{aligned}$$

$$\mathbb{E}\left[\alpha_t^2\right] \le 2\left(P^2 + y_{\max}^2\right) = M$$

ii) We have the following

$$\sqrt{\mathbb{E}\left[\|l^{'}(\mathbf{w}_t; x_t, y_t)\|^2\right]} \le \sqrt{\mathbb{E}\left[\left(A\|\mathbf{w}_t\|^{1/2} + B\right)^2\right]} \le A\sqrt{\mathbb{E}\left[\|\mathbf{w}_t\|\right]} + B \le A\sqrt{P} + B$$

Note that we have used the inequality $\mathbb{E}\left[\|\mathbf{w}_t\|\right] \le \sqrt{\mathbb{E}\left[\|\mathbf{w}_t\|^2\right]} \le P$.

iii) Using Minkowski inequality, we yield

$$\sqrt{\mathbb{E}\left[\|g_t\|^2\right]} \le \lambda\sqrt{\mathbb{E}\left[\|\mathbf{w}_t\|^2\right]} + \sqrt{\mathbb{E}\left[\|l^{'}(\mathbf{w}_t; x_t, y_t)\|^2\right]} \le \lambda P + A\sqrt{P} + B$$

iv) We have the following

$$\begin{aligned}
\sqrt{\mathbb{E}\left[\|h_t\|^2\right]} &\le \sqrt{\mathbb{E}\left[\|g_t\|^2\right]} + \delta_\Phi\sqrt{\mathbb{E}\left[\alpha_t^2\right]} \\
&\le \sqrt{G} + \delta_\Phi\sqrt{\mathbb{E}\left[\|l^{'}(\mathbf{w}_t; x_t, y_t)\|^2\right]} = \sqrt{G} + \delta_\Phi\left(A\sqrt{P} + B\right)
\end{aligned}$$

■

Lemma 22. *There exists a positive constant W such that $\mathbb{E}\left[\|\mathbf{w}_t - \mathbf{w}^*\|^2\right] \le W$ for all t.*

Proof We first remind the definitions of the relevant quantities

$$\begin{aligned}
&g_t = \lambda\mathbf{w}_t + l^{'}(\mathbf{w}_t; x_t, y_t) = \lambda\mathbf{w}_t + \alpha_t\Phi(x_t) \\
&h_t = g_t + Z_t\Delta_t \text{ where } \Delta_t = \alpha_t(\Phi(c_{i_t}) - \Phi(x_t))
\end{aligned}$$

We now prove by induction in t. We derive as follows

$$\begin{aligned}\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2 &= \left\|\prod_S(\mathbf{w}_t-\eta_t h_t)-\mathbf{w}^*\right\|^2 \leq \|\mathbf{w}_t-\eta_t h_t-\mathbf{w}^*\|^2\\ &= \|\mathbf{w}_t-\mathbf{w}^*\|+\eta_t^2\|h_t\|^2-2\eta_t\langle\mathbf{w}_t-\mathbf{w}^*,g_t+Z_t\Delta_t\rangle\end{aligned}$$

where $S=\mathbb{R}^D$ or $\mathcal{B}\left(\mathbf{0},y_{\max}\lambda^{-1/2}\right)$.

Taking conditional expectation w.r.t $\mathbf{w}_t$, we gain

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2\right] \leq& \mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]\\ &+\eta_t^2\mathbb{E}\left[\|h_t\|^2\right]-2\eta_t\left\langle\mathbf{w}_t-\mathbf{w}^*,f'(\mathbf{w}_t)\right\rangle-2\eta_t\langle\mathbf{w}_t-\mathbf{w}^*,Z_t\Delta_t\rangle\\ \leq& \mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]-2\eta_t\lambda\|\mathbf{w}_t-\mathbf{w}^*\|^2+\eta_t^2\mathbb{E}\left[\|h_t\|^2\right]-2\eta_t\langle\mathbf{w}_t-\mathbf{w}^*,Z_t\Delta_t\rangle\end{aligned}$$

Here we note that we have used $\left\langle\mathbf{w}_t-\mathbf{w}^*,f'(\mathbf{w}_t)\right\rangle\geq\lambda\|\mathbf{w}_t-\mathbf{w}^*\|^2$. It comes from the following derivation

$$f(\mathbf{w}^*)-f(\mathbf{w}_t)\geq\left\langle f'(\mathbf{w}_t),\mathbf{w}^*-\mathbf{w}_t\right\rangle+\frac{\lambda}{2}\|\mathbf{w}_t-\mathbf{w}^*\|^2$$

$$\begin{aligned}\left\langle f'(\mathbf{w}_t),\mathbf{w}_t-\mathbf{w}^*\right\rangle &\geq f(\mathbf{w}_t)-f(\mathbf{w}^*)+\frac{\lambda}{2}\|\mathbf{w}_t-\mathbf{w}^*\|^2 \geq \left\langle f'(\mathbf{w}^*),\mathbf{w}_t-\mathbf{w}^*\right\rangle+\lambda\|\mathbf{w}_t-\mathbf{w}^*\|^2\\ &\geq\lambda\|\mathbf{w}_t-\mathbf{w}^*\|^2 \qquad \text{thanks to } \left\langle f'(\mathbf{w}^*),\mathbf{w}_t-\mathbf{w}^*\right\rangle\geq 0\end{aligned}$$

Taking expectation again, we gain

$$\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2\right]\leq\frac{t-2}{t}\mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]+\frac{H}{\lambda^2t^2}+\frac{2W^{1/2}M^{1/2}\delta_\Phi}{\lambda t}$$

Choosing $W=\left(\frac{M^{1/2}\delta_\phi+\left(M\delta_\phi^2+2H\right)^{1/2}}{2\lambda}\right)^2$, we gain if $\mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]\leq W$ then $\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2\right]\leq W$. The reason is that $W=\left(\frac{M^{1/2}\delta_\phi+\left(M\delta_\phi^2+2H\right)^{1/2}}{2\lambda}\right)^2$ is the solution of the equation

$$W=\frac{t-2}{t}W+\frac{H}{\lambda^2t}+\frac{2W^{1/2}M^{1/2}\delta_\Phi}{\lambda t} \text{ or } 2W-\frac{2W^{1/2}M^{1/2}\delta_\Phi}{\lambda}-\frac{H}{\lambda^2}=0$$

. Hence, if $\mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]\leq W$, we arrive at

$$\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2\right]\leq\frac{t-2}{t}W+\frac{H}{\lambda^2t}+\frac{2W^{1/2}M^{1/2}\delta_\Phi}{\lambda t}=W$$

■

We now show the proof of Theorem 5.

Proof of Theorem 5

We first remind the definitions of the relevant quantities

$$\begin{aligned}g_t &= \lambda\mathbf{w}_t+l'(\mathbf{w}_t;x_t,y_t)=\lambda\mathbf{w}_t+\alpha_t\Phi(x_t)\\ h_t &= g_t+Z_t\Delta_t \text{ where } \Delta_t=\alpha_t\left(\Phi(c_{i_t})-\Phi(x_t)\right)\end{aligned}$$

We then derive as follows

$$\begin{aligned}\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2 &= \left\|\prod_S(\mathbf{w}_t-\eta_t h_t)-\mathbf{w}^*\right\|^2 \leq \|\mathbf{w}_t-\eta_t h_t-\mathbf{w}^*\|^2\\ &= \|\mathbf{w}_t-\mathbf{w}^*\|+\eta_t^2\|h_t\|^2-2\eta_t\langle\mathbf{w}_t-\mathbf{w}^*,g_t+Z_t\Delta_t\rangle\end{aligned}$$

where $S=\mathbb{R}^D$ or $\mathcal{B}\left(\mathbf{0},y_{\max}\lambda^{-1/2}\right)$.

$$\langle\mathbf{w}_t-\mathbf{w}^*,g_t\rangle\leq\frac{\|\mathbf{w}_t-\mathbf{w}^*\|^2-\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2}{2\eta_t}+\frac{\eta_t}{2}\|h_t\|^2-\langle\mathbf{w}_t-\mathbf{w}^*,Z_t\Delta_t\rangle$$

Taking conditional expectation w.r.t $\mathbf{w}_t$, we obtain

$$\begin{aligned}\left\langle\mathbf{w}_t-\mathbf{w}^*,f'(\mathbf{w}_t)\right\rangle \leq& \frac{\mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]-\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2\right]}{2\eta_t}\\ &+\frac{\eta_t}{2}\mathbb{E}\left[\|h_t\|^2\right]-\langle\mathbf{w}_t-\mathbf{w}^*,\mathbb{E}\left[Z_t\Delta_t\right]\rangle\end{aligned}$$

$$\begin{aligned}f(\mathbf{w}_t)-f(\mathbf{w}^*)+\frac{\lambda}{2}\|\mathbf{w}_t-\mathbf{w}^*\|^2 \leq& \frac{\mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]-\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2\right]}{2\eta_t}\\ &+\frac{\eta_t}{2}\mathbb{E}\left[\|h_t\|^2\right]-\langle\mathbf{w}_t-\mathbf{w}^*,\mathbb{E}\left[Z_t\Delta_t\right]\rangle\end{aligned}$$

Taking expectation again, we achieve

$$\begin{aligned}\mathbb{E}\left[f(\mathbf{w}_t)-f(\mathbf{w}^*)\right] \leq& \frac{\lambda}{2}(t-1)\mathbb{E}\left[\|\mathbf{w}_t-\mathbf{w}^*\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\|\mathbf{w}_{t+1}-\mathbf{w}^*\|^2\right]\\ &+\frac{\eta_t}{2}\mathbb{E}\left[\|h_t\|^2\right]-\mathbb{E}\left[\langle\mathbf{w}_t-\mathbf{w}^*,Z_t\Delta_t\rangle\right]\end{aligned}$$

i) If Z_t is independent with $\mathbf{w}_t$, we derive as

$$\begin{aligned}\mathbb{E}\left[f\left(\mathbf{w}_t\right)-f\left(\mathbf{w}^*\right)\right] &\leq \frac{\lambda}{2}\left(t-1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]\\ &+\frac{\eta_t}{2}\mathbb{E}\left[\left\|h_t\right\|^2\right]-\mathbb{E}\left[\left\langle Z_t\left(\mathbf{w}_t-\mathbf{w}^*\right),\Delta_t\right\rangle\right]\\ &\leq \frac{\lambda}{2}\left(t-1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]+\frac{\eta_t}{2}\mathbb{E}\left[\left\|h_t\right\|^2\right]\\ &+\mathbb{E}\left[Z_t^2\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2}\\ &\leq \frac{\lambda}{2}\left(t-1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]+\frac{\eta_t}{2}\mathbb{E}\left[\left\|h_t\right\|^2\right]\\ &+\mathbb{E}\left[Z_t^2\right]^{1/2}\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2}\\ &\leq \frac{\lambda}{2}\left(t-1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]+\frac{\eta_t}{2}H \qquad (5)\\ &+\mathbb{P}\left(Z_t=1\right)^{1/2}\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2}\end{aligned}$$

Taking sum over $1,2,\ldots,T$ and using the inequalities in Lemmas 21 and 22, we yield

$$\sum_{t=1}^{T}\mathbb{E}\left[f\left(\mathbf{w}_t\right)\right]-Tf\left(\mathbf{w}^*\right)\leq\frac{H}{2\lambda}\sum_{t=1}^{T}\frac{1}{t}+\sum_{t=1}^{T}\mathbb{P}\left(Z_t=1\right)^{1/2}\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2} \qquad (6)$$

$$T\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right)-f\left(\mathbf{w}^*\right)\right]\leq\frac{H}{2\lambda}\sum_{t=1}^{T}\frac{1}{t}+\sum_{t=1}^{T}\mathbb{P}\left(Z_t=1\right)^{1/2}\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2}$$

$$\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right)-f\left(\mathbf{w}^*\right)\right]\leq\frac{H\left(\log\left(T\right)+1\right)}{2\lambda T}+\frac{\delta_\Phi M^{1/2}W^{1/2}}{T}\sum_{t=1}^{T}\mathbb{P}\left(Z_t=1\right)^{1/2}$$

ii) If Z_t is independent with $\mathbf{w}_t$ and (x_t,y_t), we derive as

$$\begin{aligned}\mathbb{E}\left[f\left(\mathbf{w}_t\right)-f\left(\mathbf{w}^*\right)\right] &\leq \frac{\lambda}{2}\left(t-1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]\\ &+\frac{\eta_t}{2}H-\mathbb{E}\left[Z_t\left\langle\mathbf{w}_t-\mathbf{w}^*,\Delta_t\right\rangle\right]\\ &\leq \frac{\lambda}{2}\left(t-1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]\\ &+\frac{\eta_t}{2}H-\mathbb{E}\left[Z_t\right]\mathbb{E}\left[\left\langle\mathbf{w}_t-\mathbf{w}^*,\Delta_t\right\rangle\right]\\ &\leq \frac{\lambda}{2}\left(t-1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]+\frac{\eta_t}{2}H \qquad (7)\\ &+\mathbb{P}\left(Z_t=1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2}\end{aligned}$$

Taking sum over $1,2,\ldots,T$ and using the inequalities in Lemmas 21 and 22, we yield

$$\sum_{t=1}^{T}\mathbb{E}\left[f\left(\mathbf{w}_t\right)\right]-Tf\left(\mathbf{w}^*\right)\leq\frac{H}{2\lambda}\sum_{t=1}^{T}\frac{1}{t}+\sum_{t=1}^{T}P\left(Z_t=1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2} \qquad (8)$$

$$T\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right)-f\left(\mathbf{w}^*\right)\right]\leq\frac{H}{2\lambda}\sum_{t=1}^{T}\frac{1}{t}+\sum_{t=1}^{T}P\left(Z_t=1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2}$$

$$\begin{aligned}\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right)-f\left(\mathbf{w}^*\right)\right] &\leq \frac{H\left(\log\left(T\right)+1\right)}{2\lambda T}+\frac{\delta_\Phi M^{1/2}W^{1/2}}{T}\sum_{t=1}^{T}P\left(Z_t=1\right)\\ &\leq \frac{H\left(\log\left(T\right)+1\right)}{2\lambda T}+\delta_\Phi M^{1/2}W^{1/2}\end{aligned}$$

iii) In general, we derive as

$$\begin{aligned}\mathbb{E}\left[f\left(\mathbf{w}_t\right)-f\left(\mathbf{w}^*\right)\right] &\leq \frac{\lambda}{2}\left(t-1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right]\\ &+\frac{\eta_t}{2}H-\mathbb{E}\left[Z_t\left\langle\mathbf{w}_t-\mathbf{w}^*,\Delta_t\right\rangle\right]\\ &\leq \frac{\lambda}{2}\left(t-1\right)\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]-\frac{\lambda}{2}t\mathbb{E}\left[\left\|\mathbf{w}_{t+1}-\mathbf{w}^*\right\|^2\right] \qquad (9)\\ &+\frac{\eta_t}{2}H+\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2}\end{aligned}$$

Taking sum over $1,2,\ldots,T$ and using the inequalities in Lemmas 21 and 22, we yield

$$\sum_{t=1}^{T}\mathbb{E}\left[f\left(\mathbf{w}_t\right)\right]-Tf\left(\mathbf{w}^*\right)\leq\frac{H}{2\lambda}\sum_{t=1}^{T}\frac{1}{t}+\sum_{t=1}^{T}\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2} \qquad (10)$$

$$T\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right)-f\left(\mathbf{w}^*\right)\right]\leq\frac{H}{2\lambda}\sum_{t=1}^{T}\frac{1}{t}+\sum_{t=1}^{T}\mathbb{E}\left[\left\|\mathbf{w}_t-\mathbf{w}^*\right\|^2\right]^{1/2}\mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2}$$

$$\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T\right)-f\left(\mathbf{w}^*\right)\right]\leq\frac{H\left(\log\left(T\right)+1\right)}{2\lambda T}+\delta_\Phi M^{1/2}W^{1/2}$$

■

Proof of Theorem 15

Let us denote the model size, i.e., the number of vectors in support set, after the iteration t by S_t. We also define N_t by the binary random variable which specifies whether the incoming instance (x_t,y_t) locates in a new cell of the coverage, that is, $N_t=1$ indicating the current cell P_{i_t} is a new cell. We assume that Z_t is independent with (x_t,y_t) and so does with N_t. Since a new instance is added to the support set if either a new cell is discovered or the old cell is found but approximation is not performed, we reach the following

$$S_t\leq S_{t-1}+N_t+\left(1-Z_t\right)\left(1-N_t\right)$$

Taking expectation, we obtain

$$\begin{aligned}\mathbb{E}[S_t] &\leq \mathbb{E}[S_{t-1}] + \mathbb{E}[N_t] + (1 - \mathbb{E}[Z_t])(1 - \mathbb{E}[N_t]) \\ &\leq \mathbb{E}[S_{t-1}] + \mathbb{E}[N_t] + (1 - p_t)(1 - \mathbb{E}[N_t]) \\ &\leq \mathbb{E}[S_{t-1}] + \mathbb{E}[N_t] + q_t(1 - \mathbb{E}[N_t])\end{aligned}$$

$$\mathbb{E}[S_t] - \mathbb{E}[S_{t-1}] \leq \mathbb{E}[N_t] + q_t(1 - \mathbb{E}[N_t])$$

Summing over the above when $t = 1, \ldots, T$, we have

$$\begin{aligned}\mathbb{E}[S_T] &\leq \sum_{t=1}^{T} \mathbb{E}[N_t] + \sum_{t=1}^{T} q_t(1 - \mathbb{E}[N_t]) = \sum_{t=1}^{T} q_t + \sum_{t=1}^{T} p_t \mathbb{E}[N_t] \\ &\leq \sum_{t=1}^{T} q_t + \sum_{t=1}^{T} \mathbb{E}[N_t] \leq \sum_{t=1}^{T} q_t + \mathbb{E}[M_T] \qquad (11)\end{aligned}$$

where we have denoted $\mathbb{P}(Z_t = 1) = p_t$, $\mathbb{P}(Z_t = 0) = q_t$, and $M_T = \sum_{t=1}^{T} N_t$ indicates the number of cells discovered so far.

We consider some specific cases and investigate the model size $\mathbb{E}[S_T]$ in these cases.

i) $p_t = \mathbb{P}(Z_t = 1) = 1, \forall t$, that is, we always do approximation. From Eq. (11), we obtain

$$\mathbb{E}[S_T] \leq \mathbb{E}[M_T] \leq |\mathcal{P}|$$

ii) $p_t = \mathbb{P}(Z_t = 1) = \max\left(0, 1 - \frac{\beta}{t}\right), \forall t$. It follows that

$$q_t = 1 - p_t \leq 1 - \left(1 - \frac{\beta}{t}\right) = \frac{\beta}{t}$$

From Eq. (11), we gain

$$\begin{aligned}\mathbb{E}[S_T] &\leq \beta \sum_{t=1}^{T} \frac{1}{t} + \mathbb{E}[M_T] \leq \beta\left(1 + \int_1^T \frac{1}{t} dt\right) + \mathbb{E}[M_T] \\ &\leq \beta(\log T + 1) + \mathbb{E}[M_T]\end{aligned}$$

iii) $p_t = \mathbb{P}(Z_t = 1) = \max\left(0, 1 - \frac{\beta}{t^\rho}\right), \forall t$ where $0 < \rho < 1$. It follows that

$$q_t = 1 - p_t \leq 1 - \left(1 - \frac{\beta}{t^\rho}\right) = \frac{\beta}{t^\rho}$$

From Eq. (11), we gain

$$\mathbb{E}[S_T] \leq \beta \sum_{t=1}^{T} \frac{1}{t^\rho} + \mathbb{E}[M_T] \leq \beta\left(1 + \int_1^T t^{-\rho} dt\right) + \mathbb{E}[M_T] \leq \frac{\beta T^{1-\rho}}{1-\rho} + \mathbb{E}[M_T]$$

iv) $p_t = \mathbb{P}(Z_t = 1) = \max\left(0, 1 - \frac{\beta}{t^\rho}\right), \forall t$ where $\rho > 1$. It follows that

$$q_t = 1 - p_t \leq 1 - \left(1 - \frac{\beta}{t^\rho}\right) = \frac{\beta}{t^\rho}$$

From Eq. (11), we gain

$$\mathbb{E}[S_T] \leq \beta \sum_{t=1}^{T} \frac{1}{t^\rho} + \mathbb{E}[M_T] \leq \beta\zeta(\rho) + \mathbb{E}[M_T] \leq \beta\zeta(\rho) + |\mathcal{P}|$$

where $\zeta(.)$ is ζ- Riemann function defined by the integral $\zeta(s) = \frac{1}{\Gamma(s)} \int_0^{+\infty} \frac{t^{s-1}}{e^s - 1} dt$. ■

We now show the proof of Theorem 8. To realize this proof, we use the famous inequality, namely Hoeffding which for completeness we state below.

Theorem. *(Hoeffding inequality) Let the independent variables $X_1, \ldots, X_n$ where $a_i \leq X_i \leq b_i$ for each $i \in [n]$. Let $S = \sum_{i=1}^{n} X_i$ and $\Delta_i = b_i - a_i$. The following hold*

i) $\mathbb{P}(S - \mathbb{E}[S] > \varepsilon) \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n} \Delta_i^2}\right)$.

ii) $\mathbb{P}(|S - \mathbb{E}[S]| > \varepsilon) \leq 2\exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^{n} \Delta_i^2}\right)$

Proof of Theorem 8

From Eqs. (6, 8, 10), we achieve

$$\frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[f(\mathbf{w}_t)] - f(\mathbf{w}^*) \leq \frac{H(\log(T) + 1)}{2\lambda T} + d_T$$

Let us denote $X = f(\mathbf{w}_r) - f(\mathbf{w}^*)$, where r is uniformly sampled from $\{1, 2, \ldots, T\}$. We have

$$\mathbb{E}_r[X] = \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[f(\mathbf{w}_t)] - f(\mathbf{w}^*) \leq \frac{H(\log(T) + 1)}{2\lambda T} + d_T$$

It follows that

$$\mathbb{E}[X] = \mathbb{E}_{(x_t, y_t)_{t=1}^{T}}[\mathbb{E}_r[X]] \leq \frac{H(\log(T) + 1)}{2\lambda T} + d_T$$

Let us denote $\Delta_T = \max_{1 \leq t \leq T} (f(\mathbf{w}_t) - f(\mathbf{w}^*))$ which implies that $0 < f(\mathbf{w}_r) - f(\mathbf{w}^*) < \Delta_T$. Applying Hoeffding inequality for the random variable X, we gain

$$\mathbb{P}(X - \mathbb{E}[X] > \varepsilon) \leq \exp\left(-\frac{2\varepsilon^2}{\Delta_T^2}\right)$$

$$\mathbb{P}\left(X - \frac{H(\log(T) + 1)}{2\lambda T} - d_T > \varepsilon\right) \leq \exp\left(-\frac{2\varepsilon^2}{\Delta_T^2}\right)$$

$$\mathbb{P}\left(X \leq \frac{H(\log(T) + 1)}{2\lambda T} + d_T + \varepsilon\right) > 1 - \exp\left(-\frac{2\varepsilon^2}{\Delta_T^2}\right)$$

Choosing $\delta = \exp\left(-\frac{2\varepsilon^2}{\Delta_T^2}\right)$ or $\varepsilon = \Delta_T \sqrt{\frac{1}{2} \log \frac{1}{\delta}}$, then with the probability at least $1 - \delta$, we have

$$f(\mathbf{w}_r) - f(\mathbf{w}^*) \leq \frac{H(\log(T) + 1)}{2\lambda T} + d_T + \Delta_T \sqrt{\frac{1}{2} \log \frac{1}{\delta}}$$

■

Proof of Theorem 9

We denote $W_T^\alpha = \mathbb{E}\left[\left\|\mathbf{w}_{(1-\alpha)T+1} - \mathbf{w}^*\right\|^2\right]$. Our proof proceeds as follows.

i) If Z_t is independent with $\mathbf{w}_t$, taking sum in Eq. (5) when $t = (1-\alpha)T+1, \ldots, T$, we gain

$$\begin{aligned}\sum_{t=(1-\alpha)T+1}^{T} \mathbb{E}\left[f\left(\mathbf{w}_t\right)\right] - \alpha T f\left(\mathbf{w}^*\right) \leq & \frac{\lambda\left(1-\alpha\right)T}{2} W_T^\alpha + \frac{H}{2\lambda} \sum_{t=(1-\alpha)T+1}^{T} \frac{1}{t} \\ & + \sum_{t=1}^{T} \mathbb{P}\left(Z_t = 1\right)^{1/2} \mathbb{E}\left[\left\|\mathbf{w}_t - \mathbf{w}^*\right\|^2\right]^{1/2} \mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2} \quad (12) \\ \leq & \frac{\lambda\left(1-\alpha\right)T}{2} W_T^\alpha + \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda} \\ & + \delta_\Phi M^{1/2} W^{1/2} \sum_{t=(1-\alpha)T+1}^{T} \mathbb{P}\left(Z_t = 1\right)^{1/2}\end{aligned}$$

where we have used the inequality $\sum_{t=(1-\alpha)T+1}^{T} \frac{1}{t} \leq \log\left(1/\left(1-\alpha\right)\right)$.

$$\begin{aligned}\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T^\alpha\right) - f\left(\mathbf{w}^*\right)\right] \leq & \frac{\lambda\left(1-\alpha\right)}{2\alpha} W_T^\alpha + \frac{\delta_\Phi M^{1/2} W^{1/2}}{\alpha T} \sum_{t=(1-\alpha)T+1}^{T} \mathbb{P}\left(Z_t = 1\right)^{1/2} \\ & + \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T}\end{aligned}$$

ii) If Z_t is independent with $\mathbf{w}_t$ and (x_t, y_t), taking sum in Eq. (7) when $t = (1-\alpha)T+1, \ldots, T$, we gain

$$\begin{aligned}\sum_{t=(1-\alpha)T+1}^{T} \mathbb{E}\left[f\left(\mathbf{w}_t\right)\right] - \alpha T f\left(\mathbf{w}^*\right) \leq & \frac{\lambda\left(1-\alpha\right)T}{2} W_T^\alpha + \frac{H}{2\lambda} \sum_{t=(1-\alpha)T+1}^{T} \frac{1}{t} \\ & + \sum_{t=(1-\alpha)T+1}^{T} \mathbb{P}\left(Z_t = 1\right) \mathbb{E}\left[\left\|\mathbf{w}_t - \mathbf{w}^*\right\|^2\right]^{1/2} \mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2} \quad (13) \\ \leq & \frac{\lambda\left(1-\alpha\right)T}{2} W_T^\alpha + \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda} \\ & + \delta_\Phi M^{1/2} W^{1/2} \sum_{t=(1-\alpha)T+1}^{T} \mathbb{P}\left(Z_t = 1\right)\end{aligned}$$

$$\begin{aligned}\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T^\alpha\right) - f\left(\mathbf{w}^*\right)\right] \leq & \frac{\lambda\left(1-\alpha\right)}{2\alpha} W_T^\alpha + \frac{\delta_\Phi M^{1/2} W^{1/2}}{\alpha T} \sum_{t=(1-\alpha)T+1}^{T} \mathbb{P}\left(Z_t = 1\right) \\ & + \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T}\end{aligned}$$

iii) In general, taking sum in Eq. (9) when $t = (1-\alpha)T+1, \ldots, T$, we gain

$$\begin{aligned}\sum_{t=(1-\alpha)T+1}^{T} \mathbb{E}\left[f\left(\mathbf{w}_t\right)\right] - \alpha T f\left(\mathbf{w}^*\right) \leq & \frac{\lambda\left(1-\alpha\right)T}{2} W_T^\alpha + \frac{H}{2\lambda} \sum_{t=(1-\alpha)T+1}^{T} \frac{1}{t} \\ & + \sum_{t=(1-\alpha)T+1}^{T} \mathbb{E}\left[\left\|\mathbf{w}_t - \mathbf{w}^*\right\|^2\right]^{1/2} \mathbb{E}\left[\left\|\Delta_t\right\|^2\right]^{1/2} \\ \leq & \frac{\lambda\left(1-\alpha\right)T}{2} W_T^\alpha + \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda} + \delta_\Phi M^{1/2} W^{1/2} \alpha T \quad (14)\end{aligned}$$

$$\mathbb{E}\left[f\left(\overline{\mathbf{w}}_T^\alpha\right) - f\left(\mathbf{w}^*\right)\right] \leq \frac{\lambda\left(1-\alpha\right)}{2\alpha} W_T^\alpha + \delta_\Phi M^{1/2} W^{1/2} + \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T}$$

■

Proof of Theorem 10

The proof of this theorem is similar to that of Theorem 8 which relies on Hoeffding inequality.

From Eqs. (12, 13, 14), we achieve

$$\frac{1}{\alpha T} \sum_{t=(1-\alpha)T+1}^{T} \mathbb{E}\left[f\left(\mathbf{w}_t\right)\right] - f\left(\mathbf{w}^*\right) \leq \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T} + d_T$$

Let us denote $X = f\left(\mathbf{w}_r\right) - f\left(\mathbf{w}^*\right)$, where r is uniformly sampled from $\{(1-\alpha)T+1, 2, \ldots, T\}$. We have

$$\mathbb{E}_r\left[X\right] = \frac{1}{\alpha T} \sum_{t=(1-\alpha)T+1}^{T} \mathbb{E}\left[f\left(\mathbf{w}_t\right)\right] - f\left(\mathbf{w}^*\right) \leq \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T} + d_T$$

It follows that

$$\mathbb{E}\left[X\right] = \mathbb{E}_{(x_t, y_t)_{t=1}^T}\left[\mathbb{E}_r\left[X\right]\right] \leq \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T} + d_T$$

Let us denote $\Delta_T^\alpha = \max_{(1-\alpha)T+1 \leq t \leq T} \left(f\left(\mathbf{w}_t\right) - f\left(\mathbf{w}^*\right)\right)$ which implies that $0 < f\left(\mathbf{w}_r\right) - f\left(\mathbf{w}^*\right) < \Delta_T^\alpha$. Applying Hoeffding inequality for the random variable X, we gain

$$\mathbb{P}\left(X - \mathbb{E}\left[X\right] > \varepsilon\right) \leq \exp\left(-\frac{2\varepsilon^2}{\left(\Delta_T^\alpha\right)^2}\right)$$

$$\mathbb{P}\left(X - \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T} - d_T > \varepsilon\right) \leq \exp\left(-\frac{2\varepsilon^2}{\left(\Delta_T^\alpha\right)^2}\right)$$

$$\mathbb{P}\left(X \leq \frac{H\log\left(1/\left(1-\alpha\right)\right)}{2\lambda\alpha T} + d_T + \varepsilon\right) > 1 - \exp\left(-\frac{2\varepsilon^2}{\left(\Delta_T^\alpha\right)^2}\right)$$

Choosing $\delta = \exp\left(-\frac{2\varepsilon^2}{(\Delta_T^\alpha)^2}\right)$ or $\varepsilon = \Delta_T^\alpha\sqrt{\frac{1}{2}\log\frac{1}{\delta}}$, then with the probability at least $1-\delta$, we have

$$f(\mathbf{w}_r) - f(\mathbf{w}^*) \leq \frac{H\log(1/(1-\alpha))}{2\lambda\alpha T} + d_T + \Delta_T^\alpha\sqrt{\frac{1}{2}\log\frac{1}{\delta}}$$

■

Proof of Theorem 13

It is apparent that $f(\mathbf{w})$ is L-strongly smooth w.r.t $\|.\|_2$. Therefore, we have

$$f(\mathbf{w}_r) - f(\mathbf{w}^*) \leq f'(\mathbf{w}^*)^\mathsf{T}(\mathbf{w}_r - \mathbf{w}^*) + \frac{L}{2}\|\mathbf{w}_r - \mathbf{w}^*\|^2 \leq \frac{L}{2}\|\mathbf{w}_r - \mathbf{w}^*\|^2$$

It follows that $\Delta_T^\alpha \leq \frac{1}{2}LM_T^\alpha$. Hence we gain the conclusion. ■

12. Proofs of Bound for L2 Loss

We now consider the upper bound of $\|\mathbf{w}^*\|$ in the case that ℓ_2 loss is being used for regression problem. Concretely, we have the following theorem whose proof is similar to that of Theorem 1 in Shalev-Shwartz et al. (2007, 2011).

Theorem 23. *If* $\mathbf{w}^* = argmin_{\mathbf{w}}\left(\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^N\left(y_i - \mathbf{w}^\mathsf{T}\Phi(x_i)\right)^2\right)$ *then* $\|\mathbf{w}^*\| \leq y_{max}\lambda^{-1/2}$.

Proof Let us consider the equivalent constrains optimization problem

$$\min_{\mathbf{w},\boldsymbol{\xi}}\left(\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{N}\sum_{i=1}^N\xi_i^2\right)$$
$$\text{s.t.: } \xi_i = y_i - \mathbf{w}^\mathsf{T}\Phi(x_i),\ \forall i$$

The Lagrange function is of the following form

$$\mathcal{L}(\mathbf{w},\boldsymbol{\xi},\boldsymbol{\alpha}) = \frac{\lambda}{2}\left\|\mathbf{w}^2\right\| + \frac{1}{N}\sum_{i=1}^N\xi_i^2 + \sum_{i=1}^N\alpha_i\left(y_i - \mathbf{w}^\mathsf{T}\Phi(x_i) - \xi_i\right)$$

Setting the derivatives to 0, we gain

$$\nabla_{\mathbf{w}}\mathcal{L} = \lambda\mathbf{w} - \sum_{i=1}^N\alpha_i\Phi(x_i) = 0 \rightarrow \mathbf{w} = \lambda^{-1}\sum_{i=1}^N\alpha_i\Phi(x_i)$$

$$\nabla_{\xi_i}\mathcal{L} = \frac{2}{N}\xi_i - \alpha_i = 0 \rightarrow \xi_i = \frac{N\alpha_i}{2}$$

Substituting the above to the Lagrange function, we gain the dual form

$$\begin{aligned}\mathcal{W}(\boldsymbol{\alpha}) &= -\frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_{i=1}^N y_i\alpha_i - \frac{N}{4}\sum_{i=1}^N\alpha_i^2 \\ &= -\frac{1}{2\lambda}\left\|\sum_{i=1}^N\alpha_i\Phi(x_i)\right\|^2 + \sum_{i=1}^N y_i\alpha_i - \frac{N}{4}\sum_{i=1}^N\alpha_i^2\end{aligned}$$

Let us denote $(\mathbf{w}^*, \boldsymbol{\xi}^*)$ and $\boldsymbol{\alpha}^*$ be the primal and dual solutions, respectively. Since the strong duality holds, we have

$$\frac{\lambda}{2}\|\mathbf{w}^*\|^2 + \frac{1}{N}\sum_{i=1}^N\xi_i^{*2} = -\frac{\lambda}{2}\|\mathbf{w}^*\|^2 + \sum_{i=1}^N y_i\alpha_i^* - \frac{N}{4}\sum_{i=1}^N\alpha_i^{*2}$$

$$\begin{aligned}\lambda\|\mathbf{w}^*\|^2 &= \sum_{i=1}^N y_i\alpha_i^* - \frac{N}{4}\sum_{i=1}^N\alpha_i^{*2} - \frac{1}{N}\sum_{i=1}^N\xi_i^{*2} \\ &\leq \sum_{i=1}^N\left(y_i\alpha_i^* - \frac{N}{4}\alpha_i^{*2}\right) \leq \sum_{i=1}^N\frac{y_i^2}{N} \leq y_{\max}^2\end{aligned}$$

We note that we have used $g(\alpha_i^*) = y_i\alpha_i^* - \frac{N}{4}\alpha_i^{*2} \leq g\left(\frac{2y_i}{N}\right) = \frac{y_i^2}{N}$. Hence, we gain the conclusion. ■

Lemma 24. *Assume that ℓ_2 loss is using, the following statement holds*

$$\|\mathbf{w}_{T+1}\| \leq \lambda^{-1}\left(y_{max} + \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{w}_t\|\right)$$

where $y_{max} = \max_{y\in\mathcal{Y}}|y|$.

Proof We have the following

$$\mathbf{w}_{t+1} = \begin{cases}\prod_S\left(\frac{t-1}{t}\mathbf{w}_t - \eta_t\alpha_t\Phi(x_t)\right) & \text{if } Z_t = 0\\ \prod_S\left(\frac{t-1}{t}\mathbf{w}_t - \eta_t\alpha_t\Phi(c_{i_t})\right) & \text{otherwise}\end{cases}$$

It follows that

$$\|\mathbf{w}_{t+1}\| \leq \frac{t-1}{t}\|\mathbf{w}_t\| + \frac{1}{\lambda t}|\alpha_t| \quad \text{since } \|\Phi(x_t)\| = \|\Phi(c_{i_t})\| = 1$$

It happens that $l'(\mathbf{w}_t; x_t, y_t) = \alpha_t\Phi(x_t)$. Hence, we gain

$$|\alpha_t| = \left|y_t - \mathbf{w}_t^{\mathsf{T}}\Phi(x_t)\right| \leq y_{\max} + \|\mathbf{w}_t\|\,\|\Phi(x_t)\| \leq y_{\max} + \|\mathbf{w}_t\|$$

It implies that

$$t\|\mathbf{w}_{t+1}\| \leq (t-1)\|\mathbf{w}_t\| + \lambda^{-1}(y_{\max} + \|\mathbf{w}_t\|)$$

Taking sum when $t = 1, 2, \ldots, T$, we achieve

$$T\|\mathbf{w}_{T+1}\| \leq \lambda^{-1}\left(Ty_{\max} + \sum_{t=1}^{T}\|\mathbf{w}_t\|\right)$$

$$\|\mathbf{w}_{T+1}\| \leq \lambda^{-1}\left(y_{\max} + \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{w}_t\|\right) \tag{15}$$

■

Theorem 25. *If $\lambda > 1$ then $\|\mathbf{w}_{T+1}\| \leq \frac{y_{max}}{\lambda-1}\left(1 - \frac{1}{\lambda^T}\right) < \frac{y_{max}}{\lambda-1}$ for all T.*

Proof First we consider the sequence $\{s_T\}_T$ which is identified as $s_{T+1} = \lambda^{-1}(y_{\max} + s_T)$ and $s_1 = 0$. It is easy to find the formula of this sequence as

$$s_{T+1} - \frac{y_{\max}}{\lambda-1} = \lambda^{-1}\left(s_T - \frac{y_{\max}}{\lambda-1}\right) = \ldots = \lambda^{-T}\left(s_1 - \frac{y_{\max}}{\lambda-1}\right) = \frac{-\lambda^{-T}y_{\max}}{\lambda-1}$$

$$s_{T+1} = \frac{y_{\max}}{\lambda-1}\left(1 - \frac{1}{\lambda^T}\right)$$

We prove by induction by T that $\|\mathbf{w}_T\| \leq s_T$ for all T. It is obvious that $\|\mathbf{w}_1\| = s_1 = 0$. Assume that $\|\mathbf{w}_t\| \leq s_t$ for $t \leq T$, we verify it for $T+1$. Indeed, we have

$$\begin{aligned}\|\mathbf{w}_{T+1}\| &\leq \lambda^{-1}\left(y_{\max} + \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{w}_t\|\right) \leq \lambda^{-1}\left(y_{\max} + \frac{1}{T}\sum_{t=1}^{T}s_t\right)\\ &\leq \lambda^{-1}(y_{\max} + s_T) = s_{T+1}\end{aligned}$$

■

In addition, from Eq. (15) in case that $\lambda \leq 1$ we cannot bound $\|\mathbf{w}_{T+1}\|$. Concretely, we have the following theorem.

Theorem 26. *If $\{z_T\}_T$ is a sequence such that $z_{T+1} = \lambda^{-1}\left(y_{max} + \frac{1}{T}\sum_{t=1}^{T}z_t\right)$ with $z_1 = 0$ then in case that $\lambda \leq 1$ this sequence is not upper-bounded.*

Proof Let us denote $s_t = y_{\max}^{-1}z_t$. It is obvious that $s_1 = 0$ and $s_{T+1} = \lambda^{-1}\left(1 + \frac{1}{T}\sum_{t=1}^{T}s_t\right)$. We now prove by induction by T that

$$s_T \geq \sum_{t=1}^{T-1}\frac{1}{t} \text{ for all } T \geq 2$$

With $T = 2$, we have $s_2 = \lambda^{-1} \geq 1$. Assume that this holds for all $2 \leq t \leq T$, we verify it for $T+1$.

$$\begin{aligned}s_{T+1} &= \lambda^{-1}\left(1 + \frac{1}{T}\sum_{t=1}^{T}s_t\right) \geq 1 + \frac{1}{T}\sum_{t=1}^{T}s_t\\ &\geq 1 + \frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{t-1}\frac{1}{n} \geq 1 + \sum_{t=1}^{T-1}\frac{T-t}{Tt}\\ &\geq 1 + \sum_{t=1}^{T}\left(\frac{1}{t} - \frac{1}{T}\right) = \sum_{t=1}^{T}\frac{1}{t}\end{aligned}$$

The final conclusion is obvious from the fact $\sum_{t=1}^{T}\frac{1}{t} > \log(T+1)$ and hence can exceed any positive number. ■

References

M. Badoiu and K. L. Clarkson. Optimal core-sets for balls. In *DIMACS Workshop on Computational Geometry*, 2002.

G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Tracking the best hyperplane with a simple budget perceptron. *Machine Learning*, 69(2-3):143–167, 2007.

C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Transaction Intelligence System Technology*, 2(3):27:1–27:27, May 2011. ISSN 2157-6904.

C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal Machine Learning Research*, 2:265–292, March 2002. ISSN 1532-4435.

K. Crammer, J. Kandola, and Y. Singer. Online classification on a budget. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2004.

K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal Machine Learning Research*, 7:551–585, 2006.

F. Cucker and S. Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2002.

O. Dekel, S. Shalev-Shwartz, and Y. Singer. The forgetron: A kernel-based perceptron on a fixed budget. In *Advances in Neural Information Processing Systems (NIPS)*, pages 259–266, 2005.

M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In *International Conference on Machine Learning (ICML)*, ICML '08, pages 264–271, New York, NY, USA, 2008. ACM.

Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, December 1999.

J. Hensman, N. Fusi, and N. D Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, page 282. Citeseer, 2013.

T. Joachims. Advances in kernel methods. chapter Making Large-scale Support Vector Machine Learning Practical, pages 169–184. MIT Press, Cambridge, MA, USA, 1999. ISBN 0-262-19416-3.

J. Kivinen, A. J. Smola, and R. C. Williamson. Online Learning with Kernels. *IEEE Transactions on Signal Processing*, 52:2165–2176, August 2004.

T. Le, P. Duong, M. Dinh, D. T Nguyen, V. Nguyen, and D. Phung. Budgeted semi-supervised support vector machine. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, June 2016a.

T. Le, T. D. Nguyen, V. Nguyen, and D. Phung. Dual space gradient descent for online learning. In *Advances in Neural Information Processing (NIPS)*, pages 4583–4591, December 2016b.

T. Le, V. Nguyen, T. D. Nguyen, and Dinh Phung. Nonparametric budgeted stochastic gradient descent. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 654–572, May 2016c.

T. Le, K. Nguyen, V. Nguyen, D. T. Nguyen, and D. Phung. Gogp: Fast online regression with gaussian processes. In *International Conference on Data Mining (ICDM)*, 2017.

J. Lu, S. C.H. Hoi, J. Wang, P. Zhao, and Z.-Y. Liu. Large scale online kernel learning. *Journal Machine Learning Research*, 2015.

L. Ming, W. Shifeng, and Z. Changshui. On the sample complexity of random fourier features for online learning: How many random fourier features do we need? *ACM Transaction Knowledge Discovery Data*, 8(3):13:1–13:19, June 2014. ISSN 1556-4681.

F. Orabona, J. Keshet, and B. Caputo. Bounded kernel-based online learning. *Journal Machine Learning Research*, 10:2643–2666, December 2009. ISSN 1532-4435.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances In Neural Infomration Processing Systems (NIPS)*, 2007.

A. Rakhlin, O. Shamir, and K. Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *International Conference on Machine Learning (ICML)*, pages 449–456, 2012.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. ISBN 026218253X.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

S. Shalev-Shwartz and N. Srebro. Svm optimization: inverse dependence on training set size. In *International Conference on Machine Learning (ICML)*, pages 928–935. ACM, 2008.

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 14(1):567–599, 2013.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *International Conference on Machine Learning (ICML)*, pages 807–814, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3.

S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

I. Steinwart. Sparseness of support vector machines. *Journal Machine Learning Research*, 4:1071–1105, December 2003. ISSN 1532-4435.

I. W. Tsang, J. T. Kwok, P. Cheung, and N. Cristianini. Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, 6:363–392, 2005.

I. W. Tsang, A. Kocsor, and J. T. Kwok. Simpler core vector machines with enclosing balls. In *International Conference on Machine Learning (ICML)*, ICML '07, pages 911–918, 2007.

Z. Wang and S. Vucetic. Twin vector machines for online learning on a budget. In *SIAM International Conference on Data Mining (SDM)*, pages 906–917, 2009.

Z. Wang and S. Vucetic. Online passive-aggressive algorithms on a budget. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 908–915, 2010.

Z. Wang, K. Crammer, and S. Vucetic. Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale svm training. *Journal of Machine Learning Research*, 13(1):3103–3131, 2012.

K. Zhang, L. Lan, Z. Wang, and F. Moerchen. Scaling up kernel svm on limited resources: A low-rank linearization approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1425–1434, 2012.

P. Zhao, J. Wang, P. Wu, R. Jin, and S. C. H. Hoi. Fast bounded online gradient descent algorithms for scalable kernel-based online learning. *CoRR*, 2012.

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pages 928–936, 2003.

Journal of Machine Learning Research 18 (2017) 1-29 Submitted 6/14; Revised 11/16; Published 10/17

Efficient Sampling from Time-Varying Log-Concave Distributions

Hariharan Narayanan harin@uw.edu
Department of Statistics and Department of Mathematics
University of Washington

Alexander Rakhlin rakhlin@wharton.upenn.edu
Department of Statistics
University of Pennsylvania

Editor: Robert McCulloch

Abstract

We propose a computationally efficient random walk on a convex body which rapidly mixes with respect to a fixed log-concave distribution and closely tracks a time-varying log-concave distribution. We develop general theoretical guarantees on the required number of steps; this number can be calculated on the fly according to the distance from and the shape of the next distribution. We then illustrate the technique on several examples. Within the context of exponential families, the proposed method produces samples from a posterior distribution which is updated as data arrive in a streaming fashion. The sampling technique can be used to track time-varying truncated distributions, as well as to obtain samples from a changing mixture model, fitted in a streaming fashion to data. In the setting of linear optimization, the proposed method has oracle complexity with best known dependence on the dimension for certain geometries. In the context of online learning and repeated games, the algorithm is an efficient method for implementing no-regret mixture forecasting strategies. Remarkably, in some of these examples, only one step of the random walk is needed to track the next distribution.[1]

1. Introduction

Let $\mathcal{K}$ be a compact convex subset of $\mathbb{R}^d$ with non-empty interior. Let $\mu_0, \ldots, \mu_t, \ldots$ be a sequence of probability measures with support on $\mathcal{K}$. Suppose each probability distribution μ_t has a density

$$\frac{d\mu_t(x)}{dx} = \frac{e^{-s_t(x)}}{Z_t}\ , \qquad Z_t = \int_{x\in\mathcal{K}} e^{-s_t(x)}dx \tag{1}$$

with respect to the Lebesgue measure, where each $s_t(x)$ is a convex function on $\mathcal{K}$. This paper proposes a Markov Chain Monte Carlo method for sequentially sampling from these distributions. The method comes with strong mixing time guarantees, and is shown to be applicable to a variety of problems. Observe that, by definition, the distributions μ_t are *log-concave*, and thus our work falls within the emerging body of literature on sampling from log-concave distributions.

The problem of sampling from distributions arises in many areas of statistics, most notably in Bayesian inference (Robert and Casella, 2004). In particular, Sequential Monte Carlo methods (Doucet et al., 2001) aim to sample from time-varying distributions. The need for such methods arises, for instance, in the case of online arrival of data: it is desirable to be able to update the posterior distribution at a low computational cost. If the distributions are changing "slowly" with time, sequential methods can re-use samples from the previous distribution and perform certain re-weighting to track the next distribution, thus saving computational resources. These ideas are exploited in particle filtering methods (see (Chopin, 2002; Doucet et al., 2001) and references therein). Beyond Bayesian inference, other applications of sampling from distributions include simulated annealing, global optimization, and regret minimization.

The main critique of the MCMC methods is, in many situations, the lack of mixing time analysis. In practice, the number of steps of the chain required to obtain an honest sample from a distribution is mostly calculated based on heuristics. There is a growing body of literature that presents exceptions to these heuristic approaches (Diaconis, 2013). Coupling methods, spectral gap methods, as well as the more recent study of positive Ricci curvature, yield geometric decrease of the distance to the desired stationary distribution – a property known as *geometric ergodicity*. The most well-understood cases in this context are those with a finite or countable state space (see (Meyn and Tweedie, 2009; Diaconis, 2009)). In contrast, we are interested in a random walk on a non-discrete set.

This paper is focused on a particular circle of problems defined via log-concave distributions. These distributions constitute an important subset of the set of unimodal distributions, a fact that has been recognized within Statistics (see e.g. (Walther, 2009)). We are not the first to study mixing times for such distributions: this line of work started with the breakthrough paper of (Dyer et al., 1991), followed by a series of improvements (Frieze et al., 1994; Lovász, 1999; Lovász and Vempala, 2006, 2007). However, the recent advances in (Kannan and Narayanan, 2012) on sampling from convex bodies give an edge to obtaining stronger guarantees. In particular, a variant of the Dikin walk studied in this paper is analyzed in (Narayanan, 2016) for the case of log-linear distributions. Extending the study of this random walk, we show rapid mixing to a log-concave distribution, and further show that we can provably track a *changing* log-concave distribution with a small number (or even only *one step*) of a random walk, provided that the distribution changes slowly enough. *Such a result seems out of reach with other random walk methods due to the lack of scale-free bounds on conductance.*

We assume that we can compute a *self-concordant barrier* (see Section 5 and Appendix 8) for the set $\mathcal{K}$, a requirement that is satisfied in many cases of interest. For instance, the self-concordant barrier can be readily computed in closed form if $\mathcal{K}$ is defined via linear and quadratic constraints. While the availability of the barrier is a stronger assumption than, for instance, access to a separation oracle for $\mathcal{K}$, the barrier gives a better handle on the geometry of the space and yields fast mixing of the Markov chain.

In Section 5, we illustrate the method within several diverse application domains. As one of the examples, we consider the problem of updating the posterior with respect to a conjugate prior in an exponential family, where the parameter is taking values in a space of a

1. An extended abstract containing partial results appeared in the proceedings of the NIPS 2010 conference (Narayanan and Rakhlin, 2010).

©2017 Hariharan Narayanan and Alexander Rakhlin.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/14-223.html.

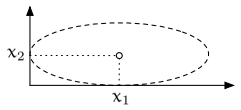
fixed dimensionality given by the sufficient statistics. The constraints then constitute a prior knowledge about the possible location of the parameter. As another example, we consider sampling from a time-varying truncated distribution, as well as the extension to sampling from mixture models fitted to streaming data. We employ the sampling technique to the classical problem of linear optimization via simulated annealing. The final example concerns the problem of regret minimization where the log-concave distribution arises naturally from the exponential weighting scheme.

The paper is organized as follows. In the next section we study the geometry of the set $\mathcal{K}$ induced by a self-concordant barrier and prove a key isoperimetric inequality in the corresponding Riemannian metric. The Markov chain for a given log-concave distribution is defined in Section 3. Conditions on the size of a step are introduced in Section 3.1, and a lower bound on the conductance of the chain is proved in Section 3.2. Section 4 contains main results about tracking time-varying distributions given appropriate measures of change between time steps. Section 5 is devoted to applications. Finally, Sections 6 and 7 contain all the remaining proofs.

2. Geometry Induced by the Self-Concordant Barrier

The Markov chain studied in this paper uses as a proposal a Gaussian distribution with a covariance that approximates well the local geometry of the set $\mathcal{K}$ at the current point. This local geometry plays a crucial role in the theory of interior point methods for optimization, yet for our purposes a handle on the local geometry yields a good lower bound on *conductance* of the Markov chain. Further intriguing similarities between optimization and sampling will be pointed out throughout the paper.

We refer to (Nemirovskii, 2004) for an introduction to the theory of interior point methods, a subject centered around the notion of a self-concordant barrier. Once we have defined a self-concordant barrier for $\mathcal{K}$, the local geometry is defined through the Hessian of the barrier at the current point. For the reader unfamiliar with the literature on self-concordant barriers, it is useful to think of the simple 2-dimensional example of $F(x) = -\log x_1 - \log x_2$ defined on the positive quadrant $\mathcal{K} = [0,\infty)^2$, with $x = (x_1, x_2)$. The Hessian of F at x is $\begin{bmatrix} 1/x_1^2 & 0 \\ 0 & 1/x_2^2 \end{bmatrix}$, and a unit ball centered at x and reshaped according to this matrix is the ellipse that touches the axes, as shown below. That is, the ellipsoid corresponds well to the



local geometry of $\mathcal{K}$ at the point x.

For a function F on the interior $int(\mathcal{K})$ having continuous derivatives of order k, for vectors $h_1, \ldots, h_k \in \mathbb{R}^d$ and $x \in int(\mathcal{K})$, for $k \geq 1$, we recursively define

$$D^k F(x)[h_1, \ldots, h_k] \triangleq \lim_{\epsilon \to 0} \frac{D^{k-1}(x + \epsilon h_k)[h_1, \ldots, h_{k-1}] - D^{k-1}(x)[h_1, \ldots, h_{k-1}]}{\epsilon}$$

where $D^0 F(x) \triangleq F(x)$. Let F be a self-concordant barrier of $\mathcal{K}$ with a parameter ν (see Appendix 8 for the precise definition and Section 5 for examples). The value of the parameter ν depends on the shape of $\mathcal{K}$ and the choice of F. The barrier parameter will enter many of the results in this paper, but it is worth mentioning that there always exists a barrier with $\nu = O(d)$.

The barrier induces a Riemannian metric whose metric tensor is the Hessian of F (Nesterov and Todd, 2008). In other words, the metric tensor on the tangent space at x assigns to a vector v the length

$$\|v\|_x^2 \triangleq D^2 F(x)[v, v],$$

and to a pair of vectors v, w, the inner product

$$\langle v, w \rangle_x \triangleq D^2 F(x)[v, w] \ .$$

The unit ball in $\|\cdot\|_x$ around a point x is called the *Dikin ellipsoid* (Nemirovskii, 2004), and it describes well the local geometry of $\mathcal{K}$ at x in the following sense: (i) the unit Dikin ellipsoid at any point is contained in $\mathcal{K}$, and (ii) the Dikin ellipsoid of radius $r = 2(1 + 3\nu)$ contains $\mathrm{sym}(\mathcal{K}, x) = \{\mathcal{K} - x\} \cap -\{\mathcal{K} - x\}$.

The random walk, introduced in the next section, is *anisotropic*, i.e. the steps change in size and shape from point to point. It is then useful to connect the properties of this random walk directly to the Riemannian distance that is defined in terms of the Hessian of F. More precisely, for $x, y \in \mathcal{K}$, let $\rho(x, y)$ be the Riemannian distance $\rho(x, y) = \inf_\Gamma \int_z \|d\Gamma\|_z$ where the infimum is taken over all rectifiable paths Γ from x to y. Let $\mathcal{M}$ be the metric space whose point set is $\mathcal{K}$ and metric is ρ, and define $\rho(S_1, S_2) = \inf_{x \in S_1, y \in S_2} \rho(x, y)$. The first main ingredient of the analysis is an isoperimetric inequality.

Theorem 1 *Let S_1 and S_2 be measurable subsets of $\mathcal{K}$ and μ a probability measure supported on $\mathcal{K}$ that possesses a density whose logarithm is concave. Then it holds that*

$$\mu((\mathcal{K} \setminus S_1) \setminus S_2) \geq \frac{1}{2(1 + 3\nu)} \rho(S_1, S_2) \mu(S_1) \mu(S_2).$$

The theorem ensures that two subsets well-separated in ρ distance must have a large mass between them. A lower bound on conductance of our Markov chain will follow from this isoperimetric inequality. We remark that convexity of the set $\mathcal{K}$ is crucial for the above property. A classical example of a non-convex shape with a "bottleneck" is a dumbbell. For this body, the above statement clearly fails, and a "local" random walk on such a body gets trapped in either of the two parts for a long time.

2.1 Connections to Interior Point Methods

Interestingly, the idea of tracking a changing distribution with only one step of a random walk parallels the technique of following a central path in the theory of interior point methods for optimization, as discussed in (Narayanan and Rakhlin, 2010). In the analysis of interior point methods, the local (according to the Dikin ellipsoid introduced in the next section) quadratic convergence of the Damped Newton step counters the slowly changing "temperature parameter" of the barrier to ensure sufficiency of one optimization step; in our method, the geometric ergodicity of the scale-free random walk (which is based on the

local shape of the Dikin ellipsoid) balances the additive change in the distribution due to the changing temperature. Once again, the scale-free property of the Dikin walk, introduced below, is crucial for drawing this parallel between one-step interior point methods and our one-step random walk.

We remark that a different parallel between sampling and interior point methods has been recently outlined in (Abernethy and Hazan, 2016). The authors show that the *mean* of the log-concave distribution coincides with the central path when the barrier is chosen to be entropic. Since one still needs to sample from the distribution, this does not imply an "algorithmic equivalence" of the two methods unless one proves a scale-free bound on conductance, as we do in this paper.

3. The Markov Chain

Let $\mathcal{B}$ be the Borel σ-field on $\mathcal{K}$. Given an initial probability measure on $\mathcal{K}$, a Markov chain is specified by a collection of one-step transition probabilities

$$\{\mathsf{P}(x, B), x \in \mathcal{K}, B \in \mathcal{B}\}$$

such that $x \mapsto \mathsf{P}(x, B)$ is a measurable map for any $B \in \mathcal{B}$ and $\mathsf{P}_x(\cdot) \triangleq \mathsf{P}(x, \cdot)$ is a probability measure on $\mathcal{K}$ for any $x \in \mathcal{K}$.

For $x \in int(\mathcal{K})$, let G_x^r denote the unique Gaussian probability density function on $\mathbb{R}^d$ such that

$$G_x^r(y) \propto \exp\left(-\frac{d\|x-y\|_x^2}{r^2} + V(x)\right), \qquad V(x) \triangleq \frac{1}{2}\ln\det D^2F(x)$$

and r is a parameter that is chosen according to a condition specified below. The covariance of this distribution is given by the Hessian of F at point x, and thus the contour lines are scaled Dikin ellipsoids.

The Markov chain considered in this paper is based on the Dikin Walk introduced by Kannan and Narayanan (2012). Adapted to sampling from log-concave distributions in this paper, the Markov chain is parametrized by a convex function s and a step size r. Rather than writing out the unwieldy explicit form of the transition kernel P_x, we can give it implicitly as the following random walk:

With probability 1/2, set $w := x$.

With probability 1/2, sample z from G_x^r and

If $z \notin \mathcal{K}$, let $w := x$.

If $z \in \mathcal{K}$, let $w := \begin{cases} z & \text{with prob.} \quad \min\left(1,\ \frac{G_z^r(x)\exp(s(x))}{G_x^r(z)\exp(s(z))}\right) \\ x & \text{otherwise.} \end{cases}$

The Markov chain is *lazy*, as it stays at the current point with probability at least 1/2. This ensures uniqueness of the stationary distribution (Lovász and Simonovits, 1993). Furthermore, a simple calculation shows that the detailed balance conditions are satisfied with

respect to a stationary distribution μ whose density (with respect to the Lebesgue measure) is proportional to $\exp(-s(x))$. Indeed, to see that $\mu(x)\mathsf{P}_x(dz) = \mu(z)\mathsf{P}_z(dx)$, it suffices to observe that

$$\exp(-s(x))G_x^r(z)\min\left(1,\ \frac{G_z^r(x)\exp(s(x))}{G_x^r(z)\exp(s(z))}\right) = \exp(-s(z))G_z^r(x)\min\left(1,\ \frac{G_x^r(z)\exp(s(z))}{G_z^r(x)\exp(s(x))}\right).$$

Therefore the Markov chain is reversible and has the desired stationary measure μ.

The value of r has a specific meaning: most of the y's sampled from G_x^r are within a thin "Dikin shell" of radius proportional to $(\mathbb{E}\|x-y\|_x^2)^{1/2} = r$ by measure-concentration arguments. We will therefore refer to r as the effective "step size". An important and non-trivial result from the theory of interior point methods is that the unit Dikin ellipsoid is contained in the set $\mathcal{K}$ and gives a good approximation to the local geometry of the set (see Figure 1 below). Thanks to this fact, the sampling procedure has in general better mixing properties than the Ball Walk (Lovász and Simonovits, 1993; Vempala, 2005).

3.1 Step Size Conditions

The analysis of the Markov chain requires the steps r to be not too large to ensure that different enough transition probability functions happen only for far away points. The precise upper bounds on r depend on the convex function $s(x)$ and can be calculated on the fly when we move to the setting of a time-varying function. We give four conditions:

Sufficient Condition 1 (Linear Functions) *If s is linear, we may set $r = 1/d$.*

Sufficient Condition 2 (Lipschitz Functions) *For a function s that is L-Lipschitz with respect to the Euclidean norm, we may set the step size $r = \min\left\{\frac{1}{d}, \frac{1}{L}\right\}$.*

Sufficient Condition 3 (Smooth Functions) *Suppose s has Lipschitz-continuous gradients: there exists $\sigma > 0$ such that $\|\nabla s(x) - \nabla s(y)\| \le \sigma\|x-y\|$. We may then set the step size to be $\min\left\{\frac{1}{d}, \frac{1}{\sqrt{\sigma}}\right\}$.*

These three conditions can be shown to follow from a more general sufficient step size condition that is based on "local" information:

Sufficient Condition 4 (General Condition) *Fix constants $C, C' > 0$. Given the convex function $s(x)$, the step size $r \le \min\left\{\frac{1}{d}, r^*\right\}$ is a valid choice if there exists a linear function $< g, x >$ such that*

$$r^* \le \sup\left\{\ r\ :\ \forall z, w \in \mathcal{K}\ \ \textit{with}\ \ \|z-w\|_z \le C'r, \quad \Big|s(z) - s(w) - \langle g, z-w\rangle\Big| < C\right\}$$

The condition says that for two points, with one being inside the $O(r)$-Dikin ellipsoid around the other point, the function is within a constant of being linear. It follows from the last condition that, for instance, if $s(x) = \langle b, x\rangle + a(x)$ is a sum of a linear and a non-linear Lipschitz part, the step size is only affected by the Lipschitz constant of the non-linear part.

It is simple to verify that the step size in Condition 2 satisfies Condition 4. Indeed, for any w such that $\|z-w\|_z \le C'r$, we have $\|z-w\| \le C''rR$ (where R is the radius of the largest ball contained in $\mathcal{K}$). Take g_z and g_w to be any subgradients of s at z and w, respectively. We then have

$$|s(z) - s(w) - \langle g_w, z-w\rangle| \le \langle g_z - g_w, z-w\rangle \le 2L\|z-w\| \le 2 \ .$$

Notice that for Condition 3, the above calculation becomes

$$\langle g_z - g_w, z-w\rangle \le \sigma\|z-w\|^2 \le 1 \ .$$

In the remainder of this paper, C will denote a universal constant that may change from line to line. The exact value of the final constant in Lemma 4 below can be traced in the proofs; we omit this calculation for the sake of brevity.

3.2 Conductance of the Markov Chain

In order to show rapid mixing of the proposed Markov chain, we prove a lower bound on its *conductance*

$$\Phi \triangleq \inf_{\mu(S_1)\le \frac{1}{2}} \frac{\int_{S_1} \mathsf{P}_x(\mathcal{K}\setminus S_1)d\mu(x)}{\mu(S_1)}, \tag{2}$$

where P_x is the one-step transition function defined earlier. Once such a lower bound is established, the following general result on the reduction of distance between distributions will imply exponentially fast convergence.

Theorem 2 ((Lovász and Simonovits, 1993)) *Let γ_0 be the initial distribution for a lazy reversible ergodic Markov chain whose conductance is Φ and stationary measure is γ. For every bounded f, let $\|f\|_\gamma \triangleq \sqrt{\int_{\mathcal{K}} f(x)^2 d\gamma(x)}$. For any fixed f, let Ef be the map that takes x to $\int_{\mathcal{K}} f(y)d\mathsf{P}_x(y)$. Then if $\int_{\mathcal{K}} f(x)d\gamma(x) = 0$, it holds that*

$$\|E^k f\|_\gamma \le \left(1 - \frac{\Phi^2}{2}\right)^{k/2} \|f\|_\gamma \ .$$

To prove a lower bound on conductance Φ, we first relate the Riemannian metric ρ to the proposed Markov Chain. Intuitively, the following result says that for close-by points, their transition distributions cannot be far apart in the total variation distance d_{TV}.

Lemma 3 *If $x, y \in \mathcal{K}$ and $\rho(x,y) \le \frac{r}{C\sqrt{d}}$ for some constant C, then*

$$d_{TV}(\mathsf{P}_x, \mathsf{P}_y) \le 1 - \frac{1}{C'}$$

for some constant C'.

Lemma 3 together with the isoperimetric inequality of Theorem 1 give a lower bound on conductance of the Markov Chain.

Lemma 4 *Let μ be a log-concave distribution with support on $\mathcal{K}$ whose density with respect to the Lebesgue measure is proportional to $\exp\{-s(x)\}$, and suppose an appropriate step size condition (Section 3.1) for the Markov chain is satisfied. Then there exists a constant $C > 0$ such that the conductance of the above Markov chain is bounded below as*

$$\Phi \ge \frac{r}{C\nu\sqrt{d}} \ .$$

We remark that the step size r enters the lower bound on Φ. While we would like the steps to be large, the conditions outlined earlier dictate a limitation on how large r can be. In particular, we always have $r \le 1/d$. The step size needs to be even smaller for functions s for which a linear approximation is poor.

We also remark that Lemma 4 establishes a *scale-free* bound on the conductance, that is, a bound that does not depend on the measure of the set S_1 in the definition (2). Such a scale-free conductance is needed for the geometric ergodicity of the chain.

4. Tracking the Distributions

Having specified the Markov chain and the step size, we now turn to the problem of tracking a sequence of distributions $\mu_1, \dots, \mu_t, \dots$. For each $t \ge 1$, define a Markov chain with parameters r_t and s_t, and let its transition kernel be denoted by $\mathsf{P}_t(x, B)$ for $x \in \mathcal{K}$ and $B \in \mathcal{B}$. Let Φ_t denote the conductance of this chain. The chain will be run for τ_t steps starting from the end of the chain at time $t-1$. Formally, let the i-th step of the t-th chain be denoted by the random variable $X_{t,i}$. Define $\tau_0 = 0$ and let $\sigma_{0,0}$ be the initial distribution of $X_{0,0}$. Then $X_{t,i}$ has distribution

$$\sigma_{0,0}\mathsf{P}_1^{\tau_1} \cdots \mathsf{P}_{t-1}^{\tau_{t-1}} \mathsf{P}_t^i$$

and we have made the identification $X_{s,\tau_s} = X_{s+1,0}$, gluing the successive chains together. Let the distribution of $X_{t,i}$ be denoted by $\sigma_{t,i}$. By the definition of the chain, $\sigma_{t,i}$ is a distribution with bounded density, supported on $\mathcal{K}$.

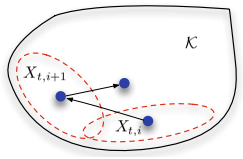


Figure 1: Steps of the Dikin Walk. The next point is sampled from a Gaussian distribution with a shape (contours depicted with dashed lines) corresponding to Dikin ellipsoids. These ellipsoids approximate well the local geometry.

4.1 Measuring the Change

Let $\|\cdot\|_t$ denote the $\mathcal{L}_2$ norm with respect to the measure μ_t, defined as $\|f\|_t = \left(\int_{\mathcal{K}} f^2 d\mu_t\right)^{1/2}$ for a measurable function $f : \mathcal{K} \to \mathbb{R}$. Further, let $\|\cdot\|_{\mathcal{K}}$ denote the supremum norm

$\|f\|_{\mathcal{K}} = \sup_{x\in\mathcal{K}} |f(x)|$ and let

$$\beta_{t+1} = \max\left\{\|d\mu_t/d\mu_{t+1}\|_{\mathcal{K}}, \|d\mu_{t+1}/d\mu_t\|_{\mathcal{K}}\right\} . \tag{3}$$

This ratio provides an upper bound on the point-wise change of the density function. A straightforward way to upper bound β_{t+1} is by writing

$$\sup_{x\in\mathcal{K}} \frac{e^{-s_t(x)}}{e^{-s_{t+1}(x)}} \frac{\int_{\mathcal{K}} e^{-s_{t+1}(x)}dx}{\int_{\mathcal{K}} e^{-s_t(x)}dx} \le \sup_{x\in\mathcal{K}} e^{2|s_t(x)-s_{t+1}(x)|}$$

and, hence,

$$\log\beta_{t+1} \le \ 2\|s_t(x) - s_{t+1}(x)\|_{\mathcal{K}} . \tag{4}$$

Another way to measure the change in successive distributions is with respect to the $\mathcal{L}_2$ norm:

$$\alpha_{t+1} = \|d\mu_t/d\mu_{t+1}\|_{t+1} \ . \tag{5}$$

In contrast to the point-wise change, the ratio α_{t+1} is more difficult to calculate. In this respect, the following result, which follows from the proof of (Lovász and Vempala, 2006; Kalai and Vempala, 2006), will be useful:

Lemma 5 *Let s_t be a convex function and $s_{t+1} = (1-\delta)^{-1} s_t$. Let μ_t and μ_{t+1} be defined as in* (1)*. Then*

$$\alpha_{t+1} \le \left(1 + \frac{\delta^2}{1-2\delta}\right)^{d/2}$$

In particular, if $\delta \le d^{-1/2} \le 1/3$, then $\alpha_{t+1} \le 5$.

We remark that the ratio between μ_t and μ_{t+1} measured in the supremum norm may be exponentially large, while the $\mathcal{L}_2$ change is small. As in (Lovász and Vempala, 2006; Kalai and Vempala, 2006), this fact will be crucial in this paper when we study simulated annealing.

4.2 Tracking the Distributions: Main Results

Denote the error in approximating the stationary distribution at the end of t-th chain by

$$\xi_t \ \triangleq \ \left\|\frac{d\sigma_{t,\tau_t}}{d\mu_t} - 1\right\|_t \tag{6}$$

and let

$$\Delta_t \ \triangleq \ \frac{r_t^2}{C\nu^2 d} .$$

Theorem 6 *The errors ξ_t satisfy the recurrence*

$$\xi_t \le (1-\Delta_t)^{\tau_t}(\beta_t^{3/2}\xi_{t-1} + \sqrt{\beta_t}(\beta_t - 1)) \tag{7}$$

for any $t \ge 1$.

Proof [Proof of Theorem 6] We iteratively apply Theorem 2 with $f = \frac{d\sigma_{t,j}}{d\mu_t} - 1$ and the stationary distribution $\gamma = \mu_t$, and observe that Ef takes $\sigma_{t,j}$ to $\sigma_{t,j+1}$. Then from Lemma 4, for $t \ge 1$ and $i \ge 1$,

$$\left\|\frac{d\sigma_{t,i}}{d\mu_t} - 1\right\|_t \le \left\|\frac{d\sigma_{t,0}}{d\mu_t} - 1\right\|_t \cdot (1-\Delta_t)^i$$

Using the first part of Lemma 11 (see Section 6)

$$\left\|\frac{d\sigma_{t,0}}{d\mu_t} - 1\right\|_t \le \beta_t^{3/2}\left\|\frac{d\sigma_{t,0}}{d\mu_{t-1}} - 1\right\|_{t-1} + \sqrt{\beta_t}(\beta_t - 1),$$

concluding the proof. An alternative recurrence, using the second part of Lemma 11, is

$$\xi_t \le (1-\Delta_t)^{\tau_t}(\sqrt{\beta_t}\xi_{t-1} + \sqrt{\beta_t - 1}),$$

which is better for large β_t but worse for $\beta_t \approx 1$. ■

We would like to adaptively choose τ_t to make the right-hand side (7) small. While the value of the error ξ_{t-1} at the previous round is not available for this purpose, let us maintain an upper bound u_{t-1} on this error. Thus, we may write τ_t as a function $\tau_t(u_{t-1}, s_t, r_t, \beta_t)$. Suppose at round $t = 0$ we ensure that $\xi_0 \le u_0$. Then, recursively, we may compute u_t as the upper bound in (7):

$$u_t \ge (1-\Delta_t)^{\tau_t}(\beta_t^{3/2}u_{t-1} + \sqrt{\beta_t}(\beta_t - 1)) \tag{8}$$

Then, given the initial condition, we have $\xi_t \le u_t$ for all $t \ge 0$.

Let us consider some consequences of Theorem 6. In particular, we are interested in situations when we can track the distributions with only *one step* of the random walk.

Corollary 7 *Let $\tau_t = 1$ for all $t \ge 1$ and suppose $\xi_0 \le u_0 = \sqrt{\beta_0}(\beta_0 - 1)/\Delta_0$ with $\Delta_0 = \frac{1}{Cd^3\nu^2} \le \frac{1}{2}$. Assume that β_t is non-decreasing and Δ_t is non-increasing in t, and suppose*

$$\beta_t^{3/2} \le 1 + \frac{\Delta_t^2}{1-\Delta_t} \tag{9}$$

for all $t \ge 1$. Then we have

$$\xi_t \le u_t = \frac{\sqrt{\beta_t}(\beta_t - 1)}{\Delta_t}$$

for all $t \ge 0$. In particular, (9) *is satisfied whenever $\beta_t - 1 \le 0.4\Delta_t^2$.*

The proof of the above corollary follows from the more general result:

Corollary 8 *Fix a sequence $\epsilon_0, \dots, \epsilon_t, \dots$ of positive target accuracies and assume $\xi_0 \le \epsilon_0$. It is then enough to set*

$$\tau_t = \left\lceil \frac{1}{\Delta_t}\log\left(\beta_t^{3/2}\cdot\frac{\epsilon_{t-1}}{\epsilon_t} + \frac{\sqrt{\beta_t}(\beta_t-1)}{\epsilon_t}\right)\right\rceil \tag{10}$$

in order to ensure $\xi_t \le \epsilon_t$ for each $t \ge 0$.

Proof Immediate by writing

$$u_t = (1-\Delta_t)^{\tau_t}(\beta_t^{3/2}\epsilon_{t-1} + \sqrt{\beta_t}(\beta_t - 1)) \leq \epsilon_t,$$

solving for τ_t, and using the approximation $\log(1/(1-\Delta_t)) \geq \log(1+\Delta_t) \geq \Delta_t$. ■

We now consider the case when one has control on the $\mathcal{L}_2$ norm α_t of the change between successive distributions. First, observe that closeness of the distributions in the norm $\|\cdot\|_t$ implies closeness in total variation distance as

$$\int |d\sigma_{t,i} - d\mu_t| = \int \left|\frac{d\sigma_{t,i}}{d\mu_t} - 1\right| d\mu_t \leq \left\|\frac{d\sigma_{t,i}}{d\mu_t} - 1\right\|_t . \tag{11}$$

Proposition 9 *Fix a sequence $\epsilon_0, \ldots, \epsilon_t, \ldots$ of positive target accuracies and assume $d_{TV}(\sigma_{0,0}, \mu_0) \leq \epsilon_0$. Suppose we set*

$$\tau_t = \left\lceil \frac{1}{\Delta_t} \log\left(\frac{\alpha_t}{\epsilon_t}\right) \right\rceil . \tag{12}$$

Then the total variation distance between σ_{t,τ_t} and μ_t is bounded as

$$d_{TV}(\sigma_{t,\tau_t}, \mu_t) \leq \sum_{s=0}^{t} \epsilon_s \tag{13}$$

for each $t \geq 0$.

Proof For any $t \geq 1$, let us write

$$\sigma_{t,\tau_t} = \mu_t + \gamma_t \tag{14}$$

with a signed measure $\gamma_t = \sigma_{t,\tau_t} - \mu_t$. By way of induction, suppose (13) holds for time t. Consider the operator E_{t+1} corresponding to the random walk of the $t+1$-st chain. The operator acts on a function f by taking f to $\int_{\mathcal{K}} f(y) d\mathsf{P}_{t+1}(x,y)$. Then applying Theorem 2 to the function $d\mu_t/d\mu_{t+1} - 1$, we have

$$\left\| E_{t+1}^{\tau_{t+1}} \left(\frac{d\mu_t}{d\mu_{t+1}} - 1 \right) \right\|_{t+1} \leq \epsilon_{t+1}$$

by the choice of τ_{t+1} and the definition of α_{t+1}. That is, upon the action of $E_{t+1}^{\tau_{t+1}}$, μ_t is mapped to μ_{t+1} within an error of at most ϵ_{t+1} in the $\mathcal{L}_2$ sense (and, hence, in the total variation sense). Since the operator E_{t+1} is non-expanding in the $\mathcal{L}_1$ sense, total variation of γ_t does not increase under the action of $E_{t+1}^{\tau_{t+1}}$. In view of the inductive hypothesis for step t, we conclude $d_{TV}(\sigma_{t+1,\tau_{t+1}}, \mu_{t+1}) \leq \sum_{s=0}^{t} \epsilon_s + \epsilon_{t+1}$, as desired. ■

5. Applications

Before diving into the applications of the random walk, let us give several examples of sets $\mathcal{K}$ for which the self-concordant barrier F and its Hessian can be easily calculated. In the following examples, assume that $\mathcal{K}$ has non-empty interior.

Example 1 *Suppose $\mathcal{K}$ is given by m linear constraints of the form $\langle a_j, x\rangle \leq b_j$, $j = 1, \ldots, m$. Then $F(x) = -\sum_{j=1}^{m} \log(b_j - \langle a_j, x\rangle)$ is a self-concordant barrier with parameter $\nu = m$. The Hessian is easily computable:*

$$D^2F(x) = \sum_{j=1}^{m} \frac{a_j a_j^\mathsf{T}}{(b_j - \langle a_j, x\rangle)^2} .$$

Example 2 *Let $\mathcal{K} = \{x \in \mathbb{R}^d : f_j(x) \leq 0, j = 1, \ldots, m\}$ where each f_j is a convex quadratic form. Then $F(x) = -\sum_{j=1}^{m} \log(-f_j(x))$ is a self-concordant barrier with parameter m. As an example, the function $-\log(R - \|x\|^2)$ is a self-concordant barrier for the unit Euclidean sphere $\{x : \|x\|^2 - 1 \leq 0\}$, with parameter $\nu = 1$, and the Hessian is given by*

$$D^2F(x) = \frac{2}{1-\|x\|^2} I + \frac{4}{(1-\|x\|^2)^2} xx^\mathsf{T} .$$

Importantly, there always exists a self-concordant barrier with $\nu = O(d)$; yet, for some convex sets (such as the sphere) the parameter can even be constant.

Self-concordant barriers can be combined: if F_j is ν_j-self-concordant for $\mathcal{K}_j$, $j = 1, \ldots, m$, then $\sum_j F_j$ is $\sum_j \nu_j$-self-concordant for the intersection $\cap_i \mathcal{K}_i$, given that it has nonempty interior. Thus, closed forms for the Hessian of the barrier, required for defining G_x^r in our Markov chain, can be calculated for many sets $\mathcal{K}$ of interest. We refer to (Nemirovskii, 2004; Nesterov and Nemirovskii, 1994) for further powerful methods for constructing the barriers.

5.1 Sampling from Posterior in Exponential Families

Suppose data $y_1, y_2, \ldots \in \mathcal{Y}$ are distributed i.i.d. according to a member of an exponential family with natural parameter x:

$$p(y|x) = \exp\{\langle x, T(y)\rangle - A(x)\} h(y)$$

where $A(x) = \int h(y) \exp\{\langle x, T(y)\rangle\}$ is a convex function and $T : \mathcal{Y} \mapsto \mathbb{R}^d$ is a sufficient statistic. Suppose $x \in \mathcal{K}$; that is, we have some knowledge about the support of the parameter. We have in mind the situation where data arrive one at a time and we are interested in sampling from the associated posterior distributions. The likelihood function after seeing $y_1, \ldots, y_t$ is

$$\ell(x) \propto \exp\left\{ \left\langle x, \sum_{i=1}^{t} T(y_i) \right\rangle - tA(x) \right\}$$

and, together with a conjugate prior $\pi_{\kappa_1,\kappa_2}(x) \propto \exp\{\langle x, \kappa_1\rangle - \kappa_2 A(x)\}$ for some $(\kappa_1, \kappa_2) \in \mathbb{R}^{d+1}$, we obtain the posterior distribution at time t

$$p_t(x|y) \propto \exp\left\{\left\langle x, \kappa_1 + \sum_{i=1}^{t} T(y_i)\right\rangle - (t+\kappa_2)A(x)\right\} .$$

We apply the sampling technique to this scenario by defining

$$s_0(x) = -\langle x, \kappa_1\rangle + \kappa_2 A(x), \qquad s_t(x) = -\left\langle x, \kappa_1 + \sum_{i=1}^{t} T(y_i)\right\rangle + (t+\kappa_2)A(x) .$$

It remains to calculate the number of steps required to track the distributions as additional data arrive one-by-one. Let L be the Lipschitz constant of $A(x)$ over $\mathcal{K}$ with respect to Euclidean norm, and let us assume L to be finite. Then Condition 4 is satisfied with $r = \min\left\{\frac{1}{(t+\kappa_2)L}, \frac{1}{d}\right\}$. Furthermore, we may set

$$\beta_t = \sup_{x\in\mathcal{K}} \exp\left\{2\left|\langle x, T(y_t)\rangle - A(x)\right|\right\},$$

a quantity that depends on the observed data. Importantly, we do not need to provide an a priori data-independent bound of this type, which might not be finite.

Suppose we would like to maintain a constant level $\epsilon > 0$ of accuracy at each step t. Corollary 8 guarantees this accuracy if each chain is run for

$$\tau_t = \mathcal{O}\left(\nu^2 d \max\{(t+\kappa_2)^2 L^2, d^2\} + \log(1/\epsilon)\right) .$$

One of the features of this bound is a relatively benign dependence on the dimension d, especially if the geometry of the set $\mathcal{K}$ allows the parameter $\nu = \mathcal{O}(1)$, as in the case of a sphere. On the negative side, the number of steps needed after seeing t data points is proportional to t^2. Such an adverse dependence, however, is to be expected as the posterior distribution becomes concentrated very quickly.

We now demonstrate that stronger results can be achieved under additional assumptions via Condition 3. Suppose that A is smooth: there exists $H \succeq 0$ such that

$$A(x) \le A(w) + \langle \nabla A(x), w - x\rangle + (w-x)^{\mathsf{T}} H (w-x)$$

for any $w, x \in \mathcal{K}$. This is a natural assumption, as the second derivative of the log normalization function A corresponds to the variance of the random variable with the given parameter; furthermore, A is differentiable. Let $\lambda_{\max}$ be the largest eigenvalue of H. Then the condition yields $r_t = \frac{C}{\sqrt{(t+\kappa_2)\lambda_{\max}}}$. To obtain ϵ-accuracy, it suffices to set

$$\tau_t = \mathcal{O}\left(\nu^2 d \max\{(t+\kappa_2)\lambda_{\max}, d^2\} + \log(1/\epsilon)\right),$$

which has only linear dependence on the size of the data seen so far.

We remark that each step of the random walk requires evaluation of the log-partition function $A(x)$. If this function is not available in closed form, we may approximate the value $A(x)$ for each query x. In order to do this, we may run an additional sampling procedure with $s'(x) = \langle x, T(y)\rangle$. Alternatively, we may appeal to known methods for this problem, such as Hit-and-Run (Vempala, 2005).

5.2 Examples of Sampling from Drifting Truncated Distributions

In the previous example, we employed the Markov chain to sample a parameter from a log-concave posterior. We now turn to the question of sampling from a log-concave distribution restricted to a convex set. This problem has a long history (see e.g. (Devroye, 1986; Gilks and Wild, 1992)), and it is recognized that sampling from truncated distributions is difficult even for nice forms such as the Normal distribution. One successful approach to this problem is the Gibbs sampling method (Robert, 1995; Damien and Walker, 2001), yet the rate of convergence is not generally available. The MCMC method of this paper yields a provably fast algorithm for such situations. Furthermore, we can track a drifting distribution over $\mathcal{K}$ with a small number of steps.

For illustration purposes, we study a simple example of a truncated Normal distribution; the same techniques, however, apply more generally. To simplify calculations, suppose the distributions μ_t are defined to be $\mathcal{N}(\mathbf{c}_t, \frac{1}{d}I)$ over a convex compact set $\mathcal{K} \subset \mathbb{R}^d$ and suppose the mean $\mathbf{c}_t$ is drifting within a Euclidean ball of radius R. With the definition in (1) we have $s_t(x) = \frac{1}{2}\|x - \mathbf{c}_t\|^2$. Define the drift $\delta_t = \|\mathbf{c}_t - \mathbf{c}_{t-1}\|$. In view of (4),

$$\log \beta_t \le \sup_{x\in\mathcal{K}} \|\mathbf{c}_t - \mathbf{c}_{t-1}\| \cdot \|2x - \mathbf{c}_t - \mathbf{c}_{t-1}\| \le C_{R,\mathcal{K}}\delta_t$$

where $C_{R,\mathcal{K}}$ depends on the radius R and the radius of a smallest Euclidean ball enclosing $\mathcal{K}$. In the same manner, the Lipschitz constant of $s_t(x)$ over $\mathcal{K}$ can be upper bounded by $L_{R,\mathcal{K}}$ that depends solely on the two radii. We may thus set the step size to be $r_t = \min\{\frac{1}{d}, \frac{1}{L_{R,\mathcal{K}}}\}$. If we aim for a fixed target accuracy ϵ for all t, by Corollary 8, it is enough to make

$$\tau_t = \left\lceil \frac{1}{\Delta_t} \log\left(\beta_t^{3/2} + \frac{\sqrt{\beta_t}(\beta_t - 1)}{\epsilon}\right)\right\rceil \tag{15}$$

steps. In the case that the drift δ_t is small enough, only one step is sufficient. To quantify the regime when this happens, observe that $\beta_t \le \exp\{C_{R,\mathcal{K}}\delta\} \le 1 + C\delta_t$, and it is then enough to require

$$\delta_t = \mathcal{O}\left(\Delta_t^2\right) = \mathcal{O}\left(\frac{\min\{1/d^2, 1/L_{R,\mathcal{K}}^2\}}{\nu^2 d}\right)$$

in view of (9). It is quite remarkable that the one-step random walk can track the changing distribution up to the accuracy $\mathcal{O}\left(\delta_t \frac{\nu^2 d}{r_t^2}\right)$, proportional to the size of the drift. Of course, better accuracy can be achieved by performing more steps, as per Corollary 8.

Another related application is to modeling with mixtures of log-concave distributions. Such models have been successful in clustering (McLachlan and Peel, 2000; Walther, 2009), with a mixture of normal distributions being a classical example (Fraley and Raftery, 2002). A mixture of parametric log-concave distributions can be written as $\sum_{i=1}^{k} \alpha_i \pi_i(\theta_i; x)$; here α_i are positive mixing weights summing to one, and π_i are a distributions on $\mathcal{K}$ parametrized by θ_i. A classical method for fitting models to data is the EM algorithm. Given that the parameters $\{\theta_i\}_{i=1}^k$ and the mixing weights $\{\alpha_i\}_{i=1}^k$ have been estimated from data, one may require random samples from this model for integration or other purposes. Given our procedure for sampling from a single log-concave distribution, one may simply pick the mixture according to the weights α_i and then sample from the component. The situation

becomes interesting in the case of online arrival of data, when we need to re-compute the EM solution in light of additional data. By the arguments of (Rakhlin and Caponnetto, 2006; Caponnetto and Rakhlin, 2006), the solution to clustering problems (the analysis was performed for square loss) is *stable* in the following sense: addition of $o(\sqrt{n})$ new data to a sample of size n is unlikely to drastically move the solution (the argument is based on uniqueness of the maximum of an empirical process). This in turn implies that the parameters $\{\theta_i\}$ are unlikely to change by a large amount, and we may thus use the method of sampling from a drifting distribution described earlier. We also remark that the method can be easily parallelized since the Markov chains for the k components do not interact.

5.3 Simulated Annealing for Convex Optimization

Let $f(x)$ be a proper convex 1-Lipschitz function. The aim of convex optimization is to find $\tilde{x}$ with the property $f(\tilde{x}) - \min_{x\in\mathcal{K}} f(x) \leq \epsilon$ for a given target accuracy $\epsilon > 0$. We consider the special case of linear function $f(x) = \langle \ell, x\rangle$, known as Linear Optimization. Complexity of an optimization procedure is often measured in terms of *oracle calls* – queries about the unknown function. A query about the function value is known as the zero-th order information, while a query about a subgradient at a point – as the first order information. In the case that the oracle answer is given without noise, it is known that the complexity scales as $\mathcal{O}\left(\mathrm{poly}(d, \log(1/\epsilon))\right)$. The state-of-the-art result here is the method of (Kalai and Vempala, 2006; Lovász and Vempala, 2006) which attains the $d^{4.5}$ dependence on the dimension.

We now apply our machinery to obtain a $\mathcal{O}\left(\nu^2 d^{3.5}\log(1/\epsilon)\right)$ method. In particular, this yields an improved $d^{3.5}$ dependence on the dimension for the case when $\mathcal{K}$ has a favorable geometry: there exists a self-concordant barrier with a parameter $\nu = \mathcal{O}(1)$.

We use the annealing scheme of (Kalai and Vempala, 2006). To this end, we set $s_t = \left(1 - d^{-1/2}\right)^{-t} f$ and observe that the assumption of Lemma 5 is satisfied with $\delta = d^{-1/2}$. Since functions are linear, we may set the step size $r_t = 1/d$ for all t. Hence, $\alpha_t \leq 5$ whenever $d > 8$ (and a different constant can be obtained for smaller d from the proof). By Proposition 9 with a constant accuracy $\epsilon_t = \epsilon \cdot (\sqrt{d}\log(d/\epsilon))^{-1}$, by making

$$\tau_t = \left\lceil C d^3 \nu^2 \log\left(\frac{5\sqrt{d}\log(d/\epsilon)}{\epsilon}\right)\right\rceil \tag{16}$$

steps for $t = 1, \ldots, k$, we guarantee

$$d_{TV}(\sigma_{k,\tau_k}, \mu_k) \leq k\epsilon(\sqrt{d}\log(d/\epsilon))^{-1} \,. \tag{17}$$

According to (Kalai and Vempala, 2006, Lemma 4.1), if X is chosen from a distribution with density proportional to $\exp\{-T^{-1}\langle \ell, x\rangle\}$, with $\|\ell\| = 1$ and some temperature $T > 0$, then

$$\mathbb{E}(\langle \ell, X\rangle) - \min_{x\in\mathcal{K}} \langle \ell, x\rangle \leq dT.$$

Hence, we take the desired temperature to be $T = \epsilon/d$, and the number of chains that permits the annealing schedule to reach this temperature can be calculated as $k = \sqrt{d}\log(\frac{d}{\epsilon})$. In view of (17), the final output of the procedure is an ϵ-accurate solution to the optimization problem. The complexity of the method is then $\mathcal{O}(d^{3.5}\nu^2\log^2(d/\epsilon))$.

This result can be extended to Lipschitz convex functions beyond linear optimization. However, the step size condition for convex Lipschitz functions requires the steps to be $\mathcal{O}(1/\epsilon)$ towards the end of the annealing schedule. This in turn implies only a suboptimal $\tilde{\mathcal{O}}(\nu^2 d/\epsilon^2)$ complexity. It is an open question of whether Dikin Walk can handle such annealing schedules in a more graceful manner.

5.4 Sequential Prediction

Another application of the proposed sampling technique is to the problem of *sequential prediction* with convex cost functions. Within this setting, the learner (or, the Statistician) is tasked with making a series of predictions while observing a sequence of outcomes on which we place no distributional assumptions. The goal of the learner is to incur cost comparable to that of a fixed strategy chosen in hindsight after observing the data. Initially studied by Hannan (Hannan, 1957), Blackwell (Blackwell, 1956), and Cover (Cover, 1965), the problem of achieving low *regret* for all sequences has received much attention in the last two decades, and we refer the reader to (Cesa-Bianchi and Lugosi, 2006) for a comprehensive treatment. As we show in this section, a strategy that exponentially down-weighs the decisions with large costs is a good regret-minimization strategy, and this exponential form is amenable to the sampling technique of this paper whenever the costs are convex.

More specifically, let $\mathcal{K} \subset \mathbb{R}^d$ be a convex compact set of decisions of the learner. Let $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$ be a sequence of unknown cost functions $\boldsymbol{\ell}_t : \mathcal{K} \to \mathbb{R}$. On round t, the learner chooses a distribution (or, a *mixed strategy*) μ_{t-1} supported on $\mathcal{K}$ and "plays" a decision $Y_t \sim \mu_{t-1}$.[2] Nature then reveals the next cost function $\boldsymbol{\ell}_t$. For example, in the well-studied problem of sequential probability assignment, the Statistician predicts the probability $x_t \in [0,1] = \mathcal{K}$ of the next outcome $\{0,1\}$ and incurs the cost $\boldsymbol{\ell}_t(x_t) = |x_t - y_t|$ with respect to the actual outcome y_t. A randomized strategy Y_t then incurs a cost $\boldsymbol{\ell}_t(Y_t)$. The goal of the learner is to minimize *expected regret*

$$\mathrm{Reg}_T(U) \triangleq \mathbb{E}\left[\sum_{t=1}^T \boldsymbol{\ell}_t(Y_t) - \sum_{t=1}^T \boldsymbol{\ell}_t(U)\right]$$

with respect to all randomized strategies defined by $p_U \in \mathcal{P}$, for some collection of distributions $\mathcal{P}$. A procedure that guarantees sublinear growth of regret with respect to any distribution $p_U \in \mathcal{P}$ and for any sequence of cost functions $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$ will be called *consistent* with respect to $\mathcal{P}$.

Let $L_t(x) = \sum_{s=1}^t \boldsymbol{\ell}_s(x)$ denote the cumulative cost functions, and let $\eta > 0$ be a parameter called *the learning rate*. Fix $R(x)$ to be some convex function that defines the prior, let

$$s_t(x) = \eta L_t(x) + R(x), \qquad s_0(x) = R(x) \tag{18}$$

and define the probability distributions μ_t as in (1). It turns out that this choice of μ_t is indeed a good regret-minimization strategy, as we show next. The method is similar to the Mixture Forecaster used in the prediction context (Yamanishi, 1998; Vovk, 2001; Azoury and Warmuth, 2001; Kakade and Ng, 2005), and for a discrete set of decisions it is known

2. The index $t-1$ on μ_{t-1} reflects the fact that Y_t is chosen without the knowledge of $\boldsymbol{\ell}_t$.

as the celebrated Exponential Weights Algorithm (Vovk, 1990; Littlestone and Warmuth, 1994).

Let $D(p||q)$ stand for the Kullback-Leibler (KL) divergence between distributions p and q.

Lemma 10 *For each $t \geq 1$, let Y_t be a random variable with distribution μ_{t-1} as defined in (1). The expected regret with respect to U with distribution p_U is*

$$Reg_T(U) = \eta^{-1}\left(D(p_U||\mu_0) - D(p_U||\mu_T)\right) + \eta^{-1}\sum_{t=1}^{T} D(\mu_{t-1}||\mu_t).$$

Specializing to the case $\boldsymbol{\ell}_t : \mathcal{K} \mapsto [0,1]$ for all t,

$$Reg_T(U) \leq \eta^{-1} D(p_U||\mu_0) + T\eta/8.$$

If the KL divergence between the comparator distribution p_U and the prior μ_0 is bounded for all $p_U \in \mathcal{P}$, the second statement of the lemma yields consistency and a $\mathcal{O}(\sqrt{T})$ rate of regret growth. To bound the divergence between a continuous initial μ_0 and a point distribution at some $x^* \in \mathcal{K}$, the analysis can be carried out in two stages: comparison to a "small-covariance" Gaussian centered at x^*, followed by an observation that the loss of the "small-covariance" Gaussian strategy is not very different from the loss of the deterministic strategy x^*. The analysis can be found in (Cesa-Bianchi and Lugosi, 2006, p. 326) and gives a near-optimal $\mathcal{O}(\sqrt{T\log T})$ regret bound.

The easy proof of Lemma 10 appeared in (Narayanan and Rakhlin, 2010) and we include it in Section 6 for completeness. Having exhibited a good prediction strategy, a natural question is whether there exists a computationally efficient algorithm for producing a random draw from a distribution close to the desired mixed strategy μ_{t-1}. To this end, we use the sampling method proposed in this paper.

As a concrete example, consider linear functions $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$ and let $R \equiv 0$. For simplicity assume boundedness $\boldsymbol{\ell}_t : \mathcal{K} \mapsto [0,1]$. In this case, we may choose $\eta = \mathcal{O}(1/\sqrt{T})$. Then

$$\beta_t \leq \exp\left\{2\eta\|\boldsymbol{\ell}_t\|_{\mathcal{K}}\right\} \leq 1 + C\eta$$

for large enough T. Further, we set $r_t = 1/d$ according to Condition 1, and the requirement (9) is seen to be satisfied for large enough T. With these choices of the parameters, the sequence of distributions $\mu_1, \ldots, \mu_t$ can be tracked with only one step of a random walk per iteration. The quality of this approximation is $\mathcal{O}\left(\eta d^3\nu^2\right)$ at each step. Therefore, regret of the proposed random walk method is within $\mathcal{O}\left(T\eta d^3\nu^2\right)$ from the ideal procedure of Lemma 10, as can be seen by writing

$$|\mathbb{E}\boldsymbol{\ell}_t(Y_t) - \mathbb{E}\boldsymbol{\ell}_t(X_{t-1,1})| \leq \int_{x\in\mathcal{K}} |\boldsymbol{\ell}_t(x)| \cdot |d\sigma_{t-1,1}(x) - d\mu_{t-1}(x)| \leq C\eta d^3\nu^2 \ .$$

By choosing $\eta = \frac{1}{d^{3/2}\nu\sqrt{T}}$,

$$\mathrm{Reg}_T(U) \leq Cd^{3/2}\nu D(p_U||\mu_0)\sqrt{T}. \tag{19}$$

A similar results holds for nonzero R, under the assumption that the L_2 distance between $d\mu_0(x) \propto \exp\{-R(x)\}dx$ and the uniform distribution on $\mathcal{K}$ is bounded.

We now discuss interesting parallels between the proposed randomized method and the known deterministic optimization-based regret minimization methods. First, the statement of Lemma 10 bears striking similarity to upper bounds on regret in terms of Bregman divergences for the Follow the Regularized Leader and Mirror Descent methods (Rakhlin, 2008; Beck and Teboulle, 2003), (Cesa-Bianchi and Lugosi, 2006, Therem 11.1). Yet, the randomized method operates in the (infinite-dimensional) space of distributions while the deterministic methods work directly with the set $\mathcal{K}$. Second, deterministic methods of online convex optimization face the difficulty of projections back to the set $\mathcal{K}$. This issue does not arise when dealing with distributions, but instead translates into the *difficulty of sampling*. We find these parallels between sampling and optimization intriguing. Third, a single step of the proposed random walk requires sampling from a Gaussian distribution with covariance given by the Hessian of the self-concordant barrier. This step can be implemented efficiently whenever the Hessian can be computed. The computation time exactly matches (Abernethy et al., 2008, Algorithm 2): it is the same as time spent inverting a Hessian matrix, which is $\mathcal{O}(d^3)$ or less. Finally, as already mentioned, the idea of following a time-varying distribution is inspired by the method of following the central path in the theory of interior point methods (Nesterov and Nemirovskii, 1994; Nemirovskii, 2004). Similarly to the fast convergence of the chain under the lower bound on conductance, one has fast quadratic local convergence of interior point methods. One may therefore make parallels between conductance and local curvature. A further investigation of these connections is needed, especially in view of the recent developments on positive Ricci curvature of Markov chains (Ollivier, 2009).

6. Proofs

Lemma 11 *For any t and $i \geq 0$, it holds that*

$$\left\|\frac{d\sigma_{t,i}}{d\mu_t} - 1\right\|_t \leq \beta_t^{3/2}\left\|\frac{d\sigma_{t,i}}{d\mu_{t-1}} - 1\right\|_{t-1} + \sqrt{\beta_t}(\beta_t - 1)$$

and, alternatively,

$$\left\|\frac{d\sigma_{t,i}}{d\mu_t} - 1\right\|_t \leq \beta_t^{1/2}\left\|\frac{d\sigma_{t,i}}{d\mu_{t-1}} - 1\right\|_{t-1} + \sqrt{\beta_t - 1}$$

Proof Let us use the shorthand $d\sigma = d\sigma_{t+1,i}$ and $\beta = \beta_{t+1}$. Using (3), we may write

$$\begin{aligned}\left\|\frac{d\sigma}{d\mu_{t+1}} - 1\right\|_{t+1} &\leq \sqrt{\beta}\left\|\frac{d\sigma}{d\mu_{t+1}} - 1\right\|_t \\ &\leq \sqrt{\beta}\left(\left\|\frac{d\sigma}{d\mu_{t+1}} - 1\right\|_t - \left\|\frac{d\sigma}{d\mu_t} - 1\right\|_t + \left\|\frac{d\sigma}{d\mu_t} - 1\right\|_t\right) \ .\end{aligned}$$

By the triangle inequality,

$$\left\|\frac{d\sigma}{d\mu_{t+1}} - 1\right\|_t - \left\|\frac{d\sigma}{d\mu_t} - 1\right\|_t \leq \left\|\frac{d\sigma}{d\mu_{t+1}} - \frac{d\sigma}{d\mu_t}\right\|_t \ .$$

For any function $f : \mathcal{K} \to \mathbb{R}$, let $f^+(x) = \max(0, f(x))$ and $f^-(x) = \min(0, f(x))$. In view of (3),

$$\begin{aligned}\left\|\frac{d\sigma}{d\mu_{t+1}} - \frac{d\sigma}{d\mu_t}\right\|_t^2 &= \left\|\left(\frac{d\sigma}{d\mu_{t+1}} - \frac{d\sigma}{d\mu_t}\right)^+\right\|_t^2 + \left\|\left(\frac{d\sigma}{d\mu_{t+1}} - \frac{d\sigma}{d\mu_t}\right)^-\right\|_t^2 \\ &\le \left\|\frac{d\sigma}{d\mu_t}(\beta-1)\mathbf{1}\left[1 < \frac{d\mu_t}{d\mu_{t+1}}\right]\right\|_t^2 + \left\|\frac{d\sigma}{d\mu_t}\left(1-\frac{1}{\beta}\right)\mathbf{1}\left[1 \ge \frac{d\mu_t}{d\mu_{t+1}}\right]\right\|_t^2 \\ &\le (\beta-1)^2\left\|\frac{d\sigma}{d\mu_t}\right\|_t^2 .\end{aligned}$$

Therefore,

$$\left\|\frac{d\sigma}{d\mu_{t+1}} - 1\right\|_t - \left\|\frac{d\sigma}{d\mu_t} - 1\right\|_t \le (\beta-1)\left\|\frac{d\sigma}{d\mu_t}\right\|_t \le (\beta-1)\left(1 + \left\|\frac{d\sigma}{d\mu_t} - 1\right\|_t\right) .$$

The first statement follows by rearranging the terms.

Alternatively, we can obtain an inequality that is slightly weaker for $\beta - 1 \approx 0$ and stronger for large β by simply writing

$$\begin{aligned}\left\|\frac{d\sigma}{d\mu_{t+1}} - 1\right\|_{t+1}^2 &= \int_{\mathcal{K}}\left(\frac{d\sigma}{d\mu_{t+1}} - 1\right)^2 d\mu_{t+1} \\ &= \int_{\mathcal{K}}\frac{d\sigma^2}{d\mu_{t+1}} - 1 = \int_{\mathcal{K}}\frac{d\sigma^2}{d\mu_t^2}\frac{d\mu_t}{d\mu_{t+1}}d\mu_t - 1 .\end{aligned}$$

Using β as an upper bound on the one-sided change $\|d\mu_t/d\mu_{t+1}\|_{\mathcal{K}}$ leads to

$$\beta\int_{\mathcal{K}}\frac{d\sigma^2}{d\mu_t^2}d\mu_t - 1 = \beta\left\|\frac{d\sigma}{d\mu_t} - 1\right\|_t^2 + \beta - 1$$

and subadditivity of the square root function concludes the proof. ■

Proof [**Proof of Theorem 1**]

Given interior points x, y in $int(\mathcal{K})$, suppose p, q are the ends of the chord in $\mathcal{K}$ containing x, y and p, x, y, q lie in that order. Denote the *cross ratio* by

$$\sigma(x, y) = \frac{|x-y||p-q|}{|p-x||q-y|},$$

and for two sets S_1 and S_2 let

$$\sigma(S_1, S_2) \triangleq \inf_{x\in S_1, y\in S_2}\sigma(x, y).$$

A result due to Lovász and Vempala (2007) states the following. If S_1 and S_2 are measurable subsets of $\mathcal{K}$ and μ a probability measure supported on $\mathcal{K}$ that possesses a density whose logarithm is concave, then

$$\mu((\mathcal{K}\setminus S_1)\setminus S_2) \ge \sigma(S_1, S_2)\mu(S_1)\mu(S_2).$$

This is a non-trivial isoperimetric inequality which says that for any partition of the convex set $\mathcal{K}$ into S_1, S_2 and S_3, the "volume" of S_3 is large relative to that of S_1 and S_2 whenever S_1 and S_2 are separated. Given this isoperimetric result, to prove the theorem it only remains to show that the σ-distance can be lower bounded (up to a multiplicative constant) by the Riemannian metric ρ. The proof of this fact goes through the Hilbert (projective) metric, which is defined by

$$d_H(x, y) \triangleq \ln\left(1 + \sigma(x, y)\right).$$

Further, for $x \in \mathcal{K}$ and a vector v, let

$$|v|_x \triangleq \sup_{x\pm\alpha v\in\mathcal{K}}\alpha.$$

The following two relations between the introduced notions hold. The first one (see Nesterov and Nemirovskii (Nesterov and Nemirovskii, 1994, Theorem 2.3.2 (iii))) is

$$|h|_x \le \|h\|_x \le 2(1+3\nu)|h|_x \tag{20}$$

for all $h \in \mathbb{R}^d$ and $x \in int(\mathcal{K})$, where ν is the self-concordance parameter of F. The second relation (see Nesterov and Todd (Nesterov and Todd, 2008, Lemma 3.1)) states that

$$\|x-y\|_x - \|x-y\|_x^2 \le \rho(x, y) \le -\ln(1-\|x-y\|_x). \tag{21}$$

whenever $\|x - y\|_x < 1$.

For any z on the segment $\overline{xy}$ an easy computation shows that $d_H(x, z) + d_H(z, y) = d_H(x, y)$. Therefore it suffices to prove the result infinitesimally. From (21), $\lim_{y\to x}\frac{\rho(x,y)}{\|x-y\|_x} = 1$, and a direct computation shows that

$$\lim_{y\to x}\frac{d_H(x, y)}{|x-y|_x} = \lim_{y\to x}\frac{\sigma(x, y)}{|x-y|_x} \ge 1.$$

Hence, in view of (20), the Hilbert metric and the Riemannian metric satisfy

$$\rho(x, y) \le 2(1+3\nu)d_H(x, y).$$

Using $\ln(1 + x) \le x$ concludes the proof. ■

Proof [**Proof of Lemma 4**] The argument roughly follows the standard path, which is explained, for instance, in (Vempala, 2005). Let S_1 be a measurable subset of $\mathcal{K}$ such that $\mu(S_1) \le \frac{1}{2}$ and $S_2 = \mathcal{K}\setminus S_1$ be its complement. Fix a $C > 1$ and let

$$S_1' = S_1 \cap \{x \big| \mathsf{P}_x(S_2) \le 1/C\} \quad \text{and} \quad S_2' = S_2 \cap \{y \big| \mathsf{P}_y(S_1) \le 1/C\}.$$

That is, points in the set S_1' are unlikely to transition to the set S_2, and S_2' is analogously unlikely to reach S_1 in one step. By the reversibility of the chain, which is easily checked,

$$\int_{S_1}\mathsf{P}_x(S_2)d\mu(x) = \int_{S_2}\mathsf{P}_y(S_1)d\mu(y).$$

For any $x \in S_1'$ and $y \in S_2'$,

$$d_{TV}(\mathsf{P}_x, \mathsf{P}_y) = 1 - \int_{\mathcal{K}} \min\left(\frac{d\mathsf{P}_x}{d\mu}(w), \frac{d\mathsf{P}_y}{d\mu}(w)\right) d\mu(w) \geq 1 - \frac{1}{C}.$$

That is, the transition probabilities for a pair in S_1' and S_2' must be dissimilar. But Lemma 3 implies that if $\rho(x,y) \leq \frac{r}{C\sqrt{d}}$, then $d_{TV}(\mathsf{P}_x, \mathsf{P}_y) \leq 1 - \frac{1}{C}$. Therefore

$$\rho(S_1', S_2') \geq \frac{r}{C\sqrt{d}}.$$

We conclude that the sets S_1' and S_2' must be well-separated. Therefore, the isoperimetric result of Theorem 1 implies that

$$\mu((\mathcal{K} \setminus S_1') \setminus S_2') \geq \frac{\rho(S_1', S_2')}{2(1+3\nu)} \min(\mu(S_1'), \mu(S_2')) \geq \frac{r}{C\nu\sqrt{d}} \min(\mu(S_1'), \mu(S_2')).$$

First suppose $\mu(S_1') \geq (1 - \frac{1}{C})\mu(S_1)$ and $\mu(S_2') \geq (1 - \frac{1}{C})\mu(S_2)$. Then,

$$\begin{aligned}
\int_{S_1} \mathsf{P}_x(S_2) d\mu(x) &= \frac{1}{2}\int_{S_1} \mathsf{P}_x(S_2) d\mu(x) + \frac{1}{2}\int_{S_2} \mathsf{P}_x(S_1) d\mu(x) \\
&\geq \frac{1}{2C}\mu((\mathcal{K} \setminus S_1') \setminus S_2') \\
&\geq \frac{r}{2C^2\nu\sqrt{d}} \min(\mu(S_1'), \mu(S_2')) \\
&\geq \frac{1 - 1/C}{2C^2} \frac{r}{\nu\sqrt{d}} \min(\mu(S_1), \mu(S_2)),
\end{aligned}$$

proving the result. Otherwise, without loss of generality, suppose $\mu(S_1') \leq (1 - \frac{1}{C})\mu(S_1)$. Then

$$\begin{aligned}
\int_{S_1} \mathsf{P}_x(S_2) d\mu(x) &= \frac{1}{2}\int_{S_1} \mathsf{P}_x(S_2) d\mu(x) + \frac{1}{2}\int_{S_2} \mathsf{P}_x(S_1) d\mu(x) \\
&\geq \frac{1}{2}\int_{S_1 \setminus S_1'} \mathsf{P}_x(S_2) d\mu(x) \geq \frac{\mu(S_1)}{2C^2},
\end{aligned}$$

concluding the proof. ■

Proof [**Proof of Lemma 5**] The proof closely follows that in (Kalai and Vempala, 2006). By definition,

$$\|d\mu_t / d\mu_{t+1}\|_{t+1}^2 = \int_{\mathcal{K}} \left(\frac{d\mu_t}{d\mu_{t+1}}\right)^2 d\mu_{t+1} = \int_{\mathcal{K}} \frac{d\mu_t^2}{d\mu_{t+1}} = \int_{\mathcal{K}} \frac{\exp\{-2s_t\}}{Z_t^2} \cdot \frac{Z_{t+1}}{\exp\{-s_{t+1}\}}.$$

Writing out the normalization terms,

$$\|d\mu_t / d\mu_{t+1}\|_{t+1}^2 = \frac{\int_{\mathcal{K}} \exp\{-s_{t+1}\} \int_{\mathcal{K}} \exp\{s_{t+1} - 2s_t\}}{\left(\int_{\mathcal{K}} \exp\{-s_t\}\right)^2} = \frac{Y(1)Y(-1+2(1-\delta))}{Y(1-\delta)Y(1-\delta)}$$

where $Y(a) = \int_{\mathcal{K}} \exp\{-as_{t+1}\}$. As shown in (Kalai and Vempala, 2006, Lemma 3.1), the function $a^d Y(a)$ is log-concave in a, and thus

$$\frac{Y(a)Y(b)}{Y\left(\frac{a+b}{2}\right)^2} \leq \left(\frac{\left(\frac{a+b}{2}\right)^2}{ab}\right)^d.$$

Applying this inequality with $a = 1$ and $b = -1 + 2(1-\delta)$,

$$\|d\mu_t / d\mu_{t+1}\|_{t+1}^2 \leq \left(1 + \frac{\delta^2}{1-2\delta}\right)^d.$$

In particular, if $\delta \leq d^{-1/2} \leq 1/3$ (that is, $d > 8$), we obtain an upper bound of $\exp\left\{\frac{d}{d-2\sqrt{d}}\right\} \leq 21$. ■

Proof [**Proof of Lemma 10**] Observe that $D(\mu_{t-1}||\mu_t)$ can be written as

$$\int_{\mathcal{K}} d\mu_{t-1} \log \frac{q_{t-1}Z_t}{Z_{t-1}q_t} = \log \frac{Z_t}{Z_{t-1}} + \int_{\mathcal{K}} \eta \boldsymbol{\ell}_t(x) d\mu_{t-1}(x) = \log \frac{Z_t}{Z_{t-1}} + \eta \mathbb{E}\boldsymbol{\ell}_t(Y_t). \tag{22}$$

Rearranging, canceling the telescoping terms, and using the fact that $Z_0 = 1$

$$\eta \mathbb{E} \sum_{t=1}^T \boldsymbol{\ell}_t(Y_t) = \sum_{t=1}^T D(\mu_{t-1}||\mu_t) - \log Z_T.$$

Let U be a random variable with a probability distribution p_U. Then

$$-\sum_{t=1}^T \mathbb{E}\boldsymbol{\ell}_t(U) = \eta^{-1} \int_{\mathcal{K}} -\eta L_T(u) dp_U(u) = \eta^{-1} \int_{\mathcal{K}} dp_U(u) \log \frac{q_T(u)}{q_0(u)}$$

Combining,

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^T \boldsymbol{\ell}_t(Y_t) - \sum_{t=1}^T \boldsymbol{\ell}_t(U)\right] &= \eta^{-1} \int_{\mathcal{K}} dp_U(u) \log \frac{q_T(u)/Z_T}{q_0(u)} + \eta^{-1} \sum_{t=1}^T D(\mu_{t-1}||\mu_t) \\
&= \eta^{-1} \left(D(p_U||\mu_0) - D(p_U||\mu_T)\right) + \eta^{-1} \sum_{t=1}^T D(\mu_{t-1}||\mu_t).
\end{aligned}$$

Now, from Eq. (22), the KL divergence can be also written as

$$D(\mu_{t-1}||\mu_t) = \log \frac{\int_{\mathcal{K}} e^{-\eta \boldsymbol{\ell}_t(x)} q_{t-1}(x) dx}{\int_{\mathcal{K}} q_{t-1}(x) dx} + \eta \mathbb{E}\boldsymbol{\ell}_t(Y_t) = \log \mathbb{E} e^{-\eta(\boldsymbol{\ell}_t(Y_t) - \mathbb{E}\boldsymbol{\ell}_t(Y_t))}$$

By representing the divergence in this form, one can obtain upper bounds via known methods, such as *log-Sobolev inequalities* (e.g. (Boucheron et al., 2003)). In the simplest case of bounded loss, it is easy to show that $D(\mu_{t-1}||\mu_t) \leq O(\eta^2)$, and the particular constant $1/8$ can be obtained by, for instance, applying Lemma A.1 in (Cesa-Bianchi and Lugosi, 2006). This proves the second part of the lemma. ■

7. Smooth Variation of the Transition Kernel

In this section, we study the transition $x \to y$. For this purpose, it is enough to assume that x is the origin and that the Dikin ellipsoid at x is a unit Euclidean ball. This can be achieved by an affine transformation, leading to no loss of generality since the resulting statement about measures on $\mathcal{K}$ is invariant with respect to affine transformations. Hence, in what follows, for the particular x we have $<\cdot,\cdot>_x = <\cdot,\cdot>$ and $\|\cdot\|_x = \|\cdot\|$. Since x is the origin, we have $\mathbb{E}\|z\|_x^2 = r^2$ for z sampled from G_x^r. Further, without loss of generality, we may also assume $s(x) = 0$.

Proof [**Proof of Lemma 3**]

In view of the first inequality in Eq. (21),

$$\|x-y\|_x - \|x-y\|_x^2 \le \rho(x,y) \le \frac{r}{C\sqrt{d}}.$$

Without loss of generality, assume $\frac{r}{C\sqrt{d}} \le \frac{1}{8}$. First, we claim that $\|x-y\|_x$ must be small. For the sake of contradiction, suppose $\|x-y\|_x > 1/2$ and consider a point y' with $\|x-y'\|_x = 1/2$ and lying on the geodesic path between x and y with respect to the Riemannian metric. Clearly, $\rho(x,y') \le \frac{r}{C\sqrt{d}} \le \frac{1}{8}$, yet by Eq. (21) we have $\frac{1}{4} \le \rho(x,y')$, contradicting our assumption. Hence, $\|x-y\|_x \le 1/2$, and, therefore, $\|x-y\|_x \le \frac{2r}{C\sqrt{d}}$.

It remains to show that if $x, y \in \mathcal{K}$ and

$$\|x-y\|_x \le \frac{2r}{C\sqrt{d}},$$

then

$$d_{TV}(\mathsf{P}_x, \mathsf{P}_y) = 1 - \frac{1}{C}.$$

By definition, we have that

$$1 - d_{TV}(\mathsf{P}_x, \mathsf{P}_y) = \mathbb{E}_z\left[\min\left\{1, \frac{G_y^r(z)}{G_x^r(z)}, \frac{G_z^r(x)\exp(s(x))}{G_x^r(z)\exp(s(z))}, \frac{G_z^r(y)\exp(s(y))}{G_x^r(z)\exp(s(z))}\right\}\right],$$

where the expectation is taken over a random point z having density G_x^r. Thus, it suffices to prove that for some $C > 1$

$$\mathbb{P}\left[\min\left\{\frac{G_y^r(z)}{G_x^r(z)}, \frac{G_z^r(x)\exp(s(x))}{G_x^r(z)\exp(s(z))}, \frac{G_z^r(y)\exp(s(y))}{G_x^r(z)\exp(s(z))}\right\} > \frac{1}{C}\right] \ge \frac{1}{C}.$$

By our assumption, x is the origin and $D^2F(x) = I$, the latter implying that $V(x) = 0$. Thus,

$$\frac{G_y^r(z)}{G_x^r(z)} = \exp\left\{-\frac{d\|y-z\|_y^2}{r^2} + V(y) + \frac{d\|z\|^2}{r^2}\right\},$$

$$\frac{G_z^r(x)\exp(s(x))}{G_x^r(z)\exp(s(z))} = \exp\left\{-\frac{d\|z\|_z^2}{r^2} + V(z) + \frac{d\|z\|^2}{r^2} + (s(x)-s(z))\right\},$$

and

$$\frac{G_z^r(y)\exp(s(y))}{G_x^r(z)\exp(s(z))} = \exp\left\{-\frac{d\|y-z\|_z^2}{r^2} + V(z) + \frac{d\|z\|^2}{r^2} + (s(y)-s(z))\right\}.$$

Thus, it remains to prove that there exists a constant C such that

$$\mathbb{P}\Big[\max\Big\{d\|y-z\|_y^2 - r^2V(y), \quad d\|z\|_z^2 + r^2(s(z)-s(x)) - r^2V(z), \\ d\|z-y\|_z^2 + r^2(s(z)-s(y)) - r^2V(z)\Big\} < d\|z\|^2 + r^2C\Big] \ge \frac{1}{C}.$$

This fact is shown in technical Lemmas 13 and 14 below. ■

In proving the technical lemmas, we will use the fact that $\|x-y\|_x \le \frac{2r}{C\sqrt{d}}$ as shown above, and that $\|x-z\|_x$ (for z sampled from G_x^r) is likely to be bounded above by a multiple of r by straightforward concentration arguments.

Lemma 12 *There exists a constant $C > 0$ such that*

$$\mathbb{P}\left[\max\left(-V(y), -V(z)\right) < C\right] > 0.9$$

Proof Fix a constant c. First, notice that over a Euclidean ball of radius c/d around the origin, the Hessians $D^2F(u)$ are lower-bounded by a factor of $(1-c/d)^2$ from the Hessian at the origin (the identity) by (24). Hence, the determinant function can decrease from 1 by at most a constant factor. Thus $-V(u) < C'$ for some constant C' for any u with $\|x-u\|_x \le c/d$. Now recall that y is deterministically within the $1/d$ ball, while z is in the ball of radius c/d with high probability. ■

Lemma 13 *Under step size Condition 4, for any*

$$\mathbb{P}\left[\max\left\{s(z)-s(x), s(z)-s(y)\right\} < C\right] > 0.32.$$

Proof Since with large enough probability $\|x-y\|_x < C'r$ and $\|x-z\|_x < C'r$, we also have $\|z-y\|_x < 2C'r$. Then, by (24), the norms at z and x are within a multiplicative constant, and thus the pairs (z,x) and (z,y) are subject to the step size choice specified in the condition. That is, there exists a g such that

$$s(z) - s(x) = s(z) - s(x) - \langle g, z-x\rangle + \langle g, z-x\rangle \le C + \langle g, z-x\rangle$$

and similarly

$$s(z) - s(y) = s(z) - s(y) - \langle g, z-y\rangle + \langle g, z-y\rangle \le C + \langle g, z-y\rangle$$

Then, assuming (without loss of generality) $x = 0$,

$$\mathbb{P}\left[\max\left\{\langle g, z-x\rangle, \langle g, z-y\rangle\right\} < 0\right] = \mathbb{P}\left[\langle g, z\rangle \le \min\left\{0, \langle g, y\rangle\right\}\right].$$

Observe that $\langle g, z\rangle$ is a Gaussian random variable whose standard deviation is larger than $\|g\|\,\|y\|$. Therefore,

$$\mathbb{P}\left[\langle g, z\rangle \le \min\left\{0, \langle g, y\rangle\right\}\right] \ge \operatorname{erfc}\left(1/\sqrt{2}\right) > 0.32,$$

where $\operatorname{erfc}(x) \triangleq \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}dt$ is the usual complementary error function. ■

The following probabilistic upper bound completes the proof.

Lemma 14 *There exists a constant $C > 0$ such that*

$$\mathbb{P}\left[\max\left\{\|y-z\|_y^2, \|z\|_z^2, \|z-y\|_z^2\right\} - \|z\|^2 < \frac{Cr^2}{d}\right] > 0.9$$

Proof [**Proof of Lemma 14**] Since $\|y\| < \frac{Cr}{\sqrt{d}}$, $\|y\|_y$ and $\|y\|_z$ are less than $\frac{Cr}{\sqrt{d}}$. So it suffices to show that

$$\mathbb{P}\left[\max\left\{\|z\|_y^2 - \|z\|^2, \|z\|_z^2 - \|z\|^2, \langle y,z\rangle_y, \langle y,z\rangle_z\right\} < \frac{Cr^2}{d}\right] > 0.9$$

We proceed to do so by proving probabilistic upper bounds on each of the terms

$$\text{(a) } \|z\|_y^2 - \|z\|^2\,, \qquad \text{(b) } \|z\|_z^2 - \|z\|^2\,, \qquad \text{(c) } \langle y,z\rangle_y\,, \quad \text{and} \quad \text{(d) } \langle y,z\rangle_z$$

separately, and finally applying the union bound. We first prove an upper bound on (a) and (b). Note that $r \le \frac{1}{d}$ and thus $r^3 \le \frac{r^2}{d}$. It suffices to observe that by (24)

$$\|z\|_z^2 - \|z\|^2 \le \left(\left(\frac{1}{1-\|z\|}\right)^2 - 1\right)\|z\|^2 \le 8\|z\|^3,$$

whenever $\|z\| < 1/2$. Similarly, for $\|y\| < 1/2$,

$$\|z\|_y^2 - \|z\|^2 \le \left(\left(\frac{1}{1-\|y\|}\right)^2 - 1\right)\|z\|^2 \le 8\|z\|^3.$$

There exists a constant C such that the quantity $\|z\|^3$ is bounded by Cr^3 with probability at least 0.99.

We now turn to bounding (c) and (d). Let $[0,u]$ denote the line segment between the origin and u. By the mean-value theorem,

$$\langle y,z\rangle_y = \langle y,z\rangle + (\langle y,z\rangle_y - \langle y,z\rangle) \le \langle y,z\rangle + \sup_{y'\in[0,y]} D^3F(y')[y,y,z]$$

$$\langle y,z\rangle_z = \langle y,z\rangle + (\langle y,z\rangle_z - \langle y,z\rangle) \le \langle y,z\rangle + \sup_{z'\in[0,z]} D^3F(z')[y,z,z]$$

Observe that

$$\langle y,z\rangle \le \frac{C\|y\|\|z\|}{\sqrt{d}}$$

with probability at least 0.99 by a measure-concentration argument. Indeed, most of the vectors z are almost perpendicular to the given vector y. Now, using (23),

$$\sup_{y'\in[0,y]} D^3F(y')[y,y,z] \le \sup_{y'\in[0,y]} 2\|y\|_{y'}^2\|z\|_{y'} \le \frac{Cr^2}{d}$$

and

$$\sup_{z'\in[0,z]} D^3F(z')[y,z,z] \le \sup_{z'\in[0,z]} 2\|y\|_{z'}\|z\|_{z'}^2 \le \frac{Cr^3}{\sqrt{d}} \le \frac{Cr^2}{d}$$

with probability at least 0.99. Therefore, there exists a constant $C > 0$ such that

$$\mathbb{P}\left[\langle y,z\rangle_y < \frac{Cr^2}{d}\right] > 0.98$$

and the same statement holds for $\langle y,z\rangle_z$. We also have that

$$\mathbb{P}\left[\frac{\|y\|\|z\|}{\sqrt{d}} + \sup_{z'\in[0,z]} 2\|y\|_{z'}\|z\|_{z'}^2 \le \frac{Cr^2}{d}\right] > 0.99$$

Therefore,

$$\mathbb{P}\left[\langle y,z\rangle_z < \frac{Cr^2}{d}\right] > 0.98.$$

■

8. Self-concordant barriers

Let $\mathcal{K}$ be a convex subset of $\mathbb{R}^d$ that is not contained in any $(d-1)$-dimensional affine subspace and $int(\mathcal{K})$ denote its interior. Following Nesterov and Nemirovskii, we call a real-valued function $F : int(\mathcal{K}) \to \mathbb{R}$, a regular self-concordant barrier if it satisfies the conditions stated below. For convenience, if $x \notin int(\mathcal{K})$, we define $F(x) = \infty$.

1. (Convex, Smooth) F is a convex thrice continuously differentiable function on $int(\mathcal{K})$.
2. (Barrier) For every sequence of points $\{x_i\} \in int(\mathcal{K})$ converging to a point $x \notin int(\mathcal{K})$, $\lim_{i\to\infty} f(x_i) = \infty$.
3. (Differential Inequalities) For all $h \in \mathbb{R}^d$ and all $x \in int(\mathcal{K})$, the following inequalities hold.
   (a) $D^2F(x)[h,h]$ is 2-Lipschitz continuous with respect to the local norm, which is equivalent to
   $$D^3F(x)[h,h,h] \le 2(D^2F(x)[h,h])^{\frac{3}{2}}.$$
   (b) $F(x)$ is ν-Lipschitz continuous with respect to the local norm defined by F,
   $$|DF(x)[h]|^2 \le \nu D^2F(x)[h,h].$$
   We call the smallest positive integer ν for which this holds, *the self-concordance parameter* of the barrier.

The following results can be found, for instance, in (Nesterov and Nemirovskii, 1994; Nemirovskii, 2004; Nemirovski and Todd, 2008). First,

$$|D^3F(x)[h_1,\ldots,h_k]| \le 2\|h_1\|_x\|h_2\|_x\|h_3\|_x\,. \tag{23}$$

Second, if $\delta = \|h\|_x < 1$, then

$$(1-\delta)^2D^2F(x) \preceq D^2F(x+h) \preceq (1-\delta)^{-2}D^2F(x)\,. \tag{24}$$

References

J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Proceedings of The Twenty First Annual Conference on Learning Theory*, 2008.

J. D. Abernethy and E. Hazan. Faster convex optimization: Simulated annealing with an efficient universal barrier. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2520–2528, 2016.

K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, June 2001.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, 2003.

D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pac. J. Math.*, 6:1–8, 1956.

S. Boucheron, G. Lugosi, and P. Massart. Concentration inequalities using the entropy method. *Annals of Probability*, 31:1583–1614, 2003.

A. Caponnetto and A. Rakhlin. Stability properties of empirical risk minimization over Donsker classes. *Journal of Machine Learning Research*, 6:2565–2583, 2006.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89(3):539–552, 2002.

T. Cover. Behaviour of sequential predictors of binary sequences. In *Proc. 4th Prague Conf. Inform. Theory, Statistical Decision Functions, Random Processes*, 1965.

P. Damien and S. Walker. Sampling truncated normal, beta, and gamma densities. *Journal of Computational and Graphical Statistics*, 10(2), 2001.

L. Devroye. *Non-uniform random variate generation (1986)*. Springer Verlag, 1986.

P. Diaconis. The markov chain monte carlo revolution. *Bulletin of the American Mathematical Society*, 46(2):179–205, 2009.

P. Diaconis. Some things we've learned (about Markov chain Monte Carlo). *Bernoulli*, 19 (4):1294–1305, 2013.

A. Doucet, N. De Freitas, N. Gordon, et al. *Sequential Monte Carlo methods in practice*. Springer New York, 2001.

M. Dyer, A. Frieze, and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the ACM (JACM)*, 38(1):1–17, 1991.

C. Fraley and A. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

A. Frieze, R. Kannan, and N. Polson. Sampling from log-concave distributions. *The Annals of Applied Probability*, pages 812–837, 1994.

W. Gilks and P. Wild. Adaptive rejection sampling for gibbs sampling. *Applied Statistics*, pages 337–348, 1992.

J. Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

S. Kakade and A. Ng. Online bounds for Bayesian algorithms. In *Proceedings of Neural Information Processing Systems (NIPS 17)*, 2005.

A.T. Kalai and S. Vempala. Simulated annealing for convex optimization. *Mathematics of Operations Research*, 31(2):253–266, 2006.

R. Kannan and H. Narayanan. Random walks on polytopes and an affine interior point method for linear programming. *Mathematics of Operations Research*, 37(1):1–20, 2012.

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

L. Lovász. Hit-and-run mixes fast. *Mathematical Programming*, 86(3):443–461, 1999. ISSN 0025-5610.

L. Lovász and M. Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Structures and Algorithms*, 4(4):359–412, 1993.

L. Lovász and S. Vempala. Simulated annealing in convex bodies and an $o^*(n^4)$ volume algorithm. *J. Comput. Syst. Sci.*, 72(2):392–417, 2006.

L. Lovász and S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, 2007.

GJ McLachlan and D Peel. Finite mixture models. 2000.

S. Meyn and R. L. Tweedie. *Markov chains and stochastic stability*. Cambridge University Press, 2009.

H. Narayanan. Randomized interior point methods for sampling and optimization. *The Annals of Applied Probability*, 26(1):597–641, 2016.

H. Narayanan and A. Rakhlin. Random walk approach to regret minimization. In *Advances in Neural Information Processing Systems*, 2010.

A. S. Nemirovski and M. J. Todd. Interior-point methods for optimization. *Acta Numerica*, pages 191—234, 2008.

A. S. Nemirovskii. Interior point polynomial time methods in convex programming, 2004.

Y. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*. SIAM, Philadelphia, 1994.

Y.E. Nesterov and M. J. Todd. On the Riemannian geometry defined by self-concordant barriers and interior-point methods. *Foundations of Computational Mathematics*, 2(4): 333–361, 2008.

Y. Ollivier. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis*, 256(3):810–864, 2009.

A. Rakhlin. Lecture notes on online learning, 2008. http://stat.wharton.upenn.edu/~rakhlin/papers/online_learning.pdf.

A. Rakhlin and A. Caponnetto. Stability of K-means clustering. In *Advances in Neural Information Processing Systems 19*, pages 1121–1128. MIT Press, 2006.

C. P. Robert. Simulation of truncated normal variables. *Statistics and computing*, 5(2): 121–125, 1995.

C. P. Robert and G. Casella. *Monte Carlo statistical methods*, volume 319. 2004.

S. Vempala. Geometric random walks: A survey. *In Combinatorial and computational geometry. Math. Sci. Res. Inst. Publ*, 52:577–616, 2005.

V. Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 372–383. Morgan Kaufmann, 1990.

V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69:213–248, 2001.

G. Walther. Inference and modeling with log-concave distributions. *Statistical Science*, pages 319–327, 2009.

K. Yamanishi. Minimax relative loss analysis for sequential prediction algorithms using parametric hypotheses. In *COLT' 98*, pages 32–43, New York, NY, USA, 1998. ACM.

Journal of Machine Learning Research 18 (2017) 1-24 Submitted 1/16; Revised 8/17; Published 11/17

Document Neural Autoregressive Distribution Estimation

Stanislas Lauly STANISLAS.LAULY@USHERBROOKE.CA
Département d'informatique
Université de Sherbrooke
Sherbrooke, Québec, Canada

Yin Zheng YINZHENG@TENCENT.COM
Tencent AI Lab
Shenzhen, Guangdong, China

Alexandre Allauzen ALLAUZEN@LIMSI.FR
LIMSI-CNRS
Université Paris Sud
Orsay, France

Hugo Larochelle HUGO.LAROCHELLE@USHERBROOKE.CA
Département d'informatique
Université de Sherbrooke
Sherbrooke, Québec, Canada

Editor: Ruslan Salakhutdinov

Abstract

We present an approach based on feed-forward neural networks for learning the distribution over textual documents. This approach is inspired by the Neural Autoregressive Distribution Estimator (NADE) model which has been shown to be a good estimator of the distribution over discrete-valued high-dimensional vectors. In this paper, we present how NADE can successfully be adapted to textual data, retaining the property that sampling or computing the probability of an observation can be done exactly and efficiently. The approach can also be used to learn deep representations of documents that are competitive to those learned by alternative topic modeling approaches. Finally, we describe how the approach can be combined with a regular neural network N-gram model and substantially improve its performance, by making its learned representation sensitive to the larger, document-level context.

Keywords: Neural networks, Deep learning, Topic models, Language models, Autoregressive models

1. Introduction

One of the most common problems in machine learning is to estimate a distribution $p(\mathbf{v})$ of multidimensional data from a set of examples $\{\mathbf{v}^{(t)}\}_{t=1}^{T}$. Indeed, good estimates of $p(\mathbf{v})$ implicitly require modeling the dependencies between the variables in $\mathbf{v}$, which is required to extract meaningful representations of this data or make predictions about this data. We are particularly interested in the case where $\mathbf{v}$ is not a sequence, where the order in the vector is not relevant (random order).

The biggest challenge one faces in distribution estimation is the well-known curse of dimensionality. In fact, this issue is particularly important in distribution estimation, even more so than in other machine learning problems. This is because a good distribution estimator must provide an accurate value of $p(\mathbf{v})$ for any given $\mathbf{v}$ (i.e. not only for likely values of $\mathbf{v}$), while the number of possible values grows exponentially as the number of dimensions of the input vector $\mathbf{v}$ increases.

One example of such a model that has been successful at tackling the curse of dimensionality is the restricted Boltzmann machine (RBM) (Hinton, 2002). The RBM and other models derived from it (e.g. the Replicated Softmax of Salakhutdinov and Hinton (2009)) are frequently trained as models of the probability distribution of high-dimensional observations and then used as feature extractors. Unfortunately, one problem with these models is that for moderately large models, calculating their estimate of $p(\mathbf{v})$ is intractable. Indeed, this calculation requires computing the partition function which normalizes the model distribution. The consequences of this property of the RBM are that approximation must be taken to train it by maximum likelihood, and that its estimation of $p(\mathbf{v})$ cannot be entirely trusted.

In an attempt to tackle these issues of the RBM, the Neural Autoregressive Distribution Estimator (NADE) was introduced by Larochelle and Murray (2011). NADE's parametrization is inspired by the RBM, but uses feed-forward neural networks and the framework of autoregression for modeling the probability distribution of binary variables in high-dimensional vectors. Importantly, computing the probability of an observation under NADE can be done exactly and efficiently.

In this paper, we describe a variety of ways to extend NADE to model data from text documents. Why work with text, where the word order is important, when we said previously that we are interested in the order not being relevant? Because the text is often represented with bag-of-words, and it has no information about the order of the words. The advantage here is that we can model a good part of the text semantically while ignoring its syntax.

We start by describing Document NADE (DocNADE), a single hidden layer feed-forward neural network model for bag-of-words observations, i.e. orderless sets of words. This requires adapting NADE to vector observations $\mathbf{v}$, where each of element v_i represents a word and where the order of the dimensions is random. Each word is represented with a lower-dimensional, real-valued embedding vector, where similar words should have similar embeddings. This is in line with much of the recent work on using feed-forward neural network models to learn word vector embeddings (Bengio et al., 2003; Mnih and Hinton, 2007, 2009; Mikolov, 2013) to counteract the curse of dimensionality. However, in DocNADE, the word representations are trained to reflect the topics (i.e. semantics) of documents only, as opposed to their syntactical properties, due to the orderless nature of bags-of-words.

We then describe how to train a deep version of DocNADE. First described by Zheng et al. (2015) in the context of image modeling, here we empirically evaluate them for text documents and show that they are competitive to existing topic models, both in terms of perplexity and document retrieval performances.

Finally, we present how the topic-level modeling ability of DocNADE can be used to obtain a useful representation of context (semantic information about the past) for language modeling. We empirically demonstrate that by learning a topical representation (representation based on subjects, topics) of previous sentences, we can improve the perplexity performance of an N-gram neural language model.

2. Document NADE (DocNADE)

DocNADE is derived from the Neural Autoregressive Distribution Estimation (NADE). We first describe NADE in this Section.

©2017 Stanislas Lauly, Yin Zheng, Alexandre Allauzen and Hugo Larochelle.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-017.html.

2.1 Neural Autoregressive Distribution Estimation (NADE)

NADE, introduced by Larochelle and Murray (2011), is a tractable distribution estimator for modeling the distribution over high-dimensional binary vectors. Let us consider a binary vector of D observations, $\mathbf{v} \in \{0,1\}^D$, where the order has no meaning. We use the notation v_i to denote the i-th component of $\mathbf{v}$ and $\mathbf{v}_{<i} \in \{0,1\}^{i-1}$ to contain the first $i-1$ components of $\mathbf{v}$. $\mathbf{v}_{<i}$ is the sub-vector $[v_1, \ldots, v_{i-1}]^\top$:

$$\mathbf{v} = [\underbrace{v_1, \ \ldots, v_{i-1}}_{\mathbf{v}_{<i}}, v_i, \ \ldots, v_D]^T. \tag{1}$$

The NADE model estimates the probability of the vector $\mathbf{v}$ by applying the probability chain rule as follows:

$$p(\mathbf{v}) = \prod_{i=1}^{D} p(v_i|\mathbf{v}_{<i}). \tag{2}$$

The peculiarity of NADE lies in the neural architecture designed to estimate the conditional probabilities involved in Equation 2. To predict the component i, the model first computes its hidden layer of dimension H

$$\mathbf{h}_i(\mathbf{v}_{<i}) = \mathbf{g}\left(\mathbf{c} + \mathbf{W}_{:,<i} \cdot \mathbf{v}_{<i}\right), \tag{3}$$

leading to the following probability model:

$$p(v_i = 1|\mathbf{v}_{<i}) = \mathrm{sigm}\left(b_i + \mathbf{V}_{i,:} \cdot \mathbf{h}_i(\mathbf{v}_{<i})\right). \tag{4}$$

In these two equations, $\mathrm{sigm}(x) = 1/(1+\exp(-x))$ denotes the sigmoid activation function while function $\mathbf{g}(\cdot)$ could be any activation function. Larochelle and Murray (2011) used the sigmoid function for $\mathbf{g}(\cdot)$. $\mathbf{W} \in \mathbb{R}^{H \times D}$ and $\mathbf{V} \in \mathbb{R}^{D \times H}$ are the parameter matrices along with the associated bias terms $\mathbf{b} \in \mathbb{R}^D$ and $\mathbf{c} \in \mathbb{R}^H$. The notation $\mathbf{W}_{:,<i}$ represents a matrix made of the $i-1$ first columns of $\mathbf{W}$. The vector $\mathbf{V}_{i,:}$ is made from the i-th row of $\mathbf{V}$ and is composed of weights associated to a logistic classifier, meaning that each row of $\mathbf{V}$ represent a different logistic classifier.

Instead of a single projection of the input vector (like for a typical autoencoder, the left part of Figure 1), the NADE model relies on a set of separate hidden layers $\mathbf{h}_i(\mathbf{v}_{<i})$ that each represent the previous inputs in a latent space (right part of Figure 1). The connections between the input dimension v_i and each hidden layer $\mathbf{h}_i(\mathbf{v}_{<i})$ are tied as shown with the blue lines in Figure 1, allowing the model to compute all the hidden layers for one input in $O(DH)$. The parameters $\{\mathbf{b}, \mathbf{c}, \mathbf{W}, \mathbf{V}\}$ are learned by minimizing the average negative log-likelihood using stochastic gradient descent.

2.2 From NADE to DocNADE

The Document NADE model (DocNADE) aims at learning meaningful representations of texts from a collection of unlabeled documents. Implemented as a feed-forward architecture, it extends NADE to provide an efficient and meaningful generative model of document bags-of-words. This model embeds, as NADE did, a set of hidden layers. Their role is to capture salient statistical patterns in the

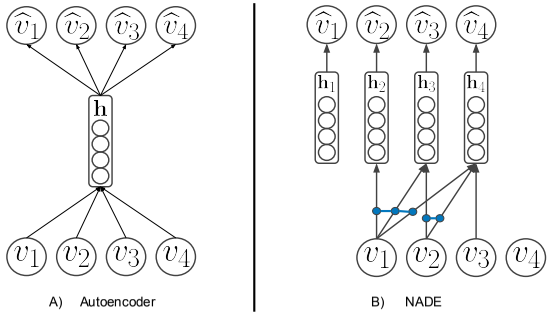


Figure 1: **A)** Typical structure for an autoencoder. **B)** Illustration of NADE. Colored lines identify the connections that share parameters and $\widehat{v}_i$ is a shorthand for the autoregressive conditional $p(v_i|\mathbf{v}_{<i})$. The observations v_i are binary.

co-occurrence of words within documents and can be considered as modeling hidden topics. This is called topic modeling (Steyvers and Griffiths, 2007), where a topic model gives a mathematical representation of a document in the form of a vector (of real values). In our case each element of the vector is a neuron of the hidden layer and can be associated to a hidden subject where its value would represent how much that subject is part of the text. This representation can be seen as a composition of the subjects.

To represent a document for DocNADE, we transform a bag-of-words into a sequence $\mathbf{v}$ of arbitrary size D. Each element of $\mathbf{v}$ corresponds to a multinomial observation (representing a word) over a fixed vocabulary of size V:

$$\mathbf{v} = [v_1, v_2, \ \ldots, v_D], \ \ v_i \in \{1, 2, \ \ldots, V\}. \tag{5}$$

Therefore v_i represents the index in the vocabulary of the i-th word of the document. For now, we assume that an ordering of the words is given, but we will discuss the general case of orderless bags-of-words in the next Section (2.2.1) and show that we can simply use a random order for the words during training.

The DocNADE model uses a word representation matrix $\mathbf{W} \in \mathbb{R}^{H \times V}$, where each column $\mathbf{W}_{:,v_i}$ of the matrix is a vector representing one of the words in the vocabulary (see Figure 2). This matrix makes the connection possible between the index v_i of a word and its embedding.

The main approach taken by DocNADE is similar to NADE, but differs significantly in the design of parameter tying. The probability of a document $\mathbf{v}$ is estimated using the probability chain rule, but the architecture is modified to cope with large vocabularies. Each word observation v_i of the document $\mathbf{v}$ leads to a hidden layer $\mathbf{h}_i$, which represents the past observations $\mathbf{v}_{<i}$ (see figure 3).

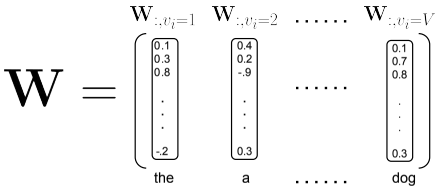


Figure 2: Word representation matrix $\mathbf{W}$, where each column of the matrix is a vector representing a word of the vocabulary.

This hidden layer is computed as follows:

$$\mathbf{h}_i(\mathbf{v}_{<i}) = \mathbf{g}\left(\mathbf{c} + \sum_{k<i} \mathbf{W}_{:,v_k}\right), \tag{6}$$

where each column of the matrix $\mathbf{W}$ acts as a vector of size H that represents a word. The embedding of the ith word in the document is thus the column of index v_i in the matrix $\mathbf{W}$.

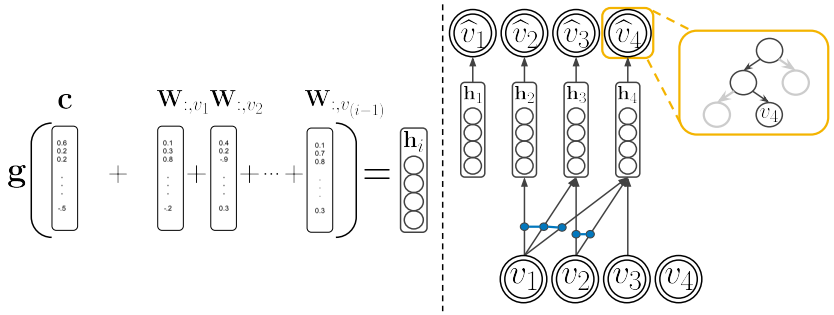


Figure 3: (**Left**) Representation of the computation of a hidden layer $\mathbf{h}_i$, function $\mathbf{g}(\cdot)$ can be any activation function. (**Right**) Illustration of DocNADE. Connections between each multinomial observation v_i and hidden units are also shared, and each conditional $p(v_i|\mathbf{v}_{<i})$ is decomposed into a tree of binary logistic regressions.

Notice that by sharing the word representation matrix across positions in the document, each hidden layer $\mathbf{h}_i(\mathbf{v}_{<i})$ is in fact independent of the order of the words within $\mathbf{v}_{<i}$. We made the hypothesis that even if the word order affect semantics, most of its information depends on the set of words used. DocNADE ignores in which order the previously observed words appeared. The implications of this choice is that the learned hidden representation will not model the syntactic structure of the document and focus entirely on its document-level semantics, i.e. its topics. It also means that the first word at the beginning of the sequence has the same influence as the last one, and

its importance compared to the other words does not deteriorate over time. This phenomenon can be observed in the left part of Figure 3. With this illustration we see that if we swap the first word v_1 with the last one v_{i-1} we would obtain the same hidden representation $\mathbf{h}_i$.

It is also worth noticing that we can compute $\mathbf{h}_{i+1}$ recursively by keeping track of the pre-activation of the previous hidden layer $\mathbf{h}_i$ as follows:

$$\mathbf{h}_{i+1}(\mathbf{v}_{<i+1}) = \mathbf{g}\left(\mathbf{W}_{:,v_i} + \underbrace{\mathbf{c} + \sum_{k<i} \mathbf{W}_{:,v_k}}_{\text{Precomputed for } \mathbf{h}_i(\mathbf{v}_{<i})}\right) \tag{7}$$

The weight sharing between the hidden layers enables us to compute all the hidden layers $\mathbf{h}_i(\mathbf{v}_{<i})$ for a document in $O(DH)$.

To compute the probability of a full document $p(\mathbf{v})$, we need to estimate all conditional probabilities $p(v_i|\mathbf{v}_{<i})$. A straightforward solution would be to compute each $p(v_i|\mathbf{v}_{<i})$ using a softmax layer with a shared weight matrix and bias with each of the hidden layers $\mathbf{h}_i$. However, the computational cost of this approach is prohibitive, as it scales linearly with the vocabulary size.[1] To overcome this issue, we represent a distribution over the vocabulary by a probabilistic binary tree, where the leaves correspond to the words. This approach is widely used in the field of neural probabilistic language models (Morin and Bengio, 2005; Mnih and Hinton, 2009). Each word is represented by a path in the tree, going from the root to the leaf associated to that word. A binary logistic regression unit is associated to each node in the tree and gives the probability of the binary choice, going left or right. A word probability can therefore be estimated by the path's probability in this tree, resulting in a complexity in $O(\log V)$ for a balanced tree. In our experiments, we used a randomly generated full binary tree with V leaves, each assigned to a unique word in the vocabulary.

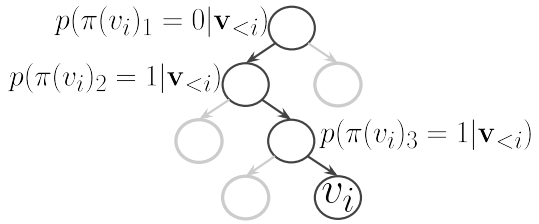


Figure 4: Path of the word v_i in a binary tree. We compute the probability of the left/right choice (0/1) for each node of the path.

More formally, let's denote by $\mathbf{l}(v_i)$ the sequence of nodes composing the path, from the root of the tree to the leaf corresponding to the word v_i. Then, $\boldsymbol{\pi}(v_i)$ is the sequence of left/right decisions of the nodes in $\mathbf{l}(v_i)$. For example, the root of the tree is always the first element $l(v_i)_1$ and the value

1. For most natural language processing tasks, the vocabulary size exceeds $10,000$.

$\pi(v_i)_1$ will be 0 if the word is in the left sub-tree and 1 if it is in the right sub-tree (see figure 4). The matrix $\mathbf{V}$ stores by row the weights associated to each logistic classifier. There is one logistic classifier per node in the tree, but only a fraction of them is used for each word. Let $\mathbf{V}_{l(v_i)_m,:}$ and $b_{l(v_i)_m}$ be the weights and bias of the logistic unit associated with the node $n(v_i)_m$. The probability $p(v_i|\mathbf{v}_{<i})$ given the tree and the hidden layer $\mathbf{h}_i(\mathbf{v}_{<i})$ is computed by the following formulas:

$$p(v_i = w|\mathbf{v}_{<i}) = \prod_{m=1}^{|\boldsymbol{\pi}(v_i)|} p(\pi(v_i)_m = \pi(w)_m|\mathbf{v}_{<i}), \text{ with} \tag{8}$$

$$p(\pi(v_i)_m = 1|\mathbf{v}_{<i}) = \text{sigm}\left(b_{l(v_i)_m} + \mathbf{V}_{l(v_i)_m,:} \cdot \mathbf{h}_i(\mathbf{v}_{<i})\right). \tag{9}$$

This hierarchical architecture allows us to efficiently compute the probability of each word in a document and therefore the probability of the whole document with the probability chain rule (see Equation 2). As in the NADE model, the parameters of the model $\{\mathbf{b}, \mathbf{c}, \mathbf{W}, \mathbf{V}\}$ are learnt by minimizing the negative log-likelihood using stochastic gradient descent.

Since there is $\log(V)$ logistic regression units for a word (one per node), each of them has a time complexity of $O(H)$, the complexity of computing the probability for a document of D words is in $O(\log(V)DH)$.

For using DocNADE to extract features from a complete document, we propose to use

$$\mathbf{h}(\mathbf{v}) = \mathbf{h}_{D+1}(\mathbf{v}_{<D+1}) = \mathbf{g}\left(\mathbf{c} + \sum_{k=1}^{D} \mathbf{W}_{:,v_k}\right) \tag{10}$$

which would be the hidden layer computed to obtain the conditional probability of a hypothetical $(D+1)$-th word appearing in the document. In other words, $\mathbf{h}(\mathbf{v})$ is the representation of all the words in the document. Notice that we do not average the document representation to become invariant to document length. Averaging did not result in any performance improvement in our preliminary experiments. We conjecture that it is because every word count n_i in the data was replaced by $\log(1 + n_i)$, which reduces the variability of the input values and makes the model more invariant to the document length. However, averaging sometimes is useful, such as for the language model approach (see Section 4).

Algorithm 1 gives the pseudo-code for computing the negative log-likelihood cost used during training and the hidden layer used for representing the whole document under DocNADE.

2.2.1 Training from bag-of-words counts

So far, we have assumed that the ordering of the words in the document is known. However, a document often takes the form of word-count vectors in which the original word order, required for specifying the sequence of conditionals $p(v_i|\mathbf{v}_{<i})$, has been lost.

It is however still possible to successfully train a DocNADE despite the absence of this information. The idea is to assume that each observed document $\mathbf{v}$ was generated by initially sampling a *seed* document $\widetilde{\mathbf{v}}$ from DocNADE, whose words were then shuffled using a randomly generated ordering to produce $\mathbf{v}$. With this approach, we can express the probability distribution of $\mathbf{v}$ by computing the marginal over all possible seed document:

$$p(\mathbf{v}) = \sum_{\widetilde{\mathbf{v}} \in \mathcal{V}(\mathbf{v})} p(\mathbf{v}, \widetilde{\mathbf{v}}) = \sum_{\widetilde{\mathbf{v}} \in \mathcal{V}(\mathbf{v})} p(\mathbf{v}|\widetilde{\mathbf{v}}) p(\widetilde{\mathbf{v}}) \tag{11}$$

Algorithm 1 Computing of the cost for training and the hidden layer for representing an entire document (DocNADE model)

Input: Multinomial observation $\mathbf{v}$

$\mathbf{a} \leftarrow \mathbf{c}$
$NLL \leftarrow 0$
For $i = 1$ to D **Do**
  $\mathbf{h}_i \leftarrow \mathbf{g}(\mathbf{a})$

  $p(v_i) \leftarrow 1$
  For $m = 1$ to $|\boldsymbol{\pi}(v_i)|$ **Do**
    $p(\pi(v_i)_m = 1|\mathbf{v}_{<i}) \leftarrow \text{sigm}\left(b_{l(v_i)_m} + \mathbf{V}_{l(v_i)_m,:} \cdot \mathbf{h}_i(\mathbf{v}_{<i})\right)$
    $p(v_i) \leftarrow p(v_i)\left(p(\pi(v_i)_m = 1|\mathbf{v}_{<i})^{\pi(v_i)_m} + (1 - p(\pi(v_i)_m = 1|\mathbf{v}_{<i})^{1-\pi(v_i)_m})\right)$
  End for

  $NLL \leftarrow NLL - \log p(v_i)$
  $\mathbf{a} \leftarrow \mathbf{a} + \mathbf{W}_{:,v_i}$
End for

$\mathbf{h}_{final} \leftarrow \mathbf{g}(\mathbf{a})$

Return NLL, $\mathbf{h}_{final}$

where $p(\widetilde{\mathbf{v}})$ is modeled by DocNADE. $\widetilde{\mathbf{v}}$ is the same as the observed document $\mathbf{v}$ but with a different (random) word sequence order, and $\mathcal{V}(\mathbf{v})$ is the set of all the documents $\widetilde{\mathbf{v}}$ that has the same word count $\mathbf{n}(\mathbf{v}) = \mathbf{n}(\widetilde{\mathbf{v}})$. With the assumption of orderings being uniformly sampled, we can replace $p(\mathbf{v}|\widetilde{\mathbf{v}})$ with $\frac{1}{|\mathcal{V}(\mathbf{v})|}$ giving us:

$$p(\mathbf{v}) = \sum_{\widetilde{\mathbf{v}} \in \mathcal{V}(\mathbf{v})} \frac{1}{|\mathcal{V}(\mathbf{v})|} p(\widetilde{\mathbf{v}}) = \frac{1}{|\mathcal{V}(\mathbf{v})|} \sum_{\widetilde{\mathbf{v}} \in \mathcal{V}(\mathbf{v})} p(\widetilde{\mathbf{v}}). \tag{12}$$

In practice, one approach to training the DocNADE model over $\widetilde{\mathbf{v}}$ is to artificially generate ordered documents by uniformly sampling words, without replacement, from the bags-of-words in the dataset. This would be equivalent to taking each original document and shuffling the order of its words. This approach can be shown to minimize a stochastic upper bound on the true negative log-likelihood of documents. As we will see, experiments show that convincing performance could still be reached.

With this training procedure, DocNADE shows its ability to learn and predict a new word in a document at a random position while preserving the overall semantic properties of the document. The model is therefore learning not to insert intruder words, i.e. words that do not belong with the others. After training, a document's learned representation should contain valuable information to identify intruder words for this document. It is interesting to note that the detection of such intruder words has been used previously as a task in user studies to evaluate the quality of the topics learned by LDA, though at the level of single topics and not whole documents (Chang et al., 2009).

The goal with DocNADE is to have a simple, powerful, fast way to train model that can also be used with data that have lost the information about ordering. We want to learn topic-level semantics, where the order of the words is less important than the existence of words is, instead of learning syntax, where the order of the words is more important. In this context, it makes no sense to apply an RNN-like model on randomly shuffled words, since there's no position varying information to model.

3. Deep Document NADE

The single hidden layer version of DocNADE already achieves very competitive performance for topic modeling (Larochelle and Lauly, 2012). Extending it to a deep architecture with multiple hidden layer could however enable better performance, as suggested by the recent and impressive success of deep neural networks. Unfortunately, deriving a deep version of DocNADE cannot be achieved solely by adding hidden layers to the definition of the conditionals $p(v_i|\mathbf{v}_{<i})$. Indeed, computing $p(\mathbf{v})$ requires computing each $p(v_i|\mathbf{v}_{<i})$ conditional (one for each word), and it is no longer possible to use Equation 7 to compute the sequence of all hidden layers in $O(DH)$ when there are multiple deep hidden layers.

In this Section, we describe an alternative training procedure that enables us the introduction of multiple stacked hidden layers. This procedure was first introduced by Zheng et al. (2015) to model images, which was itself borrowing from the training scheme introduced by Uria et al. (2014).

As mentioned in Section 2.2.1, DocNADE can be trained on random permutations of the words from training documents. As noticed by Uria et al. (2014), the use of many orderings during training can be seen as the instantiation of many different DocNADE models that share a single set of parameters. Thus, training DocNADE with random permutations also amounts to minimizing the negative log-likelihood averaged across all possible orderings, for each training example $\mathbf{v}$.

In the context of deep NADE models, a key observation is that training on all possible orderings implies that for a given context $\mathbf{v}_{<i}$, we encourage the model to be equally good at predicting any of the remaining words appearing next. Thus, we redesign the training algorithm such that, instead of sampling a complete ordering of all words for each update, we instead sample a single context $\mathbf{v}_{<i}$ and perform an update of the conditionals using that context. This is done as follows. For a given document, after generating vector $\mathbf{v}$ by shuffling the words from the document, a split point i is randomly drawn. From this split point we obtain two parts of the document, $\mathbf{v}_{<i}$ and $\mathbf{v}_{\geq i}$:

$$\mathbf{v} = [\underbrace{v_1, \ \ldots, v_{i-1}}_{\mathbf{v}_{<i}}, \underbrace{v_i, \ \ldots, v_D}_{\mathbf{v}_{\geq i}}]^\top. \tag{13}$$

The vector $\mathbf{v}_{<i}$ is considered as the input and the vector $\mathbf{v}_{\geq i}$ contains the targets to be predicted by the model. Since a training update relies on the computation of a single latent representation, that of $\mathbf{v}_{<i}$ for the drawn value of i, deeper hidden layers can be added at a reasonable increase in computation. We call a deep DocNADE trained with this procedure a DeepDocNADE.

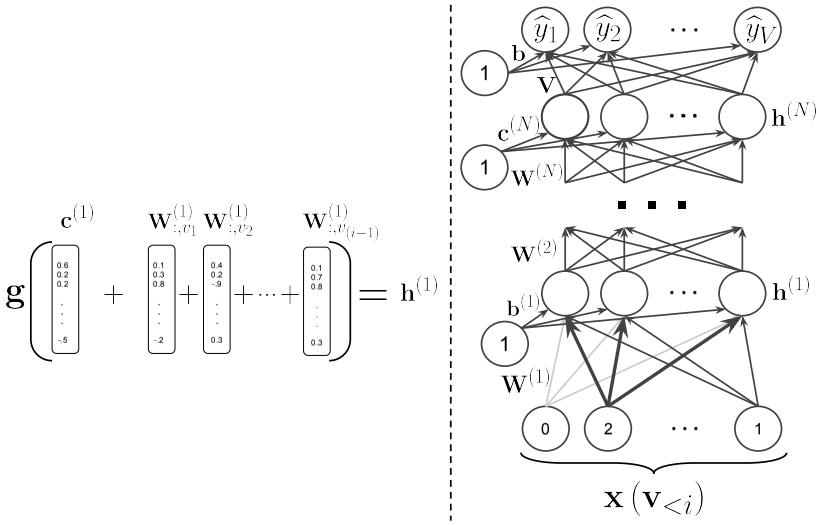


Figure 5: (**Left**) Representation of the computation by summation of the first hidden layer $\mathbf{h}^{(1)}$, which is equivalent to multiplying the bag-of-words $\mathbf{x}\left(\mathbf{v}_{<i}\right)$ with the word representation matrix $\mathbf{W}^{(1)}$. (**Right**) Illustration of DeepDocNADE architecture.

In DeepDocNADE the conditionals $p(v_i|\mathbf{v}_{<i})$ are modeled as follows. The first hidden layer $\mathbf{h}^{(1)}(\mathbf{v}_{<i})$ represents the conditioning context $\mathbf{v}_{<i}$ as in the single hidden layer DocNADE:

$$\mathbf{h}^{(1)}\left(\mathbf{v}_{<i}\right) = \mathbf{g}\left(\mathbf{c}^{(1)} + \sum_{k<i} \mathbf{W}^{(1)}_{:,v_k}\right) = \mathbf{g}\left(\mathbf{c}^{(1)} + \mathbf{W}^{(1)} \cdot \mathbf{x}\left(\mathbf{v}_{<i}\right)\right) \tag{14}$$

where $\mathbf{x}\left(\mathbf{v}_{<i}\right)$ is the histogram vector representation (bag-of-words) of the word sequence $\mathbf{v}_{<i}$, and the exponent is used as an index over the hidden layers and its parameters, with (1) referring to the first layer (see Figure 5). The matrix $\mathbf{W}^{(1)}$ is the word representation matrix of DeepDocNADE. Notice that like DocNADE, the first hidden layer $\mathbf{h}^{(1)}(\mathbf{v}_{<i})$ of DeepDocNADE is independent of the order of the words within $\mathbf{v}_{<i}$. We can now easily add new hidden layers as in a regular deep feed-forward neural network:

$$\mathbf{h}^{(n)}(\mathbf{v}_{<i}) = \mathbf{g}(\mathbf{c}^{(n)} + \mathbf{W}^{(n)} \cdot \mathbf{h}^{(n-1)}(\mathbf{v}_{<i})), \tag{15}$$

for $n = 2, \ldots, N$, where N is the total number of hidden layers. From the last hidden layer $\mathbf{h}^{(N)}$, we compute the conditional $p(v_i = w|\mathbf{v}_{<i})$, for any word w:

$$\widehat{y}_w = p(v_i = w|\mathbf{v}_{<i}) = \text{softmax}(\mathbf{V}_{w,:} \cdot \mathbf{h}^{(N)}(\mathbf{v}_{<i}) + b_w), \tag{16}$$

$$\widehat{\mathbf{y}} = \text{softmax}(\mathbf{V} \cdot \mathbf{h}^{(N)}(\mathbf{v}_{<i}) + \mathbf{b}), \tag{17}$$

where $\mathbf{V} \in \mathbb{R}^{V \times H^{(N)}}$ is the output parameter matrix and $\mathbf{b}$ the bias vector. The notation $\widehat{\mathbf{y}}$ is a vector that represents the probability of all the vocabulary words computed at once. Each element of the vector, $\hat{y}_1$ to $\hat{y}_V$, represents one of the words in the vocabulary. We use $\hat{y}_w$ to make the connection with $v_i = w$ (the index of the word w at time i).

Finally, the loss function used to update the model for the given context $\mathbf{v}_{<i}$ is:

$$L(\mathbf{v}) = \frac{D_\mathbf{v}}{D_\mathbf{v} - i + 1} \sum_{w \in \mathbf{v}_{\geq i}} -\log p(v_i = w | \mathbf{v}_{<i}), \tag{18}$$

where $D_\mathbf{v}$ is the number of words in $\mathbf{v}$, and the summation iterates over all the words w present in $\mathbf{v}_{\geq i}$. Thus, as described earlier, the model predicts each remaining word after the splitting position i as if it was actually at position i. The factor in front of the summation comes from the fact that the complete log-likelihood would contain $D_\mathbf{v}$ log-conditionals and that we are averaging over $D_\mathbf{v} - i + 1$ possible choices for the word at position i. For a more detailed presentation, see Zheng et al. (2015). The average loss function of Equation 18 is optimized with stochastic gradient descent.[2]

Note that to compute the probability $p(v_i = w | \mathbf{v}_{<i})$, a probabilistic tree could again be used. However, since all probabilities needed for an update are based on a single context $\mathbf{v}_{<i}$, a single softmax layer is sufficient to compute all necessary quantities. Therefore the computational burden of a conventional softmax is not as prohibitive as for DocNADE, especially with an efficient implementation on the GPU. For this reason, in our experiments with DeepDocNADE we opted for a regular softmax.

4. DocNADE Language Model

While topic models such as DocNADE can be useful in learning topic representations of documents, they are very poor models of language. In DocNADE, this is due to the fact that, when assigning a probability to the next word in a sentence, it ignores the order in which the previously observed words appeared. Yet, this ordering of words conveys a lot of information regarding a syntactic role of the next word or the finer semantics within the sentence. In fact, most of that information is predictable from the last few words, which is why N-gram language models remain a dominating approach to language modeling.

In this Section, we propose a new model that extends DocNADE to mitigate the influence of both short and long term dependencies in a single model, which we refer to as the DocNADE language model or DocNADE-LM. The solution we propose enhances the bag-of-words representation with the explicit inclusion of n-gram dependencies.

Figure 6 depicts the overall architecture. This model can be seen as an extension of the seminal work on neural language models by Bengio et al. (2003) that includes the representation of a document's larger context. It can also be seen as a neural extension of the cache-based language model introduced in (Kuhn and Mori, 1990), where the n-gram probability is interpolated with the

2. A document is usually represented as bag-of-words. Generating a word vector $\mathbf{v}$ from its bag-of-words, shuffling the word count vector $\mathbf{v}$, splitting it, and then regenerating the histogram $\mathbf{x}\,(\mathbf{v}_{<i})$ and $\mathbf{x}\,(\mathbf{v}_{\geq i})$ is unfortunately fairly inefficient for processing samples in a mini-batch fashion. Hence, in practice, we split the original histogram $\mathbf{x}\,(\mathbf{v})$ directly by uniformly sampling, for each word individually, how many are put on the left of the split (the others are put on the right of the split). This procedure, used also by Zheng et al. (2015), is only an approximation to the correct procedure mentioned in the main text, but results in substantial speedup while also yielding good performance.

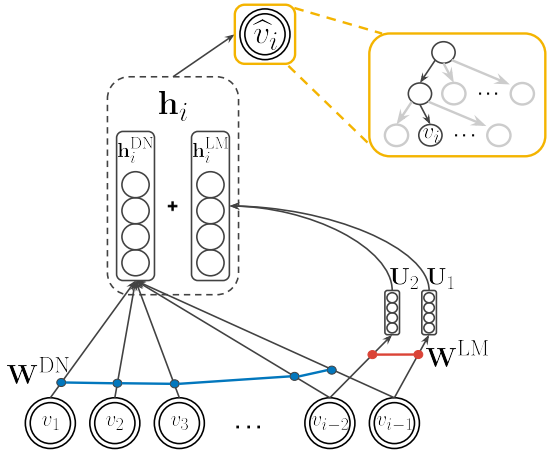


Figure 6: Illustration of the conditional $p(v_i|\mathbf{v}_{<i})$ in a trigram NADE language model. Compared to DocNADE, this model incorporates the architecture of a neural language model, that first maps previous words (2 for a trigram model) to vectors using an embedding matrix $\mathbf{W}^{\mathrm{LM}}$ before connecting them to the hidden layer using regular (untied) parameter matrices ($\mathbf{U_1}$, $\mathbf{U_2}$ for a trigram). In our experiments, each conditional $p(v_i|\mathbf{v}_{<i})$ exploits decomposed into a tree of logistic regressions, the hierarchical softmax.

word distribution observed in a dynamic cache. This cache of a fixed size keeps track of previously observed words to include long term dependencies in the prediction and preserve semantic consistency beyond the scope of the n-gram. The DocNADE language model maintains an unbounded cache and defines a proper, jointly trained solution to mitigate these two kinds (long and short) of dependencies.

As in DocNADE, a document $\mathbf{v}$ is modeled as a sequence of multinomial observations. The sequence size is arbitrary and this time the order is important, each element v_i consists of the i-th word of the sequence and represent an index in a vocabulary of size V. The conditional probability of a word given its history $p(v_i|\mathbf{v}_{<i})$ is now expressed as a smooth function of a hidden layer $\mathbf{h}_i(\mathbf{v}_{<i})$ used to predict word v_i. The peculiarity of the DocNADE language model lies in the definition of this hidden layer, which now includes two terms:

$$\mathbf{h}_i(\mathbf{v}_{<i}) = \mathbf{g}(\mathbf{b} + \mathbf{h}_i^{\mathrm{DN}}(\mathbf{v}_{<i}) + \mathbf{h}_i^{\mathrm{LM}}(\mathbf{v}_{<i})). \tag{19}$$

The first term borrows from the DocNADE model by aggregating embeddings for all the previous words in the history:

$$\mathbf{h}_i^{\mathrm{DN}}(\mathbf{v}_{<i}) = \frac{1}{i-1} \sum_{k<i} \mathbf{W}^{\mathrm{DN}}_{:,v_k}, \tag{20}$$

where $i - 1$ is the number of words used to create $\mathbf{h}_i^{\mathrm{DN}}(\mathbf{v}_{<i})$. We average the hidden representation by the number of words to get good results for the language modeling task, so that the

representation can become invariant to the document's length. The hidden representation $\mathbf{h}_i^{\text{DN}}(\mathbf{v}_{<i})$ models the semantic aspects of the text that have been observed so far. As we explained earlier for DocNADE, the order of the words within $\mathbf{v}_{<i}$ is not modeled. The hidden layer $\mathbf{h}_i^{\text{DN}}(\mathbf{v}_{<i})$ can be seen as a context representation of the past where the influence of a word relative to other words does not diminish over time.

The second contribution derives from a neural n-gram language model as follows:

$$\mathbf{h}_i^{\text{LM}}(\mathbf{v}_{<i}) = \sum_{k=1}^{n-1} \mathbf{U}_k \cdot \mathbf{W}^{\text{LM}}_{:,v_{i-k}} \ . \tag{21}$$

In this formula, the history for the word v_i is restricted to the $n-1$ preceding words, following the common n-gram assumption. The term $\mathbf{h}_i^{\text{LM}}$ hence represents the continuous representation of these $n-1$ words, in which word embeddings are linearly transformed by the ordered matrices $\mathbf{U}_1, \mathbf{U}_2, ..., \mathbf{U}_{n-1}$. Moreover, $\mathbf{b}$ gathers the bias terms for the hidden layer. In this model, two sets of word embeddings are defined, $\mathbf{W}^{\text{DN}}$ and $\mathbf{W}^{\text{LM}}$, which are respectively associated to the DocNADE and neural language model parts. For simplicity, we assume both are of the same size H.

Given hidden layer $\mathbf{h}_i(\mathbf{v}_{<i})$, conditional probabilities $p(v_i|\mathbf{v}_{<i})$ can be estimated, and thus $p(\mathbf{v})$. For the aforementioned reason explained in DocNADE, the output layer is structured as a probabilistic tree for efficient computations. Specifically, we decided to use a variation of the probabilistic binary tree, known as a hierarchical softmax layer. In this case, instead of having binary nodes with multiple levels in the tree, we have only two levels where all words have their leaf at level two and each node is a multiclass (i.e. softmax) logistic regression with roughly $\sqrt{V}$ classes (one for each children). Computing probabilities in such a structured layer can be done using only two matrix multiplications, which can be efficiently computed on the GPU.

With a hidden layer of size H, the complexity of computing the softmax at one node is $O(H\sqrt{V})$. If we have D words in a given document, the complexity of computing all necessary probabilities from the hidden layers is thus $O(DH\sqrt{V})$. It also requires $O(DH)$ computations to compute the hidden representations for the DocNADE part and $O(nH^2)$ for the language model part. The full complexity for computing $p(\mathbf{v})$ and the updates for the parameters is thus $O(DH\sqrt{V}+DH+nH^2)$.

Once again, the loss function of the model is the negative log-likelihood, and we minimize it by using stochastic gradient descent over documents, to learn the values of the parameters $\{\mathbf{b}, \mathbf{c}, \mathbf{V}, \mathbf{W}^{\text{LM}}, \mathbf{W}^{\text{DN}}, \mathbf{U}_1, ..., \mathbf{U}_n\}$.

5. Related Work

As NADE was inspired by the RBM, DocNADE can be seen as related to the Replicated Softmax model (Salakhutdinov and Hinton, 2009), an extension of the RBM to document modeling. Here, we describe in more detail the Replicated Softmax, along with its relationship with DocNADE.

Much like the RBM, the Replicated Softmax models observations using a latent, stochastic binary layer $\mathbf{h}$. Here, the observation is a document $\mathbf{v}$ and interacts with the hidden layer $\mathbf{h}$ through an energy function similar to RBM's:

$$E(\mathbf{v}, \mathbf{h}) = -D\, \mathbf{c}^\top \mathbf{h} + \sum_{i=1}^{D} -\mathbf{h}^\top \mathbf{W}_{:,v_i} - b_{v_i} = -D\, \mathbf{c}^\top \mathbf{h} - \mathbf{h}^\top \mathbf{W}\mathbf{n}(\mathbf{v}) - \mathbf{b}^\top \mathbf{n}(\mathbf{v}), \tag{22}$$

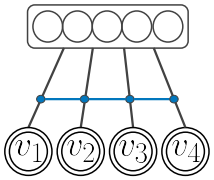


Figure 7: Replicated Softmax model. Each multinomial observation v_i is a word. Connections between each multinomial observation v_i and hidden units are shared.

where $\mathbf{n}(\mathbf{v})$ is a bag-of-words vector of size V (the size of the vocabulary) containing the word count of each word in the vocabulary for document $\mathbf{v}$. $\mathbf{h}$ is a stochastic, binary hidden layer vector, and $\mathbf{W}_{:,v_i}$ is the v_i-th column vector of matrix $\mathbf{W}$. $\mathbf{c}$ and $\mathbf{b}$ are the bias vectors for the visible and the hidden layers. We see here that the larger $\mathbf{v}$ is, the bigger the number of terms in the sum over i is, resulting in a high energy value. For this reason, the hidden bias term $\mathbf{c}^\top \mathbf{h}$ is multiplied by D, to be commensurate with the contribution of the visible layer.. We can see also that connection parameters are shared across different positions i in $\mathbf{v}$, as illustrated in Figure 7.

The conditional probabilities of the hidden and the visible layer factorize as in the RBM, in the following way:

$$p(\mathbf{h}|\mathbf{v}) = \prod_j p(h_j|\mathbf{v}) \ , \quad p(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{D} p(v_i|\mathbf{h}) \tag{23}$$

where the factors $p(h_j|\mathbf{v})$ and $p(v_i|\mathbf{h})$ are such that

$$p(h_j = 1|\mathbf{v}) = \text{sigm}(Dc_j + \sum_i W_{jv_i}) \tag{24}$$

$$p(v_i = w|\mathbf{h}) = \frac{\exp(b_w + \mathbf{h}^\top \mathbf{W}_{:,w})}{\sum_{w'} \exp(b_{w'} + \mathbf{h}^\top \mathbf{W}_{:,w'})} \tag{25}$$

The normalized exponential part in $p(v_i = w|\mathbf{h})$ is simply the softmax function. To train this model, we minimize the negative log-likelihood (NLL). Its gradient for a document $\mathbf{v}^{(t)}$ with respect to the parameters $\theta = \{\mathbf{W}, \mathbf{c}, \mathbf{b}\}$ is calculated as follows:

$$-\frac{\partial \log p(\mathbf{v}^{(t)})}{\partial \theta} = \mathbf{E}_{\mathbf{h}|\mathbf{v}^{(t)}} \left[\frac{\partial}{\partial \theta} E(\mathbf{v}^{(t)}, \mathbf{h}) \right] - \mathbf{E}_{\mathbf{v},\mathbf{h}} \left[\frac{\partial}{\partial \theta} E(\mathbf{v}, \mathbf{h}) \right]. \tag{26}$$

As with the conventional RBM, the second expectation in Equation 26 is computationally intractable. The gradient of the negative log-likelihood is therefore approximated by replacing the second expectation with an estimated value obtained by contrastive divergence (Hinton, 2002). This approach consists of performing K steps of block Gibbs sampling, starting at $\mathbf{v}^{(t)}$ and using Equations 24 and 25, to obtain a point estimate of the expectation over $\mathbf{v}$. A large values of K must be used to reduce the bias of gradient estimates and obtain good estimates of the distribution. This approximation is used to perform stochastic gradient descent.

During Gibbs sampling, the Replicated Softmax model must compute and sample from $p(v_i = w|\mathbf{h})$, which requires the computation of a large softmax. Most importantly, the computation of the softmax most be repeated K times for each update, which can be prohibitive, especially for large vocabularies. Unlike DocNADE, this softmax cannot simply be replaced by a structured softmax.

DocNADE is closely related to how the Replicated Softmax approximates, through mean-field inference, the conditionals $p(v_i = w|\mathbf{v}_{<i})$. Computing the conditional $p(v_i = w|\mathbf{v}_{<i})$ with the Replicated Softmax is intractable. However, we could use mean-field inference to approximate the full conditional $p(v_i, \mathbf{v}_{>i}, \mathbf{h}|\mathbf{v}_{<i})$ by introducing a factorized approximate distribution:

$$q(v_i, \mathbf{v}_{>i}, \mathbf{h}|\mathbf{v}_{<i}) = \prod_{k \geq i} q(v_k|\mathbf{v}_{<i}) \prod_j q(h_j|\mathbf{v}_{<i}) \,, \tag{27}$$

where $q(v_k = w|\mathbf{v}_{<i}) = \mu_{kw}(i)$ and $q(h_j = 1|\mathbf{v}_{<i}) = \tau_j(i)$. We would find the parameters $\mu_{kw}(i)$ and $\tau_j(i)$ that minimize the KL divergence between $q(v_i, \mathbf{v}_{>i}, \mathbf{h}|\mathbf{v}_{<i})$ and $p(v_i, \mathbf{v}_{>i}, \mathbf{h}|\mathbf{v}_{<i})$ by applying the following message passing equations until convergence:

$$\tau_j(i) \leftarrow \text{sigm}\left(D\, c_j + \sum_{k \geq i} \sum_{w'=1}^{V} W_{jw'} \mu_{kw'}(i) + \sum_{k<i} W_{jv_k} \right), \tag{28}$$

$$\mu_{kw}(i) \leftarrow \frac{\exp(b_w + \sum_j W_{jw} \tau_j(i))}{\sum_{w'} \exp(b_{w'} + \sum_j W_{jw'} \tau_j(i))}. \tag{29}$$

with $k \in \{i, \ldots, D\}$, $j \in \{1, \ldots, H\}$ and $w \in \{1, \ldots, V\}$. The conditional $p(v_i = w|\mathbf{v}_{<i})$ is then estimated with $\mu_{kw}(i)$ for all i. We note that one iteration of mean-field (with $\mu_{kw'}(i)$ initialized to 0) in the Replicated Softmax corresponds to the conditional $p(v_i = w|\mathbf{v}_{<i})$ computed by DocNADE with a single hidden layer and a flat softmax output layer.

In our experiment, we show that DocNADE compares favorably to Replicated Softmax.

5.1 Related work discussion

With DocNADE, the conditional probability of each word in a document is computed given the set of previously observed words. The probability distribution of the whole document is modeled using the probability chain rule. DocNADE automatically learns the underlying word representations and can compute the probability of a document exactly and efficiently. Similarly, the Replicated Softmax (Salakhutdinov and Hinton, 2009) also models the distribution, but because calculating the needed partition function for moderately large models is intractable, an approximation must be made to compute the probability of a document, making the estimate not entirely trustable. There are also other works that can learn good word representations like DocNADE, such as typical autoencoder, word2vec, doc2vec, etc. For example, people could use continuous bag-of-words autoencoder to learn good representations. The popular word2vec approaches, CBOW and Skip-gram (Mikolov et al., 2013), also learn word representations. The CBOW task is to predict a word given a surrounding context of N words where the Skip-gram does the opposite, predicting the N context words surrounding a specific word. To generate a document representation, Doc2vec (Le and Mikolov, 2014) (also called paragraph-vectors) modifies word2vec by adding a document-unique feature vector during training. For inference, a new document is presented, and all the model parameters are fixed to calculate the document vector. The advantage of DocNADE over all these above proposals is that it can model the probability of the documents as well as learn representations of words. Hence, the representations are more suitable for document retrieval tasks.

6. Topic modeling experiments

To compare the topic models, two quantitative measures are used. The first one evaluates the generative ability of different models, by computing the perplexity on held-out texts. The second one compares the quality of document representations for information retrieval.

Two different datasets are used for the experiments in this Section, a small one and a relatively big one, 20 Newsgroups and RCV1-V2 (Reuters Corpus Volume I) respectively. The 20 Newsgroups corpus has 18,786 documents (postings) partitioned into 20 different classes (newsgroups). RCV1-V2 is a much bigger dataset composed of 804,414 documents (newswire stories) manually categorized into 103 classes (topics). The two datasets were preprocessed by stemming the text and removing common stop-words. The 2,000 most frequent words of the 20 Newsgroups training set and the 10,000 most frequent words of the RCV1-V2 training set were used to create the dictionary for each dataset. Also, every word count n_i, used to represent the number of times a word appears in a document, was replaced with $log(1 + n_i)$ rounded to the nearest integer, following Salakhutdinov and Hinton (2009).

For these experiments, all the hidden representations of all the models are composed of 50 dimensional features. Early stopping on the validation set is used for avoiding overfitting and for model selection.

6.1 Generative Model Evaluation

For the generative model evaluation, we follow the experimental setup proposed by Salakhutdinov and Hinton (2009) for 20 Newsgroups and RCV1-V2 datasets. We use the exact same split for the sake of comparison. The setup consists of respectively 11,284 and 402,207 training examples for 20 Newsgroups and RCV1-V2. We randomly extract 1,000 and 10,000 documents from the training sets of 20 Newsgroups and RCV1-V2, respectively, to build a validation set. The average perplexity per word is used for comparison. This perplexity is estimated using the 50 first documents of a separate test set, as follows:

$$\exp\left(-\frac{1}{T} \sum_t \frac{1}{|\mathbf{v}^t|} \log p(\mathbf{v}^t) \right), \tag{30}$$

where T is the total number of examples in the test set and $\mathbf{v}^t$ is the t-th test document.[3]

Table 1 presents the perplexities per word results for 20 Newsgroups and RCV1-V2. We compare 5 different models: the Latent Dirichlet Allocation (LDA) (Blei et al., 2003), the Replicated Softmax, the recently proposed fast Deep AutoRegressive Networks (fDARN) (Mnih and Gregor, 2014), DocNADE and DeepDocNADE (DeepDN in the table). Each model uses 50 latent topics (size of the representation of a document). For the experiments with DeepDocNADE, we provide the performance when using 1, 2, and 3 hidden layers. As shown in Table 1, DeepDocNADE results in the best generative performances. Our best DeepDocNADE models were trained with the Adam optimizer (Kingma and Ba, 2014) and with $\tanh$ activation function. The hyper-parameters of Adam were selected on the validation set.

3. Note that there is a difference between the sizes, for the training sets and test sets of 20 Newsgroups and RCV1-V2 reported in this paper and the one reported in the original data paper of Salakhutdinov and Hinton (2009). The correct values are the ones given in this Section, which was confirmed after personal communication with Salakhutdinov and Hinton (2009).

Dataset	LDA	Replicated Softmax	fDARN	DocNADE	DeepDN (1layer)	DeepDN (2layer)	DeepDN (3layer)
20 News	1091	953	917	896	**835**	877	923
RCV1-v2	1437	988	724	742	579	552	**539**

Table 1: Test perplexity per word for models with a document representation of size 50 (50 topics). The results for LDA and Replicated Softmax were taken from Salakhutdinov and Hinton (2009).

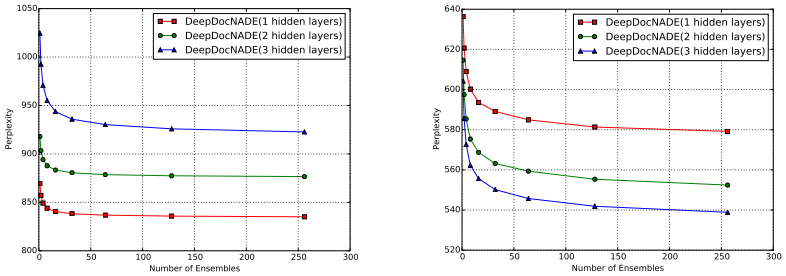


Figure 8: Perplexity obtained with different numbers of word orderings m for 20 Newsgroups on the left and RCV1-V2 on the right.

Furthermore, following Uria et al. (2014), an ensemble approach is used to compute the probability of documents, where each component of the ensembles are the same DeepDocNADE model evaluated on a different word ordering. The perplexity with M ensembles is then

$$\exp\left(-\frac{1}{T}\sum_{t}\frac{1}{|\mathbf{v}^t|}\log\left(\frac{1}{M}\sum_{m}p(\mathbf{v}^{(t,m)})\right)\right), \tag{31}$$

where T is the total number of examples, M is the number of ensembles (word orderings) and $\mathbf{v}^{(t,m)}$ denotes the m-th word ordering for the t-th document. We try $M = \{1, 2, 4, 16, 32, 64, 128, 256\}$, with the results in Table 1 using $M = 256$. For the 20 Newsgroups dataset, adding more hidden layers to DeepDocNADE fails to further improve the performance. We hypothesize that the relatively small size of this dataset makes it hard to successfully train a deep model. However, the opposite is observed on the RCV1-V2 dataset, which is more than an order of magnitude larger than 20 Newsgroups. In this case, DeepDocNADE outperforms fDARN and DocNADE, with a relative perplexity reduction of 20%, 24% and 26% with respectively 1,2 and 3 hidden layers.

To illustrate the impact of M on the performance of DeepDocNADE, Figure 8 shows the perplexity on both datasets using the different values for M that we tried. We can observe that beyond $M = 128$, this hyper-parameter has only a minimal impact on the perplexity.

weapons	**medical**	**companies**	**define**	**israel**	**book**	**windows**
weapon	treatment	demand	defined	israeli	reading	dos
shooting	medecine	commercial	definition	israelis	read	microsoft
firearms	patients	agency	refer	arab	books	version
assault	process	company	make	palestinian	relevent	ms
armed	studies	credit	examples	arabs	collection	pc

Table 2: The five nearest neighbors in the word representation space learned by DocNADE.

6.2 Document Retrieval Evaluation

A document retrieval evaluation task was also used to evaluate the quality of the document representation learned by each model. As in the previous Section, the datasets under consideration are 20 Newsgroups and RCV1-V2. The experimental setup is the same for the 20 Newsgroups dataset, while for the RCV1-V2 dataset, we reproduce the same setup as the one used by Srivastava et al. (2013). The training set contains 794,414 examples and 10,000 examples constituted the test set.

For DocNADE and DeepDocNADE, the representation of a document is obtained simply by computing the top-most hidden layer (Equation 10) when feeding all the words of a document as input.

The retrieval task follows the setup by Srivastava et al. (2013). The documents in the training and validation sets are used as the database for retrieval, while the test set is used as the query set. The similarity between a query and all examples in the database is computed using the cosine similarity between their vector representations. For each query, documents in the database are then ranked according to this similarity, and precision/recall (PR) curves are computed, by comparing the label of the query documents with those of the database documents. Since documents often have multiple labels (specifically those in RCV1-V2), the PR curves for each of its labels are computed individually and then averaged for each query. Finally, we report the global average of these (query-averaged) curves to compare models against each other. Training and model selection is otherwise performed as done in the generative modeling evaluation.

As shown in Figure 9, DeepDocNADE always yields very competitive results, on both datasets, and outperforming the other models in most cases. Specifically, for the 20 Newsgroups dataset, DeepDocNADE with 2 and 3 hidden layers always perform better than the other methods. DeepDocNADE with 1 hidden layer also performs better than the other baselines when retrieving the top few documents (e.g. when recall is smaller than 0.2).

6.3 Qualitative Inspection of Learned Representations

In this Section, we want to assess if the DocNADE approach for topic modeling can capture meaningful semantic properties of texts.

First, one way to explore the semantic properties of trained models is through their learned word embeddings. Each of the columns of the matrix W represents a word w where $W_{:,w}$ is the vector representation of w. Table 2 shows the five nearest words for some chosen words according to their embeddings, from a DocNADE model. We observe for each example the semantic consistency of the word representations. Similar results are observed for DeepDocNADE models.

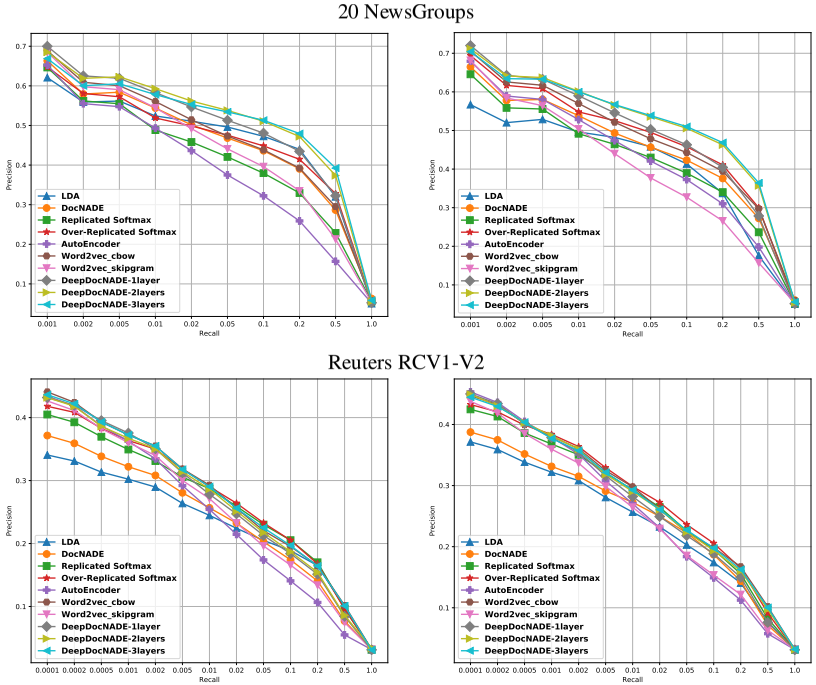


Figure 9: Precision-Recall curves for document retrieval task. On the left are the results using a hidden layer size of 128, while the plots on the right are for a size of 512.

We have also attempted to measure whether the hidden units of the first hidden layer of DocNADE and DeepDocNADE models modeled distinct topics. Understanding the function represented by hidden units in neural networks is not a trivial affair, but we considered the following simple approach. For a given hidden unit, its connections to words are interpreted as the importance of the word for the associated topic. We thus select the words having the strongest positive connections for a hidden unit, i.e. for the i-th hidden unit we chose the words w that have the highest connection values $W_{i,w}$.

With this approach, fifty topics were obtained from a DocNADE model (with the sigmoid activation function) trained on 20 Newsgroups. Four of the fifty topics are shown in Table 3 and can be readily interpreted as topics representing: religion, space, sports and security. Note that those four topics are actual (sub)categories in 20 Newsgroups.

That said, we have had less success understanding the topics extracted when using the $\tanh$ activation function or when using DeepDocNADE. It thus seems that these models learn a latent

Hidden unit topics			
jesus	shuttle	season	encryption
atheism	orbit	players	escrow
christianity	lunar	nhl	pgp
christ	spacecraft	league	crypto
athos	nasa	braves	nsa
atheists	space	playoffs	rutgers
bible	launch	rangers	clipper
christians	saturn	hockey	secure
sin	billion	pitching	encrypted
atheist	satellite	team	keys

Table 3: Illustration of some topics learned by DocNADE. A topic i is visualized by picking the 10 words w with strongest connection W_{iw}.

representation that does not align its dimensions with concepts that are easily interpretable, even though it is clearly capturing well the statistics of documents.

7. Language Modeling Experiments

In this Section, we test whether our proposed approach to incorporating a DocNADE component to a neural language model can improve the performance of a neural language model. Specifically, we considered treating a text corpus as a sequence of documents. We used the APNews dataset, as provided by Mnih and Hinton (2009). Unfortunately, information about the original segmentation into documents of the corpus was not available in the data as provided by Mnih and Hinton (2009), thus we simulated the presence of documents by grouping one or more adjacent sentences, for training and evaluating DocNADE-LM, making sure the generated documents were non-overlapping. This approach still allows us to test whether DocNADE-LM is able to effectively leverage the larger context of words in making its predictions.

Since language models are generative models, the perplexity measured on some held-out texts is widely used for evaluation. Following Mnih and Hinton (2007) and Mnih and Hinton (2009), we use the APNews dataset containing Associated Press news stories from 1995 and 1996. The dataset is again split into training, validation and test sets with respectively 633,143, 43,702 and 44,601 sentences. The vocabulary is composed of 17,964 words.

For these experiments, all the hidden representations of all the models are composed of 100 dimensional features. Early stopping on the validation set is used for avoiding overfitting and for model selection. The perplexity scores of Table 4 are computed on the test set.

The LM model in Table 4 corresponds to a regular neural (feed-forward) network language model. It is equivalent to using only the language model part of DocNADE-LM. These results are meant to measure whether the DocNADE part of DocNADE-LM indeed improves performances.

We also compare against the log-bilinear language (LBL) model of Mnih and Hinton (2007)). While we used a hierarchical softmax to compute the conditional word probabilities for the LM model (see Section 4), the LBL model uses a full softmax output layer that uses the same word representation matrix at the input and output. This latter model is therefore slower to train. Later, Mnih

Models	Number of grouped sentences	Perplexity
KN6	1	123.5
LBL	1	117.0
HLBL	1	112.1
LSTM	1	116.26
GRU	1	109.42
LM	1	119.78
DocNADE-LM	1	111.93
DocNADE-LM	2	110.9
DocNADE-LM	3	109.8
DocNADE-LM	4	109.78
DocNADE-LM	5	109.22

Table 4: Test perplexity per word for models with 100 topics. The results for HLBL and LBL were taken from Mnih and Hinton (2009).

and Hinton (2009) also proposed an adaptive approach to learning a structured softmax layer, thus we also compare against their best approach. All the aforementioned baselines are 6-gram models, taking in consideration the five previous words to predict the next one. We also compare with a more traditional 6-gram model using Kneser-Ney smoothing, taken from Mnih and Hinton (2007). Finally we compare our results against recurrent language model using ether LSTM from Graves (2013) or GRU from Cho et al. (2014). These models are trained with stochastic gradient descent, using full backpropagation through time. They are considered as states of the art for modeling sequences.

From Table 4, we see that adding context to DocNADE-LM, by increasing the size of the multi-sentence segments, improves the performance of the model (compared to LM) and also surpasses the performance of the most competitive alternatives.

7.1 Qualitative Inspection of Learned Representations

In this Section we explore the semantic properties of texts learned by the DocNADE-LM model. Interestingly, we can examine the two different components (DN and LM) separately. Because the DocNADE part and the language modeling part of the model each have their own word matrix, $\mathbf{W}^{\mathrm{DN}}$ and $\mathbf{W}^{\mathrm{LM}}$ respectively, we can compare their contribution through these learned embeddings. As explained in the previous Section, each of the columns of the matrices represents a word w where $\mathbf{W}^{\mathrm{DN}}_{:,w}$ and $\mathbf{W}^{\mathrm{LM}}_{:,w}$ are two different vector representations of the same word w.

We can see from Tables 5 and 6 that the two parts of the DocNADE-LM model have learned different properties of words. An interesting example is the nearest neighbors of the word *israel*, where the DocNADE focuses on the politico-cultural relation between these words, whereas the language model part has learned the concept of countries in general.

weapons	**medical**	**companies**	**define**	**israel**	**book**	**windows**
security	doctor	industry	spoken	israeli	story	yards
humanitarian	health	corp	to	palestinian	novel	rubber
terrorists	medicine	firms	think_of	jerusalem	author	piled
deployment	physicians	products	of	lebanon	joke	fire_department
diplomats	treatment	company	bottom_line	palestinians	writers	shell

Table 5: The five nearest neighbors in the word representation space learned by the DocNADE part of the DocNADE-LM model.

weapons	**medical**	**companies**	**define**	**israel**	**book**	**windows**
systems	special	countries	talk_about	china	film	houses
aircraft	japanese	nations	place	russia	service	room
drugs	bank	states	destroy	cuba	program	vehicle
equipment	media	americans	show	north_korea	movie	restaurant
services	political	parties	over	lebanon	information	car

Table 6: The five nearest neighbors in the word representation space learned by the language model part of the DocNADE-LM model.

8. Conclusion

We have presented models inspired by NADE that can achieve state-of-the-art performances for modeling documents. For topic modeling, DocNADE had competitive results while its deep version, DeepDocNADE, outperformed the current state-of-the-art in generative document modeling, based on a test set perplexity. Good performances were also observed when we used these models as feature extractors to represent documents for the task of information retrieval. As for language modeling, the competitive performances of the DocNADE language model showed that combining contextual information by leveraging the DocNADE architecture can significantly improve the performance of a neural probabilistic N-gram language model.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155, 2003.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5):993–1022, 2003.

Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 288–296, 2009.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

R. Kuhn and R. De Mori. A cache-based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583, june 1990.

Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, pages 2708–2716, 2012.

Hugo Larochelle and Ian Murray. The Neural Autoregressive Distribution Estimator. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, volume 15, pages 29–37, Ft. Lauderdale, USA, 2011. JMLR W&CP.

Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.

Greg Corrado Jeffrey Dean Mikolov, Kai Chen. Efficient Estimation of Word Representations in Vector Space. In *Workshop Track of the 1st International Conference on Learning Representations (ICLR 2013)*, 2013.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

Andriy Mnih and Geoffrey Hinton. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning*, pages 641–648. ACM, 2007.

Andriy Mnih and Geoffrey E Hinton. A Scalable Hierarchical Distributed Language Model. In *Advances in Neural Information Processing Systems 21 (NIPS 2008)*, pages 1081–1088, 2009.

Frederic Morin and Yoshua Bengio. Hierarchical Probabilistic Neural Network Language Model. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS 2005)*, pages 246–252. Society for Artificial Intelligence and Statistics, 2005.

Ruslan Salakhutdinov and Geoffrey Hinton. Replicated Softmax: an Undirected Topic Model. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 1607–1614, 2009.

Nitish Srivastava, Ruslan R Salakhutdinov, and Geoffrey E Hinton. Modeling documents with deep boltzmann machines. *arXiv preprint arXiv:1309.6865*, 2013.

Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

Benigno Uria, Iain Murray, and Hugo Larochelle. A deep and tractable density estimator. *JMLR: W&CP*, 32(1):467–475, 2014.

Yin Zheng, Yu-Jin Zhang, and Hugo Larochelle. A deep and autoregressive approach for topic modeling of multimodal data. *TPAMI*, 2015.

Journal of Machine Learning Research 18 (2017) 1-26 Submitted 1/17; Revised 5/17; Published 11/17

Target Curricula via Selection of Minimum Feature Sets: a Case Study in Boolean Networks

Shannon Fenn shannon.fenn@newcastle.edu.au

Pablo Moscato pablo.moscato@newcastle.edu.au

School of Electrical Engineering and Computing
University of Newcastle
University Drive
Callaghan NSW 2308
Australia

Editor: Amos Storkey

Abstract

We consider the effect of introducing a curriculum of targets when training Boolean models on supervised Multi Label Classification (MLC) problems. In particular, we consider how to order targets in the absence of prior knowledge, and how such a curriculum may be enforced when using meta-heuristics to train discrete non-linear models.

We show that hierarchical dependencies between targets can be exploited by enforcing an appropriate curriculum using hierarchical loss functions. On several multi-output circuit-inference problems with known target difficulties, Feedforward Boolean Networks (FBNs) trained with such a loss function achieve significantly lower out-of-sample error, up to 10% in some cases. This improvement increases as the loss places more emphasis on target order and is strongly correlated with an easy-to-hard curricula. We also demonstrate the same improvements on three real-world models and two Gene Regulatory Network (GRN) inference problems.

We posit a simple a-priori method for identifying an appropriate target order and estimating the strength of target relationships in Boolean MLCs. These methods use *intrinsic dimension* as a proxy for target difficulty, which is estimated using optimal solutions to a combinatorial optimisation problem known as the Minimum-Feature-Set (minFS) problem. We also demonstrate that the same generalisation gains can be achieved without providing any knowledge of target difficulty.

Keywords: Multi-Label Classification, Target Curriculum, Boolean Networks, k-Feature Set

1. Introduction

When students are first taught the concept of addition, they are not simply handed an eclectic set of many digit number pairs and their summations; rather, we provide them a learning curve of carefully chosen examples which increase in difficulty (i.e. number of digits). From standard classroom practices we can glean two key sources of inspiration regarding effective learning and teaching processes of humans: the curriculum of examples, and the curriculum of targets. This paper examines the latter.

Khan et al. (2011) found that humans naturally follow an easy-to-hard example-based curriculum when teaching a single-target concept to an unknown learner. Methods of scheduling examples this way have been explored in the machine learning community for some time, with positive results for non-convex scenarios (Bengio et al., 2009).

The typical explanation for the success of example curricula in non-convex scenarios is that it imposes a form of transfer learning where the simpler concept is discovered first and subsequently informs the more complex concept (Bengio et al., 2009). This leads neatly into a consideration of the second inspiration from the classroom: target curricula.

We consider the problem of learning discrete models for MLC problems with Boolean inputs, where closed-form solutions and gradient-guided optimisation techniques are unavailable. We use Feedforward Boolean Networks (FBNs)—defined in Section 1.2—as a case study for implementing target-curriculum regularisation with simple modifications to the typical L_1/L_2-Loss. We also introduce methods for estimating target overlap and target difficulty order in Boolean MLCs by finding a lower bound to the intrinsic dimension of the induced function. The intrinsic dimension has been presented as a proxy for the overall complexity of arbitrary data sets (Granata and Carnevale, 2016) suggesting its promise as a metric for ordering targets.

1.1 Contributions

We provide three novel contributions:

- we show that using hierarchical loss functions improves generalisation performance of FBNs trained on several circuit-inference problems possessing a natural target progression,
- we describe a simple method for estimating the complexity and overlap of Boolean targets by lower bounding their intrinsic dimensions using solutions to the minFS problem, and
- we demonstrate the success of the combined method on the same problem instances when knowledge of the target difficulty is withheld.

In the remainder of this section we define FBNs as a learning model, give background on MLC and outline existing work on sample and target curricula. In Section 2 we define hierarchical loss functions as well as the minFS problem and how it can be used to detect target overlap and estimate an appropriate curriculum. Finally, in Section 3 we present results on several problems.

1.2 Boolean Networks

This paper focuses on Boolean MLC problems where the targets possess differing complexities. In this section we provide definitions of the problem domain, its relevance and the representation of interest to this work: the Feedforward Boolean Network.

A Boolean MLC problem is one where inputs and targets take values in $\mathcal{B} \in \{0, 1\}$. An intuitive model for such a problem is a digital circuit, or FBN. A FBN with l inputs and m outputs computes a function of the form $f : \mathcal{B}^l \to \mathcal{B}^m$ by chaining computations from internal nodes (commonly called gates).

©2017 Shannon Fenn and Pablo Moscato.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-007.html.

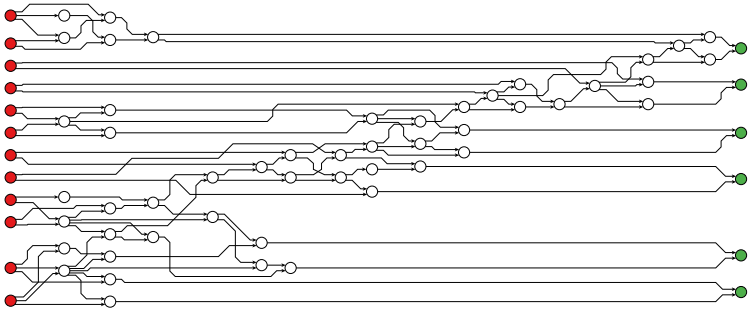


Figure 1: A 56-node FBN resulting from our learning procedure, which correctly implements the 6-bit addition function. Each node takes 2 inputs and computes the NAND function as its output. Inputs (far left) have been coloured red and outputs (far right) green. Note the ripple-carry style flow of information between sub-networks.

Internally, a FBN is a directed acyclic graph (DAG) where each node has an associated value in $\mathcal{B}$. Nodes take the values of their predecessors as input to a Boolean function which they provide as output. Input nodes are those with no predecessors and their value is provided from outside the network. The output of the network is simply the values at a particular subset of externally visible output nodes. A FBN can thus compute a function from many inputs to many outputs by combining simpler functional elements. Figure 1 shows an example FBN which correctly implements a 6-bit adder.

One advantage to working with this representation is that two singleton functionally complete operator sets exist: the negated-and (NAND) and negated-or (NOR) functions. Every Boolean mapping can be computed by some degree-2 DAG consisting purely of NAND nodes. We will use this representation for the remainder of the paper.

FBNs have a number of other advantages as learning models. They are efficient in time and memory to simulate and trivial for direct hardware implementation and logical rule conversion. Logical relationships are simple for humans to extract information from which is important if the goal of the learning procedure is knowledge-generation. While other representations have received significantly more attention, FBNs are trivial to map to existing digital hardware—an open problem for neural nets. They can use packed integer representations and bitwise arithmetic to improve model evaluation speed during training and deployment. Recent literature presents methods for binarising the internal layers of deep neural nets precisely for these reasons (Courbariaux et al., 2016).

As the name suggests, FBNs compute with purely Boolean logic, rather than using weighted sums and continuous activations. This brings challenges, the most prevalent being the lack of gradient-based optimisation procedures or closed form solutions. All optimisation techniques discussed below use general purpose combinatorial meta-heuristics, such as Simulated Annealing (SA). Herein all networks were optimised using local search in the space of graph structures—possible due to the functional completeness of the NAND operator.

1.3 Learning with Boolean Networks

Since the late publication of Turing's first description of a machine which learns by changing the connections of a NAND-gate circuit (Turing, 1948) the majority of the work on learning with a Boolean Network model has been done in the field of physics. Some time later, Patarnello and Carnevali (1987), and subsequently others (Van den Broeck and Kawai, 1990; Goudarzi et al., 2014), demonstrated that FBNs could generalise well on some problems, despite training set sizes significantly smaller than the space of all possible patterns.

Learning FBNs is a *combinatorial* optimisation problem, typically solved using meta-heuristics such as SA or genetic algorithms. Patarnello and Carnevali (1987) optimised networks by SA in the space of feedforward structures. Each move in the search procedure consisted of a random change in a single connection. Moves which also changed the node activation have been considered (Van den Broeck and Kawai, 1990; Goudarzi et al., 2014), however in light of the NAND function's functional completeness, this becomes unnecessary.

To simplify the following loss function definitions we define the $n \times m$ error matrix E, which fully characterises the mistakes made by a candidate network on a set of n example patterns with m targets. E is the element-wise absolute difference between the network output matrix Y' and the target matrix Y, with elements given by

$$E_{i,j} = \left| Y_{i,j} - Y'_{i,j} \right|,$$

where $E_{i,j} \in \mathcal{B}$ is the error for the jth target for the ith example.

Even for problems with multiple outputs, the only loss functions used to date (Patarnello and Carnevali, 1987; Teuscher et al., 2007; Goudarzi et al., 2014) have been analogues of the L_1-loss:[1]

$$L_1 = \frac{1}{m\,|\mathcal{I}|} \sum_{j=1}^{m} \sum_{i \in \mathcal{I}} E_{i,j},$$

where $\mathcal{I}$ is the set of example indices shown to the network. This loss treats all examples and targets equally and is the natural first choice for a guiding function, but for multi-target problems there are other options which we explore in Section 2.1.

So far no work has considered differences between learning single and multiple output FBNs. Results for Multi Task Learning (MTL) suggest that, with multiple related target, some networks may be favoured due to target correlation, increasing generalisation performance. To explore the possibility of further improvements, we use FBNs as a case study in the effects of target curriculum enforcement by hierarchical loss-functions on purely discrete models. In the following section we outline the relevant work on target curricula.

1.4 Example Curricula

The approach we present in this work draws strong inspiration from *example* curricula. In this subsection we give a brief outline of some relevant work to better frame our approach.

Curriculum learning by applying a ranking of *examples* can be split into two main camps: Curriculum Learning (CL) and Self-Paced Learning (SPL) (Kumar et al., 2010). In the former the ranking is determined a priori while in the latter it is learned jointly

1. With purely binary predictors and targets the L_1- and L_2-loss are identical.

with the model. Jiang et al. (2015) demonstrated a combined approach—intuitively called Self-Paced Curriculum Learning—where constraints on the example-ranking are imposed by a fixed curriculum and the ranking is learned jointly with the model but subject to the curriculum constraints. SPL and CL emerge as special cases of their framework. They present positive results in comparison to baseline methods on matrix factorization and multimedia event detection problems.

Spitkovsky et al. (2009) demonstrated the success of a curriculum of samples of *increasing complexity* on the problem of determining grammar from free-form sentences. Their proxy for complexity in this case is the length of the example sentence. They also noted the appearance of a "sweet-spot" in sentence length, the inclusion of samples above which reduced performance. The combination of both their curriculum and complexity limitation improved on the state-of-the-art.

It is important to note that the terms Curriculum Learning (CL) and Self-Paced Learning (SPL) used within this subsection related *example* curricula. It is possible to also consider a learner—in a MLC framework—to be following a curriculum of *targets* (or to be self-paced in the same respect). We primarily cover the former case of a fixed curriculum of targets, and throughout this paper "curriculum" will refer to a curriculum of targets.

1.5 Target Curricula

Learning multiple targets concurrently with a shared representation has been shown, theoretically and experimentally, to improve generalisation results when there are relationships between the targets (Caruana, 1997; Wang et al., 2016). The number of publications discussing multi-label and multi-task learning is growing rapidly and using a *curriculum* of targets, while less general, is also beginning to garner attention.

The importance of sharing knowledge between targets increases when there is a disparity in target complexity. Gülçehre and Bengio (2016) showed a visual task, on which many popular methods—including deep neural nets, SVMs and decision trees—failed. This same task became solvable when a simpler hinting target was provided (in much the same way as introduced by Abu-Mostafa, 1990). However, unless relative target difficulties are known (or suspected), we also face the issue of determining an appropriate order during learning.

Pentina et al. (2015) presented a self-paced method using adaptive linear SVMs. The implementation was model-specific, providing information transfer by using the optimal weight vector for the most recently learned task to regularise the next task's weight vector. They discover the curricula by solving all remaining tasks and selecting the best fitting, resulting in a quadratic increase in training time over solving tasks individually. While this may be acceptable for a linear model, it would be prohibitive for a FBN.

Li et al. (2017) implemented task curricula with linear models by weighting the contribution of each task-example pair to the loss function. They used a regulariser in which task-example pairs with low training error receive larger values for these weights. However, this approach does not appear applicable to inferring highly non-linear discrete models.

Lad et al. (2009) present the only model-agnostic approach to determine an optimal order of tasks that we are aware of. They use conditional entropy to produce a pairwise order matrix and solve the resulting NP-Hard Linear Ordering Problem (see Ceberio et al., 2015, for a definition). Their approach is not applicable to Boolean MLCs since the conditional

entropy difference calculation they use reduces to the difference of the entropies of the individual targets; which, for balanced binary targets, rapidly approaches zero as more training samples are given.

The impact of target order is highlighted by Read et al. (2011). Their approach uses *classifier chains*: independent learners for each binary label, trained in an arbitrary order, with the output of all prior classifiers included as input to each subsequent classifier. The authors note that the ordering of targets has a notable effect on classifier performance; which they address using ensembles of chains, each with a different random order. They note consistent superiority of the ensemble approach over a single random ordering, which suggests non-trivial variability in the impact of different orderings. Due to the large space of permutations, a principled approach to determine appropriate orderings a priori would be of value.

We have not seen work using hierarchical loss-functions (or similar regularisers) to impose a target curriculum, nor have we seen methods for determining a curriculum by use of feature selection methods. The following section describes our approach.

2. Learning Targets in Order

Here we describe three loss functions which we used to enforce a curriculum of targets, as well as a novel method for obtaining said curriculum which reduces the problem to m instances of the Minimum-Feature-Set (minFS) problem.

2.1 Hierarchical Loss Functions

Assume a two-target Boolean MLC for which we have three candidate learners, and assume we know that the first target is "easier" under some definition than the second. The first learner labels target 1 correctly and target 2 incorrectly for all examples. The second learner does the opposite, and the third learner labels 50% of the examples incorrectly for both targets. All three networks will have an equal $L1$-loss.

The rationale behind the loss functions we are about to define is that the first of the above learners should be preferred. By focusing on easier targets earlier in the training process the network is more likely to find substructures useful to later targets. The expectation is faster convergence speed, better generalisation or both.

The L_1-loss is simply the mean of the error matrix. In addition, we consider three loss functions which hierarchically aggregate the error matrix elements in progressively more strict ways:

- a linearly weighted mean: L_w,
- a locally hierarchical mean: L_{lh}, and
- a globally hierarchical mean: L_{gh}.

All definitions below assume the targets are ordered by difficulty, with the simplest target at the lowest index.

The first function, L_w, encourages the target curriculum by weighting the easier targets more highly. The weighting is linear and the resulting summation is normalised to the unit

interval:

$$L_w = \frac{2}{m\,(m-1)\,|\mathcal{I}|} \sum_{i \in \mathcal{I}} \sum_{j=1}^{m} (m-j+1)\, E_{i,j} \ .$$

We named the second function, L_{lh}, the *locally hierarchical* loss, as it enforces a hierarchy of targets on a per-example basis. The learner is rewarded for correctly labelling target j of example i if and only if it has also correctly labelled all prior targets $\{1, 2, ..., j-1\}$ *for that example*. To define this function, for each example i, we define the recurrence:

$$a_{i,1} = E_{i,1}$$
$$a_{i,k+1} = \begin{cases} E_{i,k+1}, & \text{if } a_{i,k} = 0 \\ 1, & \text{if } a_{i,k} = 1 \end{cases} ,$$

from which we can define

$$L_{lh} = \frac{1}{m\,|\mathcal{I}|} \sum_{i \in \mathcal{I}} \sum_{k=1}^{m} a_{i,k} \ .$$

The final loss, L_{gh}, follows the same principle as the former but across all examples. For this reason we named it the *globally hierarchical* loss. In this case the learner is given reward for correctly labelling target j (of any example) if and only if it has also correctly labelled all prior targets $\{1, 2, ..., j-1\}$ for all examples given. This function is effectively a "soft" equivalent to learning the targets rigidly in order. We can define L_{gh} using the mean per-target errors (the row-means of E):

$$\delta_j = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} E_{i,j} \ .$$

These can be used to define another recurrence,

$$b_0 = \delta_0$$
$$b_{k+1} = \begin{cases} \delta_{k+1}, & \text{if } b_k = 0 \\ 1, & \text{if } b_k > 0 \end{cases} ,$$

giving

$$L_{gh} = \frac{1}{m} \sum_{j=1}^{m} b_j \ .$$

Unless the difficulty order is known, or suspected, from domain specific knowledge we also require a method for estimating the target order from the training examples. We describe a method suitable for Boolean or categorical MLCs in the following section.

2.2 Measuring Target Overlap and Difficulty Order

The key requirement for ordering targets is an estimation of their relative difficulty. It is likely most suitable to use domain knowledge where available, however in cases where this is impossible or difficult to obtain we suggest a rule of thumb: order by increasing intrinsic dimension. We present an intuitive approach below using a combinatorial problem which

did not significantly increase the learning time for the instances considered. In Section 3.4 we show that the target order estimated by this method gives generalisation improvements close to those obtained using the known order (Section 3.3).

Our key assumption is that a target can be expected to be simpler than another if its intrinsic dimension (the number of input variables on which it truly depends) is smaller. This assumption is strong but there is also a strong basis for it as a rule-of-thumb in the Boolean domain, as a function's intrinsic dimension places an upper bound on the size of the smallest FBN which implements it. Furthermore, intrinsic dimension is showing promise as a method for estimating the complexity of arbitrary data (Granata and Carnevale, 2016) and as such we expect the validity of this idea may extend beyond the Boolean case.

The likelihood of effective information transfer can also be estimated by measuring the overlap in intrinsic input features. Thus estimating both target overlap, and relative complexity, is reduced to the problem of determining the minimum set of input features on which a valid function can be defined; that is, where no pattern is mapped to multiple target values. This is well known as the Minimum-Feature-Set (minFS) problem.

2.2.1 The Minimum-Feature-Set Problem

The smallest intrinsic dimension of any Boolean function which is in full agreement with given set of examples can be found using the Minimum-Feature-Set (minFS) problem. It asks: what is the smallest subset of given input features, on which a single target can still form a non-contradictory mapping? This problem, and particularly this view of it, is not in any way new to the machine learning community (Davies and Russell, 1994). However, to our knowledge, it has not been used to estimate relative target difficulty before.

This equivalent decision problem is known as the *k-Feature-Set (kFS)* problem. Formally:

k-Feature-Set

Given: A binary $n \times p$ matrix, $\mathcal{M}$, where the rows describe n examples and the columns describe p input features as well as n-element target vector, $\mathcal{T}$, and a positive integer k.

Question: $\exists S \subseteq [1, ..., p], |S| \leq k$, such that, $\forall i, j \in [1, ..., n]$ where $\mathcal{T}_i \neq \mathcal{T}_j$, $\exists l \in S$ such that $\mathcal{M}_{i,l} \neq \mathcal{M}_{j,l}$?

That is: is there a cardinality k subset, S, of the input features, such that no pair of examples which have identical values across all features in S have a different value for the target feature?

The problem is NP-complete (Davies and Russell, 1994) and assumed to not be fixed-parameter tractable, when parametrised by k, under current complexity assumptions (Cotta and Moscato, 2003). Nonetheless the exact solver used herein solved all instances in significantly less time than was required for subsequently optimising the FBN. At least one fast meta-heuristic solver exists for a generalised version of the problem, with reasonable run-times for very large instances (Rocha de Paula et al., 2015), and problem instances can be reformulated as instances of the well known Set Cover Problem, for which there are numerous powerful heuristic solvers.

We implemented instances of the minFS problem below as Mixed-Integer Programs and solved using them using the Python interface to the IBM CPLEX solver. With a solution

procedure in hand we can now define a simple method for estimating the target difficulty order and the level of target overlap.

2.2.2 Target ordering method

Given a procedure for solving the minFS problem our proposed method for ordering the targets reduces to solving m such instances; one for each target. For all instances the input feature matrix $\mathcal{M}$ is identical to the entire input matrix for the original learning problem. Each instance is solved to optimality resulting in a set of features $S_1, ..., S_m$ for each target.

An estimate of the intrinsic dimension for a target i is then simply $|S_i|$. Targets are sorted by this estimate with ties broken randomly, and the resulting total order is used in the loss functions defined in Section 2.1. These losses could be redefined to treat a subset of targets equally to avoid tie-breaking but we did not explore this possibility.

2.2.3 Target overlap score

Estimating target overlap also follows from the minFS solutions. For two targets, y_i and y_j, with solutions, S_i and S_j, the target overlap is estimated by the Szymkiewicz-Simpson overlap coefficient:

$$\sigma(i,j) = \frac{|S_i \cap S_j|}{min(|S_i|, |S_j|)} \ .$$

The overlap coefficient gives a value between 0 and 1 where 0 indicates that there is zero overlap between two sets and 1 indicates that one set is a subset of the other; values in between occur when sets are partially overlapping. This measure was chosen over the more popular Jaccard index as we wish to consider all fully overlapping cases ($A \subseteq B$) equivalent regardless of their relative sizes.

We can then define the *nestedness* score, η, as the mean overlap coefficient between successive targets (when ordered by the method in Section 2.2.2) giving the formula

$$\eta = \frac{1}{m-1} \sum_{i=2}^{m} \sigma(i, i-1) \ ,$$

which takes the value 1 when the minFS solutions for all targets forms a perfectly nested sequence of sets, and the value 0 when the sets have no pairwise overlap. A high value for η is thus a necessary but not sufficient condition for strongly inter-dependent Boolean targets. A value near zero indicates that there is unlikely to be any benefit gained from a target curriculum.

The nestedness score allows us to rule out some Boolean MLC problems for which a target curriculum would be inappropriate, and the proposed target ordering method gives us a curriculum otherwise. We show experimental results in the following section.

3. Experimental Results

It is important to see how the loss functions defined in Section 2.1 affect generalisation performance over a range of training set sizes. When optimising L_1, generalisation performance for all problems considered transitions from essentially random behaviour to near perfect

prediction as training set size increases. With this in mind we generated 1000 samples each for a range of training set sizes chosen to span that transition by a reasonable margin. We report the training set size as the fraction, $s = |\mathcal{I}| / 2^l$, of all possible patterns.

For each sample we trained one network for each loss function until they achieved zero training error. This allows us to report the difference in test error between the network optimised under L_1 and each of the remaining losses, for the same training set. Using the same training set for each loss removes the significant variation in instance difficulty as a confounding factor.

3.1 Data Sets

Here we define the problems considered. We first outline problems for which we already have a difficulty curricula. Then we consider several published models of real-world phenomena and finally we test our overall approach on the problem of discovering regulatory network update rules from discretised time-series data.[2]

3.1.1 Circuit inference test-beds

For the first test-bed we chose five circuit-inference problems: binary addition (add) and subtraction (sub) and cascaded variants of three typical circuit-learning and genetic programming problems: multiplexer (cmux), majority (cmaj) and parity (cpar). These were chosen as they all possess a natural hierarchy of targets.

Binary addition and subtraction are well known multi-output problems. For the three remaining problems, multi-output versions are constructed by cascading single-output subcircuits; each successive output is the result of applying the original function over a larger set of inputs. By this method we construct problems for which we know there is a difficulty order, and what that order is. More detailed definitions of these problems are given in Online Appendix 1.

The cascaded parity and multiplexer possess the strongest dependencies between successive targets, as each subsequent output can be computed directly from the previous output and an additional subset of inputs not already used. The cascaded majority, binary addition and binary subtraction problems have weaker dependencies, requiring extra inputs as well as the previous output and new inputs to compute the next output. This allows us to see the effect of imposing target curricula of problems with varying levels of inter-target dependency.

FBNs are a high variance learner. As such we selected problem sizes so that a meaningful effect size can be seen, but also so that 1000 instances per loss/sample-size combination could be completed in under a week. Table 1 outlines the dimensions selected, as well as the ranges in training sample size.

The hierarchy of targets is known for the above problems since they are constructed from cascading well-known circuits. This ground-truth allows us to further evaluate the methods outlined in Section 2.2 by considering the correlation between this known order and the predicted order.

2. Data and code are available at `github.com/shannonfenn/Multi-Label-Curricula-via-Minimum-Feature-Selection`.

Problem	Inputs	Targets	Training set size	Example pool size
Cascaded Parity	7	7	[8, 112]	128
Cascaded Majority	9	5	[8, 384]	512
Binary Subtraction	10	5	[8, 192]	1,024
Binary Addition	12	6	[8, 208]	4,096
Cascaded Multiplexer	15	7	[8, 180]	32,768
74181 ALU	14	8	[64, 384]	16,384
FA/BRCA pathway	28	28	[8, 96]	1,024
Mammalian cell cycle	10	9	[8, 180]	1,024
Yeast	10	9	8	9
E. Coli	10	8	4	5

Table 1: Instance sizes of the test-bed problems.

3.1.2 Boolean Models of Real Phenomena

In addition to the problems above we also conducted similar experiments using real-world models. This included a commercial integrated circuit: the 74181 4-bit Arithmetic Logic Unit (ALU), and two published biological models.

The 74181 ALU has 14 inputs and 8 outputs. It performs 32 different arithmetic and logical operations on an 8-bit input with carry in, generating a 4-bit output with carry out as well as 2 other carry-related outputs useful for faster calculations when chaining multiple ALUs. The 74181 represents a reverse engineering problem of a realistic size that is less idealised than the above problems, as there is no expected absolute ordering to the targets.

The biological models we considered were Boolean models of the Fanconi Anemia/Breast Cancer (FA/BRCA) pathway and the mammalian cell cycle (Poret and Boissel, 2014). Boolean regulatory network models like these describe the time-evolution of gene activity, levels of regulatory products and protein activity with a single Boolean variable per element. The system dynamics are then defined with Boolean update functions and an update method (synchronous, partly or fully asynchronous). Both models we consider are synchronous as this enables the inference of a single deterministic update equation by treating successive time steps as input-output pairs. More detail on all models in this section are given in Online Appendix 1.

For these three problems we ran experiments just as in Section 3.1, with the exception of the ALU problem for which we only trained 100 networks per training set size due to time constraints. Similarly, due to the large memory requirement of generating all 2^{28} patterns for the FA/BRCA model, we used a reduced example pool of 2^{10} patterns sampled uniformly. The problem dimensions, training set sizes and example pool sizes are given in Table 1. We have no suspected optimal target order for these problems, so the ordering used for all ordered losses was found using the minFS-based intrinsic dimension estimation described in Section 2.2.

3.1.3 Learning Regulatory Networks From Time-Series Data

While conducting the above experiments we observed that the primary performance gain is seen at smaller sample sizes. For this reason we chose a real-world test-bed where a key issue is the extremely limited amount of available data: inferring Boolean GRN update models from time-series gene expression data. Barman and Kwon (2017) provide binarised time-series data for an *E. coli* GRN and a cell-cycle regulatory network of fission yeast: each consisting of 10 nodes and only a handful of examples. One desirable result is a synchronous update model such as in the models described in Section 3.1.2, and one way to achieve this is to treat each sequential pair of states as an input-output pair thus deriving a FBN which predicts the next state from the current state. Again, further detail is given in Online Appendix 1.

Some preprocessing of the data were required. First we removed repeat patterns—these occur due to time disparities which we are not attempting to model. Instead we require pairs of distinct patterns as otherwise the requirements for a valid minFS instance are not met. Secondly we removed constant targets since they represent nodes for which there is no information suggesting a relationship to any other nodes: one such target was removed from the yeast data, and two from the *E. coli* data.

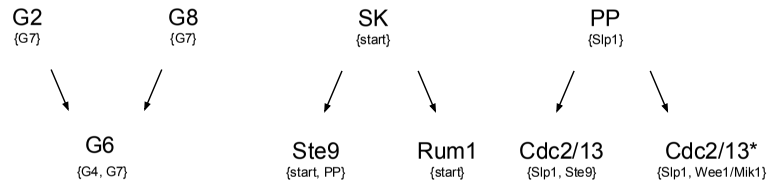
Finally, since we have very small example sets allowing rapid training, we use leave-one-out cross validation for all patterns whose removal does not introduce additional constant targets (the first was the only such pattern in both cases). For each fold we learned 1000 networks for each configuration of interest, to account for the significant variance typical in such heavily under-determined problems.

Our initial tests including all targets of the yeast and *E. coli* data sets as a single problem showed little improvement from the ordered losses. Preliminary feature set analysis suggested that there were distinct, shallow hierarchies among the targets (see Figure 2), rather than the deeper hierarchies seen in prior problems. A lack of improvement is unsurprising in this case since imposing an order upon unrelated targets is not expected to offer any benefit and to potentially detriment the performance.

Due to this we opted to learn networks only for the nodes involved in a hierarchy (2 for yeast and 1 for *E. coli*). For fair comparison we compared with the baseline L_1 network on both target sets: only those in the suspected hierarchy and all targets. Results are only shown for the former however, since the results of learning a single overall network using L_1 were notably worse. Again we used estimated target curricula for all ordered losses (see Section 2.2).

3.2 The Learning Procedure

Here we describe the local search procedure used for training. We learned each function by optimising FBNs comprised of only NAND gates using stochastic local search. As the NAND operator is functionally complete the optimisation process does not need to consider node transfer functions and can instead learn any mapping by modifying just the network structure. We set network size, n_g, for all problems at $21m$ and represented the structure using a $2 \times n_g$ integer matrix, with column i containing the indices of the two sources for the ith node.



(a) E. Coli (b) Yeast

Figure 2: Possible target hierarchies found using minFS in the yeast and *E. coli* data sets. In the *E. coli* data (a) we see the target G6 (with feature set $\{G4, G7\}$) may benefit from being learned after G2 or G8 (both have the feature set $\{G7\}$). In the yeast data (b) we see two potential hierarchies: either of Ste9 or Rum1 (both with feature set $\{start, PP\}$) may benefit from being learned after SK, and either of Cdc2/Cdc13 or Cdc2/Cdc13* (with feature sets $\{Ste9, Slp1\}$ and $\{Slp1, Wee1/Mik1\}$ respectively) may benefit from first learning PP.

We ensure the network remains feedforward by imposing a topological ordering of nodes; each node may only accept input from nodes before it in the ordering. Since any DAG has at least one topological ordering, no structure is excluded by this constraint.

Training involved performing stochastic local search in the space of valid feed-forward structure, using the Late-Acceptance Hill Climbing (LAHC) meta-heuristic (Burke and Bykov, 2008) with random restarts. This method assumes a black-box view of the learner and is similar in implementation to SA while being less sensitive to scaling in the guiding function and requiring only a single meta-parameter in place of a cooling schedule. While the particulars of the optimiser are not of relevance to this work, a brief description is given below and the pseudo-code is given in Online Appendix 1.

LAHC works similarly to typical stochastic hill climbing: a modification is made to the current solution and accepted based on some criterion. This modification involves a random change to the source of one connection in the DAG. A modification is accepted if the resulting network has equal or lower loss than the solution obtained L iterations beforehand.

The single meta-parameter, L, is the length of a history of costs which provides a dynamic error bound. In preliminary tests, the out-of-sample error was consistent over a wide range in L and so we selected the value which provided the best average convergence speed when using the $L1$-loss. The resulting value was 250 for all problems, except parity, for which it was 1000.

For MLCs with discrete inputs there are only finitely many possible patterns. Since our test-bed problems are fully defined, we construct each test set from all possible patterns except those used in that particular training set. In doing so the true out-of-sample error for each sample is known.

3.3 Results for Hierarchical Loss Functions

In this section we present results showing that the losses defined in Section 2.1 improve the out-of-sample error achieved on all test-bed problems mentioned in Section 3.1.

Learning FBNs with structure-based local search displays high variance. As such, each point shown in Figures 3, 4, and 5 represents the mean performance over 1000 different training sets of that size. We have also displayed 95% confidence intervals as transparent bands in Figure 4. For Figure 3 this interval was too small for bars or bands to be visible.

The overall test accuracy as it varies with sample size for the 7-bit cascaded parity problem is shown in the top of Figure 3. A phase transition between poor and near perfect generalisation is expected (Patarnello and Carnevali, 1987). What we see in Figure 3 is a leftward shift in this transition as we impose an increasingly strict order to the targets. Less examples are required to achieve the same generalisation if a difficulty-based target curriculum is imposed.

Even for a single problem instance the improvement is not expected to be consistent across all targets. In fact our earlier reasoning suggests the improvement should increase for successive targets. In the bottom of Figure 3 we have also shown the test accuracy improvement given by each loss, for each target of 7-bit cascaded parity. The first two targets are not shown; as expected there is little improvement for them. Beyond this, we see that the improvement increases with each successive target until the last (the same trend appears in all five problems). This peaks at the second-last target with an increase from 58% test accuracy using L_1, to 84% test accuracy using L_{gh}.

Generalisation improvements can be most readily observed when looking at the test accuracy differences between L_1 and each of L_w, L_{lh} and L_{gh} as they vary with s. Figure 4 shows the mean of these differences with respect to s along with transparent bands indicating the 95% confidence interval of the mean. We can see that the improvement differs significantly between targets but is consistently positive and statistically significant across a range of training set sizes for all five problem instances.

More importantly, the loss which most strictly enforces the learning order: L_{gh}, also confers the largest improvement, and vice versa for the loss which least strictly enforces it: L_w. This phenomena is consistent across all five problems and confirms the central idea presented in this paper.

In terms of overall generalisation accuracy we can see that training to a curriculum of targets offers a significant advantage in Boolean MLCs with target hierarchies. Furthermore we can see that a curriculum can be enforced in a combinatorial black-box optimiser using only slight modifications to the typical guiding function.

3.4 Evaluating the Discovered Curricula

Here we evaluate the effectiveness of the proposed methods for estimating target overlap and difficulty order described in Section 2 on the same test-bed problems. We also observe how the improvement conferred by hierarchical loss functions varies when random orderings are used.

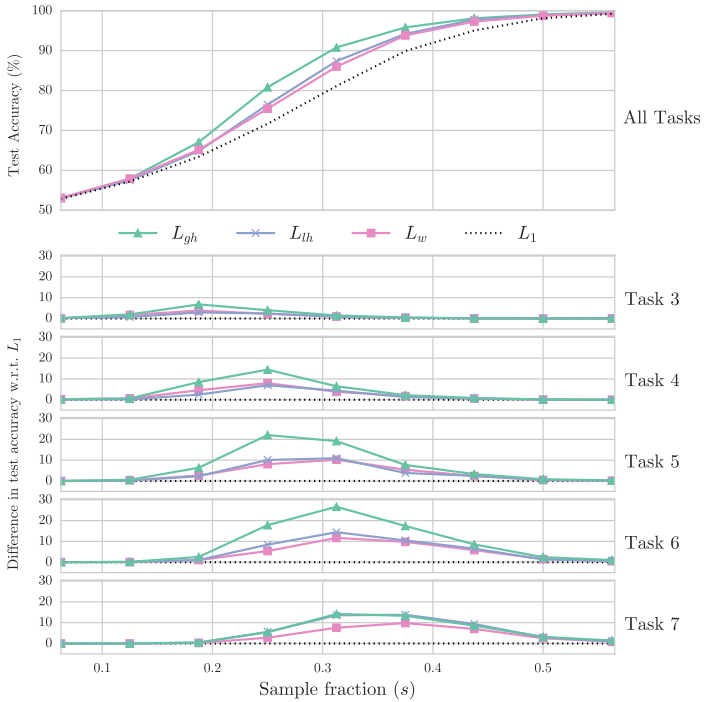


Figure 3: Generalisation performance of each loss function on the 7-bit cascaded parity problem. The top plot shows the mean test accuracy averaged across all targets for each loss while the remaining plots show the mean difference in test accuracy, on each target, between L_1 (dotted baseline) and each of L_w, L_{lh} and L_{gh}. The first two targets are not shown as they show only a negligible improvement. Confidence intervals (95%) are also not shown as they were too narrow to be visible, the widest being less than 2%. A leftward shift in the transition (top) from poor to near-perfect performance can be seen. This becomes more pronounced as the loss more strictly enforces the learning order ($L_w \to L_{lh} \to L_{gh}$) with L_{gh} achieving a 16% reduction in the number of training examples needed to achieve above 90% mean test accuracy. The differences in test accuracy for individual targets (bottom) shows an increase in improvement with target difficulty, as expected; reaching a large peak improvement—from 58% to 84% test accuracy—for the sixth target. On this particular problem the difference between L_{lh} and L_w is not statistically significant. For all loss functions the out-of-sample accuracy is significantly better than obtained with L_1 (dotted line).

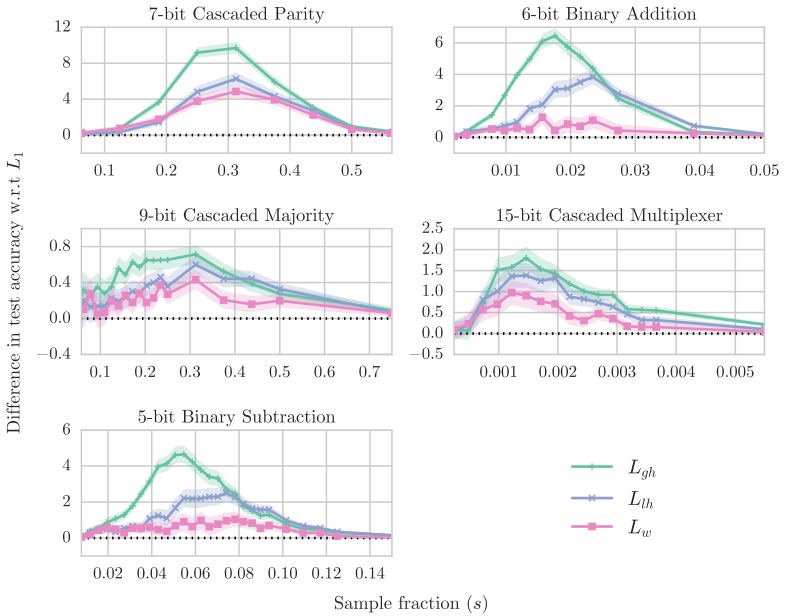


Figure 4: Mean difference in test accuracy between L_1 (dotted baseline) and each of L_w, L_{lh} and L_{gh} on each training sample and problem instance. The 95% confidence interval of the mean is shown with transparent bands. On all problems we can see a statistically significant net increase in test accuracy—though the magnitude varies widely between problems. We can also see that this improvement increases as the loss more strictly enforces the ordering ($L_w \to L_{lh} \to L_{gh}$).

3.4.1 Effectiveness of order estimation

We know the difficulty curricula for the problems in Section 3.1.1. Thus we can quantify the effectiveness of the proposed target ordering method on these, by using the rank correlation between the known and estimated orderings. Since these are permutations there are no ties to consider and we can use the simplest version of Kendall's τ:

$$\tau = \frac{P - Q}{P + Q} ,$$

where P is the number of target pairs which are placed in the same relative order under both orderings, and Q the number which are not. Being a correlation, τ takes values in $[-1, 1]$, with 1 indicating the two orderings are identical, 0 indicating they are uncorrelated and -1 that they are inverses.

We used the same experimental configuration as before but for each training instance we randomly shuffled the targets, to reduce the effect of any deterministic biases. Then we estimated an ordering using the minFS-based method and computed the rank correlation between this ordering and the order used in Section 3.3, as well as the nestedness score, η, defined in Section 2.2.3. Finally we learned one network for each instance using L_{gh} with the estimated target order. Only L_{gh} was considered since it produced the largest improvement with the known order (see Figure 4). These results are shown in Figure 5.

For each problem, Figure 5 shows the nestedness score and the rank correlation between the known and automatically discovered target orders; as well as the generalisation improvement imparted by L_{gh} when given each ordering.

We see that not all problems are easily detected as possessing overlapping hierarchical targets. On parity and majority the mean nestedness score and τ quickly reach 1.0 indicating that the proposed method is successful at automatically determining that there is an overlap, and the correct order of targets. For addition and subtraction the results are weaker but still promising however for the multiplexer there is only a weak correlation between the expected and known target orders and a low nestedness score. It is interesting that, even with a weakly correlated target order, some improvement is still gained from enforcing that order.

Except for CMUX—for which there is still an overall improvement—the third row in Figure 5 shows a high agreement between test accuracies gains using given and automatically detected target orderings. With zero prior knowledge of the respective target difficulties, the minFS-based method combined only with a slight change in the loss function, yields significant improvements in generalisation. This is promising given we are considering a learner which was already leveraging the expected benefits associated with a shared internal representation.

3.4.2 How performance correlates with target order

The next experiment we describe had two principal aims: to examine the relationship between the performance of hierarchical loss functions and the given curriculum, and to determine if the difficulty based curriculum that we have assumed as ground truth is actually optimal.

To do this we used a slightly smaller single problem—5-bit addition—and trained networks using L_{gh} with randomly generated target permutations and L_1. This gives us another

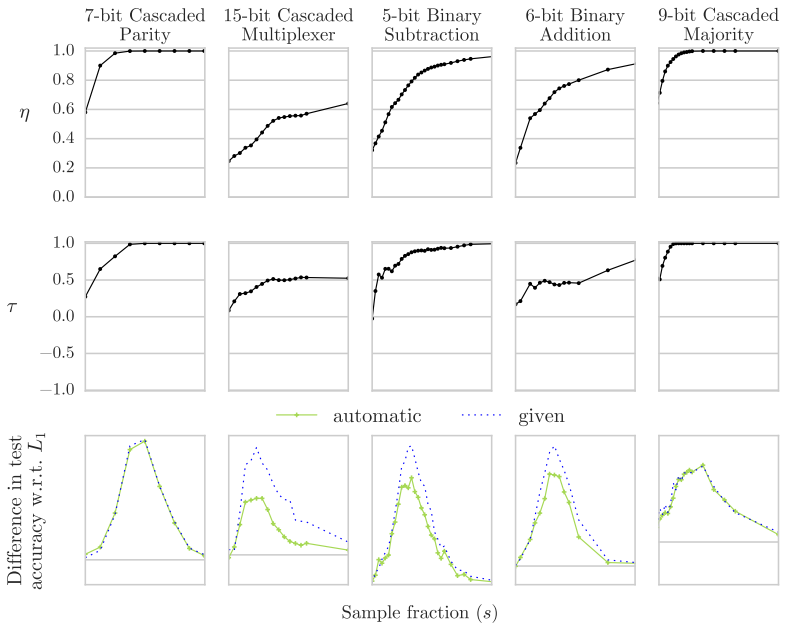


Figure 5: Mean increase in test accuracy—over networks trained using L_1—for networks trained using L_{gh} with the known target order (given) and the automatically detected target order (automatic). For all problems except CMUX both τ and η reach values above 0.5 before the respective peaks in test error difference. For these same problems the detected target order provides the nearly the same benefit as the known target order, which is no surprise given the strong correlation between the known and estimated orderings. The problem with the largest deviation, CMUX, is also the one for which τ and η are weakest. Note: the x-axes are only shared within columns, and the y-axis for the third row is not shared, since some problems have a much larger improvement than others.

important baseline—random target orderings—and also allows us to see how the performance of L_{gh} varies with τ, and if the least-to-most-significant curriculum we propose is outperformed by some other ordering.

We did not draw permutations uniformly since the rank correlation, τ, is heavily skewed over the space of permutations; the values ± 1 are only represented by 2, in combinatorially many, possibilities. However for a n-target permutation there are only $\lceil n \times (n-1)/2 \rceil$ unique values of τ, so selecting permutations uniformly distributed over τ is trivial. As such we trained 5 networks for each of 50 permutations (selected with replacement) for every value of τ over the range of sample sizes. The baseline L_1 does not vary with order so we again learned 1000 networks per sample size.

Figure 6 shows the resulting difference in test accuracy. Each line shows the mean improvement (over networks trained using L_1) as it varies with sample size for networks trained with L_{gh} for all permutations with that correlation value. One clear observation is that permutations which positively correlate (blue) with the easy-to-hard ordering give improvement while permutations which negatively correlate (red) actually detriment performance. The second key observation is that this varies with the strength of the correlation; and, most importantly, the largest improvement is given by the exact easy-to-hard curriculum ($\tau = +1$).

These results confirm our intuition that an easy-to-hard ordering is optimal—at least for this method of enforcing a target curriculum. They also agree with the results seen in Section 3.4.1 which showed that some improvement was seen for the CMUX problem even with a weak mean order correlation, but also that weaker correlation in the discovered curriculum reduced the improvement given by L_{gh}.

3.5 Real-World Problems

Here we present results for the experiments described in Sections 3.1.2 and 3.1.3. We lack an expected target order so the results for all ordered losses are with automatically detected curriculum (Section 2.2).

For the ALU, mammalian cell-cycle, and FA/BRCA pathway models we have the same regime as for the prior test-beds, so we have presented the results in the same manner. Figure 7 shows the mean difference in test accuracy between each hierarchical loss function and the baseline L_1 as sample size is varied. We also see that with the mammalian cell-cycle model there is much less differentiation between the hierarchical losses, in fact L_w appears to have given the most benefit. Overall, in all three cases imposing the target order improved test accuracy and the proposed method clearly discovered a beneficial curriculum.

For the time-series data sets the leave-one-out testing regime did not involve varying the sample size so we have instead presented the results in Tables 2 and 3. Again we see that imposing a target curriculum improved the generalisation performance of some targets later in the suspected order even for shallow hierarchies and with extremely limited sample sizes. In Table 3 we see an improvement in the target G6 and in Table 2 we see improvement for both subsequent targets—Cdc2/Cdc13 and Cdc2/Cdc13*—in one of the two estimated hierarchies (see Figure 2).

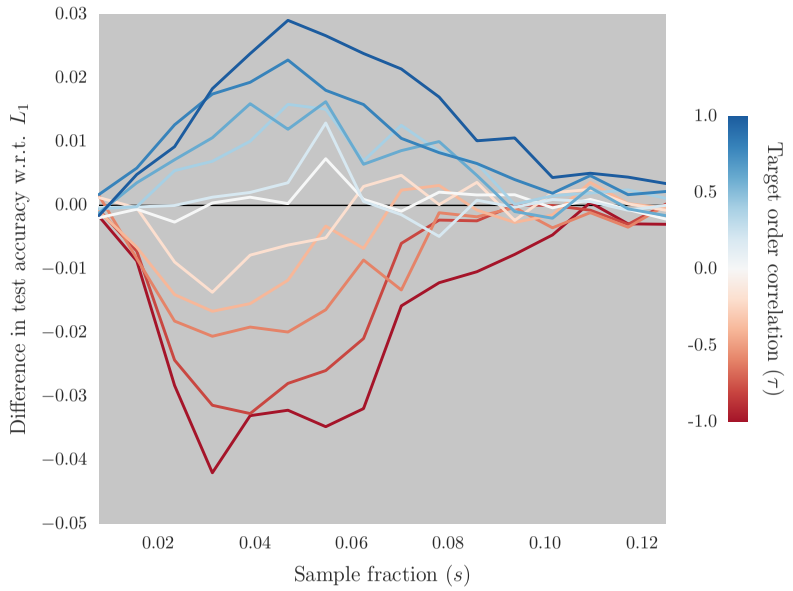


Figure 6: Mean difference in test accuracy between L_{gh} and L_1 over various training set sizes for the 5-bit addition problem for randomised target orderings. Each line represents the mean change in test accuracy for each possible level of correlation between the randomly given order and the known order. For an n-element permutation there are $\lceil n \times (n-1)/2 \rceil$ possible values of τ, hence 11 lines. The central permutations offer little benefit, and since the random permutations are skewed heavily toward $\tau = 0$ we can infer that a randomly generated ordering would offer no improvement on average. We see very clearly that that direct opposite ($\tau = -1$) of the suggested curricula impeded the learning performance by as much as the correct order improved it and that the exact easy-to-hard ordering ($\tau = 1$) conferred the largest benefit.

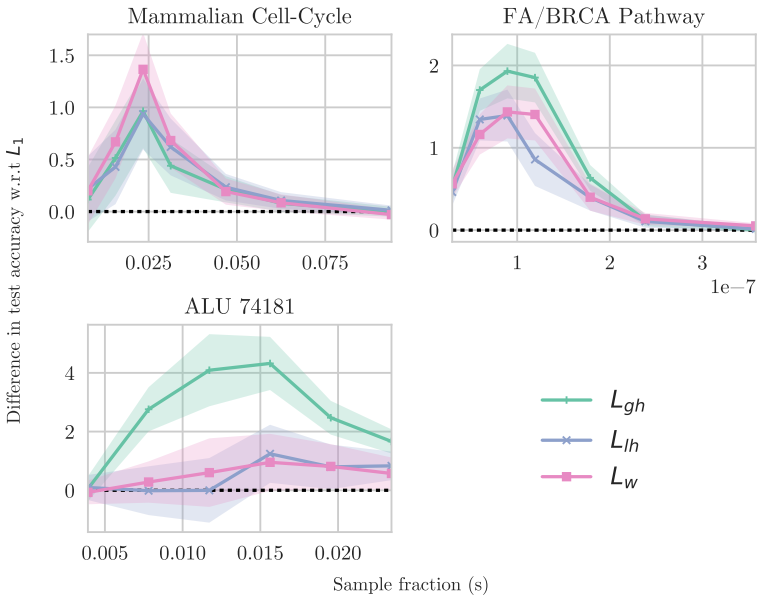


Figure 7: Mean difference in test accuracy between L_1 (dotted baseline) and each of L_w, L_{lh} and L_{gh} on each training sample for the 74181, mammalian cell-cycle and FA/BRCA pathway models. The 95% confidence interval of the mean is shown with transparent bands. On all problems we see the same net increase in test accuracy—again varying widely between problems. We see again that, except in the case of the mammalian cell-cycle model for which the losses are not clearly distinguishable, this improvement increases as the loss more strictly enforces the ordering ($L_w \to L_{lh} \to L_{gh}$).

	SK	Ste9	Rum1	PP	Cdc2/Cdc13	Cdc2/Cdc13*
L_w	0.4	0.5	0.4	1.6	1.2	0.6
L_{lh}	0.6	0.5	0.4	0.8	1.0	0.1
L_{gh}	0.9	0.5	0.3	0.5	1.3	1.4

Table 2: Results for the yeast data set. See Figure 2 for the estimated hierarchies. The SK→ {Ste9, Rum1} hierarchy has given little improvement, however we see more promise in the PP→ {Cdc2/Cdc13, Cdc2/Cdc13*} hierarchy, particularly for L_{gh}.

	G2	G8	G6
L_w	0.1	0.2	0.8
L_{lh}	0.7	-0.9	1.3
L_{gh}	0.1	0.1	1.4

Table 3: Results for the *E. coli* data set. See Figure 2 for the estimated hierarchies. The {G2, G8}→G6 hierarchy has given some improvement, however for L_{lh} we see a drop in performance on one of the base targets in the hierarchy. For L_{gh} the results are still overall positive.

4. Discussion and Future Work

We have shown that loss functions that impose a target order improve the overall generalisation performance of FBNs using a series of hierarchical multi-target circuit-inference problems. We also showed this improvement increases as the target order is more strictly enforced. Note that the losses considered do not alter the set of global optima. This approach does not restrict the space of networks but instead modifies their probability of discovery in a natural way using concepts inspired by human teaching and learning practices.

The set of circuit-inference problems described in Section 3.1.1 all possess a common element: intermediate values which are useful for computing the $m+1^{th}$ output are computed for the mth output. A human designer with access to a complete m-bit circuit would have a much easier time building an $(m+1)$-bit circuit. It is natural to assume that this ease should extend—in terms of convergence speed, generalisation or both—to a optimiser which builds such circuits in the correct order. We observed the latter but not the former.

We observed that performance improvements primarily occur around the existent generalisation phase transition. This is intuitive: there are fundamental limits with respect to training-set size that cannot be alleviated, and there are also sizes for which there is enough information present that teaching a target order confers no advantage. What we have shown is that—in the interim region between too little and plenty of training information—teaching targets in order of difficulty can make better use of what data is present. This improvement is expected in light of the positive effects of hierarchical cost functions on optimiser performance (Moscato, 1993).

The improvement on the final target from L_{gh} actually decreases (bottom of Figure 3). Initially, this appears as a cause for concern as it seems to indicate a point of diminishing returns. However this happened consistently only on the final target for all problems. Were

there a per-problem tipping point we would expect to see it before the last target, or not at all, on some problems. This is also the same for varying sizes of binary addition (see Online Appendix 2).

A more likely explanation is the gate budget coupled with a lack of constraints on which gates each target can source from. The loss function for which we see the issue occurring, L_{gh} essentially forces optimisation of each target in succession. As the process continues, it becomes increasingly more likely that any given target will have a majority of its computational nodes closer to the end of the ordering of gates. Subsequent sub-networks will thus require later and later nodes to be available in order to correctly discover the hierarchical structure of the problem which becomes an issue toward the end when the gate budget is exhausted. This could be alleviated by reserving nodes for each target.

We have also suggested an intuitive approach to determining the target overlap for Boolean MLCs, and to automatically determine an appropriate target order. For the problems considered, the bulk of the generalisation improvement conferred by the curriculum remained, even when no information on the true target order was provided to the learner.

We further examined the same loss functions in conjunction with the target curricula estimation method on three models of real world phenomena as well as on two regulatory network time-series data sets. On all three models we observed test accuracy increases similar to that seen on the circuit-inference problems. In the case of the yeast and *E. coli* data set we discovered possible target hierarchies which were shallow and as such we expected less improvement (Figure 3 shows that generalisation gains are focused primarily on later targets). Nonetheless the results are still in line with what we have observed for the previous problems. This is promising as we have not explicitly accounted for noise in either the curricula detection method or design of the loss functions.

Finally we examine the importance of the curricula itself in Section 3.4.2. We see that the easy-to-hard curricula gave the strongest improvement, that curricula which were anti-correlated with the easy-to-hard permutation actually decreased test accuracy, and that the curricula which were uncorrelated (i.e. essentially random) gave no improvement over learning with no curriculum. The fact that the peak in the distribution of permutations over τ occurs at zero means that randomly generated curricula will most commonly give no benefit and thus highlights the importance of principled methods for determining target curricula.

Not discussed here is the effect that the use of these guiding functions has on the average convergence time. We observed a variable effect on training speed and, for some problems, there is a cost in convergence speed when using target curricula. More information on training time is available in Online Appendix 2.

Another possible impact on training speed is the requirement for the target ordering method to solve an NP-hard optimisation problem. However, for all problems considered, the target order detection process had almost no impact on overall training time. For significantly larger problem instances, however, heuristic solvers would likely be required.

Noise has not been considered as a factor in this work. It is uncertain whether the minFS based methods for estimating target overlap and target schedules—or FBNs themselves—are robust to incorrect target labels. Future work should address this shortfall, perhaps by allowing the per-target error constraints in L_{gh} to be non-zero. The drop in performance on

the final target should also be addressed; possibly by using stratified networks, or annealed depth constraints, to prevent later gates being consumed by earlier targets.

While our results are only for the case of Boolean MLCs learned with FBNs, we expect they may extend to other non-convex learners, such as deep neural nets, for problems with categorical or Boolean domains, provided differentiable analogues to L_{lh} and L_{gh} are found. Future work could also examine target feature selection methods for real valued domains, as well as heuristic Boolean feature selection methods, to determine if the approach presented here is more generally applicable. A recent discussion of the use of intrinsic dimension as a proxy for the overall complexity of arbitrary data sets (Granata and Carnevale, 2016) is promising.

Acknowledgments

We thank Ms Natalie de Vries for her assistance with proof reading and discussion of the manuscript, and Drs. Alexandre Mendes and Nasimul Noman for their fruitful discussions in the early phases of the work. Pablo Moscato also wishes to thank Prof. Miguel Ángel Virasoro for a preprint copy of Patarnello and Carnevali (1987) many years ago. P.M. acknowledges funding of this research by the Australian Research Council grants Future Fellowship FT120100060 and Discovery Project DP140104183 and the Maitland Cancer Appeal.

References

Yaser S Abu-Mostafa. Learning from hints in neural networks. *Journal of Complexity*, 6 (2):192–198, June 1990.

Shohag Barman and Yung-Keun Kwon. A novel mutual information-based Boolean network inference method from time-series gene expression data. *PLOS ONE*, 12(2):e0171097, February 2017.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, pages 41–48. ACM, 2009.

Edmund K. Burke and Yuri Bykov. A late acceptance strategy in hill-climbing for examination timetabling problems. In *International Conference on the Practice and Theory of Automated Timetabling (PATAT)*, 2008.

Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

Josu Ceberio, Alexander Mendiburu, and Jose A. Lozano. The linear ordering problem revisited. *European Journal of Operational Research*, 241(3):686–696, March 2015.

Carlos Cotta and Pablo Moscato. The k-feature set problem is W[2]-complete. *Journal of Computer and System Sciences*, 67(4):686–690, December 2003.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *arXiv:1602.02830 [cs]*, February 2016.

Scott Davies and Stuart Russell. NP-Completeness of searches for smallest possible feature sets. In *AAAI Symposium on Intelligent Relevance*, pages 37–39. AAAI Press, 1994.

Alireza Goudarzi, Christof Teuscher, Natali Gulbahce, and Thimo Rohlf. Learning, generalisation, and functional entropy in random automata networks. *International Journal of Autonomous and Adaptive Communications Systems*, 7(3):295–314, 2014.

Daniele Granata and Vincenzo Carnevale. Accurate estimation of the intrinsic dimension using graph distances: Unraveling the geometric complexity of datasets. *Scientific Reports*, 6:31377, August 2016.

Çağlar Gülçehre and Yoshua Bengio. Knowledge matters: Importance of prior information for optimization. *Journal of Machine Learning Research*, 17(8):1–32, 2016.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. Self-paced curriculum learning. In *AAAI Conference on Artificial Intelligence*, pages 2694–2700, Austin, Texas, 2015. AAAI Press.

Faisal Khan, Bilge Mutlu, and Xiaojin Zhu. How do humans teach: On curriculum learning and teaching dimension. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1449–1457. Curran Associates, Inc., 2011.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *International Conference on Neural Information Processing Systems*, NIPS'10, pages 1189–1197, USA, 2010. Curran Associates Inc.

Abhimanyu Lad, Rayid Ghani, Yiming Yang, and Bryan Kisiel. Toward optimal ordering of prediction tasks. In *SIAM International Conference on Data Mining*, Proceedings, pages 884–893. SIAM, April 2009.

Changsheng Li, Junchi Yan, Fan Wei, Weishan Dong, Qingshan Liu, and Hongyuan Zha. Self-paced multi-task learning. In *AAAI Conference on Artificial Intelligence*, pages 2175–2181, 2017.

Pablo Moscato. An introduction to population approaches for optimization and hierarchical objective functions: A discussion on the role of tabu search. *Annals of Operations Research*, 41(2):85–121, June 1993.

Stefano Patarnello and Paolo Carnevali. Learning networks of neurons with Boolean logic. *Europhysics Letters*, 4(4):503, 1987.

Anastasia Pentina, Viktoriia Sharmanska, and Christoph H. Lampert. Curriculum learning of multiple tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5492–5500, June 2015.

Arnaud Poret and Jean-Pierre Boissel. An in silico target identification using Boolean network attractors: Avoiding pathological phenotypes. *Comptes Rendus Biologies*, 337 (12):661–678, December 2014.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, June 2011.

Mateus Rocha de Paula, Regina Berretta, and Pablo Moscato. A fast meta-heuristic approach for the α-β-k-feature set problem. *Journal of Heuristics*, 22(2):199–220, December 2015.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. Baby steps: How "less is more" in unsupervised dependency parsing. In *NIPS: Grammar Induction, Representation of Language and Language Learning*, 2009.

Christof Teuscher, Natali Gulbahce, and Thimo Rohlf. Learning and generalization in random Boolean networks. In *Dynamic Days 2007: International Conference on Chaos and Nonlinear Dynamics*, Boston, MA, January 2007.

Alan M. Turing. *Intelligent Machinery. Report for National Physical Laboratory. Reprinted in Ince, Dc (Editor). 1992. Mechanical Intelligence: Collected Works of A.M. Turing.* Amsterdam: North Holland, 1948.

Christian Van den Broeck and Ryoichi Kawai. Learning in feedforward Boolean networks. *Physical Review A*, 42(10):6210, 1990.

Xin Wang, Jinbo Bi, Shipeng Yu, Jiangwen Sun, and Minghu Song. Multiplicative multitask feature learning. *Journal of Machine Learning Research*, 17(80):1–33, 2016.

Journal of Machine Learning Research 18 (2017) 1-52 Submitted 9/16; Revised 10/17; Published 10/17

A General Distributed Dual Coordinate Optimization Framework for Regularized Loss Minimization

Shun Zheng* ZHENGS14@MAILS.TSINGHUA.EDU.CN
Institute for Interdisciplinary Information Sciences
Tsinghua University
Beijing, China

Jialei Wang JIALEI@UCHICAGO.EDU
Department of Computer Science
The University of Chicago
Chicago, Illinois

Fen Xia XIAFEN@EBRAIN.AI
Beijing Wisdom Uranium Technology Co., Ltd.
Beijing, China

Wei Xu WEIXU@TSINGHUA.EDU.CN
Institute for Interdisciplinary Information Sciences
Tsinghua University
Beijing, China

Tong Zhang TONGZHANG@TONGZHANG-ML.ORG
Tencent AI Lab
Shenzhen, China

Editor: Sathiya Keerthi

Abstract

In modern large-scale machine learning applications, the training data are often partitioned and stored on multiple machines. It is customary to employ the "data parallelism" approach, where the aggregated training loss is minimized without moving data across machines. In this paper, we introduce a novel distributed dual formulation for regularized loss minimization problems that can directly handle data parallelism in the distributed setting. This formulation allows us to systematically derive dual coordinate optimization procedures, which we refer to as *Distributed Alternating Dual Maximization* (DADM). The framework extends earlier studies described in (Boyd et al., 2011; Ma et al., 2017; Jaggi et al., 2014; Yang, 2013) and has rigorous theoretical analyses. Moreover, with the help of the new formulation, we develop the accelerated version of DADM (Acc-DADM) by generalizing the acceleration technique from (Shalev-Shwartz and Zhang, 2014) to the distributed setting. We also provide theoretical results for the proposed accelerated version, and the new result improves previous ones (Yang, 2013; Ma et al., 2017) whose iteration complexities grow linearly on the condition number. Our empirical studies validate our theory and show that our accelerated approach significantly improves the previous state-of-the-art distributed dual coordinate optimization algorithms.

∗. Most of the work was done during the internship of Shun Zheng at Baidu Big Data Lab in Beijing.

©2017 Shun Zheng, Jialei Wang, Fen Xia, Wei Xu, and Tong Zhang.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-463.html.

Keywords: distributed optimization, stochastic dual coordinate ascent, acceleration, regularized loss minimization, computational complexity

1. Introduction

In large-scale machine learning applications for big data analysis, it becomes a common practice to partition the training data and store them on multiple machines connected via a commodity network. A typical setting of distributed machine learning is to allow these machines to train in parallel, with each machine processing its local data with no data communication. This paradigm is often referred to as *data parallelism*. To reduce the overall training time, it is often necessary to increase the number of machines and to minimize the communication overhead. A significant challenge is to reduce the training time as much as possible when we increase the number of machines. A practical solution requires two research directions: one is to improve the underlying system design making it suitable for machine learning algorithms (Dean and Ghemawat, 2008; Zaharia et al., 2012; Dean et al., 2012; Li et al., 2014); the other is to adapt traditional single-machine optimization methods to handle data parallelism (Boyd et al., 2011; Yang, 2013; Mahajan et al., 2013; Shamir et al., 2014; Jaggi et al., 2014; Mahajan et al., 2017; Ma et al., 2017; Takáč et al., 2015; Zhang and Lin, 2015). This paper focuses on the latter.

For big data machine learning on a single machine, there are two types of algorithms: batch algorithms such as gradient descent or L-BFGS (Liu and Nocedal, 1989), and stochastic optimization algorithms such as stochastic gradient descent and their modern variance reduced versions (Defazio et al., 2014; Johnson and Zhang, 2013). It is known that batch algorithms are relatively easy to parallelize. However, on a single machine, they converge more slowly than the modern stochastic optimization algorithms due to their high per-iteration computation costs. Specifically, it has been shown that the modern stochastic optimization algorithms converge faster than the traditional batch algorithms for convex regularized loss minimization problems. The faster convergence can be guaranteed in theory and observed in practice.

The fast convergence of modern stochastic optimization methods has led to studies to extend these methods to the distributed computing setting. Specifically, this paper considers the generalization of Stochastic Dual Coordinate Ascent (SDCA) method (Hsieh et al., 2008; Shalev-Shwartz and Zhang, 2013) and its proximal variant (Shalev-Shwartz and Zhang, 2014) to handle distributed training using data parallelism. Although this problem has been considered previously (Yang, 2013; Jaggi et al., 2014; Ma et al., 2017), these earlier approaches work with the dual formulation that is the same as the traditional single-machine dual formulation, where dual variables are coupled, and hence run into difficulties when they try to motivate and analyze the derived methods under the distributed environment.

One contribution of this work is to introduce a new dual formulation specifically for distributed regularized loss minimization problems when data are distributed to multiple machines. In our new formulation, we decouple the local dual variables through introducing another dual variable β. This unique dual formulation allows us to naturally extend the proximal SDCA algorithm (ProxSDCA) of (Shalev-Shwartz and Zhang, 2014) to the setting of multi-machine distributed optimization that can benefit from data parallelism. Moreover, the analysis of the original ProxSDCA can be easily adapted to the new formu-

lation, leading to new theoretical results. This new dual formulation can also be combined with the acceleration technique of (Shalev-Shwartz and Zhang, 2014) to further improve convergence.

In the proposed formulation, each iteration of the distributed dual coordinate ascent optimization is naturally decomposed into a local step and a global step. In the local step, we allow the use of any local procedure to optimize a local dual objective function using local parameters and local data on each machine. This flexibility is similar to those of (Ma et al., 2017; Jaggi et al., 2014). For example, we may apply ProxSDCA as the local procedure. In the local step, a computer node can perform the optimization independently without communicating with each other. While in the global step, nodes communicate with each other to synchronize the local parameters and jointly update the global primal solution. Only this global step requires communication among nodes.

We summarize our main contributions as follows:

New distributed dual formulation This new formulation naturally leads to a two-step local-global dual alternating optimization procedure for distributed machine learning. We thus call the resulting procedure *Distributed Alternating Dual Maximization* (DADM). Note that DADM directly generalizes ProxSDCA, which can handle complex regularizations such as L_2-L_1 regularization.

New convergence analysis The new formulation allows us to directly generalize the analysis of ProxSDCA in (Shalev-Shwartz and Zhang, 2014) to the distributed setting. This analysis is in contrast to that of CoCoA$^+$ in (Ma et al., 2017), which employs a different way based on the Θ-approximate solution assumption of the local solver. Our analysis can lead to simplified results in the commonly used mini-batch setup.

Acceleration with theoretical guarantees Based on the new distributed dual formulation, we can naturally derive a distributed version of the accelerated proximal SDCA method (AccProxSDCA) of (Shalev-Shwartz and Zhang, 2014), which has been shown to be effective on a single machine. We call the resulting procedure *Accelerated Distributed Alternating Dual Maximization* (Acc-DADM). The main idea is to modify the original formulation using a sequence of approximations that have stronger regularizations. Moreover, we directly adapt theoretical analyses of AccProxSDCA to the distributed setting and provide guarantees for Acc-DADM. Our theorems guarantee that we can always obtain a computation speedup compared with the single-machine AccProxSDCA. These guarantees improve the theoretical results of DADM and previous methods (Yang, 2013; Ma et al., 2017) whose iteration complexities grow linearly on the condition number. Latter methods possibly fail to provide computation time improvement over the single-machine ProxSDCA when the condition number is large.

Extensive empirical studies We perform extensive experiments to compare the convergence and the scalability of the accelerated approach with that of previous state-of-the-art distributed dual coordinate ascent methods. Our empirical studies show that Acc-DADM can achieve faster convergence and better scalability than the previous state-of-the-art, in particular when the condition number is large. This phenomenon is consistent with our theory.

We organize the rest of the paper as follows. Section 2 discusses related works. Section 3 provides preliminary definitions. Section 4 to 6 present the distributed primal formulation, the distributed dual formulation and our DADM method respectively. Section 7 then provides theorems for DADM. Section 8 introduces the accelerated version and provides corresponding theoretical guarantees. Section 9 includes all proofs of this paper. Section 10 provides extensive empirical studies of our novel method. Finally, Section 11 concludes the whole paper.

2. Related Work

Several generalizations of SDCA to the distributed settings have been proposed in the literature, including DisDCA (Yang, 2013), CoCoA (Jaggi et al., 2014), and CoCoA$^+$ (Ma et al., 2017).

DisDCA was the first attempt to study distributed SDCA, and it provided a basic theoretical analysis and a practical variant that behaves well empirically. Nevertheless, their theoretical result only applies to a few specially chosen mini-batch local dual updates that differ from the practical method used in their experiments. In particular, they did not show that optimizing each local dual problem leads to convergence. This limitation makes the methods they analyzed inflexible.

CoCoA was proposed to fix the above gap between theory and practice, and it was claimed to be a framework for distributed dual coordinate ascent in that it allows any local dual solver to be used for the local dual problem, rather than the impractical choices of DisDCA. However, the actual performance of CoCoA is inferior to the practical variant proposed in DisDCA with an aggressive local update. We note that the practical variant of DisDCA did not have a solid theoretical guarantee at that time.

CoCoA$^+$ fixed this situation and may be regarded as a generalization of CoCoA. The most effective choice of the aggregation parameter leads to a version which is similar to DisDCA, but allows exact optimization of each dual problem in their theory. According to studies in (Ma et al., 2017), the resulting CoCoA$^+$ algorithm performs significantly better than the original CoCoA both theoretically and empirically. The original CoCoA$^+$ (Ma et al., 2015) can only handle problems with the L_2 regularizer, and it was generalized to general strongly convex regularizers in the long version (Ma et al., 2017). Besides, (Smith et al., 2016) extended the framework to solve the primal problem of regularized loss minimization and cover general non-strongly convex regularizers such as L_1 regularizer, and (Hsieh et al., 2015) studied parallel SDCA with asynchronous updates.

Although CoCoA$^+$ has the advantage of allowing arbitrary local solvers and flexible approximate solutions of local dual problems, its theoretical analyses do not capture the contribution of the number of machines and the mini-batch size to the iteration complexity explicitly. Moreover, the iteration complexities of both CoCoA$^+$ and DisDCA grow linearly with the condition number. Thus they probably cannot provide computation time improvement over the single-machine SDCA when the condition number is large. This paper will remedy these unsatisfied aspects by providing a different analysis based on a new distributed dual formulation. Using this formulation, we can analyze procedures that can take an arbitrary local dual solver, which is like CoCoA$^+$; moreover, we allow the dual updates to be a mini-batch, which is like DisDCA. Besides, this formulation also allows

us to naturally generalize AccProxSDCA and relevant theoretical results to the distributed setting. Our empirical results also validate the superiority of the accelerated approach.

While we focus on extending SDCA in this paper, we note that there are other approaches for parallel optimization. For example, there are direct attempts to parallelize stochastic gradient descent (Niu et al., 2011; Zinkevich et al., 2010). Some of these procedures only consider multi-core shared memory situation, which is very different from the distributed computing environment investigated in this paper. In the setting of distributed computing, data are partitioned into multiple machines, and one often needs to study communication-efficient algorithms. In such cases, one extreme is to allow exact optimization of subproblems on each local machine as considered in (Shamir et al., 2014; Zhang and Lin, 2015). Although this approach minimizes communication, the computation cost for each local solver can dominate the overall training. Therefore in practice, it is necessary to do a trade-off by using the mini-batch update approach (Takáč et al., 2013, 2015). However, it is difficult for traditional mini-batch methods to design reasonable aggregation strategies to achieve fast convergence. (Takáč et al., 2015) studied how the step size can be reduced when the mini-batch size grows in the distributed setting. (Lee and Roth, 2015) derived an analytical solution of the optimal step size for dual linear support vector machine problems. Besides, (Mahajan et al., 2013) presented a general framework for distributed optimization based on local functional approximation, which include several first-order and second-order methods as special cases. (Mahajan et al., 2017) considered each machine to handle a block of coordinates and proposed distributed block coordinate descent methods for solving ℓ_1 regularized loss minimization problems.

Different from those methods, *Distributed Alternating Dual Maximization* (DADM) proposed in this work handles the trade-off between computation and communication by developing bounds for mini-batch dual updates, which is similar to (Yang, 2013). Moreover, DADM allows other better local solvers to achieve faster convergence in practice.

3. Preliminaries

In this section, we introduce some notations used later. All functions that we consider in this paper are proper convex functions over a Euclidean space.

Given a function $f : \mathbb{R}^d \to \mathbb{R}$, we denote its **conjugate function** as

$$f^*(b) = \sup_a [b^\top a - f(a)].$$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is L**-Lipschitz** with respect to $\|\cdot\|_2$ if for all $a, b \in \mathbb{R}^d$, we have

$$|f(a) - f(b)| \leq L\|a - b\|_2.$$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is $(1/\gamma)$**-smooth** with respect to $\|\cdot\|_2$ if it is differentiable and its gradient is $(1/\gamma)$-Lipschitz with respect to $\|\cdot\|_2$. An equivalent definition is that for all $a, b \in \mathbb{R}^d$, we have

$$f(b) \leq f(a) + \nabla f(a)^\top (b - a) + \frac{1}{2\gamma}\|b - a\|_2^2.$$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is λ**-strongly convex** with respect to $\|\cdot\|_2$ if for any $a, b \in \mathbb{R}^d$, we have

$$f(b) \geq f(a) + \nabla f(a)^\top (b - a) + \frac{\lambda}{2}\|b - a\|_2^2,$$

where $\nabla f(a)$ is any subgradient of $f(a)$.

It is well known that a function f is γ-strongly convex with respect to $\|\cdot\|_2$ if and only if its conjugate function f^* is $(1/\gamma)$-smooth with respect to $\|\cdot\|_2$.

4. Distributed Primal Formulation

In this paper, we consider the following generic regularized loss minimization problem:

$$\min_{w \in \mathbb{R}^d} \left[P(w) := \sum_{i=1}^n \phi_i(X_i^\top w) + \lambda n g(w) + h(w) \right], \tag{1}$$

which is often encountered in practical machine learning problems. Here we assume each $X_i \in \mathbb{R}^{d \times q}$ is a $d \times q$ matrix, $w \in \mathbb{R}^d$ is the model parameter vector, $\phi_i(u)$ is a convex loss function defined on $\mathbb{R}^q$, which is associated with the i-th data point, $\lambda > 0$ is the regularization parameter, $g(w)$ is a strongly convex regularizer and $h(w)$ is another convex regularizer. A special case is to simply set $h(w) = 0$. Here we allow the more general formulation, which can be used to derive different distributed dual forms that may be useful for special purposes.

The above optimization formulation can be specialized to a variety of machine learning problems. As an example, we may consider the L_2-L_1 regularized least squares problem, where $\phi_i(x_i^\top w) = (w^\top x_i - y_i)^2$ for vector input data $x_i \in \mathbb{R}^d$ and real valued output $y_i \in \mathbb{R}$, $g(w) = \|w\|_2^2 + a\|w\|_1$, and $h(w) = b\|w\|_1$ for some $a, b \geq 0$.

If we set $h(w) = 0$, then it is well-known (see, for example, (Shalev-Shwartz and Zhang, 2014)) that the primal problem (1) has an equivalent single-machine dual form of

$$\max_{\alpha \in \mathbb{R}^n} \left[D(\alpha) := -\sum_{i=1}^n \phi_i^*(-\alpha_i) - \lambda n g^*\left(\frac{\sum_{i=1}^n X_i \alpha_i}{\lambda n} \right) \right], \tag{2}$$

where $\alpha = [\alpha_1, \cdots, \alpha_n], \alpha_i \in \mathbb{R}^q$ $(i = 1, ..., n)$ are dual variables, ϕ_i^* is the convex conjugate function of ϕ_i, and similarly, g^* is the convex conjugate function of g.

The stochastic dual coordinate ascent method, referred to as SDCA in (Shalev-Shwartz and Zhang, 2014), maximizes the dual formulation (2) by optimizing one randomly chosen dual variable at each iteration. Throughout the algorithm, the following primal-dual relationship is maintained:

$$w(\alpha) = \nabla g^*\left(\frac{\sum_{i=1}^n X_i \alpha_i}{\lambda n} \right), \tag{3}$$

for some subgradient $\nabla g^*(v)$.

It is known that $w(\alpha^*) = w^*$, where w^* and α^* are optimal solutions of the primal problem and the dual problem respectively. It was shown in (Shalev-Shwartz and Zhang, 2014) that the duality gap defined as $P(w(\alpha)) - D(\alpha)$, which is an upper-bound of the primal sub-optimality $P(w(\alpha)) - P(w^*)$, converges to zero. Moreover, a convergence rate can be established. In particular, for smooth loss functions, the convergence rate is linear.

We note that SDCA is suitable for optimization on a single machine because it works with a dual formulation that is suitable for a single machine. In the following, we will

generalize the single-machine dual formulation (2) to the distributed setting, and study the corresponding distributed version of SDCA.

In the distributed setting, we assume that the training data are partitioned and distributed to m machines. In other words, the index set $S = \{1, ..., n\}$ of the training data is divided into m non-overlapping partitions, where each machine $\ell \in \{1, ..., m\}$ contains its own partition $S_\ell \subseteq S$. We assume that $\cup_\ell S_\ell = S$, and we use $n_\ell := |S_\ell|$ to denote the size of the training data on machine ℓ.

Next, we can rewrite the primal problem (1) as the following constrained minimization problem that is suitable for the multi-machine distributed setting:

$$\begin{aligned} &\min_{w;\{w_\ell\}_{\ell=1}^m} \sum_{\ell=1}^m P_\ell(w_\ell) + h(w) \\ &\quad s.t. \quad w_\ell = w, \quad \text{for all } \ell \in \{1, ..., m\}, \\ &\text{where} \quad P_\ell(w_\ell) := \sum_{i \in S_\ell} \phi_i(X_i^\top w_\ell) + \lambda n_\ell g(w_\ell), \end{aligned} \tag{4}$$

where w_ℓ represents the local primal variable on each machine ℓ, P_ℓ is the corresponding local primal problem and the constraints $w_\ell = w$ are imposed to synchronize the local primal variables. Obviously this multi-machine distributed primal formulation (4) is equivalent to the original primal problem (1).

We note that the idea of objective splitting in (4) is similar to the global variable consensus formulation described in (Boyd et al., 2011). Instead of using the commonly used ADMM (Alternating Direction Method of Multipliers) method that is not a generalization of (2), in this paper we derive a distributed dual formulation based on (4) that directly generalizes (2). We further propose a framework called Distributed Alternating Dual Maximization (DADM) to solve the distributed dual formulation. One advantage of DADM over ADMM is that DADM does not need to solve the subproblems in high accuracy, and thus it can naturally enjoy the trade-off between computation and communication, which is similar to related methods such as DisDCA, CoCoA and CoCoA$^+$.

5. Distributed Dual Formulation

The optimization problem (4) can be further rewritten as:

$$\begin{aligned} &\min_{w;\{w_\ell\};\{u_i\}} \sum_{\ell=1}^m \left[\sum_{i \in S_\ell} \phi_i(u_i) + \lambda n_\ell g(w_\ell) \right] + h(w) \\ &\quad s.t \quad u_i = X_i^\top w_\ell, \text{ for all } i \in S_\ell \\ &\qquad w_\ell = w, \text{ for all } \ell \in \{1, ..., m\}. \end{aligned} \tag{5}$$

Here we introduce n dual variables $\alpha := \{\alpha_i\}_{i=1}^n$, where each α_i is the Lagrange multiplier for the constraint $u_i - X_i^\top w_\ell = 0$, and m dual variables $\beta := \{\beta_\ell\}_{\ell=1}^m$, where each β_ℓ is the Lagrange multiplier for the constraint $w_\ell - w = 0$. We can now introduce the primal-dual objective function with Lagrange multipliers as follows:

$$\begin{aligned} &J(w; \{w_\ell\}; \{u_i\}; \{\alpha_i\}; \{\beta_\ell\}) \\ :=&\sum_{\ell=1}^m \left[\sum_{i \in S_\ell} \left(\phi_i(u_i) + \alpha_i^\top (u_i - X_i^\top w_\ell) \right) + \lambda n_\ell g(w_\ell) + \beta_\ell^\top (w_\ell - w) \right] + h(w). \end{aligned}$$

Proposition 1 *Define the dual objective as*

$$D(\alpha, \beta) := \sum_{\ell=1}^m \left[\sum_{i \in S_\ell} -\phi_i^*(-\alpha_i) - \lambda n_\ell g^* \left(\frac{\sum_{i \in S_\ell} X_i \alpha_i - \beta_\ell}{\lambda n_\ell} \right) \right] - h^* \left(\sum_\ell \beta_\ell \right).$$

Then we have

$$D(\alpha, \beta) = \min_{w;\{w_\ell\};\{u_i\}} J(w; \{w_\ell\}; \{u_i\}; \{\alpha_i\}; \{\beta_\ell\}),$$

where the minimizers are achieved when the following equations are satisfied

$$\begin{aligned} \nabla \phi_i(u_i) + \alpha_i &= 0, \\ -\left(\sum_{i \in S_\ell} X_i \alpha_i - \beta_\ell \right) + \lambda n_\ell \nabla g(w_\ell) &= 0, \\ -\sum_\ell \beta_\ell + \nabla h(w) &= 0, \end{aligned} \tag{6}$$

for some subgradients $\nabla \phi_i(u_i)$, $\nabla g(w_\ell)$, and $\nabla h(w)$.

When $\beta = \{\beta_\ell\}$ are fixed, we may define the local single-machine dual formulation on each machine ℓ with respect to $\alpha_{(\ell)}$ as

$$\tilde{D}_\ell(\alpha_{(\ell)} | \beta_\ell) := \sum_{i \in S_\ell} -\phi_i^*(-\alpha_i) - \lambda n_\ell g^* \left(\frac{\sum_{i \in S_\ell} X_i \alpha_i - \beta_\ell}{\lambda n_\ell} \right), \tag{7}$$

where $\alpha_{(\ell)}$ represents local dual variables $\{\alpha_i; i \in S_\ell\}$ on machine ℓ, $\beta_\ell \in \mathbb{R}^d$ serves as a carrier for synchronization of machine ℓ. Based on Proposition 1, we obtain the following multi-machine distributed dual formulation for the corresponding primal problem (4):

$$D(\alpha, \beta) = \sum_{\ell=1}^m \tilde{D}_\ell(\alpha_{(\ell)} | \beta_\ell) - h^* \left(\sum_{\ell=1}^m \beta_\ell \right). \tag{8}$$

Moreover, we have the non-negative duality gap, and zero duality gap can be achieved when w is the minimizer of $P(w)$ and (α, β) maximizes the dual $D(\alpha, \beta)$.

Proposition 2 *Given any (w, α, β), the following duality gap is non-negative:*

$$P(w) - D(\alpha, \beta) \geq 0.$$

Moreover, zero duality gap can be achieved at (w^*, α^*, β^*), where w^* is the minimizer of $P(w)$ and (α^*, β^*) is a maximizer of $D(\alpha, \beta)$.

We note that the parameters $\{\beta_\ell\}_{\ell=1}^m$ pass the global information across multiple machines. When β_ℓ is fixed, $\tilde{D}_\ell(\alpha_{(\ell)}|\beta_\ell)$ with respect to $\alpha_{(\ell)}$ corresponds to the dual of the adjusted local primal problem:

$$\tilde{P}_\ell(w_\ell|\beta_\ell) := \sum_{i\in S_\ell} \phi_i(X_i^\top w_\ell) + \lambda n_\ell \tilde{g}_\ell(w_\ell), \tag{9}$$

where the original regularizer $\lambda n_\ell g(w_\ell)$ in $P_\ell(w_\ell)$ is replaced by the adjusted regularizer

$$\lambda n_\ell \tilde{g}_\ell(w_\ell) := \lambda n_\ell g(w_\ell) + \beta_\ell^\top w_\ell.$$

Similar to the single-machine primal-dual relationship of (3), we have the following local primal-dual relationship on each machine as:

$$w_\ell(\alpha_{(\ell)}, \beta_\ell) = \nabla g^*\left(\tilde{v}_\ell\right) = \nabla \tilde{g}_\ell^*\left(v_\ell\right), \tag{10}$$

where

$$v_\ell = \frac{\sum_{i\in S_\ell} X_i\alpha_i}{\lambda n_\ell}, \qquad \tilde{v}_\ell = v_\ell - \frac{\beta_\ell}{\lambda n_\ell}.$$

Moreover, we can define the global primal-dual relationship as

$$w(\alpha, \beta) = \nabla g^*\left(\tilde{v}\right) = \nabla \tilde{g}^*\left(v\right), \tag{11}$$

where

$$v = \frac{\sum_{i=1}^n X_i\alpha_i}{\lambda n}, \qquad \tilde{v} = v - \frac{\sum_\ell \beta_\ell}{\lambda n}.$$

We can also establish the relationship of global-local duality in Proposition 3.

Proposition 3 *Given (w, α, β) and $\{w_\ell\}$ such that $w_1 = \cdots = w_m = w$, we have the following decomposition of global duality gap as the sum of local duality gaps:*

$$P(w) - D(\alpha, \beta) \geq \sum_{\ell=1}^m \left[\tilde{P}_\ell(w_\ell|\beta_\ell) - \tilde{D}_\ell(\alpha_{(\ell)}|\beta_\ell)\right],$$

and the equality holds when $\nabla h(w) = \sum_\ell \beta_\ell$ for some subgradient $\nabla h(w)$.

Although we allow arbitrary $h(w)$, the case of $h(w) = 0$ is of special interests. This corresponds to the conjugate function

$$h^*(\beta) = \begin{cases} +\infty & \text{if } \beta \neq 0 \\ 0 & \text{if } \beta = 0 \end{cases}.$$

That is, the term $h^*\left(\sum_{\ell=1}^m \beta_\ell\right)$ is equivalent to imposing the constraint $\sum_{\ell=1}^m \beta_\ell = 0$.

Algorithm 1 Local Dual Update

Retrieve local parameters $(\alpha_\ell^{(t-1)}, \tilde{v}_\ell^{(t-1)})$
Randomly pick a mini-batch $Q_\ell \subset S_\ell$
Approximately maximize (12) w.r.t $\Delta\alpha_{Q_\ell}$
Update $\alpha_i^{(t)}$ as $\alpha_i^{(t)} = \alpha_i^{(t-1)} + \Delta\alpha_i$ for all $i \in Q_\ell$
return $\Delta v_\ell^{(t)} = \frac{1}{\lambda n_\ell}\sum_{i\in Q_\ell} X_i\Delta\alpha_i$

6. Distributed Alternating Dual Maximization

Minimizing the primal formulation (4) is equivalent to maximizing the dual formulation (8), and the latter can be achieved by repeatedly using the following alternating optimization strategy, which we refer to as **D**istributed **A**lternating **D**ual **M**aximization (**DADM**):

- **Local step**: fix β_ℓ and let each machine approximately optimize $\tilde{D}_\ell(\alpha_{(\ell)}|\beta_\ell)$ w.r.t $\alpha_{(\ell)}$ in parallel.
- **Global step**: maximize the global dual objective w.r.t β_ℓ, and set the global primal parameter w accordingly.

The above steps are applied in iterations $t = 1, 2, \ldots, T$. At the beginning of each iteration t, we assume that the local primal and dual variables on each local machine are $(\alpha_{(\ell)}^{(t-1)}, \beta_\ell^{(t-1)}, v_\ell^{(t-1)})$, then we seek to update $\alpha_{(\ell)}^{(t-1)}$ to $\alpha_{(\ell)}^{(t)}$ and $v_\ell^{(t-1)}$ to $v_\ell^{(t)}$ in the local step, and seek to update $\beta_\ell^{(t-1)}$ to $\beta_\ell^{(t)}$ in the global step.

We note that the local step can be executed in parallel w.r.t dual variables $\{\alpha_{(\ell)}\}_{\ell=1}^m$. In practice, it is often useful to optimize (7) approximately by using a randomly selected mini-batch $Q_\ell \subset S_\ell$ of size $|Q_\ell| = M_\ell$. That is, we want to find $\Delta\alpha_i^{(t)}$ with $i \in Q_\ell$ to approximately maximize the local dual objective as follows:

$$\tilde{D}_{Q_\ell}^{(t)}(\Delta\alpha_{Q_\ell}) := -\sum_{i\in Q_\ell} \phi_i^*(-\alpha_i^{(t-1)} - \Delta\alpha_i) - \lambda n_\ell g^*\left(\tilde{v}_\ell^{(t-1)} + \frac{\sum_{i\in Q_\ell} X_i\Delta\alpha_i}{\lambda n_\ell}\right). \tag{12}$$

This step is described in Algorithm 1. We can use any solver for this approximate optimization, and in our experiments, we choose ProxSDCA.

The global step is to synchronize all local solutions, which requires communication among the machines. This is achieved by optimizing the following dual objective with respect to all $\beta = \{\beta_\ell\}$:

$$\beta^{(t)} \in \arg\max_\beta D(\alpha^{(t)}, \beta). \tag{13}$$

Proposition 4 *Given v, let $w(v)$ be the unique solution of the following optimization problem*

$$w(v) = \arg\min_w \left[-\lambda n w^\top v + \lambda n g(w) + h(w)\right] \tag{14}$$

that satisfies

$$\lambda n \nabla g(w) + \nabla h(w) = \lambda n v$$

for some subgradients $\nabla g(w)$ *and* $\nabla h(w) = \rho$ *at* $w = w(v)$*. Then* $\bar{\beta}(v) = \rho$ *is a solution of*

$$\max_b \left[-\lambda n g^* \left(v - \frac{b}{\lambda n} \right) - h^*(b) \right],$$

and

$$w(v) = \nabla g^* \left(v - \frac{\bar{\beta}(v)}{\lambda n} \right).$$

Proposition 5 *Given* α*, a solution of*

$$\max_\beta D(\alpha, \beta)$$

can be obtained by setting

$$\beta_\ell = \lambda n_\ell \left(v_\ell(\alpha_{(\ell)}) - v(\alpha) + \frac{\bar{\beta}(v(\alpha))}{\lambda n} \right)$$

where $\bar{\beta}(v(\alpha))$ *is defined in Proposition 4,*

$$v(\alpha) = \frac{\sum_{i=1}^n X_i \alpha_i}{\lambda n}, \qquad v_\ell(\alpha_{(\ell)}) = \frac{\sum_{i \in S_\ell} X_i \alpha_i}{\lambda n_\ell}.$$

Moreover, if we let

$$w = w(\alpha, \beta) = w(v(\alpha)) = \nabla g^* \left(v(\alpha) - \frac{\bar{\beta}(v(\alpha))}{\lambda n} \right),$$

where $w(v)$ *is defined in Proposition 4, and*

$$w_\ell = w_\ell(\alpha_{(\ell)}, \beta_\ell) = \nabla g^* \left(v_\ell(\alpha_{(\ell)}) - \frac{\beta_\ell}{\lambda n_\ell} \right),$$

then $w = w_\ell$ *for all* ℓ*, and*

$$P(w) - D(\alpha, \beta) = \sum_{\ell=1}^m [\tilde{P}_\ell(w_\ell | \beta_\ell) - \tilde{D}_\ell(\alpha_{(\ell)} | \beta_\ell)].$$

According to Proposition 5, the solution of (13) is given by

$$\beta_\ell^{(t)} = \lambda n_\ell \left(v_\ell^{(t)} - v^{(t)} + \frac{\rho^{(t)}}{\lambda n} \right),$$

where

$$v^{(t)} = \sum_{\ell=1}^m \frac{n_\ell}{n} v_\ell^{(t)} = v^{(t-1)} + \sum_{\ell=1}^m \frac{n_\ell}{n} \Delta v_\ell^{(t)},$$

Algorithm 2 Distributed Alternating Dual Maximization (DADM)

Input: Objective $P(w)$, target duality gap ϵ, warm start variables $w^{\text{init}}, \alpha^{\text{init}}, \beta^{\text{init}}, v^{\text{init}}$, (if not specified, set $w^{\text{init}} = 0, \alpha^{\text{init}} = 0, \beta^{\text{init}} = 0, v^{\text{init}} = 0$), .
Initialize: let $w^{(0)} = w^{\text{init}}$, $\alpha^{(0)} = \alpha^{\text{init}}$, $\beta^{(0)} = \beta^{\text{init}}$, $v^{(0)} = v^{\text{init}}$.
for $t = 1, 2, \ldots$ **do**
  (**Local step**)
  for all machines $\ell = 1, 2, ..., m$ **in parallel do**
    call an arbitrary local procedure, such as Algorithm 1
  end for
  (**Global step**)
  Aggregate $v^{(t)} = v^{(t-1)} + \sum_{\ell=1}^m \frac{n_\ell}{n} \Delta v_\ell^{(t)}$
  Compute $\tilde{v}^{(t)}$ according to (15)
  Let $\Delta \tilde{v}^{(t)} = \tilde{v}^{(t)} - \tilde{v}^{(t-1)}$
  for all machines $\ell = 1, 2, ..., m$ **in parallel do**
    update local parameter $\tilde{v}_\ell^{(t)} = \tilde{v}_\ell^{(t-1)} + \Delta \tilde{v}^{(t)}$
  end for
  Stopping condition: Stop if $P(w^{(t)}) - D(\alpha^{(t)}, \beta^{(t)}) \le \epsilon$.
end for
return $w^{(t)} = \nabla g^*(\tilde{v}^{(t)})$, $\alpha^{(t)}$, $\beta^{(t)}$, $v^{(t)}$, and the duality gap $P(w^{(t)}) - D(\alpha^{(t)}, \beta^{(t)})$.

and $\rho^{(t)} = \nabla h(w^{(t)})$ is a subgradient of h at the solution $w^{(t)}$ of

$$w^{(t)} = \arg\min_w \left[-\lambda n w^\top v^{(t)} + \lambda n g(w) + h(w) \right],$$

that can achieve the first order optimality condition

$$-\lambda n v^{(t)} + \lambda n \nabla g(w^{(t)}) + \rho^{(t)} = 0$$

for some subgradient $\nabla g(w^{(t)})$.

The definition of $\tilde{v}$ implies that after each global update, we have

$$\tilde{v}_\ell^{(t)} = \tilde{v}^{(t)} = v^{(t)} - \frac{\rho^{(t)}}{\lambda n} = \nabla g(w^{(t)}), \quad \text{for all } \ell = 1, \ldots, m. \tag{15}$$

Since the objective (12) for the local step on each machine only depends on the mini-batch Q_ℓ (sampled from S_ℓ) and the vector $\tilde{v}_\ell^{(t)}$, which needs to be synchronized at each global step, we know from (15) that at each time t, we can pass the same vector $\tilde{v}^{(t)}$ as $\tilde{v}_\ell^{(t)}$ to all nodes. In practice, it may be beneficial to pass $\Delta \tilde{v}^{(t)}$ instead, especially when $\Delta \tilde{v}^{(t)}$ is sparse but $\tilde{v}^{(t)}$ is dense. Put things together, the local-global DADM iterations can be summarized in Algorithm 2.

If we consider the special case of $h(w) = 0$, the solution of (15) is simply $\tilde{v}_\ell^{(t)} = \tilde{v}^{(t)} = v^{(t)}$, and the global step in Algorithm 2 can be simplified as first aggregating updates by

$$\Delta \tilde{v}^{(t)} = \Delta v^{(t)} = \sum_{\ell=1}^m \frac{n_\ell}{n} \Delta v_\ell^{(t)},$$

and then updating local parameters in parallel. Further, if $h(w) = 0$ and the data partition is balanced, that is n_ℓ are identical for all $\ell = 1, \ldots, m$, it can be verified that the DADM procedure (ignoring the mini-batch variation) is equivalent to CoCoA$^+$. Therefore the framework presented here may be regarded as an alternative interpretation.

Moreover, when the added regularization in (1) is complex and might involves more than one non-smooth term, considering the splitting of $g(w)$ and $h(w)$ can bring computational advantages. For example, to promote both sparsity and group sparsity in the predictor we often use the sparse group lasso regularization (Friedman et al., 2010), where a combination of L_1 norm and mixed L_2/L_1 norm (group sparse norm) is introduced: $\lambda_1 \sum_{\mathcal{G}} \|w_{\mathcal{G}}\|_2 + \lambda_2 \|w\|_1 + \lambda_3/2 \|w\|_2^2$, where we add a slight L_2 regularization to make it strongly convex, as did in (Shalev-Shwartz and Zhang, 2014). The proximal mapping with respect to the sparse group lasso regularization function does not have closed form solution, thus often relies on iterative minimization steps, but there are closed form proximal mapping with respect to either L_2-L_1 norm or the group norm. Thus if we simply set $h(w) = 0$ and $\lambda g(w) = \lambda_1 \sum_{\mathcal{G}} \|w_{\mathcal{G}}\|_2 + \lambda_2 \|w\|_1 + \lambda_3/2 \|w\|_2^2$, then both the local optimization update (12) and global synchronization step (14) will not have closed form solution. However, if we assign the group norm on $h(w)$ such that $h(w) = \lambda_1 \sum_{\mathcal{G}} \|w_{\mathcal{G}}\|_2$, and hence $\lambda g(w) = \lambda_2 \|w\|_1 + \lambda_3/2 \|w\|_2^2$, the local updates steps (12) will enjoy closed form update, which makes the implementation much easier and we only need to use iterative minimization on the (rare) global synchronization step (14).

7. Convergence Analysis

Let w^* be the optimal solution for the primal problem $P(w)$ and (α^*, β^*) be the optimal solution for the dual problem $D(\alpha, \beta)$ respectively. For the primal solution $w^{(t)}$ and the dual solution $(\alpha^{(t)}, \beta^{(t)})$ at iteration t, we define the **primal sub-optimality** as

$$\epsilon_P^{(t)} := P(w^{(t)}) - P(w^*),$$

and the **dual sub-optimality** as

$$\epsilon_D^{(t)} := D(\alpha^*, \beta^*) - D(\alpha^{(t)}, \beta^{(t)}).$$

Due to the close relationship of the distributed dual formulation and the single-machine dual formulation, an analysis of DADM can be obtained by directly generalizing that of SDCA. We consider two kinds of loss functions, smooth loss functions that imply fast linear convergence and general L-Lipschitz loss functions. For the following two theorems we always assume that g is 1-strongly convex w.r.t $\|\cdot\|_2$, $\|X_i\|_2^2 \leq R$ for all i, $M_\ell = |Q_\ell|$ is fixed on each machine, and our local procedure optimizes $\tilde{D}_{Q_\ell}^{(t)}$ sufficiently well on each machine such that $\tilde{D}_{Q_\ell}^{(t)}(\Delta\alpha_{Q_\ell}) \geq \tilde{D}_{Q_\ell}^{(t)}(\Delta\tilde{\alpha}_{Q_\ell})$, where $\Delta\tilde{\alpha}_{Q_\ell}$ is given by a special choice in each theorem.

Theorem 6 *Assume that each ϕ_i is $(1/\gamma)$-smooth w.r.t $\|\cdot\|_2$ and $\Delta\tilde{\alpha}_{Q_\ell}$ is given by*

$$\Delta\tilde{\alpha}_i := s_\ell(u_i^{(t-1)} - \alpha_i^{(t-1)}), \quad \text{for all } i \in Q_\ell,$$

where $u_i^{(t-1)} := -\nabla\phi_i(X_i^\top w_\ell^{(t-1)})$ and $s_\ell := \frac{\gamma\lambda n_\ell}{\gamma\lambda n_\ell + M_\ell R} \in [0, 1]$. To reach an expected duality gap of $\mathbb{E}[P(w^{(T)}) - D(\alpha^{(T)}, \beta^{(T)})] \leq \epsilon$, every T satisfying the following condition is sufficient,

$$T \geq \left(\frac{R}{\gamma\lambda} + \max_\ell \frac{n_\ell}{M_\ell}\right) \log\left(\left(\frac{R}{\gamma\lambda} + \max_\ell \frac{n_\ell}{M_\ell}\right) \cdot \frac{\epsilon_D^{(0)}}{\epsilon}\right). \tag{16}$$

Theorem 7 *Assume that each ϕ_i is L-Lipschitz w.r.t $\|\cdot\|_2$, and $\Delta\tilde{\alpha}_{Q_\ell}$ is given by*

$$\Delta\tilde{\alpha}_i := \frac{q n_\ell}{M_\ell}(u_i^{(t-1)} - \alpha_i^{(t-1)}), \quad \text{for all } i \in Q_\ell,$$

where $-u_i^{(t-1)} := \nabla\phi_i(X_i^\top w_\ell^{(t-1)})$ and $q \in [0, \min_\ell(M_\ell/n_\ell)]$. To reach an expected normalized duality gap of $\mathbb{E}\left[\frac{P(\overline{w}) - D(\overline{\alpha}, \overline{\beta})}{n}\right] \leq \epsilon$, every T satisfying the following condition is sufficient,

$$T \geq T_0 + \tilde{n} + \frac{G}{\lambda\epsilon} \geq \max\left\{0, \lceil \tilde{n} \log(2\lambda\tilde{n}\frac{\epsilon_D^{(0)}}{nG})\rceil\right\} + \tilde{n} + \frac{5G}{\lambda\epsilon}, \tag{17}$$

where $T_0 \geq \max\{t_0, \frac{4G}{\lambda\epsilon} - 2\tilde{n} + t_0\}$, $t_0 = \max\{0, \lceil \tilde{n} \log(2\lambda\tilde{n}\frac{\epsilon_D^{(0)}}{nG})\rceil\}$, $\tilde{n} = \max_\ell(n_\ell/M_\ell)$, $G = 4RL^2$, and $\overline{w}, \overline{\alpha}, \overline{\beta}$ represent either the average vector or a randomly chosen vector of $w^{(t-1)}, \alpha^{(t-1)}, \beta^{(t-1)}$ over $t \in \{T_0+1, ..., T\}$ respectively, such as $\overline{\alpha} = \frac{1}{T-T_0}\sum_{t=T_0+1}^T \alpha^{(t-1)}$, $\overline{\beta} = \frac{1}{T-T_0}\sum_{t=T_0+1}^T \beta^{(t-1)}$, $\overline{w} = \frac{1}{T-T_0}\sum_{t=T_0+1}^T w^{(t-1)}$.

Remark 8 *Both Theorem 6 and Theorem 7 incorporate two key components: the term $\max_\ell \frac{n_\ell}{M_\ell}$ and the condition number term $\frac{1}{\lambda\gamma}$ or $\frac{L^2}{\lambda}$. When the term $\max_\ell \frac{n_\ell}{M_\ell}$ dominates the iteration complexity, we can speed up convergence and reduce the number of communications by increasing the number of machines m or the local mini-batch size M_ℓ. However, in some circumstances when the condition number is large, it will become the leading factor, and increasing m or M_ℓ will not contribute to the computation speedup. To tackle this problem, we develop the accelerated version of DADM in Section 8.*

Remark 9 *Our method is closely related to previous distributed extensions of SDCA. Theorems 6, 7 that provide theoretical guarantees for more general local updates achieve the same iteration complexity with the ones in DisDCA that only allows some special choices of local mini-batch updates. Compared with theoretical results of CoCoA$^+$ that are based on the Θ-approximate solution of the local dual subproblem, although the derived bounds are within the same scale, $\tilde{\mathcal{O}}(1/\epsilon)$ for Lipschitz losses and $\tilde{\mathcal{O}}(\log(1/\epsilon))$ for smooth losses, our bounds are different and complementary. The analysis of CoCoA$^+$ can provide better insights for more accurate solutions of the local sub-problems. While our analysis is based on the mini-batch setup, it can capture the contributions of the mini-batch size and the number of machines more explicitly.*

Remark 10 *Since the bounds are derived with a special choice of $\Delta\tilde{\alpha}_{Q_\ell}$, the actual performance of the algorithm can be significantly better than what is indicated by the bounds when the local duals are better optimized. For example, we can choose ProxSDCA in (Shalev-Shwartz and Zhang, 2014) as the local procedure and adopt the sequential update strategy as the local solver of CoCoA$^+$ does. This is also the one used in our experiments.*

Algorithm 3 Accelerated Distributed Alternating Dual Maximization (Acc-DADM).

Parameters κ, $\eta = \sqrt{\lambda/(\lambda+2\kappa)}$, $\nu = (1-\eta)/(1+\eta)$.
Initialize $v^{(0)} = y^{(0)} = w^{(0)} = 0$, $\alpha^{(0)} = 0$, $\xi_0 = (1+\eta^{-2})(P(0) - D(0,0))$.
for $t = 1, 2, \ldots, T_{\text{outer}}$ **do**

1. **Construct new objective**:
$$P_t(w) = \sum_{i=1}^{n} \phi_i(X_i^\top w) + \lambda n g(w) + h(w) + \frac{\kappa n}{2}\left\|w - y^{(t-1)}\right\|_2^2.$$

2. **Call DADM solver**:
$$(w^{(t)}, \alpha^{(t)}, \beta^{(t)}, v^{(t)}, \epsilon_t) = \text{DADM}(P_t, (\eta\xi_{t-1})/(2+2\eta^{-2}), w^{(t-1)}, \alpha^{(t-1)}, \beta^{(t-1)}, v^{(t-1)}).$$

3. **Update**:
$$y^{(t)} = w^{(t)} + \nu(w^{(t)} - w^{(t-1)}).$$

4. **Update**:
$$\xi_t = (1 - \eta/2)\xi_{t-1}.$$

end for
Return $w^{(T_{\text{outer}})}$.

8. Acceleration

Theorems 6, 7 all imply that when the condition number $\frac{1}{\gamma\lambda}$ or $\frac{L^2}{\lambda}$ is relatively small, DADM converges fast. However, the convergence may be slow when the condition number is large and dominates the iteration complexity. In fact, we observe empirically that the basic DADM method converges slowly when the regularization parameter λ is small. This phenomenon is also consistent with that of SDCA for the single-machine case. In this section, we introduce the Accelerated Distributed Alternating Dual Maximization (Acc-DADM) method that can alleviate the problem.

The procedure is motivated by (Shalev-Shwartz and Zhang, 2014), which employs an inner-outer iteration: at every iteration t, we solve a slightly modified objective, which adds a regularization term centered around the vector

$$y^{(t-1)} = w^{(t-1)} + \nu\left(w^{(t-1)} - w^{(t-2)}\right), \tag{18}$$

where $\nu \in [0, 1]$ is called the momentum parameter.

The accelerated DADM procedure (described in Algorithm 3) can be similarly viewed as an inner-outer algorithm, where DADM serves as the inner iteration. In the outer iteration, we adjust the regularization vector $y^{(t-1)}$. That is, at each outer iteration t, we define a modified local primal objective on each machine ℓ, which has the same form as the original local primal objective (9), except that $\tilde{g}_\ell(w_\ell)$ is modified to $\tilde{g}_{\ell_t}(w_\ell)$ that is defined by

$$\begin{aligned}\lambda n_\ell \tilde{g}_{\ell_t}(w_\ell) =& \lambda n_\ell g_t(w_\ell) + \beta_\ell^\top w_\ell,\\ \lambda g_t(w_\ell) =& \lambda g(w_\ell) + \frac{\kappa}{2}\|w_\ell - y^{(t-1)}\|_2^2.\end{aligned}$$

It follows that we will need to solve a modified dual at each local step with $g^*(\cdot)$ replaced by $g_t^*(\cdot)$ in the local dual problem (12). Therefore, compared to the basic DADM procedure, nothing changes other than $g^*(\cdot)$ being replaced by $g_t^*(\cdot)$ at each iteration. Specifically, when the number of machines m equals 1, this algorithm reduces to AccProxSDCA described in (Shalev-Shwartz and Zhang, 2014). Thus Acc-DADM can be naturally regarded as the distributed generalization of the single-machine AccProxSDCA. Moreover, Acc-DADM also allows arbitrary local procedures as DADM does.

Our empirical studies show that Acc-DADM significantly outperforms DADM in many cases. There are probably two reasons. One reason is the use of a modified regularizer $g_t(w)$ that is more strongly convex than the original regularizer $g(w)$ when κ is much larger than λ. The other reason is closely related to the distributed setting considered in this paper. Observe that in the modified local primal objective

$$\tilde{P}_{\ell_t}(w_\ell|\beta_\ell) := \tilde{P}_\ell(w_\ell|\beta_\ell) + \frac{\kappa n_\ell}{2}\|w_\ell - y^{(t-1)}\|_2^2,$$

the first term corresponds to the original local primal objective and the second term is an extra regularization due to acceleration that constrains w_ℓ to be close to $y^{(t-1)}$. The effect is that different local problems become more similar to each other, which stabilize the overall system.

8.1 Theoretical Results of Acc-DADM for Smooth Losses

The following theorem establishes the computation efficiency guarantees for Acc-DADM.

Theorem 11 *Assume that each ϕ_i is $(1/\gamma)$-smooth, and g is 1-strongly convex w.r.t $\|\cdot\|_2$, $\|X_i\|_2^2 \le R$ for all i, $M_\ell = |Q_\ell|$ is fixed on each machine. To obtain expected ϵ primal sub-optimality:*

$$\mathbb{E}[P(w^{(t)})] - P(w^*) \le \epsilon,$$

it is sufficient to have the following number of stages in Algorithm 3

$$T_{\text{outer}} \ge 1 + \frac{2}{\eta}\log\left(\frac{\xi_0}{\epsilon}\right) = 1 + \sqrt{\frac{4(\lambda+2\kappa)}{\lambda}}\left(\log\left(\frac{2\lambda+2\kappa}{\lambda}\right) + \log\left(\frac{P(0)-D(0,0)}{\epsilon}\right)\right),$$

and the number of inner iterations in DADM at each stage:

$$T_{\text{inner}} \ge \left(\frac{R}{\gamma(\lambda+\kappa)} + \max_\ell \frac{n_\ell}{M_\ell}\right)\left(\log\left(\frac{R}{\gamma(\lambda+\kappa)} + \max_\ell \frac{n_\ell}{M_\ell}\right) + 7 + \frac{5}{2}\log\left(\frac{\lambda+2\kappa}{\lambda}\right)\right).$$

In particular, suppose we assume $n_1 = n_2 = \ldots = n_m$, and $M_1 = M_2 = \ldots = M_m = b$, then the total vector computations for each machine are bounded by

$$\tilde{\mathcal{O}}(T_{\text{outer}} T_{\text{inner}} b) = \tilde{\mathcal{O}}\left(\left(1 + \sqrt{\frac{\kappa+\lambda}{\lambda}}\right)\left(\frac{R}{\gamma(\lambda+\kappa)} + \frac{n}{mb}\right)b\right).$$

Remark 12 *When $\kappa = 0$, then the guarantees reduce to DADM. However, DADM only enjoys linear speedup over ProxSDCA when the number of machines satisfies $m \le (n\gamma\lambda)/R$,*

and being able to obtain sub-linear speedup when $\frac{R}{\lambda\gamma} = \mathcal{O}(n)$*. Besides enjoying the properties described above as DADM, if we choose* κ *in Algorithm 3 as* $\kappa = \frac{mR}{\gamma n} - \lambda$*, and* $b = 1$*, then the total vector computations for each machine are bounded by*

$$\tilde{\mathcal{O}}\left(\sqrt{\frac{Rm}{\gamma n\lambda}}\left(\frac{n}{m}\right)\right) = \tilde{\mathcal{O}}\left(\sqrt{\frac{Rn}{\gamma\lambda m}}\right),$$

which means Acc-DADM can be much faster than DADM when the condition number is large and always obtain a reduction of computations over the single-machine AccProxSDCA by a factor of $\tilde{\mathcal{O}}(\sqrt{1/m})$*.*

8.2 Acceleration for Non-smooth, Lipschitz Losses

Theorem 11 establishes the rate of convergence for smooth loss functions, but the acceleration framework can also be used on non-smooth, Lipschitz loss functions. The main idea is to use the Nesterov's smoothing technique (Nesterov, 2005) to construct a smooth approximation of the non-smooth function $\phi_i(\cdot)$, by adding a strongly-convex regularization term on the conjugate of $\phi_i(\cdot)$:

$$\tilde{\phi}_i^*(-\alpha_i) := \phi_i^*(-\alpha_i) + \frac{\gamma}{2}\|\alpha_i\|_2^2,$$

by the property of conjugate functions (e.g. Lemma 2 in (Shalev-Shwartz and Zhang, 2014)), we know $\tilde{\phi}_i(\cdot)$, as the conjugate function of $\tilde{\phi}_i^*(\cdot)$ is $(1/\gamma)$-smooth, and

$$0 \le \tilde{\phi}_i(u_i) - \phi_i(u_i) \le \frac{\gamma L^2}{2}.$$

Then instead of the original function with non-smooth losses, we minimize the smoothed objective:

$$\min_{w\in\mathbb{R}^d}\left[\hat{P}(w) := \sum_{i=1}^n \tilde{\phi}_i(X_i^\top w) + \lambda n g(w) + h(w)\right]. \tag{19}$$

The following corollary establishes the computation efficiency guarantees for Acc-DADM on non-smooth, Lipschitz loss functions.

Corollary 13 *Assume that each* ϕ_i *is L-Lipschitz, and* g *is 1-strongly convex w.r.t* $\|\cdot\|_2$*,* $\|X_i\|_2^2 \le R$ *for all* i*,* $M_\ell = |Q_\ell|$ *is fixed on each machine. To obtain expected* ϵ *normalized primal sub-optimality:*

$$\mathbb{E}\left[\frac{P(w^{(t)})}{n}\right] - \frac{P(w^*)}{n} \le \epsilon,$$

it is sufficient to run Algorithm 3 on the smoothed objective (19)*, with*

$$\gamma = \frac{\epsilon}{L^2},$$

and the following number of stages,

$$T_{\text{outer}} \ge 1 + \frac{2}{\eta}\log\left(\frac{2\xi_0}{\epsilon}\right) = 1 + \sqrt{\frac{4(\lambda+2\kappa)}{\lambda}}\left(\log\left(\frac{2\lambda+2\kappa}{\lambda}\right) + \log\left(\frac{2(P(0)-D(0,0))}{\epsilon}\right)\right),$$

and the number of inner iterations in DADM at each stage:

$$T_{\text{inner}} \ge \left(\frac{L^2R}{\epsilon(\lambda+\kappa)} + \max_\ell \frac{n_\ell}{M_\ell}\right)\left(\log\left(\frac{L^2R}{\epsilon(\lambda+\kappa)} + \max_\ell \frac{n_\ell}{M_\ell}\right) + 7 + \frac{5}{2}\log\left(\frac{\lambda+2\kappa}{\lambda}\right)\right).$$

In particular, suppose we assume $n_1 = n_2 = \ldots = n_m$*, and* $M_1 = M_2 = \ldots = M_m = b$*, then the total vector computations for each machine are bounded by*

$$\tilde{\mathcal{O}}(T_{\text{outer}}T_{\text{inner}}b) = \tilde{\mathcal{O}}\left(\left(1 + \sqrt{\frac{\kappa+\lambda}{\lambda}}\right)\left(\frac{L^2R}{\epsilon(\lambda+\kappa)} + \frac{n}{mb}\right)b\right).$$

Remark 14 *When* $\kappa = 0$*, then the guarantees reduce to DADM for Lipschitz losses. Moreover, when* $L^2Rm \ge n\epsilon\lambda$*, if we choose* κ *in Algorithm 3 as* $\kappa = \frac{mL^2R}{n\epsilon} - \lambda$*, and* $b = 1$*, then the total vector computations for each machine are bounded by*

$$\tilde{\mathcal{O}}\left(\sqrt{\frac{L^2Rm}{n\epsilon\lambda}}\left(\frac{n}{m}\right)\right) = \tilde{\mathcal{O}}\left(L\sqrt{\frac{Rn}{\epsilon\lambda m}}\right),$$

which means Acc-DADM can be much faster than DADM when ϵ *is small and always obtain a reduction of computations over the single-machine AccProxSDCA by a factor of* $\tilde{\mathcal{O}}(\sqrt{1/m})$*.*

9. Proofs

In this section, we first present proofs about several previous propositions to establish our framework solidly. Then based on our new distributed dual formulation, we directly generalize the analysis of SDCA and adapt it to DADM in the commonly used mini-batch setup. Finally, we describe the proof for the theoretical guarantees of Acc-DADM.

9.1 Proof of Proposition 1

Proof Given any set of parameters $(w; \{w_\ell\}; \{u_i\}; \{\alpha_i\}; \{\beta_\ell\})$, we have

$$\begin{aligned}&\min_{w;\{w_\ell\};\{u_i\}} J(w; \{w_\ell\}; \{u_i\}; \{\alpha_i\}; \{\beta_\ell\}) \\ &= \underbrace{\min_{w;\{w_\ell\}} \sum_{\ell=1}^m \left[\sum_{i\in S_\ell} \min_{u_i}\left(\phi_i(u_i) + \alpha_i^\top(u_i - X_i^\top w_\ell)\right) + \lambda n_\ell g(w_\ell) + \beta_\ell^\top(w_\ell - w)\right]}_{A} + h(w),\end{aligned}$$

where the minimum is achieved at $\{u_i\}$ such that $\nabla\phi_i(u_i)+\alpha_i=0$. By eliminating u_i we obtain

$$\begin{aligned}A=&\min_{w;\{w_\ell\}}\sum_{\ell=1}^m\left[\sum_{i\in S_\ell}\left(-\phi_i^*(-\alpha_i)-\alpha_i^\top X_i^\top w_\ell\right)+\lambda n_\ell g(w_\ell)+\beta_\ell^\top(w_\ell-w)\right]+h(w)\\ =&\underbrace{\min_w\sum_{\ell=1}^m\min_{w_\ell}\left[\sum_{i\in S_\ell}-\phi_i^*(-\alpha_i)-\left(\sum_{i\in S_\ell}X_i\alpha_i-\beta_\ell\right)^\top w_\ell+\lambda n_\ell g(w_\ell)-\beta_\ell^\top w\right]+h(w)}_{B},\end{aligned}$$

where minimum is achieved at $\{w_\ell\}$ such that $-\left(\sum_{i\in S_\ell}X_i\alpha_i-\beta_\ell\right)+\lambda n_\ell\nabla g(w_\ell)=0$. By eliminating w_ℓ we obtain

$$\begin{aligned}B=&\min_w\sum_{\ell=1}^m\left[\sum_{i\in S_\ell}-\phi_i^*(-\alpha_i)-\lambda n_\ell g^*\left(\frac{\sum_{i\in S_\ell}X_i\alpha_i-\beta_\ell}{\lambda n_\ell}\right)-\beta_\ell^\top w\right]+h(w)\\ =&\underbrace{\sum_{\ell=1}^m\left[\sum_{i\in S_\ell}-\phi_i^*(-\alpha_i)-\lambda n_\ell g^*\left(\frac{\sum_{i\in S_\ell}X_i\alpha_i-\beta_\ell}{\lambda n_\ell}\right)\right]-h^*\left(\sum_\ell\beta_\ell\right)}_{D(\alpha,\beta)},\end{aligned}$$

where the minimizer is achieved at w such that $-\sum_\ell\beta_\ell+\nabla h(w)=0$. This completes the proof. ■

9.2 Proof of Proposition 2

Proof Given any w, if we take $u_i=X_i^\top w_\ell$ and $w_\ell=w$ for all i and ℓ, then $P(w)=J(w;\{w_\ell\};\{u_i\};\{\alpha_i\};\{\beta_\ell\})$ for arbitrary $(\{\alpha_i\};\{\beta_\ell\})$. It follows from Proposition 1 that

$$P(w)=J(w;\{w_\ell\};\{u_i\};\{\alpha_i\};\{\beta_\ell\})\geq D(\alpha,\beta).$$

w^* is the minimizer of $P(w)$. When $w=w^*$, we may set $u_i=u_i^*=X_i^\top w^*$ and $w_\ell=w_\ell^*=w^*$. From the first order optimality condition, we can obtain

$$\sum_i X_i\nabla\phi_i(u_i^*)+\sum_\ell\lambda n_\ell\nabla g(w_\ell^*)+\nabla h(w^*)=0.$$

If we take $\alpha_i^*=-\nabla\phi_i(u_i^*)$ and $\beta_\ell^*=\sum_{i\in S_\ell}X_i\alpha_i^*-\lambda n_\ell\nabla g(w_\ell^*)$ for some subgradients, then it is not difficult to check that all equations in (6) are satisfied. It follows that we can achieve equality in Proposition 1 as

$$P(w^*)=J(w^*;\{w_\ell^*\};\{u_i^*\};\{\alpha_i^*\};\{\beta_\ell^*\})=D(\alpha^*,\beta^*).$$

This means that zero duality gap can be achieved with w^*. It is easy to verify that (α^*,β^*) maximizes $D(\alpha,\beta)$, since for any (α,β), we have

$$\begin{aligned}D(\alpha,\beta)&\leq J(w^*;\{w_\ell^*\};\{u_i^*\};\{\alpha_i\};\{\beta_\ell\})\\&=P(w^*)\\&=D(\alpha^*,\beta^*).\end{aligned}$$

■

9.3 Proof of Proposition 3

Proof We have the decompositions

$$D(\alpha,\beta)=\sum_{\ell=1}^m\tilde{D}_\ell(\alpha_{(\ell)}|\beta_\ell)-h^*\left(\sum_\ell\beta_\ell\right),$$

and

$$P(w)=\sum_{\ell=1}^m\tilde{P}_\ell(w_\ell|\beta_\ell)-\left(\sum_\ell\beta_\ell\right)^\top w+h(w).$$

It follows that the duality gap

$$P(w)-D(\alpha,\beta)=\sum_{\ell=1}^m[\tilde{P}_\ell(w_\ell|\beta_\ell)-\tilde{D}_\ell(\alpha_{(\ell)}|\beta_\ell)]+h^*\left(\sum_\ell\beta_\ell\right)+h(w)-\left(\sum_\ell\beta_\ell\right)^\top w.$$

Note that the definition of convex conjugate function implies that

$$h^*\left(\sum_\ell\beta_\ell\right)+h(w)-\left(\sum_\ell\beta_\ell\right)^\top w\geq 0,$$

and the equality holds when $\nabla h(w)=\sum_\ell\beta_\ell$. This implies the desired result. ■

9.4 Proof of Proposition 4

Proof It is easy to check by using the duality that for any b and w:

$$\begin{aligned}&-\lambda n g^*\left(v-\frac{b}{\lambda n}\right)-h^*(b)\\ \leq&\left[-\lambda n w^\top\left(v-\frac{b}{\lambda n}\right)+\lambda n g(w)\right]+\left[-b^\top w+h(w)\right]\\ =&-\lambda n w^\top v+\lambda n g(w)+h(w),\end{aligned}$$

and the equality holds if $b=\nabla h(w)$ and $v-\frac{b}{\lambda n}=\nabla g(w)$ for some subgradients. Based on the assumptions, the equality can be achieved at $b=\bar{\beta}(v)=\nabla h(w(v))$ and $w=w(v)$. This proves the desired result by noticing that $v-\frac{b}{\lambda n}=\nabla g(w)$ implies that $w=\nabla g^*(v-b/(\lambda n))$. ■

9.5 Proof of Proposition 5

Proof Since α is fixed, we know that the problem $\max_\beta D(\alpha,\beta)$ is equivalent to

$$\max_\beta \left[\sum_{\ell=1}^m -\lambda n_\ell g^*\left(v_\ell(\alpha_{(\ell)}) - \frac{\beta_\ell}{\lambda n_\ell}\right) - h^*\left(\sum_\ell \beta_\ell\right)\right].$$

Now by using Jensen's inequality, we obtain for any (β'_ℓ):

$$\begin{aligned}
&\sum_{\ell=1}^m -\lambda n_\ell g^*\left(v_\ell(\alpha_{(\ell)}) - \frac{\beta'_\ell}{\lambda n_\ell}\right) - h^*\left(\sum_\ell \beta'_\ell\right)\\
\leq& -\lambda n g^*\left(\sum_{\ell=1}^m \frac{n_\ell}{n}\frac{\sum_{i\in S_\ell} X_i\alpha_i - \beta'_\ell}{\lambda n_\ell}\right) - h^*\left(\sum_\ell \beta'_\ell\right)\\
=& -\lambda n g^*\left(v(\alpha) - \frac{\sum_\ell \beta'_\ell}{\lambda n}\right) - h^*\left(\sum_\ell \beta'_\ell\right)\\
\leq& -\lambda n g^*\left(v(\alpha) - \frac{\bar\beta(v(\alpha))}{\lambda n}\right) - h^*\left(\bar\beta(v(\alpha))\right).
\end{aligned} \tag{20}$$

In the above derivation, the last inequality has used Proposition 4. Here the equalities can be achieved when

$$v_\ell(\alpha_{(\ell)}) - \frac{\beta'_\ell}{\lambda n_\ell} = v(\alpha) - \frac{\bar\beta(v(\alpha))}{\lambda n}$$

for all ℓ, which can be obtained with the choice of $\{\beta'_\ell\} = \{\beta_\ell\}$ given in the statement of the proposition. ■

9.6 Proof of Theorem 6

The following result is the mini-batch version of a related result in the analysis of ProxSDCA, which we apply to any local machine ℓ. The proof is included for completeness.

Lemma 15 *Assume that ϕ_i^* is γ-strongly convex w.r.t $\|\cdot\|_2$ (where γ can be zero) and g^* is 1-smooth w.r.t $\|\cdot\|_2$. Every local step, we randomly pick a mini-batch $Q_\ell \subset S_\ell$, whose size is $M_\ell := |Q_\ell|$, and optimize w.r.t dual variables $\alpha_i, i \in Q_\ell$. Then, using the simplified notation*

$$P_\ell(w_\ell^{(t-1)}) = \tilde P_\ell(w_\ell^{(t-1)}|\beta_\ell^{(t-1)}), \qquad D_\ell(\alpha_{(\ell)}^{(t-1)}) = \tilde D_\ell(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)}),$$

we have

$$\mathbb{E}\left[D_\ell(\alpha_{(\ell)}^{(t)}) - D_\ell(\alpha_{(\ell)}^{(t-1)})\right] \geq \frac{s_\ell M_\ell}{n_\ell}\mathbb{E}\left[P_\ell(w_\ell^{(t-1)}) - D_\ell(\alpha_{(\ell)}^{(t-1)})\right] - \frac{s_\ell^2 M_\ell^2}{2\lambda n_\ell^2} G_\ell^{(t)}$$

where

$$\begin{aligned}
G_\ell^{(t)} &:= \sum_{i\in S_\ell}\left[\|X_i\|_2^2 - \frac{\gamma\lambda n_\ell(1-s_\ell)}{M_\ell s_\ell}\right]\mathbb{E}\left[\|u_i^{(t-1)} - \alpha_i^{(t-1)}\|_2^2\right]\\
\Delta\tilde\alpha_i &:= \alpha_i^{(t)} - \alpha_i^{(t-1)} = s_\ell(u_i^{(t-1)} - \alpha_i^{(t-1)}), \quad \textit{for all } i \in Q_\ell,
\end{aligned}$$

and $-u_i^{(t-1)} = \nabla\phi_i(X_i^\top w_\ell^{(t-1)})$, $s_\ell \in [0,1]$.

Proof Since only the elements in Q_ℓ are updated, the improvement in the dual objective can be written as

$$\begin{aligned}
&D_\ell(\alpha_{(\ell)}^{(t)}) - D_\ell(\alpha_{(\ell)}^{(t-1)})\\
=&\left(\sum_{i\in Q_\ell} -\phi_i^*(-\alpha_i^{(t)}) - \lambda n_\ell g^*\left(v_\ell^{(t-1)} + (\lambda n_\ell)^{-1}\sum_{i\in Q_\ell} X_i\Delta\tilde\alpha_i\right)\right)\\
&-\left(\sum_{i\in Q_\ell} -\phi_i^*(-\alpha_i^{(t-1)}) - \lambda n_\ell g^*\left(v_\ell^{(t-1)}\right)\right)\\
\geq&\underbrace{\left(\sum_{i\in Q_\ell} -\phi_i^*(-\alpha_i^{(t-1)} - \Delta\tilde\alpha_i) - \nabla g^*(v_\ell^{(t-1)})^\top\left(\sum_{i\in Q_\ell} X_i\Delta\tilde\alpha_i\right) - \frac{1}{2\lambda n_\ell}\|\sum_{i\in Q_\ell} X_i\Delta\tilde\alpha_i\|_2^2\right)}_{A}\\
&-\underbrace{\left(\sum_{i\in Q_\ell} -\phi_i^*(-\alpha_i^{(t-1)})\right)}_{B},
\end{aligned}$$

where we have used the fact the g^* is 1-smooth in the derivation of the inequality.

By the definition of the update in the algorithm, and the definition of $\Delta\tilde\alpha_i = s_\ell(u_i^{(t-1)} - \alpha_i^{(t-1)}), s_\ell \in [0,1]$, we have

$$\begin{aligned}
A \geq& \sum_{i\in Q_\ell} -\phi_i^*(-(\alpha_i^{(t-1)} + s_\ell(u_i^{(t-1)} - \alpha_i^{(t-1)}))\\
&-\nabla g^*(v^{(t-1)})^\top\left(\sum_{i\in Q_\ell} X_i s_\ell(u_i^{(t-1)} - \alpha_i^{(t-1)})\right)\\
&-\frac{1}{2\lambda n_\ell}\|\sum_{i\in Q_\ell} X_i s_\ell(u_i^{(t-1)} - \alpha_i^{(t-1)})\|_2^2
\end{aligned} \tag{21}$$

From now on, we omit the superscript $(t-1)$. Since ϕ_i^* is γ-strongly convex w.r.t $\|\cdot\|_2$, we have

$$\begin{aligned}
\phi_i^*(-(\alpha_i + s_\ell(u_i - \alpha_i))) &= \phi^*(s_\ell(-u_i) + (1-s_\ell)(-\alpha_i))\\
&\leq s_\ell\phi^*(-u_i) + (1-s_\ell)\phi_i^*(-\alpha_i) - \frac{\gamma}{2}s_\ell(1-s_\ell)\|u_i - \alpha_i\|_2^2
\end{aligned} \tag{22}$$

Bringing Eq. (22) into Eq. (21), we get

$$\begin{aligned}
A \geq & \sum_{i\in Q_\ell}\left(-s_\ell\phi_i^*(-u_i)-(1-s_\ell)\phi_i^*(-\alpha_i)+\frac{\gamma}{2}s_\ell(1-s_\ell)\|u_i-\alpha_i\|_2^2\right)\\
& -w_\ell^\top\left(\sum_{i\in Q_\ell}s_\ell X_i(u_i-\alpha_i)\right)-\frac{1}{2\lambda n_\ell}\|\sum_{i\in Q_\ell}s_\ell X_i(u_i-\alpha_i)\|_2^2\\
\geq & \underbrace{\left(\sum_{i\in Q_\ell}-\phi_i^*(-\alpha_i)\right)}_{B}+\sum_{i\in Q_\ell}\left(s_\ell\left(w_\ell^\top X_i(-u_i)-\phi_i^*(-u_i)\right)+s_\ell\phi_i^*(-\alpha_i)+s_\ell w_\ell^\top X_i\alpha_i\right)\\
& +\sum_{i\in Q_\ell}\frac{\gamma}{2}s_\ell(1-s_\ell)\|u_i-\alpha_i\|_2^2-\sum_{i\in Q_\ell}\frac{M_\ell\|X_i(u_i-\alpha_i)\|_2^2}{2\lambda n_\ell}s_\ell^2,
\end{aligned}$$

where we get the second inequality according to the fact that $\|\sum_{i\in Q_\ell}a_i\|_2^2\leq\sum_{i\in Q_\ell}M_\ell\|a_i\|_2^2$.

Since we choose $-u_i=\nabla\phi_i(X_i^\top w_\ell)$, for some subgradients $\nabla\phi_i(X_i^\top w_\ell)$, which yields $w_\ell^\top X_i(-u_i)-\phi_i^*(-u_i)=\phi_i(X_i^\top w_\ell)$, then we obtain

$$\begin{aligned}
A-B\geq & \sum_{i\in Q_\ell}s_\ell\left[\phi_i(X_i^\top w_\ell)+\phi_i^*(-\alpha_i)+w_\ell^\top X_i\alpha_i\right]\\
& +\sum_{i\in Q_\ell}s_\ell\|u_i-\alpha_i\|_2^2\left[\frac{\gamma(1-s_\ell)}{2}-\frac{s_\ell M_\ell\|X_i\|_2^2}{2\lambda n_\ell}\right].\\
= & \sum_{i\in Q_\ell}s_\ell\left[\phi_i(X_i^\top w_\ell)+\phi_i^*(-\alpha_i)+w_\ell^\top X_i\alpha_i\right]\\
& +\frac{M_\ell}{2\lambda n_\ell}\sum_{i\in Q_\ell}s_\ell^2\|u_i-\alpha_i\|_2^2\left[\frac{\gamma\lambda n_\ell(1-s_\ell)}{M_\ell s_\ell}-\|X_i\|_2^2\right].
\end{aligned}\tag{23}$$

Recall that with $w_\ell=\nabla g^*(\tilde{v}_\ell)$, we have $g(w)+g^*(\tilde{v})=w^\top\tilde{v}$. Then we derive the local duality gap as

$$\begin{aligned}
& P_\ell(w_\ell)-D_\ell(\alpha_{(\ell)})\\
= & \sum_{i\in S_\ell}\phi_i(X_i^\top w_\ell)+\lambda n_\ell g(w_\ell)+\beta_\ell^\top w_\ell-\left(\sum_{i\in S_\ell}-\phi_i^*(-\alpha_i)-\lambda n_\ell g^*\left(\frac{\sum_{i\in S_\ell}X_i\alpha_i-\beta_\ell}{\lambda n_\ell}\right)\right)\\
= & \sum_{i\in S_\ell}\left(\phi_i(X_i^\top w_\ell)+\phi_i^*(-\alpha_i)+w_\ell^\top X_i\alpha_i\right)
\end{aligned}$$

Then, taking the expectation of Eq. (23) w.r.t the random choice of mini-batch set Q_ℓ at round t, we obtain

$$\begin{aligned}
E_t[A_\ell-B_\ell]\geq & \frac{M_\ell}{n_\ell}\sum_{i\in S_\ell}s_\ell\left[\phi_i(X_i^\top w_\ell)+\phi_i^*(-\alpha_i)+w_\ell^\top X_i\alpha_i\right]\\
& +\frac{M_\ell^2}{2\lambda n_\ell^2}\sum_{i\in S_\ell}s_\ell^2\|u_i-\alpha_i\|_2^2\left[\frac{\gamma\lambda n_\ell(1-s_\ell)}{M_\ell s_\ell}-\|X_i\|_2^2\right]\\
= & \frac{s_\ell M_\ell}{n_\ell}\sum_{i\in S_\ell}\left[\phi_i(X_i^\top w_\ell)+\phi_i^*(-\alpha_i)+w_\ell^\top X_i\alpha_i\right]\\
& -\frac{M_\ell^2}{2\lambda n_\ell^2}\sum_{i\in S_\ell}s_\ell^2\|u_i-\alpha_i\|_2^2\left[\|X_i\|_2^2-\frac{\gamma\lambda n_\ell(1-s_\ell)}{M_\ell s_\ell}\right].
\end{aligned}$$

Take expectation of both sides w.r.t the randomness in previous iterations, we have

$$\mathbb{E}[A_\ell-B_\ell]\geq\frac{s_\ell M_\ell}{n_\ell}\mathbb{E}\left[P_\ell(w_\ell)-D_\ell(\alpha_{(\ell)})\right]-\frac{s_\ell^2M_\ell^2}{2\lambda n_\ell^2}G_\ell^{(t)},$$

where

$$G_\ell^{(t)}:=\sum_{i\in S_\ell}\left[\|X_i\|_2^2-\frac{\gamma\lambda n_\ell(1-s_\ell)}{M_\ell s_\ell}\right]\mathbb{E}\left[\|u_i-\alpha_i\|_2^2\right].$$

■

Proof of Theorem 6.

Proof We will apply Lemma 15 with

$$s_\ell=\frac{1}{1+\frac{RM_\ell}{\gamma\lambda n_\ell}}=\frac{\gamma\lambda n_\ell}{\gamma\lambda n_\ell+M_\ell R}\in[0,1], i\in S_\ell.$$

Recall that $\|X_i\|_2^2\leq R$ for all $i\in S_\ell$, then we have

$$\|X_i\|_2^2-\frac{\gamma\lambda n_\ell(1-s_\ell)}{M_\ell s_\ell}\leq 0,\ \text{for all } i\in S_\ell,$$

which implies that $G_\ell^{(t)}\leq 0$ for all ℓ. It follows that for all ℓ after the local update step we have:

$$\begin{aligned}
& \mathbb{E}\left[\tilde{D}_\ell(\alpha_{(\ell)}^{(t)}|\beta_\ell^{(t-1)})-\tilde{D}_\ell(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)})\right]\\
\geq & \frac{s_\ell M_\ell}{n_\ell}\mathbb{E}\left[\tilde{P}_\ell(w_\ell^{(t-1)}|\beta_\ell^{(t-1)})-\tilde{D}_\ell(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)})\right].
\end{aligned}\tag{24}$$

Now we note that after the global step at iteration $t-1$, the choices of $w^{(t-1)}$ and $\beta^{(t-1)}$ in DADM is according to the choice of Proposition 4 and Proposition 5. It follows from

Proposition 5 that the following relationship between the global and local duality gap at the beginning of the t-th iteration is satisfied:

$$P(w^{(t-1)}) - D(\alpha^{(t-1)}, \beta^{(t-1)}) = \sum_{\ell} \left[\tilde{P}_\ell(w_\ell^{(t-1)}|\beta_\ell^{(t-1)}) - \tilde{D}_\ell(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)}) \right].$$

Using this decomposition and summing over ℓ in (24), we obtain

$$\mathbb{E}\left[D(\alpha^{(t)}, \beta^{(t-1)}) - D(\alpha^{(t-1)}, \beta^{(t-1)})\right] \geq q\mathbb{E}\left[P(w^{(t-1)}) - D(\alpha^{(t-1)}, \beta^{(t-1)})\right],$$

where

$$q = \min_\ell \frac{s_\ell M_\ell}{n_\ell} = \min_\ell \frac{\gamma\lambda M_\ell}{\gamma\lambda n_\ell + M_\ell R}.$$

Since $D(\alpha^{(t)}, \beta^{(t)}) \geq D(\alpha^{(t)}, \beta^{(t-1)})$, we obtain

$$\mathbb{E}\left[D(\alpha^{(t)}, \beta^{(t)}) - D(\alpha^{(t-1)}, \beta^{(t-1)})\right] \geq q\mathbb{E}\left[P(w^{(t-1)}) - D(\alpha^{(t-1)}, \beta^{(t-1)})\right].$$

Let (α^*, β^*) be the optimal solution of the dual problem, we have defined the dual suboptimality as $\epsilon_D^{(t)} := D(\alpha^*, \beta^*) - D(\alpha^{(t)}, \beta^{(t)})$. Let $\epsilon_G^{(t-1)} = P(w^{(t-1)}) - D(\alpha^{(t-1)}, \beta^{(t-1)})$, and we know that $\epsilon_D^{(t-1)} \leq \epsilon_G^{(t-1)}$. It follows that

$$\mathbb{E}[\epsilon_D^{(t-1)}] \geq \mathbb{E}[\epsilon_D^{(t-1)} - \epsilon_D^{(t)}] \geq q\mathbb{E}[\epsilon_G^{(t-1)}] \geq q\mathbb{E}[\epsilon_D^{(t-1)}].$$

Therefore we have

$$q\mathbb{E}[\epsilon_G^{(t)}] \leq \mathbb{E}[\epsilon_D^{(t)}] \leq (1-q)\mathbb{E}[\epsilon_D^{(t-1)}] \leq (1-q)^t \epsilon_D^{(0)} \leq e^{-qt}\epsilon_D^{(0)}.$$

To obtain an expected duality gap of $E[\epsilon_G^{(T)}] \leq \epsilon$, every T, which satisfies

$$T \geq \frac{1}{q}\log\left(\frac{1}{q}\frac{\epsilon_D^{(0)}}{\epsilon}\right),$$

is sufficient. This proves the desired bound. ■

9.7 Proof of Theorem 7

Now, we consider L-Lipschitz loss functions and use the following basic lemma for L-Lipschitz losses taken from (Shalev-Shwartz and Zhang, 2013, 2014).

Lemma 16 *Let $\phi : \mathbb{R}^q \to \mathbb{R}$ be an L-Lipschitz function w.r.t $\|\cdot\|_2$, then we have $\phi^*(\alpha) = \infty$, for any $\alpha \in \mathbb{R}^q$ s.t. $\|\alpha\|_2 > L$.*

Proof of Theorem 7.
Proof Applying Lemma 15 with $\gamma = 0$, then we have

$$G_\ell^{(t)} = \sum_{i \in S_\ell} \|X_i\|_2^2 \ \mathbb{E}\left[\|u_i^{(t-1)} - \alpha_i^{(t-1)}\|_2^2\right].$$

According to Lemma 16, we know that $\|u_i^{(t-1)}\|_2 \leq L$ and $\|\alpha_i^{(t-1)}\|_2 \leq L$, thus we have

$$\|u_i^{(t-1)} - \alpha_i^{(t-1)}\|_2^2 \leq 2\left(\|u_i^{(t-1)}\|_2^2 + \|\alpha_i^{(t-1)}\|_2^2\right) \leq 4L^2.$$

Recall that $\|X_i\|_2^2 \leq R$, then we have $G_\ell^{(t)} \leq G_\ell$, where $G_\ell = 4n_\ell RL^2$. Combining this into Lemma 15, we have

$$\begin{aligned}
&\mathbb{E}\left[\tilde{D}_\ell(\alpha_{(\ell)}^{(t)}|\beta_\ell^{(t-1)}) - \tilde{D}_\ell(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)})\right] \\
\geq &\frac{s_\ell M_\ell}{n_\ell}\mathbb{E}\left[\tilde{P}_\ell(w_\ell^{(t-1)}|\beta_\ell^{(t-1)}) - \tilde{D}_\ell(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)})\right] - \frac{s_\ell^2 M_\ell^2}{2\lambda n_\ell^2}G_\ell. \qquad (25)
\end{aligned}$$

Now we also note that after the global step at iteration $t-1$, the choices of $w^{(t-1)}$ and $\beta^{(t-1)}$ in DADM is according to the choice of Proposition 4 and Proposition 5. It follows from Proposition 5 that the following relationship of global and local duality gap at the beginning of the t-th iteration is satisfied:

$$P(w^{(t-1)}) - D(\alpha^{(t-1)}, \beta^{(t-1)}) = \sum_{\ell} \left[\tilde{P}_\ell(w_\ell^{(t-1)}|\beta_\ell^{(t-1)}) - \tilde{D}_\ell(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)}) \right].$$

Summing the inequality (25) over ℓ, combining with the above decomposition and bringing $D(\alpha^{(t)}, \beta^{(t)}) \geq D(\alpha^{(t)}, \beta^{(t-1)})$ into it, we get

$$\mathbb{E}[D(\alpha^{(t)}, \beta^{(t)}) - D(\alpha^{(t-1)}, \beta^{(t-1)})] \geq q\mathbb{E}[P(w^{(t-1)}) - D(\alpha^{(t-1)}, \beta^{(t-1)})] - \sum_{\ell=1}^{m} \frac{q^2}{2\lambda}G_\ell, \quad (26)$$

where $q \in [0, \min_\ell \frac{M_\ell}{n_\ell}]$, $q = \frac{s_\ell M_\ell}{n_\ell}$ and $s_\ell \in [0,1]$ is chosen so that all $\frac{s_\ell M_\ell}{n_\ell}(\ell = 1, ..., m)$ are equal.

Let (α^*, β^*) be the optimal solution for the dual problem $D(\alpha, \beta)$, and we have defined the dual suboptimality as $\epsilon_D^{(t)} := D(\alpha^*, \beta^*) - D(\alpha^{(t)}, \beta^{(t)})$. Note that the duality gap is an upper bound of the dual suboptimality, $P(w^{(t-1)}) - D(\alpha^{(t-1)}, \beta^{(t-1)}) \geq \epsilon_D^{(t-1)}$. Then (26) implies that

$$\mathbb{E}\left[\frac{\epsilon_D^{(t)}}{n}\right] \leq (1-q)\mathbb{E}\left[\frac{\epsilon_D^{(t-1)}}{n}\right] + q^2\frac{G}{2\lambda}, \text{ where } G = \frac{1}{n}\sum_{\ell=1}^{m} G_\ell = 4RL^2$$

Starting from this recursion, we can now apply the same analysis for L-Lipschitz loss functions of the single-machine SDCA in (Shalev-Shwartz and Zhang, 2013) to obtain the following desired inequality:

$$\mathbb{E}\left[\frac{\epsilon_D^{(t)}}{n}\right] \leq \frac{2G}{\lambda(2\tilde{n} + t - t_0)}, \qquad (27)$$

for all $t \geq t_0 = \max(0, \lceil \tilde{n}\log(\frac{2\lambda\epsilon_D^{(0)}\tilde{n}}{nG})\rceil)$, where $\tilde{n} = \max_\ell(n_\ell/M_\ell)$. Further applying the same strategies in (Shalev-Shwartz and Zhang, 2013) based on (27) proves the desired bound. ■

9.8 Proof of Theorem 11

Our proof strategy follows (Shalev-Shwartz and Zhang, 2014) and (Frostig et al., 2015), which both used acceleration techniques of (Nesterov, 2004) on top of approximate proximal point steps, the main differences compared with (Shalev-Shwartz and Zhang, 2014) and (Frostig et al., 2015) are here we warm start with two groups dual variables (α and β) where (Shalev-Shwartz and Zhang, 2014) warm start only with α as it consider the single machine setting, and (Frostig et al., 2015) warm start from primal variables w.

Proof The proof consists of the following steps:

- In Lemma 17 we show that one can construct a quadratic lower bound of the original objective $P(w)$ from an approximate minimizer of the proximal objective $P_t(w)$.
- Using the quadratic lower bound we construct an estimation sequence, based on which in Lemma 18 we prove the accelerated convergence rate for the outer loops.
- We show in Lemma 19 that by warm start the iterates from the last stage, the dual sub-optimality for the next stage is small.

Based on Lemma 19, we know the contraction factor between the initial dual sub-optimality and the target primal-dual gap at stage t can be upper bounded by

$$\begin{aligned}\frac{D_t(\alpha_{\text{opt}}^{(t)},\beta_{\text{opt}}^{(t)})-D_t(\alpha^{(t-1)},\beta^{(t-1)})}{(\eta\xi_{t-1})/(2+2\eta^{-2})}\leq&\frac{\epsilon_{t-1}}{(\eta\xi_{t-1})/(2+2\eta^{-2})}+\frac{36\kappa\xi_{t-3}}{\lambda(\eta\xi_{t-1})/(2+2\eta^{-2})}\\ \leq&\frac{1}{1-\eta/2}+\frac{36(2+2\eta^{-2})}{\eta(1-\eta/2)^2}\cdot\frac{\kappa}{\lambda}\\ \leq&2+\frac{36}{\eta^5(1-\eta^2)}\end{aligned}$$

where the last step we used the fact that $\eta^{-4}=(\eta^{-2}-1)(\eta^{-2}+1)+1>(\eta^{-2}-1)(\eta^{-2}+1)=\frac{2\kappa(\eta^{-2}+1)}{\lambda}$. Thus using the results from plain DADM (Theorem 6), we know the number of inner iterations in each stage is upper bounded by

$$\begin{aligned}&\chi\left(\log(\chi)+\log\left(\frac{D_t(\alpha_{\text{opt}}^{(t)},\beta_{\text{opt}}^{(t)})-D_t(\alpha^{(t-1)},\beta^{(t-1)})}{(\eta\xi_{t-1})/(2+2\eta^{-2})}\right)\right)\\ \leq&\chi\left(\log(\chi)+7+\frac{5}{2}\log\left(\frac{\lambda+2\kappa}{\lambda}\right)\right),\end{aligned}$$

where $\chi=\frac{R}{\gamma(\lambda+\kappa)}+\max_\ell\frac{n_\ell}{M_\ell}$. ■

9.9 Proof of Corollary 13

By the property of $\tilde{\phi}_i(u_i)$, for every w we have

$$0\leq\frac{\hat{P}(w)}{n}-\frac{P(w)}{n}\leq\frac{\gamma L^2}{2},$$

thus if we found a predictor $w^{(t)}$ that is $\frac{\epsilon}{2}$-suboptimal with respect to $\frac{\hat{P}(w)}{n}$:

$$\frac{\hat{P}(w^{(t)})}{n}-\min_w\frac{\hat{P}(w)}{n}\leq\frac{\epsilon}{2},$$

and we choose $\gamma=\epsilon/L^2$, we know it must be ϵ-suboptimal with respect to $\frac{P(w)}{n}$, because

$$\begin{aligned}\frac{P(w^{(t)})}{n}-\frac{P(w^*)}{n}\leq&\frac{\hat{P}(w^{(t)})}{n}-\frac{\hat{P}(w^*)}{n}+\frac{\gamma L^2}{2}\\ \leq&\frac{\hat{P}(w^{(t)})}{n}-\min_w\frac{\hat{P}(w)}{n}+\frac{\epsilon}{2}\leq\epsilon.\end{aligned}$$

The rest of the proof just follows the smooth case as proved in Theorem 11.

Dual subproblems in Acc-DADM Define: $\tilde{\lambda}=\lambda+\kappa$, $f(w)=\frac{\lambda}{\tilde{\lambda}}g(w)+\frac{\kappa}{2\tilde{\lambda}}\|w\|_2^2$. Let

$$\begin{aligned}P_t(w)=&\sum_{i=1}^n\phi_i(X_i^\top w)+\lambda ng(w)+h(w)+\frac{\kappa n}{2}\left\|w-y^{(t-1)}\right\|_2^2\\ =&\sum_{i=1}^n\phi_i(X_i^\top w)+\tilde{\lambda}n\left(f(w)-\frac{\kappa}{\tilde{\lambda}}w^\top y^{(t-1)}\right)+h(w)+\frac{\kappa n_\ell}{2}\left\|y^{(t-1)}\right\|_2^2\end{aligned}$$

be the global primal problem to solve, and

$$P_{\ell_t}(w_\ell)=\sum_{i\in S_\ell}\phi_i(X_i^\top w_\ell)+\lambda n_\ell g(w_\ell)+\frac{\kappa n_\ell}{2}\left\|w_\ell-y^{(t-1)}\right\|_2^2$$

be the separated local problem. Given each dual variable β_ℓ, we also define the adjusted local primal problem as:

$$\tilde{P}_{\ell_t}(w_\ell|\beta_\ell)=\sum_{i\in S_\ell}\phi_i(X_i^\top w_\ell)+\lambda n_\ell g(w_\ell)+\beta_\ell^\top w_\ell+\frac{\kappa n_\ell}{2}\left\|w_\ell-y^{(t-1)}\right\|_2^2,$$

it is not hard to see the adjusted local dual problem is

$$\tilde{D}_{\ell_t}(\alpha_{(\ell)}|\beta_\ell)=\sum_{i\in S_\ell}-\phi_i^*(-\alpha_i)-\tilde{\lambda}n_\ell f^*\left(\frac{\sum_{i\in S_\ell}X_i\alpha_i-\beta_\ell+\kappa n_\ell y^{(t-1)}}{\tilde{\lambda}n_\ell}\right)+\frac{\kappa n_\ell}{2}\left\|y^{(t-1)}\right\|_2^2,$$

and the global dual objective can be written as

$$D_t(\alpha,\beta)=\sum_{\ell=1}^m\tilde{D}_{\ell_t}(\alpha_{(\ell)}|\beta_\ell)-h^*\left(\sum_{\ell=1}^m\beta_\ell\right).$$

Quadratic lower bound for $P(w)$ based on approximate proximal point algorithm Since $P_t(w)=P(w)+\frac{\kappa n}{2}\left\|w-y^{(t-1)}\right\|_2^2$, and let $w_{\text{opt}}^{(t)}=\arg\min_w P_t(w)$. The following lemma shows we can construct a lower bound of $P(w)$ from an approximate minimizer of $P_t(w)$.

Lemma 17 *Let w^+ be an ϵ-approximated minimizer of $P_t(w)$, i.e.*

$$P_t(w^+) \leq P_t(w_{\text{opt}}^{(t)}) + \epsilon.$$

We can construct the following quadratic lower bound for $P(w)$, as $\forall w$

$$P(w) \geq P(w^+) + Q(w; w^+, y^{(t-1)}, \epsilon), \tag{28}$$

where

$$\begin{aligned} Q(w; w^+, y^{(t-1)}, \epsilon) =& \frac{\lambda n}{4} \left\| w - \left(y^{(t-1)} - \left(1 + \frac{2\kappa}{\lambda}\right)(y^{(t-1)} - w^+) \right) \right\|_2^2 \\ & - \frac{\kappa^2 n}{\lambda} \left\| w^+ - y^{(t-1)} \right\|_2^2 - \left(\frac{2\kappa + 2\lambda}{\lambda} \right) \epsilon. \end{aligned}$$

Proof Since $w_{\text{opt}}^{(t)}$ is the minimizer of a $(\kappa+\lambda)n$-strongly convex objective $P_t(w)$, we know $\forall w$,

$$\begin{aligned} P_t(w) \geq & P_t(w_{\text{opt}}^{(t)}) + \frac{(\kappa+\lambda)n}{2} \left\| w - w_{\text{opt}}^{(t)} \right\|_2^2 \\ \geq & P_t(w^+) + \frac{(\kappa+\lambda)n}{2} \left\| w - w_{\text{opt}}^{(t)} \right\|_2^2 - \epsilon, \end{aligned}$$

which is equivalent to

$$P(w) \geq P(w^+) + \frac{(\kappa+\lambda)n}{2} \left\| w - w_{\text{opt}}^{(t)} \right\|_2^2 - \epsilon + \frac{\kappa n}{2} \left(\left\| w^+ - y^{(t-1)} \right\|_2^2 - \left\| w - y^{(t-1)} \right\|_2^2 \right).$$

Since

$$\begin{aligned} \frac{\kappa+\lambda/2}{2} \left\| w - w^+ \right\|_2^2 =& \frac{\kappa+\lambda/2}{2} \left\| w - w_{\text{opt}}^{(t)} + w_{\text{opt}}^{(t)} - w^+ \right\|_2^2 \\ =& \frac{\kappa+\lambda/2}{2} \left(\left\| w - w_{\text{opt}}^{(t)} \right\|_2^2 + \left\| w_{\text{opt}}^{(t)} - w^+ \right\|_2^2 \right) \\ & + (\kappa+\lambda/2) \langle w - w_{\text{opt}}^{(t)}, w_{\text{opt}}^{(t)} - w^+ \rangle \\ \leq & \frac{\kappa+\lambda/2}{2} \left(\left\| w - w_{\text{opt}}^{(t)} \right\|_2^2 + \left\| w_{\text{opt}}^{(t)} - w^+ \right\|_2^2 \right) + \frac{\lambda/2}{2} \left\| w_{\text{opt}}^{(t)} - w \right\|_2^2 \\ & + \frac{(\kappa+\lambda/2)^2}{\lambda} \left\| w^+ - w_{\text{opt}}^{(t)} \right\|_2^2, \end{aligned}$$

re-organizing terms we get

$$\frac{\kappa+\lambda}{2} \left\| w_{\text{opt}}^{(t)} - w \right\|_2^2 \geq \frac{\kappa+\lambda/2}{2} \left\| w - w^+ \right\|_2^2 - \frac{(\kappa+\lambda)(\kappa+\lambda/2)}{\lambda} \left\| w^+ - w_{\text{opt}}^{(t)} \right\|_2^2$$

So

$$\begin{aligned} P(w) \geq & P(w^+) + \frac{(\kappa+\lambda/2)n}{2} \left\| w - w^+ \right\|_2^2 - \frac{(\kappa+\lambda)(\kappa+\lambda/2)n}{\lambda} \left\| w^+ - w_{\text{opt}}^{(t)} \right\|_2^2 - \epsilon \\ & + \frac{\kappa n}{2} \left(\left\| w^+ - y^{(t-1)} \right\|_2^2 - \left\| w - y^{(t-1)} \right\|_2^2 \right) \end{aligned}$$

Also noted that $\frac{(\kappa+\lambda)n}{2} \left\| w^+ - w_{\text{opt}}^{(t)} \right\|_2^2 \leq \epsilon$, we get

$$\begin{aligned} P(w) \geq & P(w^+) + \frac{(\kappa+\lambda/2)n}{2} \left\| w - w^+ \right\|_2^2 - \left(\frac{2\kappa+2\lambda}{\lambda} \right) \epsilon \\ & + \frac{\kappa n}{2} \left(\left\| w^+ - y^{(t-1)} \right\|_2^2 - \left\| w - y^{(t-1)} \right\|_2^2 \right) \end{aligned}$$

Decompose $\left\| w - w^+ \right\|_2^2$ we get

$$\left\| w - w^+ \right\|_2^2 = \left\| w - y^{(t-1)} \right\|^2 + \left\| y^{(t-1)} - w^+ \right\|_2^2 + 2\langle w - y^{(t-1)}, y^{(t-1)} - w^+ \rangle.$$

So

$$\begin{aligned} P(w) \geq & P(w^+) + \frac{(\lambda/2)n}{2} \left\| w - y^{(t-1)} \right\|_2^2 - \left(\frac{2\kappa+2\lambda}{\lambda} \right) \epsilon \\ & + \frac{(2\kappa+\lambda/2)n}{2} \left\| y^{(t-1)} - w^+ \right\|_2^2 + (\kappa+\lambda/2)n \langle w - y^{(t-1)}, y^{(t-1)} - w^+ \rangle \end{aligned}$$

Noticed that the right hand side of above inequality is a quadratic function with respect to w, and the minimum is achieved when

$$w = y^{(t-1)} - \left(1 + \frac{2\kappa}{\lambda}\right)(y^{(t-1)} - w^+),$$

with minimum value

$$-\frac{\kappa^2 n}{\lambda} \left\| w^+ - y^{(t-1)} \right\|_2^2 - \left(\frac{2\kappa+2\lambda}{\lambda} \right) \epsilon,$$

with above we finished the proof of Lemma 17. ■

Convergence proof Define the following sequence of quadratic functions

$$\psi_0(w) = P(0) + \frac{\lambda n}{4} \left\| w \right\|_2^2 - \left(\frac{2\kappa+2\lambda}{\lambda} \right)(P(0) - D(0,0)),$$

and for $t \geq 1$,

$$\psi_t(w) = (1-\eta)\psi_{t-1}(w) + \eta(P(w^{(t)}) + Q(w; w^{(t)}, y^{(t-1)}, \epsilon_t)),$$

where $\eta = \sqrt{\frac{\lambda}{\lambda+2\kappa}}$, We first calculate the explicit form of the quadratic function $\psi_t(w)$ and its minimizer $v^{(t)} = \arg\min_w \psi_t(w)$. Clearly $v^{(0)} = 0$, and noticed that $\psi_t(w)$ is always a $\frac{\lambda n}{2}$-strongly convex function, we know $\psi_t(w)$ is in the following form:

$$\psi_t(w) = \psi_t(v^{(t)}) + \frac{\lambda n}{4} \left\| w - v^{(t)} \right\|_2^2.$$

Based on the definition of $\psi_{t+1}(w)$, and $v^{(t+1)}$ is minimizing $\psi_{t+1}(w)$, based on first-order optimality condition, we know

$$\frac{(1-\eta)\lambda n}{2}(v^{(t+1)}-v^{(t)})+\frac{\eta\lambda n}{2}\left(v^{(t+1)}-\left(y^{(t)}-\left(1+\frac{2\kappa}{\lambda}\right)(y^{(t)}-w^{(t+1)})\right)\right)=0,$$

rearranging we get

$$v^{(t+1)}=(1-\eta)v^{(t)}+\eta\left(y^{(t)}-\left(1+\frac{2\kappa}{\lambda}\right)(y^{(t)}-w^{(t+1)})\right).$$

The following lemma proves the convergence rate of $w^{(t)}$ to its minimizer.

Lemma 18 *Let*

$$\epsilon_t\leq\frac{\eta}{2(1+\eta^{-2})}\xi_t,$$

and

$$\xi_t=(1-\eta/2)^t\xi_0,$$

we will have the following convergence guarantee:

$$P(w^{(t)})-P(w^*)\leq\xi_t.$$

Proof It is sufficient to prove

$$P(w^{(t)})-\min_w\psi_t(w)\leq\xi_t, \tag{29}$$

then we get

$$P(w^{(t)})-P(w^*)\leq P(w^{(t)})-\psi_t(w^*)\leq P(w^{(t)})-\min_w\psi_t(w)\leq\xi_t.$$

We prove equation (29) by induction. When $t=0$, we have

$$P(w^{(0)})-\phi_0(v^{(0)})=\left(\frac{2\kappa+2\lambda}{\lambda}\right)\epsilon_0=\xi_0,$$

which verified (29) is true for $t=0$. Suppose the claim holds for some $t\geq1$, for the stage $t+1$, we have

$$\begin{aligned}
\psi_{t+1}(v^{(t+1)})=&(1-\eta)\left(\psi_t(v^{(t)})+\frac{\lambda n}{4}\left\|v^{(t+1)}-v^{(t)}\right\|_2^2\right)\\
&+\eta(P(w^{(t+1)})+Q(v^{(t+1)};w^{(t+1)},y^{(t)},\epsilon_t))\\
=&(1-\eta)\psi_t(v^{(t)})+\frac{(1-\eta)\eta^2\lambda n}{4}\left\|v^{(t)}-\left(y^{(t)}-\left(1+\frac{2\kappa}{\lambda}\right)(y^{(t)}-w^{t+1})\right)\right\|_2^2\\
&+\eta P(w^{(t+1)})+\frac{\eta(1-\eta)^2\lambda n}{4}\left\|v^{(t)}-\left(y^{(t)}-\left(1+\frac{2\kappa}{\lambda}\right)(y^{(t)}-w^{(t+1)})\right)\right\|_2^2\\
&-\frac{\eta\kappa^2 n}{\lambda}\left\|y^{(t)}-w^{(t+1)}\right\|_2^2-\eta\left(\frac{2\kappa+2\lambda}{\lambda}\right)\epsilon_t\\
=&(1-\eta)\psi_t(v^{(t)})+\eta P(w^{(t+1)})-\frac{\eta\kappa^2 n}{\lambda}\left\|y^{(t)}-w^{(t+1)}\right\|_2^2-\eta\left(\frac{2\kappa+2\lambda}{\lambda}\right)\epsilon_t\\
&+\frac{\eta(1-\eta)\lambda n}{4}\left\|v^{(t)}-\left(y^{(t)}-\left(1+\frac{2\kappa}{\lambda}\right)(y^{(t)}-w^{(t+1)})\right)\right\|_2^2.
\end{aligned}$$

Since

$$\begin{aligned}
&-\frac{\eta\kappa^2 n}{\lambda}\left\|y^{(t)}-w^{(t+1)}\right\|_2^2+\frac{\eta(1-\eta)\lambda n}{4}\left\|v^{(t)}-\left(y^{(t)}-\left(1+\frac{2\kappa}{\lambda}\right)(y^{(t)}-w^{(t+1)})\right)\right\|_2^2\\
\geq&\left(-\frac{\eta\kappa^2 n}{\lambda}+\frac{\eta(1-\eta)\lambda n}{4}\left(1+\frac{2\kappa}{\lambda}\right)^2\right)\left\|y^{(t)}-w^{(t+1)}\right\|_2^2\\
&+\eta(1-\eta)n\left(\kappa+\frac{\lambda}{2}\right)\langle v^{(t)}-y^{(t)},y^{(t)}-w^{(t+1)}\rangle\\
\geq&\left(-\frac{\eta\kappa^2 n}{\lambda}+\frac{\eta(1-\eta)\kappa^2 n}{\lambda}\right)\left\|y^{(t)}-w^{(t+1)}\right\|_2^2\\
&+\eta(1-\eta)n\left(\kappa+\frac{\lambda}{2}\right)\langle v^{(t)}-y^{(t)},y^{(t)}-w^{(t+1)}\rangle\\
=&-\frac{\eta^2\kappa^2 n}{\lambda}\left\|y^{(t)}-w^{(t+1)}\right\|_2^2+\eta(1-\eta)n\left(\kappa+\frac{\lambda}{2}\right)\langle v^{(t)}-y^{(t)},y^{(t)}-w^{(t+1)}\rangle
\end{aligned}$$

Thus

$$\begin{aligned}
\psi_{t+1}(v^{(t+1)})\geq&(1-\eta)\psi_t(v^{(t)})+\eta P(w^{(t+1)})-\eta\left(\frac{2\kappa+2\lambda}{\lambda}\right)\epsilon_t\\
&-\frac{\eta^2\kappa^2 n}{\lambda}\left\|y^{(t)}-w^{(t+1)}\right\|_2^2+\eta(1-\eta)n\left(\kappa+\frac{\lambda}{2}\right)\langle v^{(t)}-y^{(t)},y^{(t)}-w^{(t+1)}\rangle
\end{aligned}$$

Also using (28) with $w = w^{(t)}$, we have

$$\begin{aligned}
P(w^{(t)}) \geq & P(w^{(t+1)}) + \frac{\lambda n}{4}\left\|w^{(t)} - \left(y^{(t)} - \left(1+\frac{2\kappa}{\lambda}\right)(y^{(t)} - w^{(t+1)})\right)\right\|_2^2 \\
& - \frac{\kappa^2 n}{\lambda}\left\|w^{(t+1)} - y^{(t)}\right\|_2^2 - \left(\frac{2\kappa+2\lambda}{\lambda}\right)\epsilon_t \\
\geq & P(w^{(t+1)}) + n\left(\kappa+\frac{\lambda}{2}\right)\langle w^{(t)} - y^{(t)}, y^{(t)} - w^{(t+1)}\rangle - \left(\frac{2\kappa+2\lambda}{\lambda}\right)\epsilon_t \\
& + \left(\frac{\lambda n}{4}\left(1+\frac{2\kappa}{\lambda}\right)^2 - \frac{\kappa^2 n}{\lambda}\right)\left\|w^{(t+1)} - y^{(t)}\right\|_2^2 \\
\geq & P(w^{(t+1)}) + n\left(\kappa+\frac{\lambda}{2}\right)\langle w^{(t)} - y^{(t)}, y^{(t)} - w^{(t+1)}\rangle - \left(\frac{2\kappa+2\lambda}{\lambda}\right)\epsilon_t \\
& + \kappa n\left\|w^{(t+1)} - y^{(t)}\right\|_2^2.
\end{aligned}$$

we get

$$\begin{aligned}
P(w^{(t+1)}) - \psi_{t+1}(v^{(t+1)}) \leq & (1-\eta)(P(w^{(t)}) - \psi_t(v^{(t)})) + \left(\frac{2\kappa+2\lambda}{\lambda}\right)\epsilon_t \\
& + \left(\frac{\eta^2\kappa^2 n}{\lambda} - (1-\eta)\kappa n\right)\left\|y^{(t)} - w^{(t+1)}\right\|_2^2 \\
& + (1-\eta)n\left(\kappa+\frac{\lambda}{2}\right)\langle y^{(t)} - w^{(t)}, y^{(t)} - w^{(t+1)}\rangle. \\
& + \eta(1-\eta)n\left(\kappa+\frac{\lambda}{2}\right)\langle y^{(t)} - v^{(t)}, y^{(t)} - w^{(t+1)}\rangle
\end{aligned}$$

Since

$$\begin{aligned}
(1-\eta)(P(w^{(t)}) - \psi_t(v^{(t)})) + \left(\frac{2\kappa+2\lambda}{\lambda}\right)\epsilon_t \leq & (1-\eta)\xi_t + \left(\frac{2\kappa+2\lambda}{\lambda}\right)\cdot\frac{\eta}{2(1+\eta^{-2})}\xi_t \\
= & \left(1-\eta+\frac{\eta}{2}\right)\xi_t = \xi_{t+1},
\end{aligned}$$

and

$$\frac{\eta^2\kappa^2 n}{\lambda} - (1-\eta)\kappa n = \kappa n\left(\frac{\kappa}{\lambda+2\kappa} + \sqrt{\frac{\lambda}{\lambda+2\kappa}} - 1\right) \leq 0,$$

If we set $y^{(t)} = (\eta v^{(t)} + w^{(t)})/(1+\eta)$, which is equivalent to the update rule as $y^{(t)} = w^{(t)} + \nu(w^{(t)} - w^{(t-1)})$, because in that way we have

$$\begin{aligned}
\eta v^{(t)} = & \eta((1-\eta)v^{(t-1)}) + \eta^2\left(y^{(t-1)} - \left(1+\frac{2\kappa}{\lambda}\right)(y^{(t-1)} - w^{(t)})\right) \\
= & w^{(t)} + (1-\eta)(\eta v^{(t-1)} - (1+\eta)y^{(t-1)}) \\
= & w^{(t)} - (1-\eta)w^{(t-1)},
\end{aligned}$$

thus

$$\begin{aligned}
y^{(t)} = & \frac{\eta v^{(t)} + w^{(t)}}{1+\eta} \\
= & \frac{2w^{(t)} - (1-\eta)w^{(t-1)}}{1+\eta} \\
= & w^{(t)} + \nu(w^{(t)} - w^{(t-1)}).
\end{aligned}$$

So

$$y^{(t)} - w^{(t)} + \eta(y^{(t)} - v^{(t)}) = 0,$$

combining above we obtain

$$P(w^{(t+1)}) - \psi_{t+1}(v^{(t+1)}) \leq \xi_{t+1},$$

which concludes the proof. ■

Initial dual sub-optimality in each acceleration stage In the lemma below we upper bound the quantity

$$D_t(\alpha_{\text{opt}}^{(t)}, \beta_{\text{opt}}^{(t)}) - D_t(\alpha^{(t-1)}, \beta^{(t-1)}),$$

where $\alpha_{\text{opt}}^{(t)}, \beta_{\text{opt}}^{(t)} = \arg\max_{\alpha,\beta} D_t(\alpha, \beta)$.

Lemma 19 *We have the following upper bound on the initial dual sub-optimality at stage t:*

$$D_t(\alpha_{\text{opt}}^{(t)}, \beta_{\text{opt}}^{(t)}) - D_t(\alpha^{(t-1)}, \beta^{(t-1)}) \leq \epsilon_{t-1} + \frac{36\kappa}{\lambda}\xi_{t-3}.$$

Proof On one hand, since $f(\cdot)$ is 1-strongly convex, we know $f^*(\cdot)$ is 1-smooth. Thus

$$\begin{aligned}
& \tilde{\lambda} n_\ell f^*\left(\frac{\sum_{i\in S_\ell} X_i\alpha_i^{(t-1)} - \beta_\ell^{(t-1)} + \kappa n_\ell y^{(t-1)}}{\tilde{\lambda} n_\ell}\right) \\
\leq & \tilde{\lambda} n_\ell f^*\left(\frac{\sum_{i\in S_\ell} X_i\alpha_i^{(t-1)} - \beta_\ell^{(t-1)} + \kappa n_\ell y^{(t-2)}}{\tilde{\lambda} n_\ell}\right) \\
& + \kappa n_\ell \nabla f^*\left(\frac{\sum_{i\in S_\ell} X_i\alpha_i^{(t-1)} - \beta_\ell^{(t-1)} + \kappa n_\ell y^{(t-2)}}{\tilde{\lambda} n_\ell}\right)^\top (y^{(t-1)} - y^{(t-2)}) \\
& + \frac{\kappa^2 n_\ell^2}{2\tilde{\lambda} n_\ell}\left\|y^{(t-1)} - y^{(t-2)}\right\|_2^2,
\end{aligned}$$

noted that

$$\nabla f^*\left(\frac{\sum_{i\in S_\ell} X_i\alpha_i^{(t-1)} - \beta_\ell^{(t-1)} + \kappa n_\ell y^{(t-2)}}{\tilde{\lambda} n_\ell}\right) = w^{(t-1)},$$

we see

$$\begin{aligned}&-\tilde{D}_{\ell_t}(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)})+\tilde{D}_{\ell_{t-1}}(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)})\\ \leq&\kappa n_\ell {w_\ell^{(t-1)}}^\top(y^{(t-1)}-y^{(t-2)})+\frac{\kappa^2 n_\ell^2}{2\tilde{\lambda}n_\ell}\left\|y^{(t-1)}-y^{(t-2)}\right\|_2^2\\ &+\frac{\kappa n_\ell}{2}\left\|y^{(t-2)}\right\|_2^2-\frac{\kappa n_\ell}{2}\left\|y^{(t-1)}\right\|_2^2.\end{aligned}$$

On the other hand, since

$$\begin{aligned}&\tilde{P}_{\ell_t}(w_\ell^{(t-1)}|\beta_\ell^{(t-1)})-\tilde{P}_{\ell_{t-1}}(w_\ell^{(t-1)}|\beta_\ell^{(t-1)})\\ =&\kappa n_\ell {w_\ell^{(t-1)}}^\top(y^{(t-2)}-y^{(t-1)})-\frac{\kappa n_\ell}{2}\left\|y^{(t-2)}\right\|_2^2+\frac{\kappa n_\ell}{2}\left\|y^{(t-1)}\right\|_2^2.\end{aligned}$$

Combining above we know

$$\begin{aligned}&\tilde{P}_{\ell_t}(w_\ell^{(t-1)}|\beta_\ell^{(t-1)})-\tilde{D}_{\ell_t}(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)})\\ \leq&\tilde{P}_{\ell_{t-1}}(w_\ell^{(t-1)}|\beta_\ell^{(t-1)})-\tilde{D}_{\ell_{t-1}}(\alpha_{(\ell)}^{(t-1)}|\beta_\ell^{(t-1)})+\frac{\kappa^2 n_\ell^2}{2\tilde{\lambda}n_\ell}\left\|y^{(t-1)}-y^{(t-2)}\right\|_2^2,\end{aligned}$$

Since $\kappa\leq\tilde{\lambda}$, summing over above inequality we know

$$P_t(w_\ell^{(t-1)})-D_t(\alpha^{(t-1)},\beta^{(t-1)})\leq\epsilon_{t-1}+\frac{\kappa n}{2}\left\|y^{(t-1)}-y^{(t-2)}\right\|_2^2,$$

also noted that $P_t(w_\ell^{(t-1)})\geq D_t(\alpha_{\text{opt}}^{(t)},\beta_{\text{opt}}^{(t)})$, we get

$$D_t(\alpha_{\text{opt}}^{(t)},\beta_{\text{opt}}^{(t)})-D_t(\alpha^{(t-1)},\beta^{(t-1)})\leq\epsilon_{t-1}+\frac{\kappa n}{2}\left\|y^{(t-1)}-y^{(t-2)}\right\|_2^2.$$

For the term $\left\|y^{(t-1)}-y^{(t-2)}\right\|_2^2$, based on the definition of $y^{(t-1)}$ and the fact $\eta\leq 1$, we know

$$\begin{aligned}\left\|y^{(t-1)}-y^{(t-2)}\right\|_2\leq&\left\|w^{(t-1)}-w^{(t-2)}-\eta(w^{(t-1)}-w^{(t-2)}-(w^{(t-2)}-w^{(t-3)}))\right\|_2\\ \leq&3\max_{i=\{1,2\}}\left\|w^{(t-i)}-w^{(t-i-1)}\right\|_2.\end{aligned}$$

Then we upper bound $\left\|w^{(t-i)}-w^{(t-i-1)}\right\|_2$ using objective sub-optimality, using triangle inequality and the fact that $P(w^*)$ is λn-strongly convex, we have

$$\begin{aligned}\left\|w^{(t-i)}-w^{(t-i-1)}\right\|_2\leq&\left\|w^{(t-i)}-w^*\right\|_2+\left\|w^{(t-i-1)}-w^*\right\|_2\\ \leq&\sqrt{\frac{2(P(w^{(t-i)})-P(w^*))}{\lambda n}}+\sqrt{\frac{2(P(w^{(t-i-1)})-P(w^*))}{\lambda n}}\\ \leq&2\sqrt{2\frac{\xi_{t-i-1}}{\lambda n}}.\end{aligned}$$

We know

$$\left\|y^{(t-1)}-y^{(t-2)}\right\|_2^2\leq 9\max_{i=\{1,2\}}\left\|w^{(t-i)}-w^{(t-i-1)}\right\|_2^2\leq\frac{72\xi_{t-3}}{\lambda n}.$$

Combining above we get

$$D_t(\alpha_{\text{opt}}^{(t)},\beta_{\text{opt}}^{(t)})-D_t(\alpha^{(t-1)},\beta^{(t-1)})\leq\epsilon_{t-1}+\frac{36\kappa}{\lambda}\xi_{t-3}.$$

■

10. Experiments

In this section, we apply algorithms to solve L_2-L_1 regularized loss minimization problems. We compare **Acc-DADM** to **CoCoA**$^+$ and **OWL-QN** (Andrew and Gao, 2007), as they have already been shown to be superior to other related algorithms in (Yang, 2013; Jaggi et al., 2014; Ma et al., 2017; Andrew and Gao, 2007). For **Acc-DADM** and **CoCoA**$^+$, we apply ProxSDCA of (Shalev-Shwartz and Zhang, 2014) as the local procedure and perform aggressively sequential updates, as the practical variant of DisDCA did in (Yang, 2013) and CoCoA$^+$ did in (Ma et al., 2015). For details about the updates of the local procedure (ProxSDCA), please refer to the application section of (Shalev-Shwartz and Zhang, 2014). For fair comparisons, we use same balanced data partitions and random seeds.

We implement all algorithms using OpenMPI (Graham et al., 2006) and run them on a small cluster inside a private OpenStack cloud service. To simplify the programming efforts, we use one processor to simulate one machine. We test algorithms on four real datasets with different properties (see Table 1). These datasets are publicly available from LIBSVM dataset collections[1].

Table 1: Datasets

Dataset	**Size** (n)	**Features** (d)	**Sparsity**
covtype	581,012	54	22.12%
rcv1	677,399	47,236	0.16%
HIGGS	11,000,000	28	92.11%
kdd2010	19,264,097	29,890,095	$9.8e^{-7}$

Different loss functions The optimization problem we consider to solve is

$$\min_w\frac{1}{n}\sum_{i=1}^n\phi_i(x_i^\top w)+\frac{\lambda}{2}\|w\|_2^2+\mu\|w\|_1,$$

where $x_i\in\mathbb{R}^d$ is a feature vector, $y_i\in\{-1,1\}$ is a binary class label and $\phi_i:\mathbb{R}\to\mathbb{R}$ is the associated loss function. We consider two models: Support Vector Machine (SVM)

1. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets

and Logistic Regression (LR). For SVM, we follow (Shalev-Shwartz and Zhang, 2014) and employ the smooth hinge loss $\tilde{\phi}_i$ (1-smooth) that is:

$$\tilde{\phi}_i(a) = \begin{cases} 0 & a \geq 1 \\ 1 - y_i a - 1/2 & a \leq 0 \\ \frac{1}{2}(1 - y_i a)^2 & o.w. \end{cases} \tag{30}$$

For LR, we employ the logistic loss ($\frac{1}{4}$-smooth) $\phi_i(a) = \log(1 + \exp(-y_i a))$. To apply Acc-DADM, we choose $\lambda g(w) = \frac{\lambda}{2}\|w\|_2^2 + \mu\|w\|_1$ and $h(w) = 0$. Please refer to (Shalev-Shwartz and Zhang, 2014) for detailed derivations of specific loss functions. For datasets with a medium sample size, such as *covtype* and *rcv1*, we employ 8 machines ($m = 8$). For relatively large datasets, such as *HIGGS* and *kdd2010*, we employ 20 machines ($m = 20$). In experiments, we set $\mu = 1e^{-5}$ and vary λ in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ on different datasets to see the convergence behaviours.

Mini-batch size Adjusting the mini-batch size M_ℓ corresponds to trading between computation and communication. We use $sp := \frac{M_\ell}{n_\ell}$ to denote the sampling percentage of the local procedure. In experiments, we test three sp values, 0.05 (red), 0.20 (green) and 0.80 (blue). For each case, we run algorithms for 100 passes over the data. As sp increases, the total number of communications needed to run 100 passes through the data decreases.

Acceleration parameters For all experiments, we set $\kappa = \frac{mR}{\gamma n} - \lambda$ as our theory suggests. As for ν, our theory suggests $\nu = \frac{1-\eta}{1+\eta}$ where $\eta = \sqrt{\frac{\lambda}{\lambda+2\kappa}}$. While in practice we find that $\nu = 0$ also works well and the algorithm can converge more smoothly. In Figure 1, we plot the empirical convergence results for the theory suggested ν and the choice $\nu = 0$. We observe that Acc-DADM with the theory suggested ν does enjoy the acceleration effect, though often converges with rippling behavior. Such a rippling behavior is normal for the accelerated methods, for example, as studied in (Odonoghue and Candès, 2015). Moreover, we observe simply setting $\nu = 0$ also works well in practice and produces more smooth convergence behavior. Thus we use $\nu = 0$ in later experiments for better visualization.

Figures 2, 3, 4 and 5 show detailed comparison experiments of CoCoA$^+$ and Acc-DADM. Figures 2, 3 show **the normalized duality gap versus the number of communications** and **the normalized duality gap versus time (s)** experiments for SVM respectively. Figures 4, 5 show similar plots for LR. From Figures 2, 4, we can see that larger M_ℓ corresponds to more local computations and enjoys less communications accordingly. Looking at Figures 3, 5 about the normalized duality gap versus time (s), we can see the practical effect of adjusting M_ℓ. Moreover, Figures 6, 7 show the total comparisons of OWL-QN, CoCoA$^+$ and Acc-DADM by solving LR problems. For OWL-QN, we follow the standard implementation of (Andrew and Gao, 2007) and set the memory parameter as 10. For CoCoA$^+$ and Acc-DADM, we set $sp = 1.0$ and it implies that each communication round corresponds to one pass over the data.

Our empirical studies show that Acc-DADM always yields the best results. When λ is relatively large, CoCoA$^+$ sometimes also converges as fast as Acc-DADM. However, CoCoA$^+$ slows down rapidly as λ becomes small, while Acc-DADM still enjoys fast convergence. These observations are consistent with our theory.

Scalability Scalability is an important metric for distributed algorithms. We study the scalability by observing the number of communications or the running time (s) needed to reach a certain accuracy ($1e^{-3}$ duality gap) versus the number of machines when the mini-batch size is fixed. For balanced partitions, the mini-batch size of each machine is $\frac{n}{m}sp$. To fix the mini-batch size, we need to adjust sp accordingly when m varies. In experiments, we vary sp in the range $\{0.04, 0.08, 0.16, 0.32\}$ when m grows exponentially from 4 to 32 or from 5 to 40. For each case, we run algorithms for at most 100 passes over the data. It implies that if the algorithm does not reach enough accuracy within 100 passes over the data, we record the number of communications or the running time as the final value after 100 passes.

Figures 8, 9 show scalability results when solving SVM problems. Figures 10, 11 show similar results of LR. From Figures 8, 10, where the y-axis represents the number of communications, we can see the effect of increasing machines only from the algorithm aspect regardless of different communication overheads when employing various distributed computing frameworks (Dean and Ghemawat, 2008; Zaharia et al., 2012; Li et al., 2014). Besides, we show the actual running time versus the number of machines in Figures 9, 11 with the communication time colored green. Empirical results show that Acc-DADM usually enjoys good scalability. Especially when λ is relatively small, such as $1e^{-7}$, CoCoA$^+$ may not reach enough accuracy even after 100 passes over the data, while Acc-DADM works significantly better. These observations are consistent with our theory.

Non-smooth Losses Our acceleration technique also works for non-smooth losses (see Section 8.2). Figures 12, 13 show experimental results when employing the hinge loss $\phi_i(a) = \max\{0, 1 - a\}$. In practice, we just set $\kappa = \frac{mL^2R}{n} - \lambda$ and run Algorithm 3 directly without applying the smoothing technique described in Section 8.2. We can observe that Acc-DADM also enjoys the acceleration effect as in the smooth case and converges significantly faster than CoCoA$^+$ when λ is small.

11. Conclusions

In this paper, we have introduced a novel distributed dual formulation for regularized loss minimization problems. Based on this new formulation, we studied a distributed generalization of the single-machine ProxSDCA, which we refer to as DADM. We have shown that the analysis of ProxSDCA can be easily generalized to establish the convergence of DADM. Moreover, we have adapted AccProxSDCA to the distributed setting by using this new dual formulation and provided corresponding theoretical guarantees. We performed numerous experiments on real datasets to validate our theory and show that our new approach improves the previous state-of-the-art in distributed dual optimization.

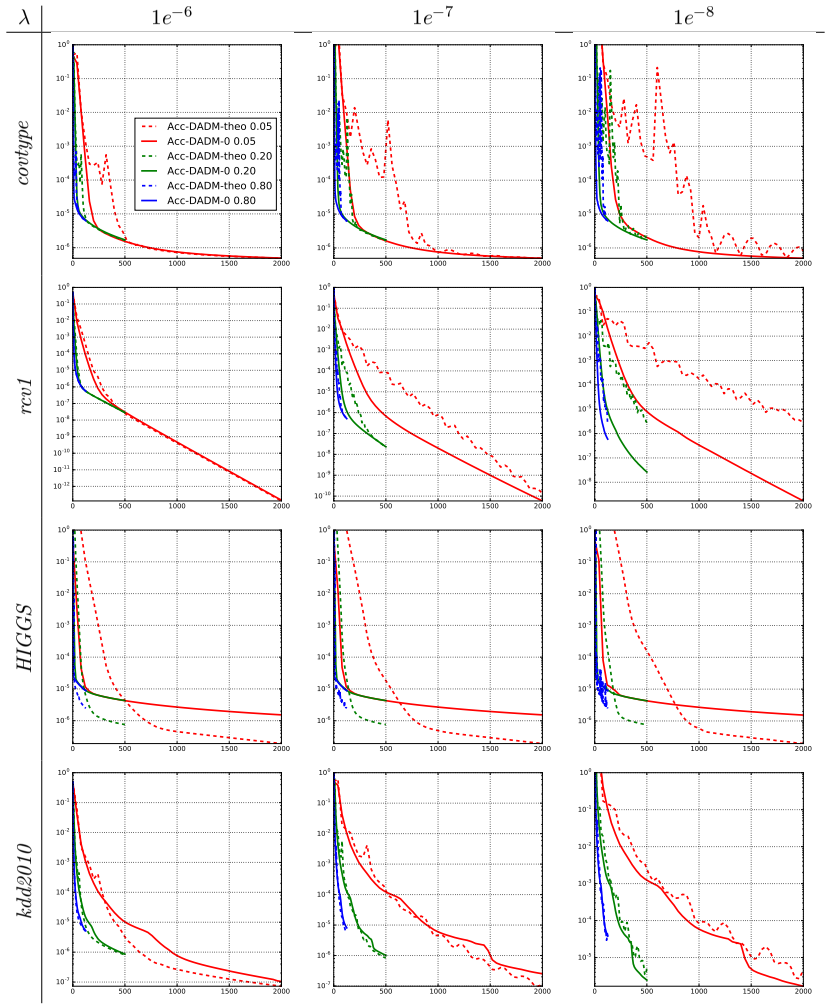


Figure 1: **The normalized duality gap versus the number of communications** of **SVM** experiments with $\mu = 1e^{-5}$, λ varing in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ and sp varing in the range $\{0.05, 0.20, 0.80\}$ on four datasets. Acc-DADM-theo represents the theory suggested ν, Acc-DADM-0 represents the empirical choice $\nu = 0$. We run methods in each case for 100 passes over the data.

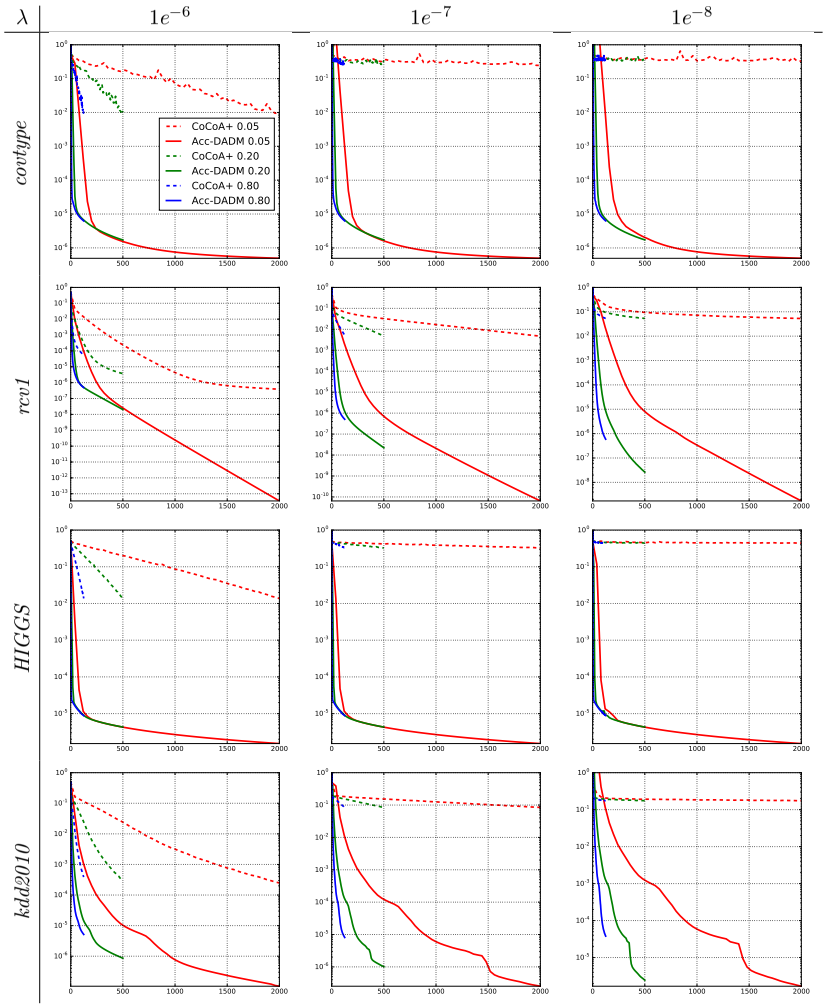


Figure 2: **The normalized duality gap versus the number of communications** of **SVM** experiments with $\mu = 1e^{-5}$, λ varing in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ and sp varing in the range $\{0.05, 0.20, 0.80\}$ on four datasets. We run methods in each case for 100 passes over the data.

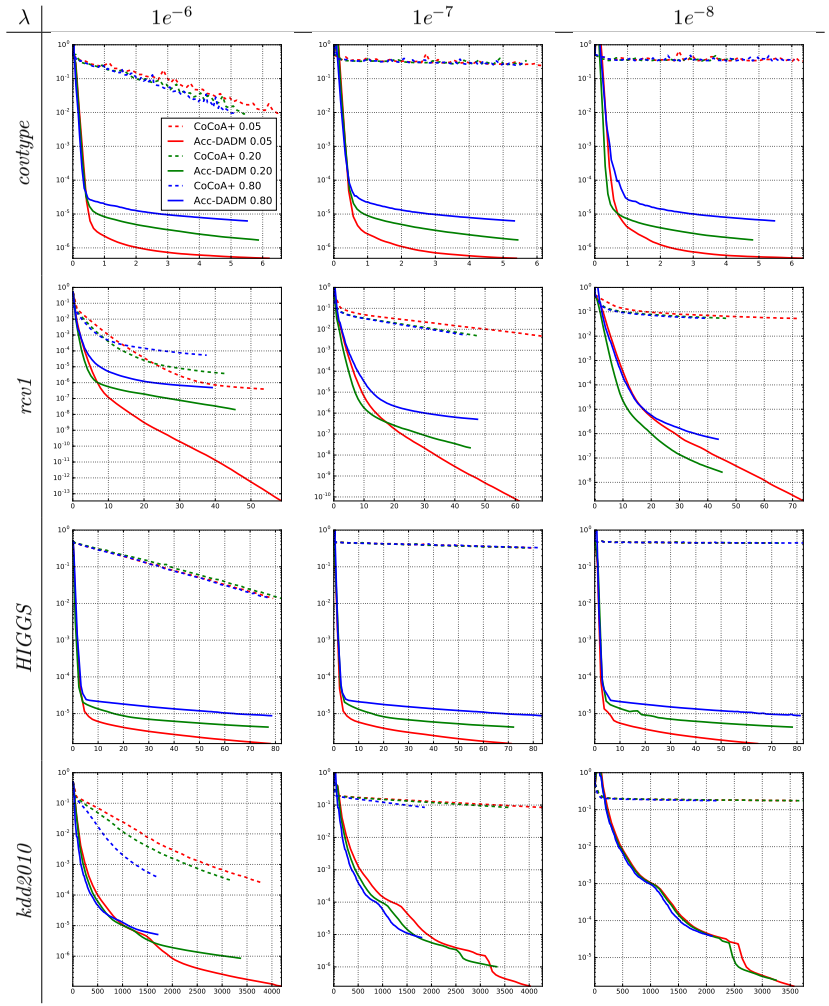


Figure 3: **The normalized duality gap versus time (s)** of **SVM** experiments with $\mu = 1e^{-5}$, λ varing in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ and sp varing in the range $\{0.05, 0.20, 0.80\}$ on four datasets. We run methods in each case for 100 passes over the data.

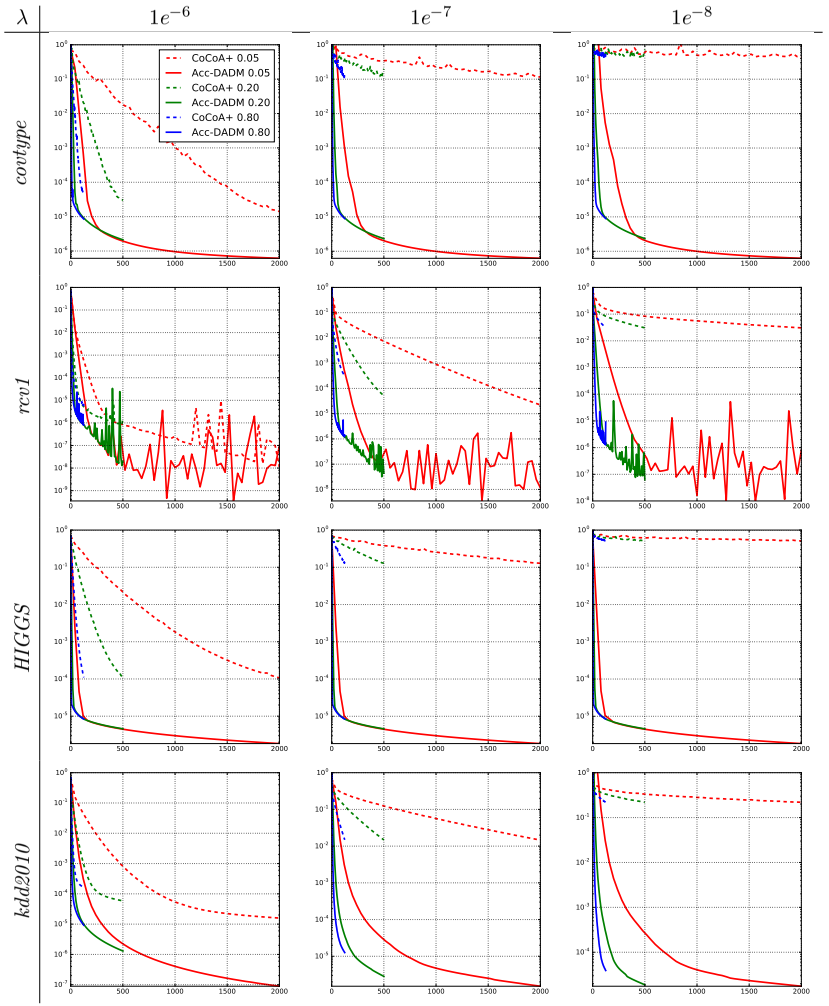


Figure 4: **The normalized duality gap versus the number of communications** of **LR** experiments with $\mu = 1e^{-5}$, λ varing in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ and sp varing in the range $\{0.05, 0.20, 0.80\}$ on four datasets. We run methods in each case for 100 passes over the data.

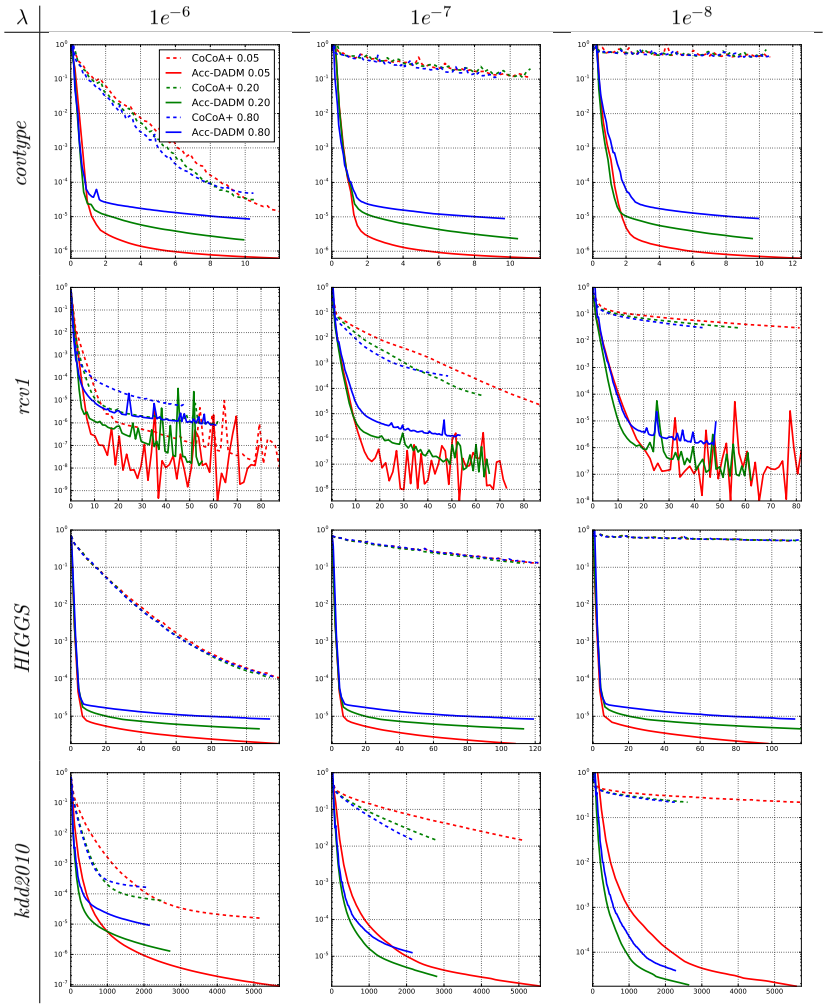


Figure 5: **The normalized duality gap versus time(s)** of LR experiments with $\mu = 1e^{-5}$, λ varing in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ and sp varing in the range $\{0.05, 0.20, 0.80\}$ on four datasets. We run methods in each case for 100 passes over the data.

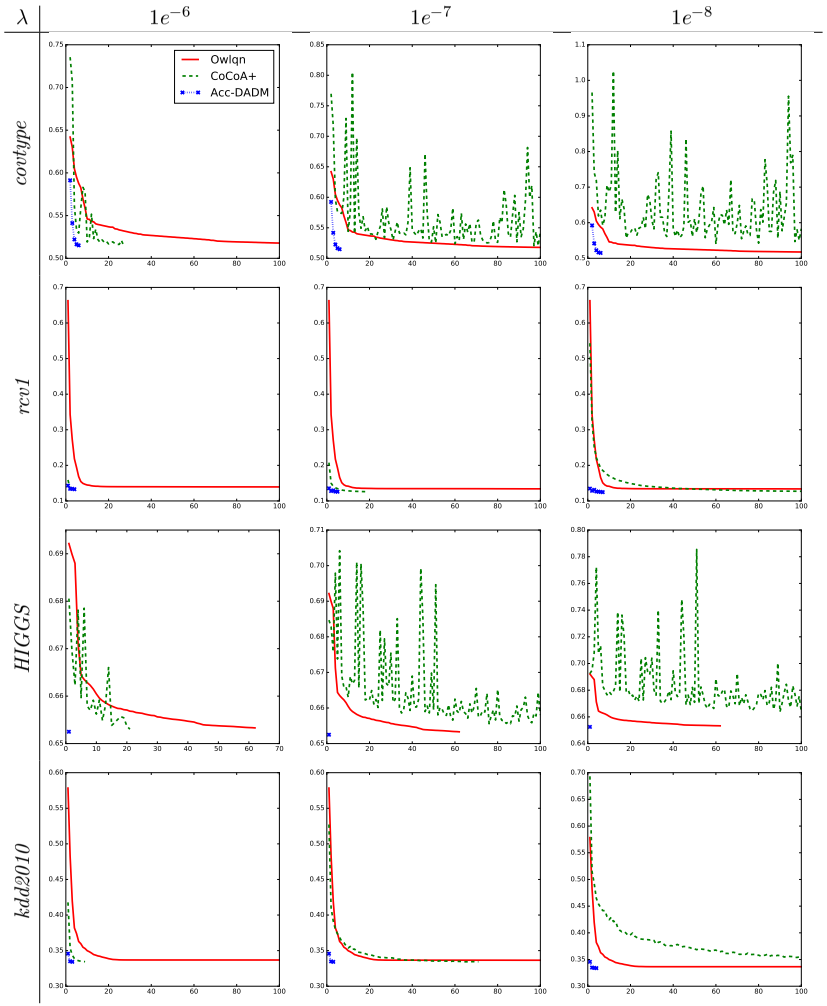


Figure 6: **The normalized primal objective versus the number of passes over the data** of **LR** experiments with $\mu = 1e^{-5}$, λ varing in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ and $sp = 1.0$ on four datasets. We terminate methods either if the stopping condition is met ($1e^{-3}$ duality gap) or after 100 passes over the data.

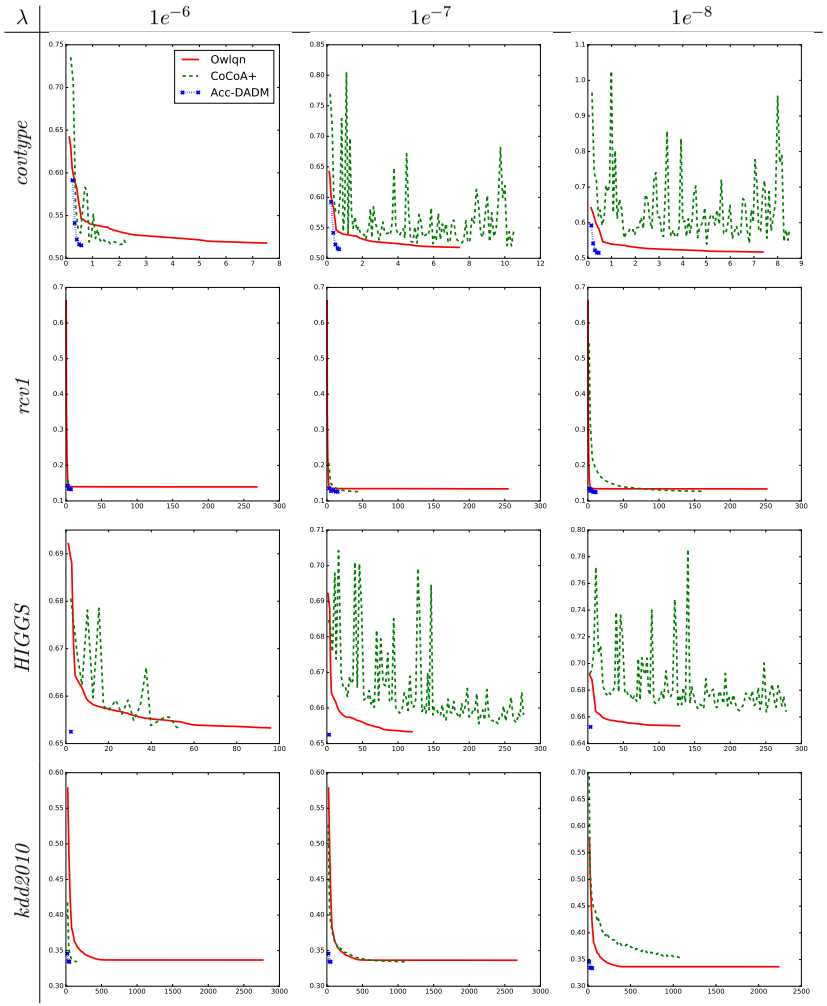


Figure 7: **The normalized primal objective versus time (s)** of **LR** experiments with $\mu = 1e^{-5}$, λ varing in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ and $sp = 1.0$ on four datasets. We terminate methods either if the stopping condition is met ($1e^{-3}$ duality gap) or after 100 passes over the data.

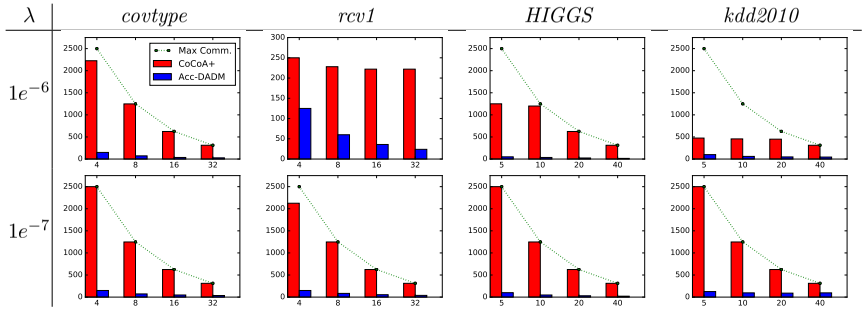


Figure 8: **The number of communications to reach $1e^{-3}$ duality gap versus the number of machines** of **SVM** experiments on four datasets. We fix the mini-batch size by varying sp as 0.04, 0.08, 0.16, 0.32 when m grows exponentially from 4 to 32 or from 10 to 80. For each case, we run algorithms for at most 100 passes over the data. *Max Comm.* represents the total number of communications needed to go through 100 passes over the data.

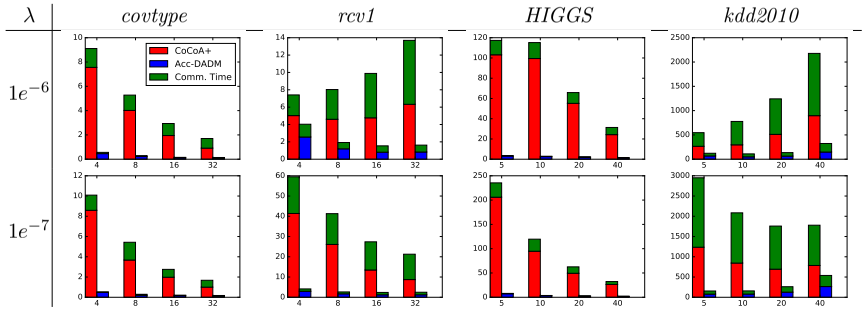


Figure 9: **Time (s) to reach $1e^{-3}$ duality gap versus the number of machines** of **SVM** experiments on four datasets. We fix the mini-batch size by varying sp as 0.04, 0.08, 0.16, 0.32 when m grows exponentially from 4 to 32 or from 10 to 80. For each case, we run algorithms for at most 100 passes over the data. *Comm. Time* represents the total communication time of the corresponding algorithm.

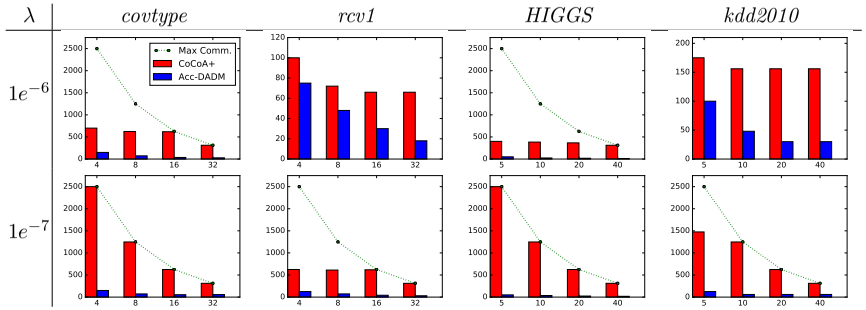


Figure 10: **The number of communications to reach $1e^{-3}$ duality gap versus the number of machines** of **LR** experiments on four datasets. We fix the mini-batch size by varying *sp* as 0.04, 0.08, 0.16, 0.32 when m grows exponentially from 4 to 32 or from 10 to 80. For each case, we run algorithms for at most 100 passes over the data. *Max Comm.* represents the total number of communications needed to go through 100 passes over the data.

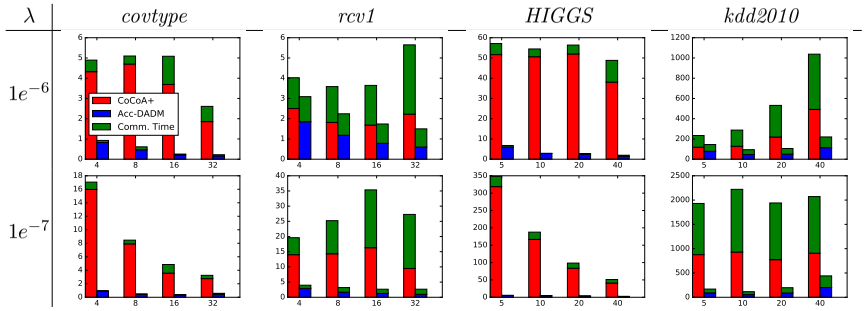


Figure 11: **Time (s) to reach $1e^{-3}$ duality gap versus the number of machines** of **LR** experiments on four datasets. We fix the mini-batch size by varying *sp* as 0.04, 0.08, 0.16, 0.32 when m grows exponentially from 4 to 32 or from 10 to 80. For each case, we run algorithms for at most 100 passes over the data. *Comm. Time* represents the total communication time of the corresponding algorithm.

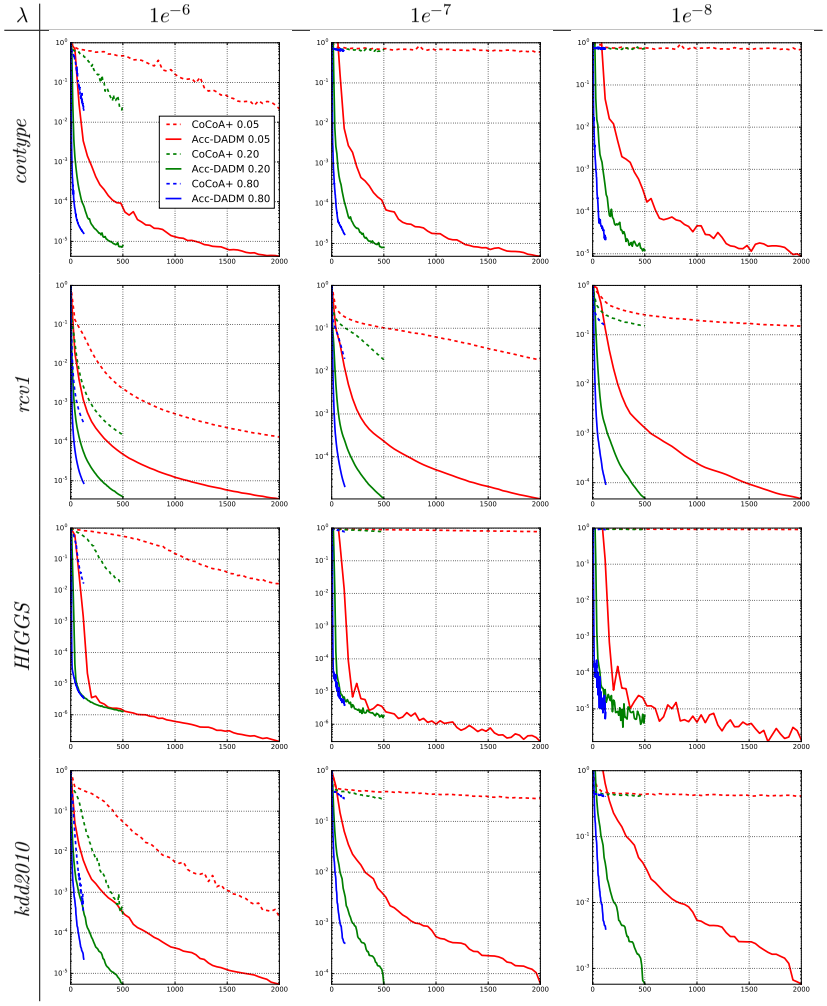


Figure 12: **The normalized duality gap versus the number of communications** of **the hinge loss** experiments with $\mu = 1e^{-5}$, λ varing in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ and *sp* varing in the range $\{0.05, 0.20, 0.80\}$ on four datasets. We run methods in each case for 100 passes over the data.

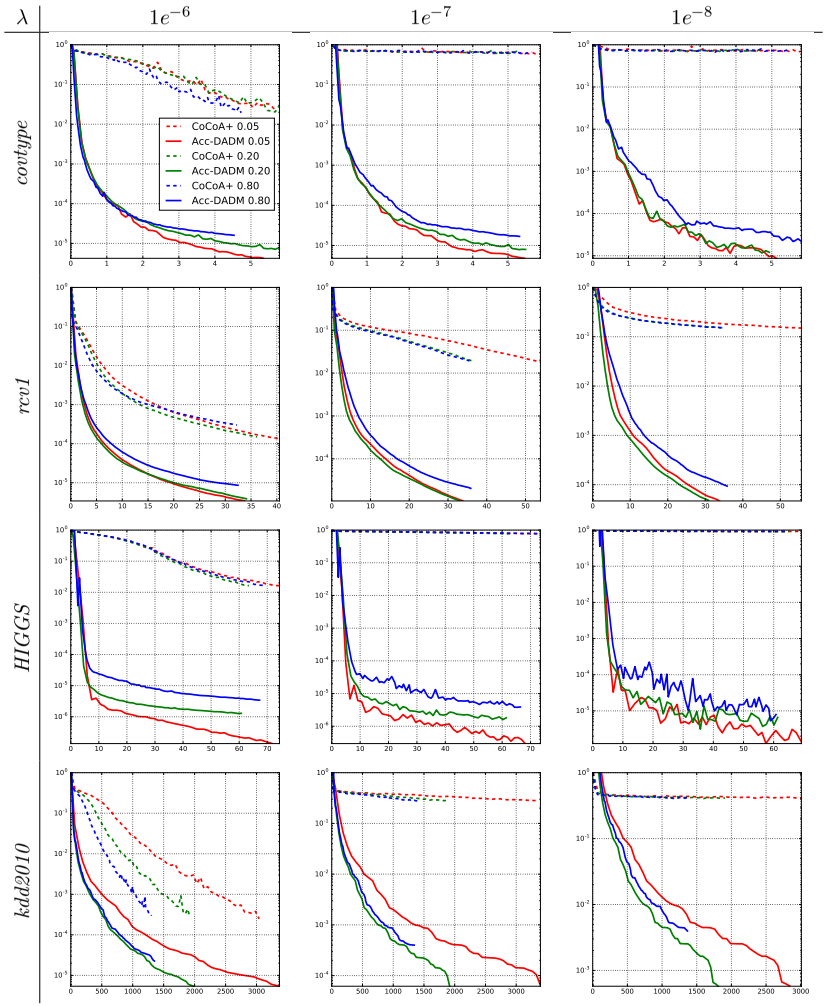


Figure 13: **The normalized duality gap versus time (s)** of **the hinge loss** experiments with $\mu = 1e^{-5}$, λ varing in the range $\{1e^{-6}, 1e^{-7}, 1e^{-8}\}$ and *sp* varing in the range $\{0.05, 0.20, 0.80\}$ on four datasets. We run methods in each case for 100 passes over the data.

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China (NSFC) grant 61532001, Tsinghua Initiative Research Program Grant 20151080475, MOE Online Education Research Center (Quantong Fund) grant 2017ZD203, and gift funds from Huawei and Ant Financial.

References

Galen Andrew and Jianfeng Gao. Scalable training of l_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40. ACM, 2007.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010.

Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: Approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2540–2548, 2015.

Richard Graham, Timothy Woodall, and Jeffrey Squyres. Open MPI: A flexible high performance MPI. *Parallel Processing and Applied Mathematics*, pages 228–239, 2006.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th International Conference on Machine Learning*, pages 408–415, 2008.

Cho-jui Hsieh, Hsiang-fu Yu, and Inderjit Dhillon. PASSCoDe: Parallel asynchronous stochastic dual co-ordinate descent. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2370–2379, 2015.

Martin Jaggi, Virginia Smith, Martin Takáč, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I. Jordan. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pages 3068–3076, 2014.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Ching-pei Lee and Dan Roth. Distributed box-constrained quadratic optimization for dual linear SVM. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 987–996, 2015.

Mu Li, David G. Andersen, Jun Woo Park, Alexander J. Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J. Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation*, pages 583–598, 2014.

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I. Jordan, Peter Richtárik, and Martin Takáč. Adding vs. averaging in distributed primal-dual optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1973–1982, 2015.

Chenxin Ma, Jakub Konečnỳ, Martin Jaggi, Virginia Smith, Michael I. Jordan, Peter Richtárik, and Martin Takáč. Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, pages 1–36, 2017.

Dhruv Mahajan, Nikunj Agrawal, S. Sathiya Keerthi, S. Sundararajan, and Léon Bottou. An efficient distributed learning algorithm based on effective local functional approximations. *arXiv preprint arXiv:1310.8418*, 2013.

Dhruv Mahajan, S. Sathiya Keerthi, and S. Sundararajan. A distributed block coordinate descent method for training l_1 regularized linear classifiers. *Journal of Machine Learning Research*, 18(91):1–35, 2017.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, 2004.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

Feng Niu, Benjamin Recht, Christopher Ré, and Stephen J. Wright. HOGWILD!: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 693–701, 2011.

Brendan Odonoghue and Emmanuel Candès. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(1):567–599, 2013.

Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1-2):1–41, 2014.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate Newton-type method. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1000–1008, 2014.

Virginia Smith, Simone Forte, Chenxin Ma, Martin Takáč, Michael I. Jordan, and Martin Jaggi. CoCoA: A general framework for communication-efficient distributed optimization. *arXiv preprint arXiv:1611.02189*, 2016.

Martin Takáč, Avleen Bijral, Peter Richtárik, and Nati Srebro. Mini-batch primal and dual methods for SVMs. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1022–1030, 2013.

Martin Takáč, Peter Richtárik, and Nathan Srebro. Distributed mini-batch SDCA. *arXiv preprint arXiv:1507.08322*, 2015.

Tianbao Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pages 629–637, 2013.

Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, pages 2–2. USENIX Association, 2012.

Yuchen Zhang and Xiao Lin. DiSCO: Distributed optimization for self-concordant empirical loss. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 362–370, 2015.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J. Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2010.

Journal of Machine Learning Research 18 (2017) 1-40 Submitted 9/16; Revised 8/17; Published 11/17

Second-Order Stochastic Optimization for Machine Learning in Linear Time

Naman Agarwal NAMANA@CS.PRINCETON.EDU
Computer Science Department
Princeton University
Princeton, NJ 08540, USA

Brian Bullins BBULLINS@CS.PRINCETON.EDU
Computer Science Department
Princeton University
Princeton, NJ 08540, USA

Elad Hazan EHAZAN@CS.PRINCETON.EDU
Computer Science Department
Princeton University
Princeton, NJ 08540, USA

Editor: Tong Zhang

Abstract

First-order stochastic methods are the state-of-the-art in large-scale machine learning optimization owing to efficient per-iteration complexity. Second-order methods, while able to provide faster convergence, have been much less explored due to the high cost of computing the second-order information. In this paper we develop second-order stochastic methods for optimization problems in machine learning that match the per-iteration cost of gradient based methods, and in certain settings improve upon the overall running time over popular first-order methods. Furthermore, our algorithm has the desirable property of being implementable in time linear in the sparsity of the input data.

Keywords: second-order optimization, convex optimization, regression

1. Introduction

In recent literature stochastic first-order optimization has taken the stage as the primary workhorse for training learning models, due in large part to its affordable computational costs which are linear (in the data representation) per iteration. The main research effort devoted to improving the convergence rates of first-order methods have introduced elegant ideas and algorithms in recent years, including adaptive regularization (Duchi et al., 2011), variance reduction (Johnson and Zhang, 2013; Defazio et al., 2014), dual coordinate ascent (Shalev-Shwartz and Zhang, 2013), and many more.

In contrast, second-order methods have typically been much less explored in large scale machine learning (ML) applications due to their prohibitive computational cost per iteration which requires computation of the Hessian in addition to a matrix inversion. These operations are infeasible for large scale problems in high dimensions.

©2017 Naman Agarwal, Brian Bullins and Elad Hazan.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-491.html.

In this paper we propose a family of novel second-order algorithms, LiSSA (Linear time Stochastic Second-Order Algorithm) for convex optimization that attain fast convergence rates while also allowing for an implementation with linear time per-iteration cost, matching the running time of the best known gradient-based methods. Moreover, in the setting where the number of training examples m is much larger than the underlying dimension d, we show that our algorithm has provably faster running time than the best known gradient-based methods.

Formally, the main optimization problem we are concerned with is the empirical risk minimization (ERM) problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^d} \left\{ \frac{1}{m}\sum_{k=1}^{m} f_k(\mathbf{x}) + R(\mathbf{x}) \right\}$$

where each $f_k(\mathbf{x})$ is a convex function and $R(\mathbf{x})$ is a convex regularizer. The above optimization problem is the standard objective minimized in most supervised learning settings. Examples include logistic regression, SVMs, etc. A common aspect of many applications of ERM in machine learning is that the loss function $f_i(\mathbf{x})$ is of the form $l(\mathbf{x}^\top \mathbf{v}_i, y_i)$ where $(\mathbf{v}_i, y_i)$ is the ith training example-label pair. We call such functions generalized linear models (GLM) and will restrict our attention to this case. We will assume that the regularizer is an ℓ_2 regularizer,[1] typically $\|\mathbf{x}\|^2$.

Our focus is second-order optimization methods (Newton's method), where in each iteration, the underlying principle is to move to the minimizer of the second-order Taylor approximation at any point. Throughout the paper, we will let $\nabla^{-2} f(\mathbf{x}) \triangleq \left[\nabla^2 f(\mathbf{x})\right]^{-1}$. The update of Newton's method at a point $\mathbf{x}_t$ is then given by

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \nabla^{-2} f(\mathbf{x}_t) \nabla f(\mathbf{x}_t). \tag{1}$$

Certain desirable properties of Newton's method include the fact that its updates are independent of the choice of coordinate system and that the Hessian provides the necessary regularization based on the curvature at the present point. Indeed, Newton's method can be shown to eventually achieve quadratic convergence (Nesterov, 2013). Although Newton's method comes with good theoretical guarantees, the complexity per step grows roughly as $\Omega(md^2 + d^\omega)$ (the former term for computing the Hessian and the latter for inversion, where $\omega \approx 2.37$ is the matrix multiplication constant), making it prohibitive in practice. Our main contribution is a suite of algorithms, each of which performs an approximate Newton update based on stochastic Hessian information and is implementable in linear $O(d)$ time. These algorithms match and improve over the performance of first-order methods in theory and give promising results as an optimization method on real world data sets. In the following we give a summary of our results. We propose two algorithms, LiSSA and LiSSA-Sample.

LiSSA: Algorithm 1 is a practical stochastic second-order algorithm based on a novel estimator of the Hessian inverse, leading to an efficient approximate Newton step (Equation 1). The estimator is based on the well known Taylor approximation of the inverse (Fact 2) and is described formally in Section 3.1. We prove the following informal theorem about LiSSA.

1. It can be seen that some of these assumptions can be relaxed in our analyses, but since these choices are standard in ML, we make these assumptions to simplify the discourse.

Theorem 1 (Informal) *LiSSA returns a point* $\mathbf{x}_t$ *such that* $f(\mathbf{x}_t) \leq \min_{\mathbf{x}^*} f(\mathbf{x}^*) + \varepsilon$ *in total time*

$$\tilde{O}\left((m + S_1\kappa)\, d \log\left(\frac{1}{\varepsilon}\right)\right)$$

where κ *is the underlying condition number of the problem and* S_1 *is a bound on the variance of the estimator.*

The precise version of the above theorem appears as Theorem 7. In theory, the best bound we can show for S_1 is $O(\kappa^2)$; however, in our experiments we observe that setting S_1 to be a small constant (often 1) is sufficient. We conjecture that S_1 can be improved to $O(1)$ and leave this for future work. If indeed S_1 can be improved to $O(1)$ (as is indicated by our experiments), LiSSA enjoys a convergence rate comparable to first-order methods. We provide a detailed comparison of our results with existing first-order and second-order methods in Section 1.2. Moreover, in Section 7 we present experiments on real world data sets that demonstrate that LiSSA as an optimization method performs well as compared to popular first-order methods. We also show that LiSSA runs in time proportional to input sparsity, making it an attractive method for high-dimensional sparse data.

LiSSA-Sample: This variant brings together efficient first-order algorithms with matrix sampling techniques (Li et al., 2013; Cohen et al., 2015) to achieve better runtime guarantees than the state-of-the-art in convex optimization for machine learning in the regime when $m > d$. Specifically, we prove the following theorem:

Theorem 2 (Informal) *LiSSA-Sample returns a point* $\mathbf{x}_t$ *such that* $f(\mathbf{x}_t) \leq \min_{\mathbf{x}^*} f(\mathbf{x}^*) + \varepsilon$ *in total time*

$$\tilde{O}\left(m + \sqrt{\kappa d}\right) d \log^2\left(\frac{1}{\varepsilon}\right) \log\log\left(\frac{1}{\varepsilon}\right).$$

The above result improves upon the best known running time for first-order methods achieved by acceleration when we are in the setting where $\kappa > m >> d$. We discuss the implication of our bounds and further work in Section 1.3.

In all of our results stated above κ corresponds to the condition number of the underlying problem. In particular we assume some strong convexity for the underlying problem. This is a standard assumption which is usually enforced by the addition of the ℓ_2 regularizer. In stating our results formally we stress on the nuances between different notions of the condition number (ref. Section 2.1), and we state our results precisely with respect to these notions. In general, all of our generalization/relaxations of the condition number are smaller than $\frac{1}{\lambda}$ where λ is the regularization parameter, and this is usually taken to be the condition number of the problem. The condition of strong convexity has been relaxed in literature by introducing proximal methods. It is an interesting direction to adapt our results in those setting which we leave for future work.

We also remark that all of our results focus on the very high accuracy regime. In general the benefits of linear convergence and second-order methods can be seen to be effective only when considerably small error is required. This is also the case for recent advances in fast first-order methods where their improvement over stochastic gradient descent becomes apparent only in the high accuracy regime. Our experiments also demonstrate that second-order methods can improve upon fast first-order methods in the regime of very high accuracy.

While it is possible that this regime is less interesting for generalization, in this paper we focus on the optimization problem itself.

We further consider the special case when the function f is self-concordant. Self-concordant functions are a sub-class of convex functions which have been extensively studied in convex optimization literature in the context of interior point methods (Nemirovski, 2004). For self-concordant functions we propose an algorithm (Algorithm 5) which achieves linear convergence with running time guarantees independent of the condition number. We prove the formal running time guarantee as Theorem 25.

We believe our main contribution to be a demonstration of the fact that second-order methods are comparable to, or even better than, first-order methods in the large data regime, in both theory and practice.

1.1 Overview of Techniques

LiSSA: The key idea underlying LiSSA is the use of the Taylor expansion to construct a natural estimator of the Hessian inverse. Indeed, as can be seen from the description of the estimator in Section 3.1, the estimator we construct becomes unbiased in the limit as we include additional terms in the series. We note that this is not the case with estimators that were considered in previous works such as that of Erdogdu and Montanari (2015), and so we therefore consider our estimator to be more natural. In the implementation of the algorithm we achieve the optimal bias/variance trade-off by truncating the series appropriately.

An important observation underlying our linear time $O(d)$ step is that for GLM functions, $\nabla^2 f_i(\mathbf{x})$ has the form $\alpha \mathbf{v}_i \mathbf{v}_i^\top$ where α is a scalar dependent on $\mathbf{v}_i^\top \mathbf{x}$. A single step of LiSSA requires us to efficiently compute $\nabla^2 f_i(\mathbf{x})\mathbf{b}$ for a given vector $\mathbf{b}$. In this case it can be seen that the matrix-vector product reduces to a vector-vector product, giving us an $O(d)$ time update.

LiSSA-Sample: LiSSA-Sample is based on Algorithm 2, which represents a general family of algorithms that couples the quadratic minimization view of Newton's method with any efficient first-order method. In essence, Newton's method allows us to reduce (up to $\log\log$ factors) the optimization of a general convex function to solving intermediate quadratic or ridge regression problems. Such a reduction is useful in two ways.

First, as we demonstrate through our algorithm LiSSA-Sample, the quadratic nature of ridge regression problems allows us to leverage powerful sampling techniques, leading to an improvement over the running time of the best known accelerated first-order method. On a high level this improvement comes from the fact that when solving a system of m linear equations in d dimensions, a constant number of passes through the data is enough to reduce the system to $O(d\log(d))$ equations. We carefully couple this principle and the computation required with accelerated first-order methods to achieve the running times for LiSSA-Sample. The result for the quadratic sub-problem (ridge regression) is stated in Theorem 15, and the result for convex optimization is stated in Theorem 16.

The second advantage of the reduction to quadratic sub-problems comes from the observation that the intermediate quadratic sub-problems can potentially be better conditioned than the function itself, allowing us a better choice of the step size in practice. We define these local notions of condition number formally in Section 2.1 and summarize the typical benefits for such algorithms in Theorem 13. In theory this is not a significant improve-

ment; however, in practice we believe that this could be significant and lead to runtime improvements.[2]

To achieve the bound for LiSSA-Sample we extend the definition and procedure for sampling via leverage scores described by Cohen et al. (2015) to the case when the matrix is given as a sum of PSD matrices and not just rank one matrices. We reformulate and reprove the theorems proved by Cohen et al. (2015) in this context, which may be of independent interest.

1.2 Comparison with Related Work

In this section we aim to provide a short summary of the key ideas and results underlying optimization methods for large scale machine learning. We divide the summary into three high level principles: first-order gradient-based methods, second-order Hessian-based methods, and quasi-Newton methods. For the sake of brevity we will restrict our summary to results in the case when the objective is strongly convex, which as justified above is usually ensured by the addition of an appropriate regularizer. In such settings the main focus is often to obtain algorithms which have provably linear convergence and fast implementations.

First-Order Methods: First-order methods have dominated the space of optimization algorithms for machine learning owing largely to the fact that they can be implemented in time proportional to the underlying dimension (or sparsity). Gradient descent is known to converge linearly to the optimum with a rate of convergence that is dependent upon the condition number of the objective. In the large data regime, stochastic first-order methods, introduced and analyzed first by Robbins and Monro (1951), have proven especially successful. Stochastic gradient descent (SGD), however, converges sub-linearly even in the strongly convex setting. A significant advancement in terms of the running time of first-order methods was achieved recently by a clever merging of stochastic gradient descent with its full version to provide *variance reduction*. The representative algorithms in this space are SAGA (Roux et al., 2012; Defazio et al., 2014) and SVRG (Johnson and Zhang, 2013; Zhang et al., 2013). The key technical achievement of the above algorithms is to relax the running time dependence on m (the number of training examples) and κ (the condition number) from a product to a sum. Another algorithm which achieves similar running time guarantees is based on dual coordinate ascent, known as SDCA (Shalev-Shwartz and Zhang, 2013).

Further improvements over SAGA, SVRG and SDCA have been obtained by applying the classical idea of *acceleration* emerging from the seminal work of Nesterov (1983). The progression of work here includes an accelerated version of SDCA (Shalev-Shwartz and Zhang, 2016); APCG (Lin et al., 2014); Catalyst (Lin et al., 2015), which provides a generic framework to accelerate first order algorithms; and Katyusha (Allen-Zhu, 2016), which introduces the concept of negative momentum to extend acceleration for variance reduced algorithms beyond the strongly convex setting. The key technical achievement of accelerated methods in general is to reduce the dependence on condition number from linear to a square root. We summarize these results in Table 1.

2. While this is a difficult property to verify experimentally, we conjecture that this is a possible explanation for why LiSSA performs better than first-order methods on certain data sets and ranges of parameters.

LiSSA places itself naturally into the space of fast first-order methods by having a running time dependence that is comparable to SAGA/SVRG (ref. Table 1). In LiSSA-Sample we leverage the quadratic structure of the sub-problem for which efficient sampling techniques have been developed in the literature and use accelerated first-order methods to improve the running time in the case when the underlying dimension is much smaller than the number of training examples. Indeed, to the best of our knowledge LiSSA-Sample is the theoretically fastest known algorithm under the condition $m >> d$. Such an improvement seems out of hand for the present first-order methods as it seems to strongly leverage the quadratic nature of the sub-problem to reduce its size. We summarize these results in Table 1.

Algorithm	Runtime	Reference
SVRG, SAGA, SDCA	$(md + O(\hat{\kappa}d)) \log\left(\frac{1}{\varepsilon}\right)$	Johnson and Zhang (2013) Zhang et al. (2013) Roux et al. (2012) Shalev-Shwartz and Zhang (2013)
LiSSA	$(md + O(\hat{\kappa}_l)S_1 d) \log\left(\frac{1}{\varepsilon}\right)$	Corollary 8
AccSDCA, Catalyst Katyusha	$\tilde{O}\left(md + d\sqrt{\hat{\kappa}m}\right) \log\left(\frac{1}{\varepsilon}\right)$	Shalev-Shwartz and Zhang (2016) Lin et al. (2015) Allen-Zhu (2016)
LiSSA-Sample	$\tilde{O}\left(md + d\sqrt{\kappa_{sample}\bar{d}}\right) \log^2\left(\frac{1}{\varepsilon}\right)$	Theorem 16

Table 1: Run time comparisons. Refer to Section 2.1 for definitions of the various notions of condition number.

Second-Order Methods: Second-order methods such as Newton's method have classically been used in optimization in many different settings including development of interior point methods (Nemirovski, 2004) for general convex programming. The key advantage of Newton's method is that it achieves a linear-quadratic convergence rate. However, naive implementations of Newton's method have two significant issues, namely that the standard analysis requires the full Hessian calculation which costs $O(md^2)$, an expense not suitable for machine learning applications, and the matrix inversion typically requires $O(d^3)$ time. These issues were addressed recently by the algorithm NewSamp (Erdogdu and Montanari, 2015) which tackles the first issue by subsampling and the second issue by low-rank projections. We improve upon the work of Erdogdu and Montanari (2015) by defining a more *natural* estimator for the Hessian inverse and by demonstrating that the estimator can be computed in time proportional to $O(d)$. We also point the reader to the works of Martens (2010); Byrd et al. (2011) which incorporate the idea of taking samples of the Hessian; however, these works do not provide precise running time guarantees on their proposed algorithm based on problem specific parameters. Second-order methods have also enjoyed success in the distributed setting (Shamir et al., 2014).

Quasi-Newton Methods: The expensive computation of the Newton step has also been tackled via estimation of the curvature from the change in gradients. These algorithms are generally known as quasi-Newton methods stemming from the seminal BFGS algo-

rithm (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970). The book of Nocedal and Wright (2006) is an excellent reference for the algorithm and its limited memory variant (L-BFGS). The more recent work in this area has focused on stochastic quasi-Newton methods which were proposed and analyzed in various settings by Schraudolph et al. (2007); Mokhtari and Ribeiro (2014); Byrd et al. (2016). These works typically achieve sub-linear convergence to the optimum. A significant advancement in this line of work was provided by Moritz et al. (2016) who propose an algorithm based on L-BFGS by incorporating ideas from variance reduction to achieve linear convergence to the optimum in the strongly convex setting. Although the algorithm achieves linear convergence, the running time of the algorithm depends poorly on the condition number (as acknowledged by the authors). Indeed, in applications that interest us, the condition number is not necessarily a constant as is typically assumed to be the case for the theoretical results in Moritz et al. (2016).

Our key observation of linear time Hessian-vector product computations for machine learning applications provides evidence that in such instances, obtaining true Hessian information is efficient enough to alleviate the need for quasi-Newton information via gradients.

1.3 Discussion and Subsequent Work

In this section we provide a brief survey of certain technical aspects of our bounds which have since been improved by subsequent work.

An immediate improvement in terms of $S_2 \sim \kappa$ (in fact suggested in the original manuscript) was achieved by Bollapragada et al. (2016) via conjugate gradient on a sub-sampled Hessian which reduces this to $\sqrt{\kappa}$. A similar improvement can also be achieved in theory through the extensions of LiSSA proposed in the paper. As we show in Section 7, the worse dependence on condition number has an effect on the running time when κ is quite large.[3] Accelerated first-order methods, such as APCG (Lin et al., 2014), outperform LiSSA in this regime. To the best of our knowledge second-order stochastic methods have so far not exhibited an improvement in that regime experimentally. We believe a more practical version of LiSSA-Sample could lead to improvements in this regime, leaving this as future work.

To the best of our knowledge the factor of $S_1 = \kappa^2$ that appears to reduce the variance of our estimator has yet not been improved despite it being $O(1)$ in our experiments. This is an interesting question to which partial answers have been provided in the analysis of Ye et al. (2017).

Significant progress has been made in the space of inexact Newton methods based on matrix sketching techniques. We refer the reader to the works of Pilanci and Wainwright (2015); Xu et al. (2016); Cohen (2016); Luo et al. (2016); Ye et al. (2017) and the references therein.

We would also like to comment on the presence of a warm start parameter $\frac{1}{\kappa M}$ in our proofs of Theorems 7 and 15. In our experiments the warm start we required would be quite small (often a few steps of gradient descent would be sufficient) to make LiSSA converge. The warm start does not affect the asymptotic results proven in Theorems 7 and 15 because getting to such a warm start is independent of ε. However, improving this warm start, especially in the context of Theorem 15, is left as interesting future work.

3. Equivalently, λ is small.

On the complexity side, Arjevani and Shamir (2016) proved lower bounds on the best running times achievable by second-order methods. In particular, they show that to get the faster rates achieved by LiSSA-Sample, it is necessary to use a non-uniform sampling based method as employed by LiSSA-Sample. We would like to remark that in theory, up to logarithmic factors, the running time of LiSSA-Sample is still the best achieved so far in the setting $m >> d$. Some of the techniques and motivations from this work were also generalized by the authors to provide faster rates for a large family of non-convex optimization problems (Agarwal et al., 2017).

1.4 Organization of the Paper

The paper is organized as follows: we first present the necessary definitions, notations and conventions adopted throughout the paper in Section 2. We then describe our estimator for LiSSA, as well as state and prove the convergence guarantee for LiSSA in Section 3. After presenting a generic procedure to couple first-order methods with Newton's method in 4, we present LiSSA-Sample and the associated fast quadratic solver in Section 5. We then present our results regarding self-concordant functions in Section 6. Finally, we present an experimental evaluation of LiSSA in Section 7.

2. Preliminaries

We adopt the convention of denoting vectors and scalars in lowercase, matrices in uppercase, and vectors in boldface. We will use $\|\cdot\|$ without a subscript to denote the ℓ_2 norm for vectors and the spectral norm for matrices. Throughout the paper we denote $\mathbf{x}^* \triangleq \operatorname{argmin}_{\mathbf{x}\in\mathcal{K}} f(\mathbf{x})$. A convex function f is defined to be α-strongly convex and β-smooth if, for all $\mathbf{x}, \mathbf{y}$,

$$\nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}) + \frac{\beta}{2}\|\mathbf{y}-\mathbf{x}\|^2 \geq f(\mathbf{y}) - f(\mathbf{x}) \geq \nabla f(\mathbf{x})^\top(\mathbf{y}-\mathbf{x}) + \frac{\alpha}{2}\|\mathbf{y}-\mathbf{x}\|^2.$$

The following is a well known fact about the inverse of a matrix A s.t. $\|A\| \leq 1$ and $A \succ 0$:

$$A^{-1} = \sum_{i=0}^{\infty}(I-A)^i. \tag{2}$$

2.1 Definition of Condition Numbers

We now define several measures for the condition number of a function f. The differences between these notions are subtle and we use them to precisely characterize the running time for our algorithms.[4]

For an α-strongly convex and β-smooth function f, the condition number of the function is defined as $\kappa(f) \triangleq \frac{\beta}{\alpha}$, or κ when the function is clear from the context. Note that by definition this corresponds to the following notion:

$$\kappa \triangleq \frac{\max_{\mathbf{x}} \lambda_{\max}(\nabla^2 f(\mathbf{x}))}{\min_{\mathbf{x}} \lambda_{\min}(\nabla^2 f(\mathbf{x}))} \ .$$

4. During initial reading we suggest the reader to skip the subtlety with these notions with the knowledge that they are all smaller than the pessimistic bound one can achieve by considering a value proportional to $O(\lambda^{-1})$, where λ is the coefficient of the ℓ_2 regularizer.

We define a slightly relaxed notion of condition number where the *max* moves out of the fraction above. We refer to this notion as a *local* condition number κ_l as compared to the *global* condition number κ defined above:

$$\kappa_l \triangleq \max_{\mathbf{x}} \frac{\lambda_{\max}(\nabla^2 f(\mathbf{x}))}{\lambda_{\min}(\nabla^2 f(\mathbf{x}))} \ .$$

It follows that $\kappa_l \leq \kappa$. The above notions are defined for any general function f, but in the case of functions of the form $f(\mathbf{x}) = \frac{1}{m}\sum_{k=1}^{m} f_k(\mathbf{x})$, a further distinction is made with respect to the component functions. We refer to such definitions of the condition number by $\hat{\kappa}$. In such cases one typically assumes the each component is bounded by $\beta_{max}(\mathbf{x}) \triangleq \max_k \lambda_{\max}(\nabla^2 f_k(\mathbf{x}))$. The running times of algorithms like SVRG depend on the following notion of condition number:

$$\hat{\kappa} = \frac{\max_{\mathbf{x}} \beta_{\max}(\mathbf{x})}{\min_{\mathbf{x}} \lambda_{\min}(\nabla^2 f(\mathbf{x}))} \ .$$

Similarly, we define a notion of local condition number for $\hat{\kappa}$, namely

$$\hat{\kappa}_l \triangleq \max_{\mathbf{x}} \frac{\beta_{\max}(\mathbf{x})}{\lambda_{\min}(\nabla^2 f(\mathbf{x}))},$$

and it again follows that $\hat{\kappa}_l \leq \hat{\kappa}$.

For our (admittedly pessimistic) bounds on the variance we also need a per-component strong convexity bound $\alpha_{\min}(\mathbf{x}) \triangleq \min_k \lambda_{\min}(\nabla^2 f_k(\mathbf{x}))$. We can now define

$$\hat{\kappa}_l^{\max} \triangleq \max_{\mathbf{x}} \frac{\beta_{\max}(\mathbf{x})}{\alpha_{\min}(\mathbf{x})} \ .$$

Assumptions: In light of the previous definitions, we make the following assumptions about the given function $f(\mathbf{x}) = \frac{1}{m}\sum_{k=1}^{m} f_k(\mathbf{x})$ to make the analysis easier. We first assume that the regularization term has been divided equally and included in $f_k(\mathbf{x})$. We further assume that each $\nabla^2 f_k(\mathbf{x}) \preceq I$.[5] We also assume that f is α-strongly convex and β-smooth, $\hat{\kappa}_l$ is the associated local condition number and $\nabla^2 f$ has a Lipschitz constant bounded by M.

We now collect key concepts and pre-existing results that we use for our analysis in the rest of the paper.

Matrix Concentration: The following lemma is a standard concentration of measure result for sums of independent matrices.[6] An excellent reference for this material is by Tropp (2012).

5. The scaling is without loss of generality, even when looking at additive errors, as this gets picked up in the log-term due to the linear convergence.
6. The theorem in the reference states the inequality for the more nuanced bounded variance case. We only state the simpler bounded spectral norm case which suffices for our purposes.

Theorem 3 (Matrix Bernstein, Tropp, 2012) *Consider a finite sequence $\{X_k\}$ of independent, random, Hermitian matrices with dimension d. Assume that*

$$\mathbf{E}[X_k] = 0 \text{ and } \|X_k\| \leq R.$$

Define $Y = \sum_k X_k$. Then we have for all $t \geq 0$,

$$\mathbf{Pr}\left(\|Y\| \geq t\right) \leq d \exp\left(\frac{-t^2}{4R^2}\right).$$

Accelerated SVRG: The following theorem was proved by Lin et al. (2015).

Theorem 4 *Given a function $f(\mathbf{x}) = \frac{1}{m}\sum_{k=1}^{m} f_k(\mathbf{x})$ with condition number κ, the accelerated version of SVRG via Catalyst (Lin et al., 2015) finds an ε-approximate minimum with probability $1-\delta$ in time*

$$\tilde{O}(md + \min(\sqrt{\kappa m}, \kappa)d) \log\left(\frac{1}{\varepsilon}\right).$$

Sherman-Morrison Formula: The following is a well-known expression for writing the inverse of rank one perturbations of matrices:

$$\left(A + \mathbf{v}\mathbf{v}^T\right)^{-1} = A^{-1} - \frac{A^{-1}\mathbf{v}\mathbf{v}^T A^{-1}}{1 + \mathbf{v}^T A^{-1}\mathbf{v}} \ .$$

3. LiSSA: Linear (time) Stochastic Second-Order Algorithm

In this section, we provide an overview of LiSSA (Algorithm 1) along with its main convergence results.

3.1 Estimators for the Hessian Inverse

Based on a recursive reformulation of the Taylor expansion (Equation 2), we may describe an unbiased estimator of the Hessian. For a matrix A, define A_j^{-1} as the first j terms in the Taylor expansion, i.e.,

$$A_j^{-1} \triangleq \sum_{i=0}^{j} (I - A)^i, \text{ or equivalently } A_j^{-1} \triangleq I + (I - A)A_{j-1}^{-1} \ .$$

Note that $\lim_{j\to\infty} A_j^{-1} \to A^{-1}$. Using the above recursive formulation, we now describe an unbiased estimator of $\nabla^{-2} f$ by first deriving an unbiased estimator $\tilde{\nabla}^{-2} f_j$ for $\nabla^{-2} f_j$.

Definition 5 (Estimator) *Given j independent and unbiased samples $\{X_1 \dots X_j\}$ of the Hessian $\nabla^2 f$, define $\{\tilde{\nabla}^{-2} f_0, \dots, \tilde{\nabla}^{-2} f_j\}$ recursively as follows:*

$$\tilde{\nabla}^{-2} f_0 = I \text{ and } \tilde{\nabla}^{-2} f_t = I + (I - X_t)\tilde{\nabla}^{-2} f_{t-1} \text{ for } t = 1, \dots, j.$$

It can be readily seen that $\mathbf{E}[\tilde{\nabla}^{-2} f_j] = \nabla^{-2} f_j$, and so $\mathbf{E}[\tilde{\nabla}^{-2} f_j] \to \nabla^{-2} f$ as $j \to \infty$, providing us with an unbiased estimator in the limit.

Algorithm 1 LiSSA: Linear (time) Stochastic Second-Order Algorithm

Input: T, $f(\mathbf{x}) = \frac{1}{m}\sum_{k=1}^{m} f_k(\mathbf{x})$, S_1, S_2, T_1
$\mathbf{x}_1 = FO(f(\mathbf{x}), T_1)$
for $t = 1$ to T **do**
  for $i = 1$ to S_1 **do**
    $X_{[i,0]} = \nabla f(\mathbf{x}_t)$
    for $j = 1$ to S_2 **do**
      Sample $\tilde{\nabla}^2 f_{[i,j]}(\mathbf{x}_t)$ uniformly from $\{\nabla^2 f_k(\mathbf{x}_t) \mid k \in [m]\}$
      $X_{[i,j]} = \nabla f(\mathbf{x}_t) + (I - \tilde{\nabla}^2 f_{[i,j]}(\mathbf{x}_t))X_{[i,j-1]}$
    end for
  end for
  $X_t = 1/S_1 \left(\sum_{i=1}^{S_1} X_{[i,S_2]}\right)$
  $\mathbf{x}_{t+1} = \mathbf{x}_t - X_t$
end for
return $\mathbf{x}_{T+1}$

Remark 6 *One can also define and analyze a simpler (non-recursive) estimator based on directly sampling terms from the series in Equation* (2)*. Theoretically, one can get similar guarantees for the estimator; however, empirically our proposed estimator exhibited better performance.*

3.2 Algorithm

Our algorithm runs in two phases: in the first phase it runs any efficient first-order method FO for T_1 steps to shrink the function value to the regime where we can then show linear convergence for our algorithm. It then takes approximate Newton steps based on the estimator from Definition 5 in place of the true Hessian inverse. We use two parameters, S_1 and S_2, to define the Newton step. S_1 represents the number of unbiased estimators of the Hessian inverse we average to get better concentration for our estimator, while S_2 represents the depth to which we capture the Taylor expansion. In the algorithm, we compute the average Newton step directly, which can be computed in linear time as observed earlier, instead of estimating the Hessian inverse.

3.3 Main Theorem

In this section we present our main theorem which analyzes the convergence properties of LiSSA. Define $FO(M, \hat{\kappa}_l)$ to be the total time required by a first-order algorithm to achieve accuracy $\frac{1}{4\hat{\kappa}_l M}$.

Theorem 7 *Consider Algorithm 1, and set the parameters as follows:* $T_1 = FO(M, \hat{\kappa}_l)$, $S_1 = O((\hat{\kappa}_l^{\max})^2 \ln(\frac{d}{\delta}))$, $S_2 \geq 2\hat{\kappa}_l \ln(4\hat{\kappa}_l)$*. The following guarantee holds for every* $t \geq T_1$ *with probability* $1 - \delta$*,*

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\| \leq \frac{\|\mathbf{x}_t - \mathbf{x}^*\|}{2}.$$

Moreover, we have that each step of the algorithm takes at most $\tilde{O}(md + (\hat{\kappa}_l^{\max})^2 \hat{\kappa}_l d^2)$ *time. Additionally if* f *is GLM, then each step of the algorithm can be run in time* $\tilde{O}(md + (\hat{\kappa}_l^{\max})^2 \hat{\kappa}_l d)$*.*

As an immediate corollary, we obtain the following:

Corollary 8 *For a GLM function* $f(\mathbf{x})$*, Algorithm 1 returns a point* $\mathbf{x}_t$ *such that with probability at least* $1 - \delta$*,*

$$f(\mathbf{x}_t) \leq \min_{\mathbf{x}^*} f(\mathbf{x}^*) + \varepsilon$$

in total time $\tilde{O}(m + (\hat{\kappa}_l^{\max})^2 \hat{\kappa}_l) d \ln\left(\frac{1}{\varepsilon}\right)$ *for* $\varepsilon \to 0$*.*

In the above theorems $\tilde{O}$ hides log factors of $\kappa, d, \frac{1}{\delta}$. We note that our bound $(\hat{\kappa}_l^{\max})^2$ on the variance is possibly pessimistic and can likely be improved to a more average quantity. However, since in our experiments setting the parameter $S_1 \sim O(1)$ suffices, we have not tried to optimize it further.

We now prove our main theorem about the convergence of LiSSA (Theorem 7).

Proof [Proof of Theorem 7]

Note that since we use a first-order algorithm to get a solution of accuracy at least $\frac{1}{4\hat{\kappa}_l M}$, we have that

$$\|\mathbf{x}_1 - \mathbf{x}^*\| \leq \frac{1}{4\hat{\kappa}_l M}. \tag{3}$$

As can be seen from Definition 5, a single step of our algorithm is equivalent to $\mathbf{x}_{t+1} = \mathbf{x}_t - \tilde{\nabla}^{-2} f(\mathbf{x}_t) \nabla f(\mathbf{x}_t)$, where $\tilde{\nabla}^{-2} f(\mathbf{x}_t)$ is the average of S_1 independent estimators $\tilde{\nabla}^{-2} f(\mathbf{x}_t)_{S_2}$. We now make use of the following lemma.

Lemma 9 *Let* $\mathbf{x}_{t+1} = \mathbf{x}_t - \tilde{\nabla}^{-2} f(\mathbf{x}_t) \nabla f(\mathbf{x}_t)$*, as per a single iteration of Algorithm 1, and suppose* S_1, S_2 *are as defined in Algorithm 1. Then if we choose* $S_2 \geq 2\hat{\kappa}_l \ln(2\hat{\kappa}_l)$ *we have the following guarantee on the convergence rate for every step with probability* $1 - \delta$*:*

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\| \leq \gamma \|\mathbf{x}_t - \mathbf{x}^*\| + M \|\nabla^{-2} f(\mathbf{x}_t)\| \|\mathbf{x}_t - \mathbf{x}^*\|^2$$

where $\gamma = 16\hat{\kappa}_l^{\max} \sqrt{\frac{\ln(d\delta^{-1})}{S_1}} + \frac{1}{16}$.

Substituting the values of S_1 and S_2, combining Equation (3) and Lemma 9, and noting that $\|\nabla^{-2} f(\mathbf{x}_t)\| \leq \hat{\kappa}_l$, we have that at the start of the Newton phase the following inequality holds:

$$\|\mathbf{x}_{t+1} - \mathbf{x}^*\| \leq \frac{\|\mathbf{x}_t - \mathbf{x}^*\|}{4} + M\hat{\kappa}_l^{\max} \|\mathbf{x}_t - \mathbf{x}^*\|^2 \leq \frac{\|\mathbf{x}_t - \mathbf{x}^*\|}{2} \ .$$

It can be shown via induction that the above property holds for all $t \geq T_1$, which concludes the proof. ■

We now provide a proof of Lemma 9.

Proof [Proof of Lemma 9] Define $\chi(\mathbf{x}_t) = \int_0^1 \nabla^2 f(\mathbf{x}^* + \tau(\mathbf{x}_t - \mathbf{x}^*))d\tau$. Note that $\nabla f(\mathbf{x}_t) = \chi(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{x}^*)$. Following an analysis similar to that of Nesterov (2013), we have that

$$\begin{aligned}\|\mathbf{x}_{t+1} - \mathbf{x}^*\| &= \|\mathbf{x}_t - \mathbf{x}^* - \tilde{\nabla}^{-2} f(\mathbf{x}_t)\nabla f(\mathbf{x}_t)\| \\ &= \|\mathbf{x}_t - \mathbf{x}^* - \tilde{\nabla}^{-2} f(\mathbf{x}_t)\chi(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{x}^*)\| \\ &\leq \|I - \tilde{\nabla}^{-2} f(\mathbf{x}_t)\chi(\mathbf{x}_t)\|\|\mathbf{x}_t - \mathbf{x}^*\|.\end{aligned}$$

Following from the previous equations, we have that

$$\frac{\|\mathbf{x}_{t+1} - \mathbf{x}^*\|}{\|\mathbf{x}_t - \mathbf{x}^*\|} \leq \|I - \tilde{\nabla}^{-2} f(\mathbf{x}_t)\chi(\mathbf{x}_t)\| = \| \underbrace{I - \nabla^{-2} f(\mathbf{x}_t)\chi(\mathbf{x}_t)}_{a} - \underbrace{\left(\tilde{\nabla}^{-2} f(\mathbf{x}_t) - \nabla^{-2} f(\mathbf{x}_t)\right)\chi(\mathbf{x}_t)}_{b} \|.$$

We now analyze the above two terms a and b separately:

$$\begin{aligned}\|a\| &= \|I - \nabla^{-2} f(\mathbf{x}_t)\chi(\mathbf{x}_t)\| \\ &\leq \|\nabla^{-2} f(\mathbf{x}_t) \int_0^1 \left(\nabla^2 f(\mathbf{x}_t) - \nabla^2 f(\mathbf{x}^* + \tau(\mathbf{x}_t - \mathbf{x}^*))d\tau\right) \| \\ &\leq M\|\nabla^{-2} f(\mathbf{x}_t)\|\|\mathbf{x}_t - \mathbf{x}^*\|.\end{aligned}$$

The second inequality follows from the Lipschitz bound on the Hessian. The second term can be bounded as follows:

$$\|b\| \leq \left(\| \left(\tilde{\nabla}^{-2} f(\mathbf{x}_t) - \nabla^{-2} f(\mathbf{x}_t)\right) \|\|\chi(\mathbf{x}_t)\|\right) \leq \gamma.$$

The previous claim follows from Lemma 10 which shows a concentration bound on the sampled estimator and by noting that due to our assumption on the function, we have that for all $\mathbf{x}$, $\|\nabla^2 f(\mathbf{x})\| \leq 1$, and hence $\|\chi(\mathbf{x})\| \leq 1$.

Putting the above two bounds together and using the triangle inequality, we have that

$$\frac{\|\mathbf{x}_{t+1} - \mathbf{x}^*\|}{\|\mathbf{x}_t - \mathbf{x}^*\|} \leq M\|\nabla^{-2} f(\mathbf{x}_t)\|\|\mathbf{x}_t - \mathbf{x}^*\| + \gamma$$

which concludes the proof. ■

Lemma 10 *Let $\tilde{\nabla}^{-2} f(\mathbf{x}_t)$ be the average of S_1 independent samples of $\tilde{\nabla}^{-2} f(\mathbf{x}_t)_{S_2}$, as defined in Definition 5 and used in the per-step update of Algorithm 1, and let $\nabla^2 f(\mathbf{x}_t)$ be the true Hessian. If we set $S_2 \geq 2\hat{\kappa}_l \ln(\hat{\kappa}_l S_1)$, then we have that*

$$\mathbf{Pr}\left(\|\tilde{\nabla}^{-2} f(\mathbf{x}_t) - \nabla^{-2} f(\mathbf{x}_t)\| > 16\hat{\kappa}_l^{\max}\sqrt{\frac{\ln(\frac{d}{\delta})}{S_1}} + 1/16\right) \leq \delta.$$

Proof [Proof of Lemma 10] First note the following statement which is a straightforward implication of our construction of the estimator:

$$\mathbf{E}[\tilde{\nabla}^{-2} f(\mathbf{x}_t)] = \mathbf{E}[\tilde{\nabla}^{-2} f(\mathbf{x}_t)_{S_2}] = \sum_{i=0}^{S_2} (I - \nabla^2 f(\mathbf{x}_t))^i.$$

We also know from Equation (2) that for matrices X such that $\|X\| \leq 1$ and $X \succ 0$,

$$X^{-1} = \sum_{i=0}^{\infty} (I - X)^i.$$

Since we have scaled the function such that $\|\nabla^2 f_k\| \leq 1$, it follows that

$$\nabla^{-2} f(\mathbf{x}_t) = \mathbf{E}\left[\tilde{\nabla}^{-2} f(\mathbf{x}_t)_{S_2}\right] + \sum_{i=S_2+1}^{\infty} (I - \nabla^2 f(\mathbf{x}_t))^i. \tag{4}$$

Also note that since $\nabla^2 f(\mathbf{x}_t) \succeq \frac{I}{\hat{\kappa}_l}$, it follows that $\|I - \nabla^2 f(\mathbf{x}_t)\| \leq 1 - \frac{1}{\hat{\kappa}_l}$. Observing the second term in the above equation,

$$\begin{aligned}\|\sum_{i=S_2}^{\infty} (I - \nabla^2 f(\mathbf{x}_t))^i\| &\leq \|(I - \nabla^2 f(\mathbf{x}_t))\|^{S_2} \left(\sum_{i=0}^{\infty} \|I - \nabla^2 f(\mathbf{x}_t)\|^i\right) \\ &\leq (1 - \frac{1}{\hat{\kappa}_l})^{S_2} \left(\sum_{i=0}^{\infty} (1 - \frac{1}{\hat{\kappa}_l})^i\right) \\ &\leq (1 - \frac{1}{\hat{\kappa}_l})^{S_2} \hat{\kappa}_l \\ &\leq \exp\left(-\frac{S_2}{\hat{\kappa}_l}\right) \hat{\kappa}_l.\end{aligned}$$

Since we have chosen $S_2 \geq 2\hat{\kappa}_l \ln(4\hat{\kappa}_l)$, we get that the above term is bounded by $\frac{1}{16}$. We will now show, using the matrix Bernstein inequality (Theorem 3), that the estimate $\tilde{\nabla}^{-2} f$ is concentrated around its expectation. To apply the inequality we first need to bound the spectral norm of each random variable. To that end we note that $\tilde{\nabla}^{-2} f_{S_2}$ has maximum spectral norm bounded by

$$\|\tilde{\nabla}^{-2} f_{S_2}\| \leq \sum_{i=0}^{S_2} (1 - 1/\hat{\kappa}_l^{\max})^i \leq \hat{\kappa}_l^{\max}.$$

We can now apply Theorem 3, which gives the following:

$$\mathbf{Pr}\left(\|\tilde{\nabla}^{-2} f - \mathbf{E}[\tilde{\nabla}^{-2} f]\| > \varepsilon\right) \leq 2d \exp\left(\frac{-\varepsilon^2 S_1}{64(\hat{\kappa}_l^{\max})^2}\right).$$

Setting $\varepsilon = 16\hat{\kappa}_l^{\max}\sqrt{\frac{ln(\frac{d}{\delta})}{S_1}}$ gives us that the probability above is bounded by δ. Now putting together the bounds and Equation (4) we get the required result. ■

3.4 Leveraging Sparsity

A key property of real-world data sets is that although the input is a high dimensional vector, the number of non-zero entries is typically very low. The following theorem shows

that LiSSA can be implemented in a way to leverage the underlying sparsity of the data. Our key observation is that for GLM functions, the rank one Hessian-vector product can be performed in $O(s)$ time where s is the sparsity of the input $\mathbf{x}_k$.

Theorem 11 *For GLM functions Algorithm 1 returns a point $\mathbf{x}_t$ such that with probability at least $1-\delta$*

$$f(\mathbf{x}_t) \leq \min_{\mathbf{x}^*} f(\mathbf{x}^*) + \varepsilon$$

in total time $\tilde{O}(ms + (\hat{\kappa}_l^{\max})^2 \hat{\kappa}_l s) \ln\left(\frac{1}{\varepsilon}\right)$ for $\varepsilon \to 0$.

We will prove the following theorem, from which Theorem 11 will immediately follow.

Theorem 12 *Consider Algorithm 1, let f be of the form described above, and let s be such that the number of non zero entries in $\mathbf{x}_i$ is bounded by s. Then each step of the algorithm can be implemented in time $O(ms + (\kappa_l^{\max})^2 \kappa_l s)$.*

Proof [Proof of Theorem 12] Proof by induction. Let $c_0 = 1$, $d_0 = 1$, $\mathbf{v}_0 = \mathbf{0}$, and consider the update rules $c_{j+1} = 1 + (1-\lambda)c_j$, $d_{j+1} = (1-\lambda)d_j$, and $\mathbf{v}_{j+1} = \mathbf{v}_j - \frac{1}{(1-\lambda)d_j}\tilde{\nabla}^2 f_{[i,j+1]}(\mathbf{x})(c_j \nabla f(\mathbf{x}) + d_j \mathbf{v}_j)$, where λ is the regularization parameter. For the base case, note that $X_{[i,0]} = c_0 \nabla f(\mathbf{x}) + d_0 \mathbf{v}_0 = \nabla f(\mathbf{x})$, as is the case in Algorithm 1. Furthermore, suppose $X_{[i,j]} = c_j \nabla f(\mathbf{x}) + d_j \mathbf{v}_j$. Then we see that

$$\begin{aligned}
X_{[i,j+1]} &= \nabla f(\mathbf{x}) + (I - \lambda I - \tilde{\nabla}^2 f_{[i,j+1]}(\mathbf{x})) X_{[i,j]} \\
&= \nabla f(\mathbf{x}) + ((1-\lambda)I - \tilde{\nabla}^2 f_{[i,j+1]}(\mathbf{x}))(c_j \nabla f(\mathbf{x}) + d_j \mathbf{v}_j) \\
&= (1 + (1-\lambda)c_j)\nabla f(\mathbf{x}) + (1-\lambda)(d_j \mathbf{v}_j) - \tilde{\nabla}^2 f_{[i,j+1]}(\mathbf{x})(c_j \nabla f(\mathbf{x}) + d_j \mathbf{v}_j) \\
&= c_{j+1} \nabla f(\mathbf{x}) + d_{j+1} \mathbf{v}_{j+1}.
\end{aligned}$$

Note that updating c_{j+1} and d_{j+1} takes constant time, and $\tilde{\nabla}^2 f_{[i,j+1]}(\mathbf{x})(c_j \nabla f(\mathbf{x}))$ and $\tilde{\nabla}^2 f_{[i,j+1]}(\mathbf{x})\mathbf{v}_j$ can each be calculated in $O(s)$ time. It can also be seen that each product gives an s-sparse vector, so computing $\mathbf{v}_{j+1}$ takes time $O(s)$. Since $\nabla f(\mathbf{x})$ can be calculated in $O(ms)$ time, and since $\mathbf{v}_0$ is 0-sparse which implies the number of non-zero entries of $\mathbf{v}_j$ is at most js, it follows that the total time to calculate X_t is $O(ms + (\kappa_l^{\max})^2 \kappa_l s)$. ■

4. LiSSA: Extensions

In this section we first describe a family of algorithms which generically couple first-order methods as sub-routines with second-order methods. In particular, we formally describe the algorithm LiSSA-Quad (Algorithm 2) and provide its runtime guarantee (Theorem 13). The key idea underlying this algorithm is that Newton's method essentially reduces a convex optimization problem to solving intermediate quadratic sub-problems given by the second-order Taylor approximation at a point, i.e., the sub-problem Q_t given by

$$Q_t(\mathbf{y}) = f(\mathbf{x}_{t-1}) + \nabla f(\mathbf{x}_{t-1})^T \mathbf{y} + \frac{\mathbf{y}^T \nabla^2 f(\mathbf{x}_{t-1}) \mathbf{y}}{2}$$

Algorithm 2 LiSSA-Quad

Input: T, $f(\mathbf{x}) = \sum_{k=1}^{m} f_k(\mathbf{x})$, ALG, ALG_{params}, T_1, ε
$\mathbf{x}_0 = \text{ALG}\ (f(\mathbf{x}), T_1)$
for $t = 1$ to T **do**
  $Q_t(\mathbf{y}) = \nabla f(\mathbf{x}_{t-1})^T \mathbf{y} + \frac{\mathbf{y}^T \nabla^2 f(\mathbf{x}_{t-1}) \mathbf{y}}{2}$
  $\mathbf{x}_t = A(Q_t, \varepsilon^2, A_{params})$
end for
return $\mathbf{x}_T$

where $\mathbf{y} \triangleq \mathbf{x} - \mathbf{x}_{t-1}$. The above ideas provide an alternative implementation of our estimator for $\nabla^{-2} f(\mathbf{x})$ used in LiSSA. Consider running gradient descent on the above quadratic Q_t, and let $\mathbf{y}_t^i$ be the ith step in this process. By definition we have that

$$\mathbf{y}_t^{i+1} = \mathbf{y}_t^i - \nabla Q_t(\mathbf{y}_t^i) = (I - \nabla^2 f(\mathbf{x}_{t-1}))\mathbf{y}_t^i - \nabla f(\mathbf{x}_{t-1}).$$

It can be seen that the above expression corresponds exactly to the steps taken in LiSSA (Algorithm 2, line 8), the difference being that we use a sample of the Hessian instead of the true Hessian. Therefore LiSSA can also be interpreted as doing a partial stochastic gradient descent on the quadratic Q_t. It is partial because we have a precise estimate of gradient of the function f and a stochastic estimate for the Hessian. We note that this is essential for the linear convergence guarantees we show for LiSSA.

The above interpretation indicates the possibility of using any first-order linearly convergent scheme for approximating the minimizer of the quadratic Q_t. In particular, consider any algorithm ALG that, given a convex quadratic function Q_t and an error value ε, produces a point $\mathbf{y}$ such that

$$\|\mathbf{y} - \mathbf{y}_t^*\| \leq \varepsilon \tag{5}$$

with probability at least $1 - \delta_{ALG}$, where $\mathbf{y}_t^* = \operatorname{argmin} Q_t$. Let the total time taken by the algorithm ALG to produce the point be $T_{ALG}(\varepsilon, \delta_{ALG})$. For our applications we require ALG to be linearly convergent, i.e. T_{ALG} is proportional to $\log(\frac{1}{\varepsilon})$ with probability at least $1 - \delta_{ALG}$.

Given such an algorithm ALG, LiSSA-Quad, as described in Algorithm 2, generically implements the above idea, modifying LiSSA by replacing the inner loop with the given algorithm ALG. The following is a meta-theorem about the convergence properties of LiSSA-Quad.

Theorem 13 *Given the function $f(\mathbf{x}) = \sum f_i(\mathbf{x})$ which is α-strongly convex, let $\mathbf{x}^*$ be the minimizer of the function and $\{\mathbf{x}_t\}$ be defined as in Algorithm 2. Suppose the algorithm ALG satisfies condition (5) with probability $1 - \delta_{ALG}$ under the appropriate setting of parameters ALG_{params}. Set the parameters in the algorithm as follows: $T_1 = T_{ALG}(1/4\alpha M)$, $T = \log\log(1/\varepsilon)$, $\delta_{ALG} = \delta/T$, where ε is the final error guarantee one wishes to achieve. Then we have that after T steps, with probability at least $1 - \delta$,*

$$\min_{t=\{1\ldots T\}} \|\mathbf{x}_t - \mathbf{x}^*\| \leq \varepsilon.$$

In particular, LiSSA-Quad(ALG) produces a point $\mathbf{x}$ *such that*

$$\|\mathbf{x} - \mathbf{x}^*\| \leq \varepsilon$$

in total time $O(T_{ALG}(\varepsilon, \delta_{ALG}) \log\log(1/\varepsilon))$ *with probability at least* $1 - \delta$ *for* $\varepsilon \to 0$.

Note that for GLM functions, the $\nabla Q_t(\mathbf{y})$ at any point can be computed in time linear in d. In particular, a full gradient of Q_t can be computed in time $O(md)$ and a stochastic gradient (corresponding to a stochastic estimate of the Hessian) in time $O(d)$. Therefore, a natural choice for the algorithm *ALG* in the above are first-order algorithms which are linearly convergent, for example SVRG, SDCA, and Acc-SDCA. Choosing a first-order algorithm FO gives us a family of algorithms LiSSA-Quad(FO), each with running time comparable to the running time of the underlying FO, up to logarithmic factors. The following corollary summarizes the typical running time guarantees for LiSSA-Quad(FO) when FO is Acc-SVRG.

Corollary 14 *Given a GLM function* $f(\mathbf{x})$*, if ALG is replaced by Acc-SVRG (Lin et al., 2015), then under a suitable setting of parameters, LiSSA-Quad produces a point* $\mathbf{x}$ *such that*

$$f(\mathbf{x}) - f(\mathbf{x}^*) \leq \varepsilon$$

with probability at least $1 - \delta$*, in total time* $\tilde{O}(m + \min\left\{\sqrt{\hat{\kappa}_l m}, \hat{\kappa}_l\right\}) d \log(1/\varepsilon) \log\log(1/\varepsilon)$.

Here the $\tilde{O}$ above hides logarithmic factors in κ, d, δ, but not in ε. Note that the above running times depend upon the condition number $\hat{\kappa}_l$ which as described in Section 2 can potentially provide better dependence compared to its global counterpart. In practice this difference could lead to faster running time for LiSSA-Quad as compared to the underlying first-order algorithm FO. We now provide a proof for Theorem 13.

Proof [Proof of Theorem 13] We run the algorithm A to achieve accuracy ε^2 on each of the intermediate quadratic functions Q_t, and we set $\delta_A = \delta/T$ which implies via a union bound that for all $t \leq T$,

$$\|\mathbf{x}_{t+1} - \mathbf{x}_t^*\| \leq \varepsilon^2 \tag{6}$$

with probability at least $1 - \delta$.

Assume that for all $t < T$, $\|\mathbf{x}_t - \mathbf{x}^*\| \geq \varepsilon$ (otherwise the theorem is trivially true). Using the analysis of Newton's method as before, we get that for all $t \leq T$,

$$\begin{aligned}
\|\mathbf{x}_{t+1} - \mathbf{x}^*\| &\leq \|\mathbf{x}_t^* - \mathbf{x}^*\| + \|\mathbf{x}_{t+1} - \mathbf{x}_t^*\| \\
&\leq \|\mathbf{x}_t - \nabla^{-2} f(\mathbf{x}_t) \nabla f(\mathbf{x}_t) - \mathbf{x}^*\| + \|\mathbf{x}_{t+1} - \mathbf{x}_t^*\| \\
&\leq \frac{M}{4\alpha} \|\mathbf{x}_t - \mathbf{x}^*\|^2 + \varepsilon^2 \\
&\leq \left(\frac{M}{4\alpha} + 1\right) \|\mathbf{x}_t - \mathbf{x}^*\|^2
\end{aligned}$$

where the second inequality follows from the analysis in the proof of Theorem 7 and Equation (6). We know that $\|\mathbf{x}_0 - \mathbf{x}_t\| \leq \sqrt{\frac{\alpha}{M}}$ from the initial run of the first-order algorithm *FO*. Applying the above inductively and using the value of T prescribed by the theorem statement, we get that $\|\mathbf{x}_T - \mathbf{x}^*\| \leq \varepsilon$. ■

5. Runtime Improvement through Fast Quadratic Solvers

The previous section establishes the reduction from general convex optimization to quadratic functions. In this section we show how we can leverage the fact that for quadratic functions the running time for accelerated first-order methods can be improved in the regime when $\kappa > m >> d$. In particular, we show the following theorem.

Theorem 15 *Given a vector* $\mathbf{b} \in \mathbb{R}^d$ *and a matrix* $A = \sum A_i$ *where each* A_i *is of the form* $A_i = \mathbf{v}_i \mathbf{v}_i^T + \lambda I$ *for some* $\mathbf{v}_i \in \mathbb{R}^d, \|\mathbf{v}_i\| \leq 1$ *and* $\lambda \geq 0$ *a fixed parameter, Algorithm 4 computes a vector* $\tilde{\mathbf{v}}$ *such that* $\|A^{-1}\mathbf{b} - \tilde{\mathbf{v}}\| \leq \varepsilon$ *with probability at least* $1 - \delta$ *in total time*

$$\tilde{O}\left(md \log\left(\frac{1}{\varepsilon}\right) + \left(d + \sqrt{\kappa_{sample}(A) d}\right) d \log^2\left(\frac{1}{\varepsilon}\right)\right).$$

$\tilde{O}()$ *contains factors logarithmic in* $m, d, \kappa_{sample}(A), \|b\|, \delta$.

$\kappa_{sample}(A)$ is the condition number of an $O(d \log(d))$ sized sample of A and is formally defined in Equation (11). We can now use Algorithm 4 to compute an approximate Newton step by setting $A = \nabla^2 f(x)$ and $\mathbf{b} = \nabla f(x)$. We therefore propose LiSSA-Sample to be a variant of LiSSA-Quad where Algorithm 4 is used as the subroutine ALG and any first-order algorithm can be used in the initial phase. The following theorem bounding the running time of LiSSA-Sample follows immediately from Theorem 13 and Theorem 15.

Theorem 16 *Given a GLM function* $f(\mathbf{x}) = \sum_i f_i(\mathbf{x})$*, let* $\mathbf{x}^* = \operatorname{argmin} f(\mathbf{x})$*. LiSSA-Sample produces a point* $\mathbf{x}$ *such that*

$$\|\mathbf{x} - \mathbf{x}^*\| \leq \varepsilon$$

with probability at least $1 - \delta$ *in total time*

$$\tilde{O}\left(\left(md \log\left(\frac{1}{\varepsilon}\right) + \left(d + \sqrt{\kappa_{sample}(f) d}\right) d \log^2\left(\frac{1}{\varepsilon}\right)\right) \log\log\left(\frac{1}{\varepsilon}\right)\right).$$

$\tilde{O}()$ *contains factors logarithmic in* $m, d, \kappa_{sample}(f), G, \delta$.

5.1 Fast Quadratic Solver - Outline

In this section we provide a short overview of Algorithm 4. To simplify the discussion, lets consider the case when we have to compute $A^{-1}\mathbf{b}$ for a $d \times d$ matrix A given as $A = \sum_{i=1}^m \mathbf{v}_i \mathbf{v}_i^T = VV^T$ where the ith column of V is $\mathbf{v}_i$. The computation can be recast as minimization of a convex quadratic function $Q(\mathbf{y}) = \frac{\mathbf{y}^T A \mathbf{y}}{2} + \mathbf{b}^T y$ and can be solved up to accuracy ε in total time $\left(m + \sqrt{\kappa(A) m}\right) d \log(1/\varepsilon)$ as can be seen from Theorem 4 Algorithm 4 improves upon the running time bound in the case when $m > d$. In the following we provide a high level outline of the procedure which is formally described as Algorithm 4.

- Given A we will compute a low complexity constant spectral approximation B of A. Specifically $B = \sum_{i=1}^{O(d\log(d))} \mathbf{u}_i\mathbf{u}_i^T$ and $B \preceq A \preceq 2B$. This is achieved by techniques developed in matrix sampling/sketching literature, especially those of Cohen et al. (2015). The procedure requires solving a constant number of $O(d\log(d))$ sized linear systems, which we do via Accelerated SVRG.
- We use B as a preconditioner and compute $BA^{-1}\mathbf{b}$ by minimizing the quadratic $\frac{\mathbf{y}^T AB^{-1}\mathbf{y}}{2} + \mathbf{b}^T\mathbf{y}$. Note that this quadratic is well conditioned and can be minimized using gradient descent. In order to compute the gradient of the quadratic which is given by $AB^{-1}\mathbf{y}$, we again use Accelerated SVRG to solve a linear system in B.
- Finally, we compute $A^{-1}\mathbf{b} = B^{-1}BA^{-1}\mathbf{b}$ using Accelerated SVRG to solve a linear system in B.

In the rest of the section we formally describe the procedure outlined and provide the necessary definitions. One key nuance we must take into account is the fact that based on our assumption we have included the regularization term into the component functions. Due to this, the Hessian does not necessarily look like a sum of rank one matrices. Of course, one can decompose the identity matrix that appears due to the regularizer as a sum of rank one matrices. However, note that the procedure above requires that each of the sub-samples must have good condition number too in order to solve linear systems on them with Accelerated SVRG. Therefore, the sub-samples generated must look like sub-samples formed from the Hessians of component functions. For this purpose we extend the procedure and the definition for leverage scores described by Cohen et al. (2015) to the case when the matrix is given as a sum of PSD matrices and not just rank one matrices. We reformulate and reprove the basic theorems proved by Cohen et al. (2015) in this context. To maintain computational efficiency of the procedure, we then make use of the fact that each of the PSD matrices actually is a rank one matrix plus the Identity matrix. We now provide the necessary preliminaries for the description of the algorithm and its analysis.

5.2 Preliminaries for Fast Quadratic Solver

For all the definitions and preliminaries below assume we are given a $d \times d$ PSD matrix $A \triangleq \sum_{i=1}^m A_i$ where A_i are also PSD matrices. Let $A \cdot B \triangleq Tr(B^T A)$ be the standard matrix dot product. Given two matrices A and B we say B is a λ-spectral approximation of A if $\frac{1}{\lambda}A \preceq B \preceq A$.

Definition 17 (Generalized Matrix Leverage Scores) *Define*

$$\tau_i(A) \triangleq A^+ \cdot A_i \tag{7}$$

$$\tau_i^B(A) \triangleq B^+ \cdot A_i. \tag{8}$$

Then we have the following facts:

Fact 18

$$\sum_{i=1}^n \tau_i(A) = Tr\left(\sum A^+ A_i\right) = Tr(A^+A) = rank(A) \leq d. \tag{9}$$

Fact 19 *If B is a λ-spectral approximation of A, then $\tau_i(A) \leq \tau_i^B(A) \leq \lambda\tau_i(A)$.*

Given A of the form $A = \sum A_i$, we define a sample of A of size r as the following. Consider a subset of indices of size r, $I = \{i_1 \dots i_r\} \subseteq [m]$. For every such sample I and given a weight vector $\mathbf{w} \in \mathbb{R}^r$, we can associate the following matrix:

$$Sample(\mathbf{w}, I) \triangleq \sum_{j\in I} w_j A_j. \tag{10}$$

When the sample is unweighted, i.e., $\mathbf{w} = \mathbf{1}$, we will simply denote the above as $Sample(I)$. We can now define $\kappa_{sample}(A, r)$ to be

$$\kappa_{sample}(A, r) \triangleq \max_{I:|I|\geq r} \kappa(Sample(r, I)). \tag{11}$$

This is the worst-case condition number of an unweighted sub-sample of size r of the matrix A. For our results we will be concerned with a $\Omega(d\log(d))$ sized sub-sample, i.e., the quantity $\kappa_{sample}(A, O(d\log(d)))$. We remind the reader that by definition κ_{sample} for Hessians of the functions we are concerned with is always larger than $1/\lambda$, where λ is the coefficient of the ℓ_2 regularizer.

The following lemma is a generalization of the leverage score sampling lemma (Lemma 4, Cohen et al. 2015). The proof is very similar to the original proof by Cohen et al. (2015) and is included in the Appendix for completeness.

Lemma 20 (Spectral Approximation via Leverage Score Sampling) *Given an error parameter $0 < \varepsilon < 1$, let $\mathbf{u}$ be a vector of leverage score overestimates, i.e., $\tau_i(A) \leq \mathbf{u}_i$, for all $i \in [m]$. Let α be a sampling rate parameter and let c be a fixed positive constant. For each matrix A_i, we define a sampling probability $p_i(\alpha) = \min\{1, \alpha\mathbf{u}_i c\log d\}$. Let I be a random sample of indices drawn from $[m]$ by sampling each index with probability $p_i(\alpha)$. Define the weight vector $\mathbf{w}(\alpha)$ to be the vector such that $\mathbf{w}(\alpha)_i = \frac{1}{p_i(\alpha)}$. By definition of weighted samples we have that*

$$Sample(\mathbf{w}(\alpha), I) = \sum_{i=1}^m \frac{1}{p_i(\alpha)} A_i \mathbf{1}_{x_i\sim p_i(\alpha)}(x_i = 1)$$

where x_i is a Bernoulli random variable with probability $p_i(\alpha)$.

If we set $\alpha = \varepsilon^{-2}$, $S = Sample(\mathbf{w}(\alpha), I)$ is formed by at most $\sum_i \min\{1, \alpha\mathbf{u}_i c\log(d)\} \leq \alpha c\log(d)\|\mathbf{u}\|_1$ entries in the above sum, and $\frac{1}{1+\varepsilon}S$ is a $\frac{1+\varepsilon}{1-\varepsilon}$ spectral approximation for A with probability at least $1 - d^{-c/3}$.

The following theorem is an analogue of the key theorem regarding uniform sampling (Theorem 1, Cohen et al. 2015). The proof is identical to the original proof and is also included in the appendix for completeness.

Theorem 21 (Uniform Sampling) *Given any $A = \sum_{i=1}^m A_i$ as defined above, let $S = \sum_{j=1}^r X_j$ be formed by uniformly sampling r matrices $X_1 \dots X_r \sim \{A_i\}$ without repetition. Define*

$$\tilde{\tau}_i^S(A) = \begin{cases} \tau_i^S(A) & \text{if } \exists\, j \;\; s.t. \;\; X_j = A_i \\ \tau_i^{S+A_i}(A) & \text{otherwise} \end{cases}.$$

Then $\tilde{\tau}_i^S(A) \geq \tau_i(A)$ *for all* i*, and*

$$\mathbb{E}\left[\sum_{i=1}^{n} \tilde{\tau}_i^S(A)\right] \leq \frac{md}{r}.$$

Unlike the case of rank one matrices as was done by Cohen et al. (2015), it is not immediately clear how to efficiently compute $\tilde{\tau}_i$ as we cannot directly apply the Sherman-Morrison formula. In our case though, since we have that each $A_i = \mathbf{v}_i\mathbf{v}_i^T + \lambda I$ we define slightly different estimates which are efficiently computable and prove that they work as well.

Theorem 22 *Suppose we are given any* $A = \sum_{i=1}^m A_i$ *where each* A_i *is of the form* $A_i = \mathbf{v}_i\mathbf{v}_i^T + \lambda I$*. Let* $S = \sum_{j=1}^m X_j$ *be formed by uniformly sampling* r *matrices* $X_1 \ldots X_r \sim \{A_i\}$ *without repetition. Define*

$$\hat{\tau}_i^S(A) = \begin{cases} \mathbf{v}_i^T S^+ \mathbf{v}_i^T + \frac{d}{r} & \text{if } \exists\, j \;\; s.t.\; X_j = A_i \\ \frac{1}{1+\frac{1}{\mathbf{v}_i^T(S+\lambda I)^+\mathbf{v}_i^T}} + \frac{d}{r} & \text{otherwise} \end{cases}.$$

Then $\hat{\tau}_i^S(A) \geq \tau_i(A)$ *for all* i*, and*

$$\mathbb{E}\left[\sum_{i=1}^{n} \hat{\tau}_i^S(A)\right] \leq O\left(\frac{md}{r}\right).$$

Proof [Proof of Theorem 22] We will prove that

$$\frac{d}{m} \geq \hat{\tau}_i^S(A) - \tilde{\tau}_i^S(A) \geq 0. \tag{12}$$

Consider the case when $\exists\, j \;\; s.t.\; X_j = A_i$. By definition of $\tilde{\tau}_i^S(A)$,

$$\begin{aligned}\tilde{\tau}_i^S(A) = \tau_i^S(A) = S^+ \cdot A_i = S^+ \cdot (\mathbf{v}_i\mathbf{v}_i^T + \lambda I) = \mathbf{v}_i^T S^+ \mathbf{v}_i + \lambda S^+ \cdot \lambda I \\ \leq \mathbf{v}_i^T S^+ \mathbf{v}_i^T + (r\lambda I)^+ \cdot I = \mathbf{v}_i^T S^+ \mathbf{v}_i^T + \frac{d}{r} = \hat{\tau}_i^S(A).\end{aligned}$$

The above follows by noting that $S \succeq r\lambda I$. It also follows from the definition that $\hat{\tau}^S(A)_i - \tilde{\tau}^S(A)_i \leq \frac{d}{r}$.

In the other case a similar inequality can be shown by noting via the Sherman-Morrison formula that

$$\mathbf{v}_i^T(S+\lambda I+\mathbf{v}_i\mathbf{v}_i^T)^+\mathbf{v}_i = \mathbf{v}_i^T\left((S+\lambda I)^+ - \frac{(S+\lambda I)^+\mathbf{v}_i\mathbf{v}_i^T(S+\lambda I)^+}{1+\mathbf{v}_i^T(S+\lambda I)^+\mathbf{v}_i}\right)\mathbf{v}_i = \frac{1}{1+\frac{1}{\mathbf{v}_i^T(S+\lambda I)^+\mathbf{v}_i^T}}.$$

This proves Equation (12). A direct application of Theorem 21 now finishes the proof. ■

Algorithm 3 REPEATED HALVING

Input: $A = \sum_{i=1}^m (\mathbf{v}_i\mathbf{v}_i^T + \lambda I)$
Output: B **an** $O(d\log(d))$ **size weighted sample of** A **and** $B \preceq A \preceq 2B$
Take a uniformly random unweighted sample of size $\frac{m}{2}$ of A to form A'
if A' has size $> O(d\log(d))$ **then**
  Recursively compute an 2-spectral approximation $\tilde{A}'$ of A'
end if
Compute estimates γ_i of generalized leverage scores $\{\hat{\tau}_i^{A'}(A)\}$ s.t. the following are satisfied

$$\gamma_i \geq \hat{\tau}_i^{A'}(A)$$
$$\sum \gamma_i \leq \sum 16\hat{\tau}_i^{A'}(A) + 1$$

Use these estimates to sample matrices from A to form B

5.3 Algorithms

In the following we formally state the two sub-procedures: Algorithm 4, which solves the required system, and Algorithm 3, which is the sampling routine for reducing the size of the system.

We prove the following theorem regarding the above algorithm REPEATED HALVING (Algorithm 3).

Theorem 23 *REPEATED HALVING (Algorithm 3) produces an* $\tilde{O}(d)$ *sized sample* B *such that* $\frac{B}{2} \preceq A \preceq 2B$ *and can be implemented in total time*

$$\tilde{O}\left(md + d^2 + \sqrt{\kappa_{sample}(O(d\log(d)))d}\right).$$

The $\tilde{O}$ in the above contains factors logarithmic in $m, d, \|A\|_F$. Note that the Frobenius norm of A is bounded by $d\|A\|$ which is bounded in our case by d.

Algorithm 4 Fast Quadratic Solver (FQS)

Input: $A = \sum_{i=1}^m (\mathbf{v}_i\mathbf{v}_i^T + \lambda I)$**,** $\mathbf{b}$**,** ε
Output : $\tilde{\mathbf{v}}$ **s.t.** $\|A^{-1}\mathbf{b} - \tilde{\mathbf{v}}\| \leq \varepsilon$
Compute B s.t. $2B \succeq A \succeq B$ using REPEATED HALVING(Algorithm 3)
$Q(\mathbf{y}) = \frac{\mathbf{y}^T AB^{-1}\mathbf{y}}{2} + \mathbf{b}^T\mathbf{y}$
Compute $\hat{\mathbf{y}}$ such that $\|\hat{\mathbf{y}} - \operatorname{argmin} Q(\mathbf{y})\| \leq \frac{\varepsilon}{4\|B^{-1}\|}$
Output $\tilde{\mathbf{v}}$ such that $\|B^{-1}\hat{\mathbf{y}} - \tilde{\mathbf{v}}\| \leq \varepsilon/2$

We first provide the proof of Theorem 15 using Theorem 23 and then provide the proof for Theorem 23. For the purpose of clarity of discourse we hide the terms that appear due to the probabilistic part of the lemmas. We can take a union bound to bound the total probability of failure, and those terms show up in the log factor.

Proof [Proof of Theorem 15] We will first prove correctness which follows from noting that $\operatorname{argmin}_{\mathbf{y}} Q(\mathbf{y}) = BA^{-1}b$. Therefore we have that

$$\varepsilon/2 \geq \|B^{-1}(\hat{\mathbf{y}} - \operatorname*{argmin}_{\mathbf{y}} Q(\mathbf{y}))\| = \|B^{-1}\hat{\mathbf{y}} - A^{-1}b\|.$$

A simple application of the triangle inequality proves that $\|A^{-1}b - \tilde{\mathbf{v}}\| \leq \varepsilon$.

Running Time Analysis: We now prove that the algorithm can be implemented efficiently. We first note the following two simple facts: since B is a 2-approximation of A, $\kappa(B) \leq 2\kappa(A)$; furthermore, the quadratic $Q(\mathbf{y})$ is 1-strongly convex and 2-smooth. Next, we consider the running time of implementing step 5. For this purpose we will perform gradient descent over the quadratic $Q(\mathbf{y})$. Note that the gradient of $Q(\mathbf{y})$ is given by

$$\nabla Q(\mathbf{y}) = AB^{-1}\mathbf{y} - \mathbf{b}.$$

Let $\mathbf{y}_0 = 0$, $\tilde{\varepsilon} = \left(\frac{\varepsilon}{4\|B^{-1}\|}\right)^2$ and G_Q be a bound on the gradient of the quadratic Q($\mathbf{y}$). Let $\mathbf{v}_t$ and $\mathbf{y}_{t+1}$ be such that

$$\|B^{-1}\mathbf{y}_{t-1} - \mathbf{v}_t\| \leq \frac{\tilde{\varepsilon}}{100\|A\|G_Q}$$

$$\mathbf{y}_{t+1} = \mathbf{y}_t - \eta\left(A\mathbf{v}_t - \mathbf{b}\right).$$

Define $h_t = Q(\mathbf{y}_t) - \min_{\mathbf{y}} Q(\mathbf{y})$. Using the standard analysis of gradient descent, we show that the following holds true for $t > 0$:

$$h_t \leq \max\{\tilde{\varepsilon}, (0.9)^t h_0\}.$$

This follows directly from the gradient descent analysis which we outline below. To make the analysis easier, we define a *true* gradient descent series:

$$\mathbf{z}_{t+1} = \mathbf{y}_t - \eta\nabla Q(\mathbf{y}_t).$$

Note that

$$\|\mathbf{z}_{t+1} - \mathbf{y}_{t+1}\| = \|A\left(\mathbf{v}_t - B^{-1}\mathbf{y}_t\right)\| \leq \frac{\tilde{\varepsilon}}{10G_Q}.$$

We now have that

$$\begin{aligned}
h_{t+1} - h_t &= Q(\mathbf{y}_{t+1}) - Q(\mathbf{y}_t) \\
&\leq \langle\nabla Q(\mathbf{y}_t), \mathbf{y}_{t+1} - \mathbf{y}_t\rangle + \frac{\beta}{2}\|\mathbf{y}_{t+1} - \mathbf{y}_t\|^2 \\
&= \langle\nabla Q(\mathbf{y}_t), \mathbf{z}_{t+1} - y_t\rangle + \langle\nabla Q(\mathbf{y}_t), \mathbf{y}_{t+1} - \mathbf{z}_{t+1}\rangle + \frac{\beta}{2}\|\mathbf{z}_{t+1} - \mathbf{y}_t + \mathbf{y}_{t+1} - \mathbf{z}_{t+1}\|^2 \\
&\leq -\eta\|\nabla Q(\mathbf{y}_t)\|^2 + \langle\nabla Q(\mathbf{y}_t), \mathbf{y}_{t+1} - \mathbf{z}_{t+1}\rangle + \beta\eta^2\|\nabla Q(\mathbf{y}_t)\|^2 + \beta\|\mathbf{y}_{t+1} - \mathbf{z}_{t+1}\|^2 \\
&\leq \frac{1}{\beta}\|\nabla Q(\mathbf{y}_t)\|^2 + \frac{2}{10\beta}\left((\|\nabla Q(\mathbf{y}_t)\| + 1)\tilde{\varepsilon}\right) \\
&\leq -\frac{\alpha}{\beta}h_t + \frac{\tilde{\varepsilon}}{50\beta}
\end{aligned}$$

where $\beta \geq 1$ and $\alpha \leq 2$ are the strong convexity and smoothness parameters of $Q(\mathbf{y})$. Therefore, we have that

$$h_{t+1} \leq 0.75h_t + 0.08\tilde{\varepsilon}.$$

Using the inductive assumption that $\|h_t\| \leq \max\{\tilde{\varepsilon}, (0.9)^t h_0\}$, it follows easily that

$$h_{t+1} \leq \max\{\tilde{\varepsilon}, (0.9)^{t+1} h_0\}.$$

Using the above inequality, it follows that for $t \geq O(\log(\frac{1}{\tilde{\varepsilon}}))$, we have that h_t is less than or equal to $\tilde{\varepsilon}$. By the 1-strong convexity of $Q(\mathbf{y})$, we have that if $h_t \leq \tilde{\varepsilon}$, then

$$\|\mathbf{y}_t - \operatorname*{argmin}_{\mathbf{y}} Q(\mathbf{y})\| \leq \sqrt{\tilde{\varepsilon}} \leq \frac{\varepsilon}{4\|B\|^{-1}}.$$

The running time of the above sub-procedure is bounded by the time to multiply a vector with A, which takes $O(md)$ time, and the time required to compute $\mathbf{v}_t$, which involves solving a linear system in B at each step. Finally, in step 6 we once again compute the solution of a linear system in B. Combining these we get that the total running time is

$$\tilde{O}(md + LIN(B, \hat{\varepsilon}))\log\left(\frac{1}{\tilde{\varepsilon}}\right)$$

where $\hat{\varepsilon} = \frac{\varepsilon}{400\|A\|\|B\|^{-1}G_Q} = \Omega\left(\frac{\varepsilon}{\kappa(A)G_Q}\right)$. Now we can bound $LIN(B, \varepsilon)$ by $\tilde{O}(d^2 + d\sqrt{\kappa(A)d})\log(1/\varepsilon)$ by using Acc-SVRG to solve the linear system and by noting that B is an $O(d\log(d))$ sized 2-approximation sample of A. The bound G_Q can be taken as a bound on the gradient of the quadratic at the start of the procedure, as the norm of gradient only shrinks along the gradient descent path. It is therefore enough to take $G_Q \leq \|b\|$, which finishes the proof. ■

Proof [Proof of Theorem 23]

The correctness of the algorithm is immediate from Theorems 22 and 20. The key challenge in the proof lies in proving that there is an efficient way to implement the algorithm, which we describe next.

Runtime Analysis: To analyze the running time of the algorithm we consider the running time of the computation required in step 7 of the algorithm. We will show that there exists an efficient algorithm to compute numbers γ_i such that

$$\gamma_i \geq \hat{\tau}_i^{A'}(A)$$

$$\sum\gamma_i \leq \sum 16\hat{\tau}_i^{A'}(A) + 8k\varepsilon\|A\|_F^2.$$

First note that by recursion, $\tilde{A}'$ is a 2-spectral approximation of A', and so we have that

$$\hat{\tau}_i^{A'}(A) \leq \hat{\tau}_i^{\tilde{A}'}(A) \leq 2\hat{\tau}_i^{A'}(A).$$

We also know that A' is an $O(d\log(d))$ sized weighted sub-sample of A', and so it is of the form

$$\tilde{A}' = \sum_{i=1}^{O(d\log(d))} \mathbf{v}_i\mathbf{v}_i^T + \lambda I = \sum_{i=1}^{O(d\log(d))} b_i b_i^T$$

where such a decomposition can be achieved easily via decomposing the identity into canonical vectors. Therefore, any such $\tilde{A}'$ can be written as BB^T, where the columns of B are b_i.

To compute $\hat{\tau}_i^{\tilde{A}'}(A)$, we need to compute the value of $\mathbf{v}_i^T(\tilde{A}')^+\mathbf{v}_i = \|B^T(\tilde{A}')^+\mathbf{v}_i\|_2^2$. Indeed, computing this directly for all vectors v_i may be inefficient, but one can compute a constant approximation using the following procedure outlined by Cohen et al. (2015).

To compute a constant approximation, we randomly sample a Gaussian matrix G consisting of k rows and compute instead the norms $\gamma'(i) = \|GB^T(\tilde{A}')^+\mathbf{v}_i\|_2^2$. By the Johnson-Lindenstrauss lemma, setting $k = O(\log(md))$, we have that with high probability,

$$1/2\|B^T(\tilde{A}')^+\mathbf{v}_i\|_2^2 \leq \gamma'(i) \leq 2\|B^T(\tilde{A}')^+\mathbf{v}_i\|_2^2.$$

Consider the following procedure for the computation of $\gamma'(i)$:

1. Compute the matrix $G' = GB^T$ first which takes $\tilde{O}(kd^2)$ time.
2. For each row G'_i of G' compute a vector G''_i such that $\|G''_i - G'_i\tilde{A}'^+\|^2 \leq \varepsilon$. This takes a total time of $kLIN(\tilde{A}', \varepsilon)$.
3. For each v_i compute $\gamma''_i \triangleq \sum_{j=1}^k < G''_j, \mathbf{v}_i >^2$. This takes a total time of $O(kmd)$.

Here, $LIN(S, \varepsilon)$ is the running time of solving a linear system in S with error ε. Therefore, substituting $k = O(\log(md))$, the total running time of the above algorithm is

$$O(md + d^2 + LIN(\tilde{A}', \varepsilon)).$$

It is now easy from the definitions of γ''_i and γ'_i to see that

$$\gamma''_i \in \left[\gamma'_i - k\varepsilon\|\mathbf{v}_i\|^2, \gamma'_i + k\varepsilon\|\mathbf{v}_i\|^2\right].$$

Setting $\gamma_i = 4(\gamma''_i + k\varepsilon\|\mathbf{v}_i\|^2)$, it follows from the definitions that

$$\gamma_i \geq \hat{\tau}_i^{A'}(A)$$

$$\sum \gamma_i \leq \sum 16\hat{\tau}_i^{A'}(A) + 8k\varepsilon\|A\|_F^2.$$

Now setting $\varepsilon = \frac{1}{8k\|V\|_F^2}$ satisfies the required inequality for γ_i. This implies that when sampling from γ_i, we will have a 2-approximation, and the number of matrices will be bounded by $O(d\log(d))$.

To solve the linear systems required above we use Accelerated SVRG (Theorem 4) which ensures that the running time of $LIN(S, \varepsilon) \leq \tilde{O}(d^2 + d\sqrt{\kappa(S)d})\log(\frac{1}{\varepsilon})$. This follows by noting that in our case S is a sum of $O(d\log(d))$ matrices. To bound the condition number of S, note that it is a 2-approximation of some unweighted sample of the matrix A of size greater than $O(d\log(d))$. Therefore, we have that $\kappa(S) \leq \kappa_{sample}(A)$.

Putting the above arguments together we get that the total running time of the procedure is bounded by

$$\tilde{O}(md + (d^2 + d\sqrt{\kappa_{sample(A)}d})),$$

which proves the theorem. ■

6. Condition Number Independent Algorithms

In this section we state our main result regarding self-concordant functions and provide the proof. The following is the definition of a self-concordant function:

Definition 24 (Self-Concordant Functions) *Let $\mathcal{K} \subseteq \mathbb{R}^n$ be a non-empty open convex set, and let $f : \mathcal{K} \mapsto \mathbb{R}$ be a C^3 convex function. Then, f is said to be self-concordant if*

$$|\nabla^3 f(\mathbf{x})[\mathbf{h}, \mathbf{h}, \mathbf{h}]| \leq 2(\mathbf{h}^\top \nabla^2 f(\mathbf{x})\mathbf{h})^{3/2},$$

where we have

$$\nabla^k f(\mathbf{x})[\mathbf{h}_1, \ldots, \mathbf{h}_k] \triangleq \frac{\partial^k}{\partial t_1 \ldots \partial t_k}|_{t_1=\cdots=t_k} f(\mathbf{x} + t_1\mathbf{h}_1 + \cdots + t_k\mathbf{h}_k).$$

Our main theorem regarding self-concordant functions is as follows:

Theorem 25 *Let $0 < r < 1$, let $\gamma \geq 1$, and let ν be a constant depending on γ, r. Set $\eta = 10(1-r)^2$, $S_1 = c_r = \frac{50}{(1-r)^4}$, where c_r is a constant depending on r, and $T = \frac{f(\mathbf{x}_1) - f(\mathbf{x}^*)}{\nu}$. Then, after $t > T$ steps, the following linear convergence guarantee holds between two full gradient steps for Algorithm 5:*

$$\mathbf{E}\left[f(w_s) - f(w^*)\right] \leq \frac{1}{2}\mathbf{E}\left[f(w_{s-1}) - f(w^*)\right].$$

Moreover, the time complexity of each full gradient step is $O(md + c_r d^2)$, where c_r is a constant depending on r (independent of the condition number).

To describe the algorithm we first present the requisite definitions regarding self-concordant functions.

6.1 Self-Concordant Functions Preliminaries

An excellent reference for this material is the lecture notes on this subject by Nemirovski (2004).

A key object in the analysis of self-concordant functions is the notion of a Dikin ellipsoid, which is the unit ball around a point $\mathbf{x}$ in the norm given by the Hessian $\|\cdot\|_{\nabla^2 f(\mathbf{x})}$ at the point. We will refer to this norm as the *local norm* around a point and denote it as $\|\cdot\|_{\mathbf{x}}$. Formally, we have:

Definition 26 (Dikin ellipsoid) *The Dikin ellipsoid of radius r centered at a point $\mathbf{x}$ is defined as*

$$W_r(\mathbf{x}) \triangleq \{\mathbf{y} \mid \|\mathbf{y} - \mathbf{x}\|_{\nabla^2 f(\mathbf{x})} \leq r\}.$$

One of the key properties of self-concordant functions is that inside the Dikin ellipsoid, the function is well conditioned with respect to the local norm at the center, and furthermore, the function is smooth. The following lemmas makes these notions formal, and the proofs of these lemmas can be found in the lecture notes of Nemirovski (2004).

Lemma 27 (Nemirovski 2004) *For all* $\mathbf{h}$ *such that* $\|\mathbf{h}\|_{\mathbf{x}} < 1$ *we have that*

$$(1 - \|\mathbf{h}\|_{\mathbf{x}})^2 \nabla^2 f(\mathbf{x}) \preceq \nabla^2 f(\mathbf{x} + \mathbf{h}) \preceq \frac{1}{(1 - \|\mathbf{h}\|_{\mathbf{x}})^2} \nabla^2 f(\mathbf{x}).$$

Lemma 28 (Nemirovski 2004) *For all* $\mathbf{h}$ *such that* $\|\mathbf{h}\|_{\mathbf{x}} < 1$ *we have that*

$$f(\mathbf{x} + \mathbf{h}) \leq f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{h} \rangle + \rho\left(\|\mathbf{h}\|_{\mathbf{x}}\right)$$

where $\rho(x) \triangleq -\ln(1 - x) - x$.

Another key quantity which is used both as a potential function as well as a dampening for the step size in the analysis of Newton's method in general is the Newton decrement which is defined as $\lambda(\mathbf{x}) \triangleq \|\nabla f(\mathbf{x})\|_{\mathbf{x}}^* = \sqrt{\nabla f(\mathbf{x})^\top \nabla^{-2} f(\mathbf{x}) \nabla f(\mathbf{x})}$. The following lemma quantifies how $\lambda(\mathbf{x})$ behaves as a potential by showing that once it drops below 1, it ensures that the minimum of the function lies in the current Dikin ellipsoid. This is the property which we use crucially in our analysis.

Lemma 29 (Nemirovski 2004) *If* $\lambda(\mathbf{x}) < 1$ *then*

$$\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}} \leq \frac{\lambda(\mathbf{x})}{1 - \lambda(\mathbf{x})}.$$

6.2 Condition Number Independent Algorithms

In this section we describe an efficient linearly convergent method (Algorithm 5) for optimization of self-concordant functions for which the running time is independent of the condition number. We have not tried to optimize the complexity of the algorithms in terms of d as our main focus is to make it condition number independent.

The key idea here is the ellipsoidal view of Newton's method, whereby we show that after making a constant number of full Newton steps, one can identify an ellipsoid and a norm such that the function is well conditioned with respect to the norm in the ellipsoid. This is depicted in Figure 1.

At this point one can run any desired first-order algorithm. In particular, we choose SVRG and prove its fast convergence. Algorithm 6 (described in the appendix) states the modified SVRG routine for general norms used in Algorithm 5.

We now state and prove the following theorem regarding the convergence of Algorithm 5.

Proof It follows from Lemma 31 that at $t = \frac{f(\mathbf{x}_1) - f(\mathbf{x}^*)}{\nu}$, the minimizer is contained in the Dikin ellipsoid of radius r, where ν is a constant depending on γ, r. This fact, coupled with Lemma 30, shows that the function satisfies the following property with respect to $W_{\mathbf{x}_t}$:

$$\forall \mathbf{x} \in W_{\mathbf{x}_t} \;\; (1 - r)^2 \nabla^2 f(\mathbf{x}_T) \preceq \nabla^2 f(\mathbf{x}) \preceq (1 - r)^{-2} \nabla^2 f(\mathbf{x}_T).$$

Using the above fact along with Lemma 32, and substituting for the parameters, concludes the proof. ■

We describe the Algorithm N-SVRG (Algorithm 6 in the appendix). Since the algorithm and the following lemmas are minor variations of their original versions, we include the proofs of the following lemmas in the appendix for completeness.

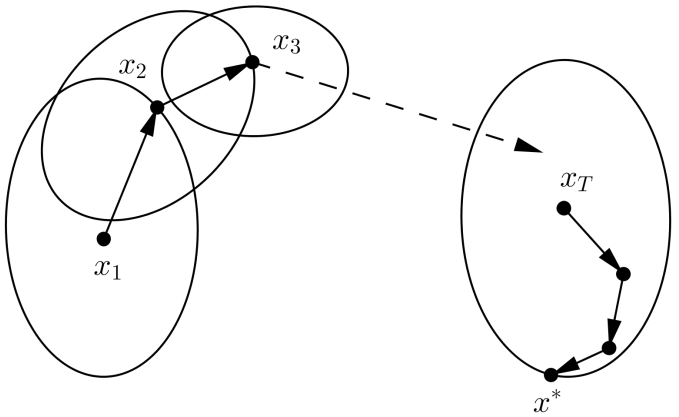


Figure 1: Visualization of the steps taken in Algorithm 5.

Lemma 30 *Let* f *be a self-concordant function over* $\mathcal{K}$*, let* $0 < r < 1$*, and consider* $\mathbf{x} \in \mathcal{K}$*. Let* $W_r(\mathbf{x})$ *be the Dikin ellipsoid of radius* r*, and let* $\alpha = (1 - r)^2$ *and* $\beta = (1 - r)^{-2}$*. Then, for all* $\mathbf{h}$ *s.t.* $\mathbf{x} + \mathbf{h} \in W_r(\mathbf{x})$*,*

$$\alpha \nabla^2 f(\mathbf{x}) \preceq \nabla^2 f(\mathbf{x} + \mathbf{h}) \preceq \beta \nabla^2 f(\mathbf{x}).$$

Lemma 31 *Let* f *be a self-concordant function over* $\mathcal{K}$*, let* $0 < r < 1$*, let* $\gamma \geq 1$*, and consider following the damped Newton step as described in Algorithm 5 with initial point* $\mathbf{x}_1$*. Then, the number of steps* t *of the algorithm before the minimizer of* f *is contained in the Dikin ellipsoid of radius* r *of the current iterate, i.e.* $\mathbf{x}^* \in W_r(\mathbf{x}_t)$*, is at most* $t = \frac{f(\mathbf{x}_1) - f(\mathbf{x}^*)}{\nu}$*, where* ν *is a constant depending on* γ, r*.*

Algorithm 5 ellipsoidCover

Input : Self-concordant $f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x})$, T, $r \in \{0, 1\}$, initial point $\mathbf{x}_1 \in \mathcal{K}$, $\gamma > 1$, S_1, η, T
Initialize: $\mathbf{x}_{\text{curr}} = \mathbf{x}_1$
while $\lambda(\mathbf{x}_{\text{curr}}) > \frac{r}{1+r}$ **do**
  $step = \gamma(1 + \lambda(\mathbf{x}_{\text{curr}}))$
  $\mathbf{x}_{\text{curr}} = \mathbf{x}_{\text{curr}} - \frac{1}{step} \nabla^{-2} f(\mathbf{x}_{\text{curr}}) \nabla f(\mathbf{x}_{\text{curr}})$
end while
$\mathbf{x}_{T+1} = \mathbf{N} - \mathbf{SVRG}(W_r(\mathbf{x}), f(\mathbf{x}), \nabla^2 f(\mathbf{x}_{\text{curr}}), S_1, \eta, T)$
return $\mathbf{x}_{T+1}$

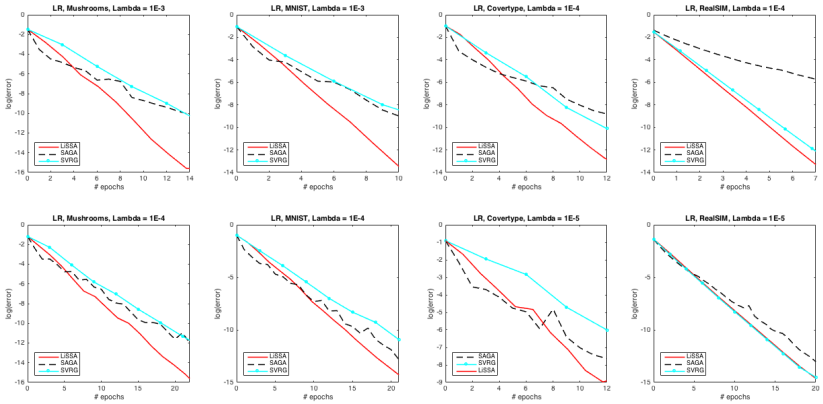


Figure 2: Performance of LiSSA as compared to a variety of related optimization methods for different data sets and choices of regularization parameter λ. $S_1 = 1$, $S_2 \sim \kappa \ln(\kappa)$.

Lemma 32 *Let f be a convex function. Suppose there exists a convex set $\mathcal{K}$ and a positive semidefinite matrix A such that for all $\mathbf{x} \in \mathcal{K}$, $\alpha A \preceq \nabla^2 f(\mathbf{x}) \preceq \beta A$. Then the following holds between two full gradient steps of Algorithm 6:*

$$\mathbf{E}\left[f(\mathbf{x}_s) - f(\mathbf{x}^*)\right] \leq \left(\frac{1}{\alpha\eta(1-2\eta\beta)n} + \frac{2\eta\beta}{(1-2\eta\beta)}\right)\mathbf{E}\left[f(\mathbf{x}_{s-1}) - f(\mathbf{x}^*)\right].$$

7. Experiments

In this section we present experimental evaluation for our theoretical results.[7] We perform the experiments for a classification task over two labels using the logistic regression (LR) objective function with the ℓ_2 regularizer. For all of the classification tasks we choose two values of λ:$\frac{1}{m}$ and $\frac{10}{m}$, where m is the number of training examples. We perform the above classification tasks over four data sets: MNIST, CoverType, Mushrooms, and RealSIM. Figure 2 displays the log-error achieved by LiSSA as compared to two standard first-order algorithms, SVRG and SAGA (Johnson and Zhang, 2013; Defazio et al., 2014), in terms of the number of passes over the data. Figure 3 presents the performance of LiSSA as compared to NewSamp (Erdogdu and Montanari, 2015) and standard Newton's method with respect to both time and iterations.

7. Our code for LiSSA can be found here: `https://github.com/brianbullins/lissa_code`.

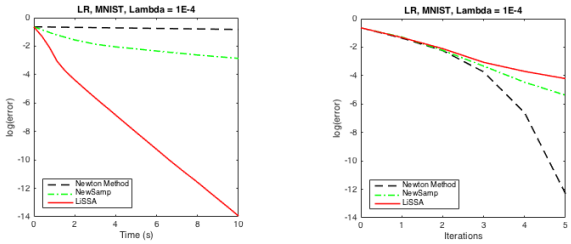


Figure 3: Convergence of LiSSA over time/iterations for logistic regression with MNIST, as compared to NewSamp and Newton's method.

7.1 Experiment Details

In this section we describe our experiments and choice of parameters in detail. Table 2 provides details regarding the data sets chosen for the experiments. To make sure our functions are scaled such that the norm of the Hessian is bounded, we scale the above data set points to unit norm.

Table 2: A description of data sets used in the experiments.

Data set	m	d	Reference
MNIST4-9	11791	784	LeCun and Cortes (1998)
Mushrooms	8124	112	Lichman (2013)
CoverType	100000	54	Blackard and Dean (1999)
RealSIM	72309	20958	McCallum (1997)

7.2 Comparison with Standard Algorithms

In Figures 2 and 4 we present comparisons between the efficiency of our algorithm with different standard and popular algorithms. In both cases we plot $\log(CurrentValue - OptimumValue)$. We obtained the optimum value for each case by running our algorithm for a long enough time until it converged to the point of machine precision.

Epoch Comparison: In Figure 2, we compare LiSSA with SVRG and SAGA in terms of the accuracy achieved versus the number of passes over the data. To compute the number of passes in SVRG and SAGA, we make sure that the inner stochastic gradient iteration in both the algorithms counts as exactly one pass. This is done because although it accesses gradients at two different points, one of them can be stored from before in both cases. The outer full gradient in SVRG counts as one complete pass over the data. We set the number of inner iterations of SVRG to $2m$ for the case when $\lambda = 1/m$, and we parameter tune the number of inner iterations when $\lambda = 10/m$. The stepsizes for all of the algorithms are parameter tuned by an exhaustive search over the parameters.

Time Comparison: For the comparison with respect to time (Figure 4), we consider the following algorithms: AdaGrad (Duchi et al., 2011), BFGS (Broyden, 1970; Fletcher, 1970;

Goldfarb, 1970; Shanno, 1970), gradient descent, SGD, SVRG (Johnson and Zhang, 2013) and SAGA (Defazio et al., 2014). The $\log(Error)$ is plotted as a function of the time elapsed from the start of the run for each algorithm. We next describe our choice of parameters for the algorithms. For AdaGrad we used the faster diagonal scaling version proposed by Duchi et al. (2011). We implemented the basic version of BFGS with backtracking line search. In each experiment for gradient descent, we find a reasonable step size using parameter tuning. For stochastic gradient descent, we use the variable step size $\eta_t = \gamma/\sqrt{t}$ which is usually the prescribed step size, and we hand tune the parameter γ. The parameters for SVRG and SAGA were chosen in the same way as before.

Choice of Parameters for LiSSA: To pick the parameters for our algorithm, we observe that it exhibits smooth behavior even in the case of $S_1 = 1$, so this is used for the experiments. However, we observe that increasing S_2 has a positive effect on the convergence of the algorithm up to a certain point, as a higher S_2 leads to a larger per-iteration cost. This behavior is consistent with the theoretical analysis. We summarize the comparison between the per-iteration convergence and the value of S_2 in Figure 5. As the theory predicts S_2 to be of the order $\kappa \ln(\kappa)$, for our experiments we determine an estimate for κ and set S_2 to around $\kappa \ln(\kappa)$. This value is typically equal to m in our experiments. We observe that setting S_2 in this way resulted in the experimental results displayed in Figure 2.

Comparison with Second-Order Methods: Here we present details about the comparison between LiSSA, NewSamp (Erdogdu and Montanari, 2015), and standard Newton's method, as displayed in Figure 3. We perform this experiment on the MNIST data set and show the convergence properties of all three algorithms over time as well as over iterations. We could not replicate the results of NewSamp on all of our data sets as it sometimes seems to diverge in our experiments. For logistic regression on the MNIST data set, we could get it to converge by setting the value of S_1 to be slightly higher. We observe as is predicted by the theory that when compared in terms of the number of iterations, NewSamp and LiSSA perform similarly, while Newton's method performs the best as it attains a quadratic convergence rate. This can be seen in Figure 3. However, when we consider the performance in terms of time for these algorithms, we see that LiSSA has a significant advantage.

Comparison with Accelerated First-Order Methods: Here we present experimental results comparing LiSSA with a popular accelerated first-order method, APCG (Lin et al., 2014), as seen in Figure 6. We ran the experiment on the RealSIM data set with three settings of $\lambda = 10^{-5},\ 10^{-6},\ 10^{-7}$, to account for the high condition number setting. We observe a trend that can be expected from the runtime guarantees of the algorithms. When λ is not too low, LiSSA performs better than APCG, but as λ gets very low we see that APCG performs significantly better than LiSSA. This is not surprising when considering that the running time of APCG grows proportional to $\sqrt{\kappa m}$, whereas for LiSSA the running time can at best be proportional to κ. We note that for accelerated first-order methods to be useful, one needs the condition number to be quite large which is not often the case for applications. Nevertheless, we believe that an algorithm with running time guarantees similar to LiSSA-Sample can get significant gains in these settings, and we leave this exploration as future work.

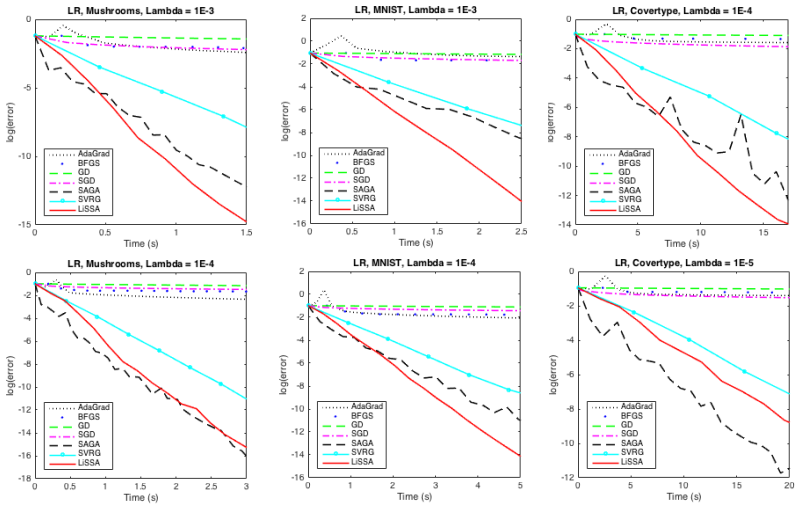


Figure 4: Performance of LiSSA as compared to a variety of related optimization methods for different data sets and choices of regularization parameter λ. $S_1 = 1$, $S_2 \sim \kappa \ln(\kappa)$.

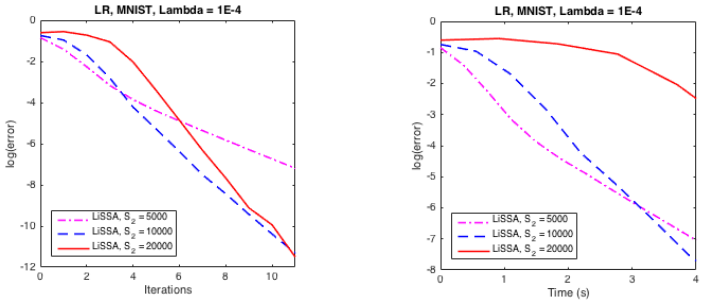


Figure 5: Differing convergence rates for LiSSA based on different choices of the S_2 parameter.

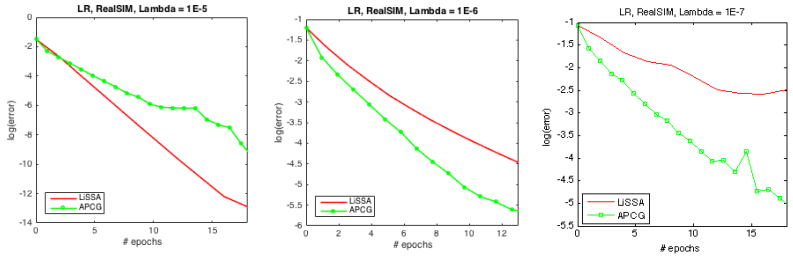


Figure 6: Comparison with accelerated first-order methods in ill-conditioned settings.

Acknowledgments

The authors would like to thank Zeyuan Allen-Zhu, Dan Garber, Haipeng Luo and David McAllester for several illuminating discussions and suggestions. We would especially like to thank Aaron Sidford and Yin Tat Lee for discussions regarding the matrix sampling approach.

Appendix A. Remaining Proofs

Here we provide proofs for the remaining lemmas.

A.1 Proof of Lemma 20

Proof The proof is almost identical to the proof of Lemma 4 by Cohen et al. (2015). As in there we will use the following lemma on matrix concentration, which appears as Lemma 11 in the work of Cohen et al. (2015).

Lemma 33 (Cohen et al. 2015) *Let $Y_1 \dots Y_k$ be independent random positive semidefinite matrices of size $d \times d$. Let $Y = \sum Y_i$ and let $Z = E[Y]$. If $Y_i \preceq R \cdot Z$, then*

$$P\left[\sum Y_i \preceq (1-\varepsilon)Z\right] \leq de^{\frac{\varepsilon^2}{2R}}$$

$$P\left[\sum Y_i \succeq (1+\varepsilon)Z\right] \leq de^{\frac{\varepsilon^2}{3R}}.$$

For every matrix A_i choose $Y_i = \frac{A_i}{p_i}$ with probability p_i and 0 otherwise. Therefore we need to bound $\sum Y_i$. Note that $E[Y_i] = A$. We will now show that

$$\forall i \;\; Y_i \preceq \frac{1}{c \log d \varepsilon^{-2}} A$$

which will finish the proof by a direct application of Lemma 33. First we will show that

$$\frac{A_i}{\tau_i(A)} \preceq A.$$

We need to show that $\forall \mathbf{x}\ \mathbf{x}^T A_i \mathbf{x} \leq \tau_i(A) \mathbf{x}^T A \mathbf{x}$. By noting that the A_i are PSD we have that if $\mathbf{x}^T A \mathbf{x} = 0$ then $\forall i, \mathbf{x}^T A_i \mathbf{x} = 0$. We can now consider $\mathbf{x} = A^{+/2}\mathbf{y}$. Therefore, we need to show that

$$\mathbf{y} A^{+/2} A_i A^{+/2} \mathbf{y} \leq \tau_i(A) \|y\|^2.$$

This is true because the maximum eigenvalue of $A^{+/2} A_i A^{+/2}$ is bounded by its $Trace(A^{+/2} A_i A^{+/2})$ (due to positive semidefiniteness), which is equal to $\tau_i(A)$ by definition. Therefore when $p_i < 1$, i.e., $\alpha c \mathbf{u}_i \log(d) < 1$, the facts $\tau_i(A) \leq \mathbf{u}_i$ immediately provide

$$Y_i \preceq \frac{1}{c \log d \varepsilon^{-2}} A.$$

When $p_i = 1$ the above does not hold, but we can essentially replace $Y_i = A_i$ with $c \log d \varepsilon^{-2}$ variables each equal to $\frac{A_i}{c \log d \varepsilon^{-2}}$, each being sampled with probability 1. This does not change $E[\sum Y_i]$ but proves concentration. Now a direct application of Lemma 33 finishes the proof. Also note that a standard Chernoff bound proves the required bound on the sample size. ■

A.2 Proof of Lemma 21

Proof The proof is almost identical to the proof of Theorem 1 by Cohen et al. (2015). We include the proof here for completeness. Define S_i to be S if $\exists j$ such that $X_j = A_i$, and $S + A_i$ otherwise. Now since $S^i \preceq A$, we have that $\tilde{\tau}_i^S(A) \geq \tau_i(A)$. To bound the expected sum we break it down into two parts:

$$\sum_i \tilde{\tau}_i^S(A) = \sum_{i \in S} \tilde{\tau}_i^S(A) + \sum_{i \notin S} \tilde{\tau}_i^S(A).$$

The first part is a sum of the leverage scores of S, and so it is bounded by d. To bound the second term, consider a random process that first selects S, then selects a random $i \notin S$ and returns $\tilde{\tau}_i^S(A)$. There are always exactly $m - r$ such i, so the value returned by this random process is, in expectation, exactly equal to $\frac{1}{m-r} E[\sum_{i \notin S} \tilde{\tau}_i^S(A)]$. This random process is also equivalent to randomly selecting a set S of size $r + 1$, then randomly choosing an $i \in S'$ and returning its leverage score. In expectation it is therefore equal to the average leverage score in S. S' has size $r + 1$, and its leverage scores sum to its rank, so we can bound its average leverage score by $\frac{d}{r+1}$. Overall, we have that

$$E[\sum_i \tilde{\tau}_i^S(A)] = d + \frac{d(m-r)}{r+1} \leq O\left(\frac{md}{r}\right).$$

■

A.3 Proof of Ellipsoidal Cover Lemma

Proof [Proof of Lemma 31] Let $\lambda(\mathbf{x})$ be the Newton decrement at $\mathbf{x}$. By Lemma 29, we know that if $\lambda(\mathbf{x}) \leq \frac{r}{1+r} < 1$, then

$$||\mathbf{x} - \mathbf{x}^*||_{\mathbf{x}} \leq \frac{\lambda(\mathbf{x})}{1 - \lambda(\mathbf{x})} = r.$$

Consider a single iteration of the algorithm at time t. If $\lambda(\mathbf{x}_t) \leq \frac{r}{1+r}$, then we may conclude that $\mathbf{x}^* \in W_r(\mathbf{x}_t)$. Therefore, it is only when $\lambda(\mathbf{x}_t) > \frac{r}{1+r}$ that $\mathbf{x}^*$ may not be contained within $W_r(\mathbf{x}_t)$. Since our update is of the form

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{1}{\gamma(1 + \lambda(\mathbf{x}_t))} \nabla^{-2} f(\mathbf{x}_t) \nabla f(\mathbf{x}_t),$$

we have the following:

$$\begin{aligned}
f(\mathbf{x}_{t+1}) &\leq f(\mathbf{x}_t) - \frac{1}{\gamma(1 + \lambda(\mathbf{x}_t))} \nabla f(\mathbf{x}_t)^\top ([\nabla^2 f(\mathbf{x}_t)]^{-1} \nabla f(\mathbf{x}_t)) + \rho\left(\frac{\lambda(\mathbf{x}_t)}{\gamma(1 + \lambda(\mathbf{x}_t))}\right) \\
&= f(\mathbf{x}_t) - \frac{(\lambda(\mathbf{x}_t))^2}{\gamma(1 + \lambda(\mathbf{x}_t))} + \rho\left(\frac{\lambda(\mathbf{x}_t)}{\gamma(1 + \lambda(\mathbf{x}_t))}\right) \\
&= f(\mathbf{x}_t) - \frac{(\lambda(\mathbf{x}_t))^2}{\gamma(1 + \lambda(\mathbf{x}_t))} - \ln\left(1 - \frac{\lambda(\mathbf{x}_t)}{\gamma(1 + \lambda(\mathbf{x}_t))}\right) - \frac{\lambda(\mathbf{x}_t)}{\gamma(1 + \lambda(\mathbf{x}_t))} \\
&= f(\mathbf{x}_t) - \frac{\lambda(\mathbf{x}_t)}{\gamma} + \ln\left(1 + \frac{\lambda(\mathbf{x}_t)}{\gamma + (\gamma - 1)\lambda(\mathbf{x}_t)}\right)
\end{aligned}$$

where the first inequality follows from Lemma 28. It now follows that

$$\begin{aligned}
f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}) &\geq \frac{\lambda(\mathbf{x}_t)}{\gamma} - \ln\left(1 + \frac{\lambda(\mathbf{x}_t)}{\gamma + (\gamma - 1)\lambda(\mathbf{x}_t)}\right) \\
&\geq \frac{\lambda(\mathbf{x}_t)}{\gamma} - \frac{\lambda(\mathbf{x}_t)}{\gamma + (\gamma - 1)\lambda(\mathbf{x}_t)} \\
&= \lambda(\mathbf{x}_t)\left(\frac{1}{\gamma} - \frac{1}{\gamma + (\gamma - 1)\lambda(\mathbf{x}_t)}\right) \\
&> \frac{r}{1+r}\left(\frac{1}{\gamma} - \frac{1}{\gamma + \frac{(\gamma-1)r}{1+r}}\right) \\
&> 0
\end{aligned}$$

where the second inequality comes from the fact that, for all $x \in \mathbb{R}$, $\ln(1 + x) \leq 1 + x$. Let $\nu = \frac{r}{1+r}\left(\frac{1}{\gamma} - \frac{1}{\gamma + \frac{(\gamma-1)r}{1+r}}\right)$. Then we see that after

$$\frac{f(\mathbf{x}_1) - f(\mathbf{x}^*)}{\nu}$$

steps, we can guarantee that we have arrived at $\mathbf{x}_t$ such that $\mathbf{x}^* \in W_r(\mathbf{x}_t)$. ■

Algorithm 6 N-SVRG

Input: Convex set $\mathcal{K}$, $f(\mathbf{x}) = \frac{1}{m}\sum_{k=1}^m f_k(\mathbf{x})$, Norm $A \succeq 0$, update frequency S_1, and learning rate η

```
Initialize x̃_0
for s = 1, 2 ... do
  x̃ = x̃_{s-1}
  μ̃ = (1/m) Σ_{k=1}^m ∇f_k(x̃)
  x_0 = x̃
  for t = 1 to S_1 do
    Randomly choose i_t ∈ {1 ... m}
    g = ∇f_{i_t}(x_{t-1}) − ∇f_{i_t}(x̃) + μ̃
    x_t = Π_K^A (x_{t-1} − η A^{-1} g)
  end for
  x̃_s = x_t for randomly chosen t ∈ {1, ..., m}
end for
```

A.4 Proof of SVRG Lemma

Proof [Proof of Lemma 32] The proof follows the original proof of SVRG (Johnson and Zhang, 2013) with a few modifications to take the general norm into account. For any i, consider

$$g_i(\mathbf{x}) = f_i(\mathbf{x}) - f_i(\mathbf{x}^*) - \langle \nabla f_i(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle.$$

We know that $g_i(\mathbf{x}^*) = \min_{\mathbf{x}} g_i(\mathbf{x})$ since $\nabla g_i(\mathbf{x}^*) = 0$. Therefore,

$$\begin{aligned} 0 = g_i(\mathbf{x}^*) &\leq \min_{\eta} \left(g_i(\mathbf{x} - \eta A^{-1} \nabla g_i(\mathbf{x}^*)) \right) \\ &\leq \min_{\eta} \left(g_i(\mathbf{x}) - \eta \|g_i(\mathbf{x})\|^2_{A^{-1}} + 0.5\beta\eta^2 \|g_i(\mathbf{x})\|^2_{A^{-1}} \right) \\ &= g_i(\mathbf{x}) - \frac{1}{2\beta} \|g_i(\mathbf{x})\|^2_{A^{-1}} \ . \end{aligned}$$

The second inequality follows from smoothness of g_i, where we note that g_i is as smooth as f_i. Summing the above inequality over i and setting $\mathbf{x}^*$ to be the minimum of f so that $\nabla f(\mathbf{x}^*) = 0$, we get

$$\frac{\sum_{i=1}^m \|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{x}^*)\|^2_{A^{-1}}}{2m\beta} \leq f(\mathbf{x}) - f(\mathbf{x}^*).$$

Define $v_t = \nabla f_{i_t}(\mathbf{x}_{t-1}) - \nabla f_{i_t}(\tilde{\mathbf{x}}) + \tilde{\mu}$. Conditioned on $\mathbf{x}_{t-1}$, we can take the expectation with respect to i_t and obtain

$$\begin{aligned} \mathbf{E}\left[\|v_t\|^2_{A^{-1}}\right] &\leq 2\mathbf{E}\left[\|\nabla f_{i_t}(\mathbf{x}_{t-1}) - \nabla f_{i_t}(\mathbf{x}^*)\|^2_{A^{-1}}\right] + 2\mathbf{E}\left[\|\nabla f_{i_t}(\tilde{\mathbf{x}}) - \nabla f_{i_t}(\mathbf{x}^*) - \nabla f(\tilde{\mathbf{x}})\|^2_{A^{-1}}\right] \\ &\leq 2\mathbf{E}\left[\|\nabla f_{i_t}(\mathbf{x}_{t-1}) - \nabla f_{i_t}(\mathbf{x}^*)\|^2_{A^{-1}}\right] + 2\mathbf{E}\left[\|\nabla f_{i_t}(\tilde{\mathbf{x}}) - \nabla f_{i_t}(\mathbf{x}^*)\|^2_{A^{-1}}\right] \\ &\leq 4\beta \left(f(\mathbf{x}_{t-1}) - f(\mathbf{x}^*) + f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*) \right). \end{aligned}$$

The above inequalities follow by noting the following three facts:

- $\|a+b\|^2_{A^{-1}} \leq 2\|a\|^2_{A^{-1}} + 2\|b\|^2_{A^{-1}}$.
- $\tilde{\mu} = \nabla f(\tilde{\mathbf{x}})$.
- $\mathbf{E}\left[\|X - \mathbf{E}X\|^2_{A^{-1}}\right] \leq \mathbf{E}\|X\|^2_{A^{-1}}$.

Now note that conditioned on $\mathbf{x}_{t-1}$, we have that $\mathbf{E}v_t = \nabla f(\mathbf{x}_{t-1})$, and so

$$\begin{aligned} \mathbf{E}\|\mathbf{x}_t - \mathbf{x}^*\|^2_A &\leq \|\mathbf{x}_{t-1} - \eta A^{-1} v_t - \mathbf{x}^*\|^2_A \\ &= \|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_A - 2\eta\langle \mathbf{x}_{t-1} - \mathbf{x}^*, AA^{-1}\mathbf{E}v_t\rangle + \eta^2 \mathbf{E}\|v_t\|^2_{A^{-1}} \\ &\leq \|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_A - 2\eta\langle \mathbf{x}_{t-1} - \mathbf{x}^*, \nabla f(\mathbf{x}_{t-1})\rangle + 4\beta\eta^2 \left(f(\mathbf{x}_{t-1}) - f(\mathbf{x}^*) + f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*) \right) \\ &\leq \|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_A - 2\eta \left(f(\mathbf{x}_{t-1}) - f(\mathbf{x}^*) \right) + 4\beta\eta^2 \left(f(\mathbf{x}_{t-1}) - f(\mathbf{x}^*) + f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*) \right) \\ &= \|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_A - 2\eta(1 - 2\eta\beta) \left(f(\mathbf{x}_{t-1}) - f(\mathbf{x}^*) \right) + 4\beta\eta^2 \left(f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*) \right). \end{aligned}$$

The first inequality uses the Pythagorean inequality for norms, the second inequality uses the derived inequality for $\mathbf{E}\|v_t\|^2_{A^{-1}}$, and the third inequality uses the convexity of $f(\mathbf{x})$. Consider a fixed stage s, so that $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}_{s-1}$ and $\tilde{\mathbf{x}}_s$ are selected after all of the updates have been completed. By summing the previous inequality over t and taking an expectation over the history, we get

$$\begin{aligned} \mathbf{E}\|\mathbf{x}_n - \mathbf{x}^*\|^2_A + 2\eta(1 - 2\eta\beta)n\mathbf{E}\left[f(\mathbf{x}_s) - f(\mathbf{x}^*)\right] &\leq \mathbf{E}\|\mathbf{x}_0 - \mathbf{x}^*\|^2_A + 4\beta n\eta^2 \mathbf{E}\left[f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*)\right] \\ &= \|\tilde{\mathbf{x}} - \mathbf{x}^*\|^2_A + 4\beta n\eta^2 \mathbf{E}\left[f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*)\right] \\ &\leq 2\frac{\mathbf{E}\left[f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*)\right]}{\alpha} + 4\beta n\eta^2 \mathbf{E}\left[f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*)\right] \\ &= 2\left(\frac{1}{\alpha} + 2\beta n\eta^2\right) \mathbf{E}\left[f(\tilde{\mathbf{x}}) - f(\mathbf{x}^*)\right]. \end{aligned}$$

The second inequality uses the strong convexity property. Therefore, we have that

$$\mathbf{E}\left[f(\mathbf{x}_s) - f(\mathbf{x}^*)\right] \leq \left(\frac{1}{\alpha\eta(1 - 2\eta\beta)n} + \frac{2\eta\beta}{(1 - 2\eta\beta)}\right) \mathbf{E}\left[f(\mathbf{x}_{s-1}) - f(\mathbf{x}^*)\right].$$

■

References

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199. ACM, 2017.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *arXiv preprint arXiv:1603.05953*, 2016.

Yossi Arjevani and Ohad Shamir. Oracle complexity of second-order methods for finite-sum problems. *arXiv preprint arXiv:1611.04982*, 2016.

Jock A. Blackard and Denis J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture*, 24(3):131–151, 1999.

Raghu Bollapragada, Richard Byrd, and Jorge Nocedal. Exact and inexact subsampled Newton methods for optimization. *arXiv preprint arXiv:1609.08502*, 2016.

Charles G. Broyden. The convergence of a class of double-rank minimization algorithms: 2. the new algorithm. *IMA Journal of Applied Mathematics*, 6(3):222–231, 1970.

Richard H. Byrd, Gillian M. Chin, Will Neveitt, and Jorge Nocedal. On the use of stochastic Hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3):977–995, 2011.

Richard H. Byrd, Samantha L. Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-Newton method for large-scale optimization. *SIAM Journal on Optimization*, 26 (2):1008–1031, 2016.

Michael B. Cohen. Nearly tight oblivious subspace embeddings by trace inequalities. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 278–287. Society for Industrial and Applied Mathematics, 2016.

Michael B. Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 181–190. ACM, 2015.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

Murat A. Erdogdu and Andrea Montanari. Convergence rates of sub-sampled Newton methods. In *Advances in Neural Information Processing Systems*, pages 3034–3042, 2015.

Roger Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13 (3):317–322, 1970.

Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109):23–26, 1970.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Yann LeCun and Corinna Cortes. The MNIST database of handwritten digits. 1998. URL `http://yann.lecun.com/exdb/mnist/`.

Mu Li, Gary L. Miller, and Richard Peng. Iterative row sampling. In *IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 127–136. IEEE, 2013.

Moshe Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Hongzhou Lin, Julien Mairal, and Zaïd Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.

Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method. In *Advances in Neural Information Processing Systems*, pages 3059–3067, 2014.

Haipeng Luo, Alekh Agarwal, Nicolò Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems*, pages 902–910, 2016.

James Martens. Deep learning via Hessian-free optimization. In *International Conference on Machine Learning*, pages 735–742, 2010.

Andrew McCallum. Real-sim, 1997. Available at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#real-sim`.

Aryan Mokhtari and Alejandro Ribeiro. RES: regularized stochastic BFGS algorithm. *IEEE Transactions on Signal Processing*, 62(23):6089–6104, 2014.

Philipp Moritz, Robert Nishihara, and Michael Jordan. A linearly-convergent stochastic L-BFGS algorithm. In *Artificial Intelligence and Statistics*, pages 249–258, 2016.

Arkadi Nemirovski. Interior point polynomial time methods in convex programming. *Lecture notes*, 2004.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate O(1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2013.

Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.

Mert Pilanci and Martin J. Wainwright. Newton sketch: A linear-time optimization algorithm with linear-quadratic convergence. *arXiv preprint arXiv:1505.02250*, 2015.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

Nicolas L. Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pages 2663–2671, 2012.

Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In *Artificial Intelligence and Statistics*, pages 436–443, 2007.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013.

Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1-2):105–145, 2016.

Ohad Shamir, Nathan Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate Newton-type method. In *International Conference on Machine Learning*, pages 1000–1008, 2014.

David F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111):647–656, 1970.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W. Mahoney. Sub-sampled Newton methods with non-uniform sampling. In *Advances in Neural Information Processing Systems*, pages 3000–3008, 2016.

Haishan Ye, Luo Luo, and Zhihua Zhang. A unifying framework for convergence analysis of approximate Newton methods. *arXiv preprint arXiv:1702.08124*, 2017.

Lijun Zhang, Mehrdad Mahdavi, and Rong Jin. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, pages 980–988, 2013.

Journal of Machine Learning Research 18 (2017) 1-49 Submitted 1/17; Revised 8/17; Published 11/17

Regularized Estimation and Testing for High-Dimensional Multi-Block Vector-Autoregressive Models

Jiahe Lin jiahelin@umich.edu
Department of Statistics
University of Michigan
Ann Arbor, MI 48109, USA

George Michailidis* gmichail@ufl.edu
Department of Statistics and the Informatics Institute
University of Florida
Gainesville, FL 32611, USA

Editor: Jie Peng

Abstract

Dynamical systems comprising of multiple components that can be partitioned into distinct *blocks* originate in many scientific areas. A pertinent example is the interactions between financial assets and selected macroeconomic indicators, which has been studied at *aggregate level*—e.g. a stock index and an employment index—extensively in the macroeconomics literature. A key shortcoming of this approach is that it ignores potential influences from other related components (e.g. Gross Domestic Product) that may impact the system's dynamics and structure and thus produces incorrect results. To mitigate this issue, we consider a multi-block linear dynamical system with Granger-causal ordering between blocks, wherein the blocks' temporal dynamics are described by vector autoregressive processes and are influenced by blocks higher in the system hierarchy. We derive the maximum likelihood estimator for the posited model for Gaussian data in the high-dimensional setting based on appropriate regularization schemes for the parameters of the block components. To optimize the underlying non-convex likelihood function, we develop an iterative algorithm with convergence guarantees. We establish theoretical properties of the maximum likelihood estimates, leveraging the decomposability of the regularizers and a careful analysis of the iterates. Finally, we develop testing procedures for the null hypothesis of whether a block "Granger-causes" another block of variables. The performance of the model and the testing procedures are evaluated on synthetic data, and illustrated on a data set involving log-returns of the US S&P100 component stocks and key macroeconomic variables for the 2001–16 period.

Keywords: Vector-autoregression; Stability; Block-coordinate descent; Consistency; Global testing

1. Introduction.

The study of linear dynamical systems has a long history in control theory (Kumar and Varaiya, 1986) and economics (Hansen and Sargent, 2013) due to their analytical tractability and ease to estimate their parameters. Such systems in their so-called *reduced form* give rise to Vector Autoregressive (VAR) models (Lütkepohl, 2005) that have been widely used in macroeconomic modeling for policy analysis (Sims, 1980, 1992), in financial econometrics (Gourieroux and Jasiak, 2001), and more recently in functional genomics (Shojaie et al., 2012), financial systemic risk analysis (Billio et al., 2012) and neuroscience (Seth, 2013).

In many applications, the components of the system under consideration can be naturally partitioned into *interacting blocks*. For example, Cushman and Zha (1997) studied the impact of monetary policy in a small open economy, where the economy under consideration is modeled as one block, while variables in other (foreign) economies as the other. Both blocks have their own autoregressive structure, and the inter-dependence between blocks is unidirectional: the foreign block *influences* the small open economy, but *not* the other way around, thus effectively introducing a *linear ordering* amongst blocks. Another example comes from the connection between the stock market and employment macroeconomic variables (Fitoussi et al., 2000; Phelps, 1999; Farmer, 2015) that focuses on the impact through a wealth effect mechanism of the former on the latter. Once again, the underlying hypothesis of interest is that the stock market influences employment, but not the other way around. In another application domain, molecular biologists conduct time course experiments on cell lines or animal models and collect data across multiple molecular compartments (transcripotmics, proteomics, metabolomics, lipidomics) in order to delineate mechanisms for disease onset and progression or to study basic biological processes. In this case, the interactions amongst the blocks (molecular compartments) are clearly delineated (transciptomic compartment influencing the proteomic and metabolomic ones), thus leading again to a linear ordering of the blocks.

The proposed model also encompasses the popular in marketing, regional science and growth theory VAR-X model, provided that the temporal evolution of the *exogenous* block of variables "X" exhibits autoregressive dynamics. For example, Nijs et al. (2001) examine the sensitivity of over 500 product prices to various marketing promotion strategies (the exogenous block), while Pauwels and Weiss (2008) examine changes in subscription rates, search engine referrals and marketing efforts of customers when switched from a free account to a fee-based structure, the latter together with customer characteristics representing the exogenous block. Pesaran et al. (2004) examine regional inter-dependencies, building a model where country specific macroeconomic indicators evolve according to a VAR model and they are influenced exogenously by key macroeconomic variables from neighboring countries/regions. Finally, Abeysinghe (2001) studies the impact of the price of oil on Gross Domestic Product growth rates for a number of countries, while controlling for other exogenous variables such as the country's consumption and investment expenditures along with its trade balance.

The proposed model gives rise to a network structure that in its most general form corresponds to a multi-partite graph, depicted in Figure 1 for 3 blocks, that exhibits a *directed acyclic structure* between the constituent blocks, and can also exhibit additional dependence between the nodes in each block. Selected properties of such multi-block structures, known as *chain graphs*, have been studied in the literature. Specifically, their maximum likelihood estimation for *independent and identically distributed* Gaussian data under a high-dimensional *sparse* regime is thoroughly investigated in Lin et al. (2016), where a provably convergent estimation procedure is introduced and its theoretical properties are established.

*. Corresponding Author. Post Address: 205 Griffin Floyd Hall, 1 University Ave, Gainesville, FL, 32611.

©2017 Jiahe Lin and George Michailidis.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-055.html.

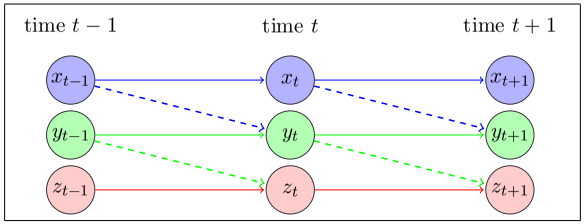


Figure 1: Diagram for a dynamic system with three groups of variables

Given the wide range of applications of multi-block VAR models, which in addition encompass the widely used VAR-X model, the key contributions of the current paper are fourfold: (i) formulating the model as a recursive dynamical system and examining its stability properties; (ii) developing a provably convergent algorithm for obtaining the regularized maximum likelihood estimates (MLE) of the model parameters under high-dimensional scaling; (iii) establishing theoretical properties of the ML estimates; and (iv) devising a testing procedure for the parameters that connect the constituent blocks of the model: if the null hypothesis is not rejected, then one is dealing with a set of independently evolving VAR models, otherwise with the posited multi-block VAR model. Finally, the model, estimation and testing procedures are illustrated on an important problem in macroeconomics, as gleaned by the background of the problem and discussion of the results provided in Section 6.

For the multi-block VAR model, we assume that the time series within each block are generated by a Gaussian VAR process. Further, the transition matrices within and across blocks can be either *sparse* or *low rank*. The posited regularized Gaussian likelihood function is not *jointly convex* in all the model parameters, which poses a number of technical challenges that are compounded by the presence of temporal dependence. These are successfully addressed and resolved in Section 3, where we provide a numerically convergent algorithm and establish the theoretical properties of the resulting ML estimates, that constitutes a key contribution in the study of multi-block VAR models.

The remainder of this paper is organized as follows. In Section 2, we introduce the model setup and the corresponding estimation procedure. In Section 3, we provide consistency properties of the obtained ML estimates under a high-dimensional scaling. In Section 4, we introduce the proposed testing framework, both for low-rank and sparse interaction matrices between the blocks. Section 5 contains selected numerical results that assess the performance of the estimation and testing procedures. Finally, an application to financial and macroeconomic data that was previously discussed as motivation for the model under consideration is presented in Section 6.

Notation. Throughout this paper, we use $|||A|||_1$ and $|||A|||_\infty$ respectively to denote the matrix induced 1-norm and infinity norm of $A \in \mathbb{R}^{m\times n}$, that is, $|||A|||_1 = \max_{1\le j\le n}\sum_{i=1}^m |a_{ij}|$, $|||A|||_\infty = \max_{1\le i\le m}\sum_{j=1}^n |a_{ij}|$, and use $\|A\|_1$ and $\|A\|_\infty$ respectively to denote the element-wise 1-norm and infinity norm: $\|A\|_1 = \sum_{i,j}|a_{ij}|$, $\|A\|_\infty = \max_{i,j}|a_{ij}|$. Moreover, we use $|||A|||_*$, $|||A|||_F$ and $|||A|||_{op}$ to denote the nuclear, Frobenius and operator norms of A, respectively. For two matrices A and B of commensurate dimensions, denote their inner product by $\langle\!\langle A, B\rangle\!\rangle = \mathrm{trace}(A'B)$. Finally, we write $A \gtrsim B$ if there exists some absolute constant c that is independent of the model parameters such that $A \ge cB$.

2. Problem Formulation.

To convey the main ideas and the key technical contributions, we consider a recursive linear dynamical system comprising of two blocks of variables, whose structure is given by:

$$\begin{aligned} X_t &= AX_{t-1} + U_t, \\ Z_t &= BX_{t-1} + CZ_{t-1} + V_t, \end{aligned} \tag{1}$$

where $X_t \in \mathbb{R}^{p_1}, Z_t \in \mathbb{R}^{p_2}$ are the variables in groups 1 and 2, respectively. The temporal intra-block dependence is captured by transition matrices A and C, while the inter-block dependence by B. Noise processes $\{U_t\}$ and $\{V_t\}$, respectively, capture additional contemporaneous intra-block dependence of X_t and Z_t, after conditioning on their respective past values. Further, we assume that U_t and V_t follow mean zero Gaussian distributions with covariance matrices given by Σ_u and Σ_v, i.e.,

$$U_t \sim \mathcal{N}(0, \Sigma_u), \quad \text{and} \quad V_t \sim \mathcal{N}(0, \Sigma_v).$$

With the above model setup, the parameters of interest are transition matrices $A \in \mathbb{R}^{p_1\times p_1}$, $B \in \mathbb{R}^{p_2\times p_1}$ and $C \in \mathbb{R}^{p_2\times p_2}$, as well as the covariances Σ_u, Σ_v. In high-dimensional settings, different combinations of structural assumptions can be imposed on these transition matrices to enable their estimation from limited time series data. In particular, the intra-block transition matrices A and C are sparse, while the inter-block matrix B can be either sparse or low rank. Note that the block of X_t variables acts as an *exogenous* effect to the evolution of the Z_t block (e.g., Cushman and Zha, 1997; Nicolson et al., 2016). Further, we assume $\Omega_u := \Sigma_u^{-1}$ and $\Omega_v := \Sigma_v^{-1}$ are sparse.

Remark 1 *For ease of exposition, we posit a* VAR(1) *modeling structure. Extensions to general multi-block structures akin to the one depicted in Figure 1 and* VAR(d) *specifications are rather straightforward and briefly discussed in Section 7.*

The triangular (recursive) structure of the system enables a certain degree of separability between X_t and Z_t. In the posited model, X_t is a stand-alone VAR(1) process, and the time series in block Z_t is "Granger-caused" by that in block X_t, but not vice versa. The second equation in (1), as mentioned in the introductory section, also corresponds to the so-called "VAR-X" model in the econometrics literature (e.g., Sims, 1980; Bianchi et al., 2010), that extends the standard VAR model to include influences from lagged values of *exogenous* variables. Consider the joint process $W_t = (X_t', Z_t')'$, it corresponds to a VAR(1) model whose transition matrix G has a block triangular form:

$$W_t = GW_{t-1} + \varepsilon_t, \qquad \text{where} \quad G := \begin{bmatrix} A & O \\ B & C \end{bmatrix}, \quad \varepsilon_t = \begin{bmatrix} U_t \\ V_t \end{bmatrix}. \tag{2}$$

The model in (2) can also be viewed from a Structural Equations Modeling viewpoint involving time series data, and also has a Moving Average representation corresponding to a

structural VAR representation with Granger causal ordering (Lütkepohl, 2005). As mentioned in the introductory section, the focus of this paper is model parameter estimation under high-dimensional scaling, rather than their cause and effect relationship. For a comprehensive discourse of causality issues for VAR models, we refer the readers to Granger (1969); Lütkepohl (2005), and references therein.

Next, we introduce the notion of stability and spectrum with respect to the system.

System Stability. To ensure that the joint process $\{W_t\}$ is stable (Lütkepohl, 2005), we require the spectral radius, denoted by $\rho(\cdot)$, of the transition matrix G to be smaller than 1, which is guaranteed by requiring that $\rho(A) < 1$ and $\rho(C) < 1$, since

$$|\lambda \mathrm{I}_{p_1 \times p_2} - G| = \begin{vmatrix} \lambda \mathrm{I}_{p_1} - A & O \\ -B & \lambda \mathrm{I}_{p_2} - C \end{vmatrix} = |\lambda \mathrm{I}_{p_1} - A||\lambda \mathrm{I}_{p_2} - C|,$$

implying that the set of eigenvalues of G is the union of the sets of eigenvalues of A and C, hence

$$\rho(A) < 1 \ , \rho(C) < 1, \qquad \Rightarrow \qquad \rho(G) = \max\{\rho(A), \rho(C)\} < 1.$$

The latter relation implies that the stability of such a recursive system imposes spectrum constraints only on the diagonal blocks that govern the intra-block evolution, whereas the off-diagonal block that governs the inter-block interaction is left unrestricted.

Spectrum of the joint process. Throughout, we assume that the spectral density of $\{W_t\}$ exists, which then possesses a special structure as a result of the block triangular transition matrix G. Formally, we define the spectral density of $\{W_t\}$ as

$$f_W(\theta) = \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \Gamma_W(h) e^{-ih\theta}, \qquad \theta \in [-\pi, \pi],$$

where $\Gamma_W(h) := \mathbb{E} W_t W_{t+h}'$. For two (generic) processes $\{X_t\}$ and $\{Z_t\}$, define their cross-covariance as $\Gamma_{X,Z}(h) = \mathbb{E} X_t Z_{t+h}'$ and $\Gamma_{Z,X}(h) = \mathbb{E} Z_t X_{t+h}'$. In general, $\Gamma_{X,Z}(h) \neq \Gamma_{Z,X}(h)$. The cross-spectra are defined as:

$$f_{X,Z}(\theta) := \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \Gamma_{X,Z}(h) e^{-ih\theta}, \quad \text{and} \quad f_{Z,X}(\theta) := \frac{1}{2\pi} \sum_{h=-\infty}^{\infty} \Gamma_{Z,X}(h) e^{-ih\theta}, \quad \theta \in [-\pi, \pi].$$

For the model given in (2), by writing out the dynamics of Z_t, the cross-spectra between X_t and Z_t are given by

$$f_{X,Z}(\theta)(\mathrm{I}_{p_2} - C' e^{-i\theta}) = f_X(\theta) B' e^{-i\theta}, \qquad \text{and} \qquad (\mathrm{I}_{p_2} - C e^{i\theta}) f_{Z,X}(\theta) = B e^{i\theta} f_X(\theta). \quad (3)$$

Similarly, we have

$$(\mathrm{I}_{p_2} - C e^{i\theta}) f_Z(\theta) = B e^{i\theta} f_{X,Z}(\theta) + f_{V,Z}(\theta). \quad (4)$$

Combining (3) and (4), the spectrum of the joint process W_t is given by

$$f_W(\theta) = \big[H_1(e^{i\theta})\big]^{-1} \Big(\big[H_2(e^{i\theta})\big] \big[\mathbf{1}_{2\times 2} \otimes f_X(\theta)\big] \big[H_2(e^{-i\theta})\big]^\top + \begin{bmatrix} O & O \\ O & \Sigma_v \end{bmatrix} \Big) \big[H_1(e^{-i\theta})\big]^{-\top}, \quad (5)$$

where $\mathbf{1}_{2\times 2}$ is a 2×2 matrix with all entries being 1, and

$$H_1(x) := \begin{bmatrix} \mathrm{I}_{p_1} & O \\ O & \mathrm{I}_{p_2} - Cx \end{bmatrix} \in \mathbb{R}^{(p_1+p_2)\times(p_1+p_2)}, \quad H_2(x) := \begin{bmatrix} \mathrm{I}_{p_1} & O \\ O & Bx \end{bmatrix} \in \mathbb{R}^{(p_1+p_2)\times(2p_1)}.$$

Equation (5) implies that the spectrum of the joint process $\{W_t\}$ can be decomposed into the sum of two parts: the first, is a function of $f_X(\theta)$, while the second part involves the idiosyncratic error process $\{V_t\}$, which only affects the right-bottom block of the spectrum. Note that since $\{W_t\}$ is a VAR(1) process, its matrix-valued characteristic polynomial is given by

$$\mathcal{G}(\theta) := \mathrm{I}_{(p_1+p_2)} - G\theta,$$

and its spectral density also takes the following form (c.f. Hannan, 1970):

$$f_W(\theta) = \frac{1}{2\pi} \big[\mathcal{G}^{-1}(e^{i\theta})\big] \Sigma_\varepsilon \big[\mathcal{G}^{-1}(e^{i\theta})\big]^*,$$

where

$$\mathcal{G}(x) = \begin{bmatrix} \mathrm{I}_{p_1} - Ax & O \\ -Bx & \mathrm{I}_{p_2} - Cx \end{bmatrix}, \qquad \Sigma_\varepsilon = \begin{bmatrix} \Sigma_u & O \\ O & \Sigma_v \end{bmatrix},$$

and $\mathcal{G}^*$ is the conjugate transpose. One can easily reach the same conclusion as in (5) by multiplying each term, followed by some algebraic manipulations.

2.1 Estimation.

Next, we outline the algorithm for obtaining the ML estimates of the transition matrices A, B and C and inverse covariance matrices Σ_u^{-1} and Σ_v^{-1} from time series data. We allow for a potential high-dimensional setting, where the ambient dimensions p_1 and p_2 of the model exceed the total number of observations T.

Given centered times series data $\{x_0, \cdots, x_T\}$ and $\{z_0, \cdots, z_T\}$, we use $\mathcal{X}^T$ and $\mathcal{Z}^T$ respectively, to denote the "response" matrix from time 1 to T, that is:

$$\mathcal{X}^T = \begin{bmatrix} x_1 & x_2 & \dots & x_T \end{bmatrix}' \quad \text{and} \quad \mathcal{Z}^T = \begin{bmatrix} z_1 & z_2 & \dots & z_T \end{bmatrix}',$$

and use $\mathcal{X}$ and $\mathcal{Z}$ without the superscript to denote the "design" matrix from time 0 to $T-1$:

$$\mathcal{X} = \begin{bmatrix} x_0 & x_1 & \dots & x_{T-1} \end{bmatrix}' \quad \text{and} \quad \mathcal{Z} = \begin{bmatrix} z_0 & z_1 & \dots & z_{T-1} \end{bmatrix}'.$$

We use $\mathcal{U}$ and $\mathcal{V}$ to denote the error matrices. To obtain estimates for the parameters of interest, we formulate optimization problems using a penalized log-likelihood function, with regularization terms corresponding to the imposed structural assumptions on the model parameters–sparsity and/or low-rankness. To solve the optimization problems, we employ block-coordinate descent algorithms, akin to those described in Lin et al. (2016), to obtain the solution.

As previously mentioned, $\{X_t\}$ is not "Granger-caused" by Z_t and hence it is a stand-alone VAR(1) process; this enables us to separately estimate the parameters governing the X_t process (A and Σ_u^{-1}) from those of the Z_t process (B, C, and Σ_v^{-1}).

Estimation of A and Σ_u^{-1}. Conditional on the initial observation x_0, the likelihood of $\{x_t\}_{t=1}^T$ is given by:

$$\begin{aligned} L(x_T, x_{T-1}, \cdots, x_1 | x_0) &= L(x_T | x_{T-1}, \cdots, x_0) L(x_{T-1} | x_{T-2}, \cdots, x_0) \cdots L(x_1 | x_0) \\ &= L(x_T | x_{T-1}) L(x_{T-1} | x_{T-2}) \cdots L(x_1 | x_0), \end{aligned}$$

where the second equality follows from the Markov property of the process. The log-likelihood function is given by:

$$\ell(A, \Sigma_u^{-1}) = \frac{T}{2} \log\det(\Sigma_u^{-1}) - \frac{1}{2} \sum_{t=1}^{T} (x_t - A x_{t-1})' \Sigma_u^{-1} (x_t - A x_{t-1}) + \text{constant}.$$

Letting $\Omega_u := \Sigma_u^{-1}$, then the penalized maximum likelihood estimator can be written as

$$(\widehat{A}, \widehat{\Omega}_u) = \underset{\substack{A \in \mathbb{R}^{p_1 \times p_2} \\ \Omega_u \in \mathbb{S}^{++}_{p_1 \times p_1}}}{\arg\min} \Big\{ \mathrm{tr}\big[\Omega_u (\mathcal{X}^T - \mathcal{X} A')'(\mathcal{X}^T - \mathcal{X} A')/T\big] - \log\det\Omega_u + \lambda_A \|A\|_1 + \rho_u \|\Omega_u\|_{1,\text{off}} \Big\}. \quad (6)$$

Algorithm 1 describes the key steps for obtaining $\widehat{A}$ and $\widehat{\Omega}_u$.

Algorithm 1: Computational procedure for estimating A and Σ_u^{-1}.

Input: Time series data $\{x_t\}_{t=1}^T$, tuning parameter λ_A and ρ_u.

1 **Initialization:** Initialize with $\widehat{\Omega}_u^{(0)} = \mathrm{I}_{p_1}$, then
$$\widehat{A}^{(0)} = \arg\min_A \big\{ \tfrac{1}{T} \big\| \mathcal{X}^T - \mathcal{X} A' \big\|_F^2 + \lambda_A \|A\|_1 \big\};$$

2 **Iterate until convergence:**

– Update $\widehat{\Omega}_u^{(k)}$ by graphical Lasso (Friedman et al., 2008) on the residuals with the plug-in estimate $\widehat{A}^{(k)}$;

– Update $\widehat{A}^{(k)}$ with the plug-in $\widehat{\Omega}_u^{(k-1)}$ and cyclically update each row with a Lasso penalty, which solves

$$\min_A \big\{ \tfrac{1}{T} \mathrm{tr}\big[\widehat{\Omega}_u^{(k-1)} (\mathcal{X}^T - \mathcal{X} A)'(\mathcal{X}^T - \mathcal{X} A)/T\big] + \lambda_A \|A\|_1 \big\}. \quad (7)$$

Output: Estimated sparse transition matrix $\widehat{A}$ and sparse $\widehat{\Omega}_u$.

Estimation of B, C and Σ_v^{-1}. Similarly, to obtain estimates of B, C and $\Omega_v := \Sigma_v^{-1}$, we formulate the optimization problem as follows:

$$\begin{aligned} (\widehat{B}, \widehat{C}, \widehat{\Omega}_v) = \underset{\substack{B \in \mathbb{R}^{p_2 \times p_1}, C \in \mathbb{R}^{p_2 \times p_2} \\ \Omega_v \in \mathbb{S}^{++}_{p_2 \times p_2}}}{\arg\min} \Big\{ & \mathrm{tr}\big[\Omega_v (\mathcal{Z}^T - \mathcal{X} B' - \mathcal{Z} C')'(\mathcal{Z}^T - \mathcal{X} B' - \mathcal{Z} C')/T\big] - \log\det\Omega_v \\ & + \lambda_B \mathcal{R}(B) + \lambda_C \|C\|_1 + \rho_v \|\Omega_v\|_{1,\text{off}} \Big\}, \end{aligned} \quad (8)$$

where the regularizer $\mathcal{R}(B) = \|B\|_1$ if B is assumed to be sparse, and $\mathcal{R}(B) = \|B\|_*$ if B is assumed to be low rank. Algorithm 2 outlines the procedure for obtaining estimates $\widehat{B}$, $\widehat{C}$

Algorithm 2: Computational procedure for estimating B, C and Σ_v^{-1}.

Input: Time series data $\{x_t\}_{t=1}^T$ and $\{z_t\}_{t=1}^T$, tuning parameters λ_B, λ_C, ρ_v.

1 **Initialization:** Initialize with $\widehat{\Omega}_v^{(0)} = \mathrm{I}_{p_2}$, then
$$(\widehat{B}^{(0)}, \widehat{C}^{(0)}) = \underset{(B,C)}{\arg\min} \big\{ \tfrac{1}{T} \big\| \mathcal{Z}^T - \mathcal{X} B' - \mathcal{Z} C' \big\|_F^2 + \lambda_B \mathcal{R}(B) + \lambda_C \|C\|_1 \big\}.$$

2 **Iterate until convergence:**

– Update $\widehat{\Omega}_v^{(k)}$ by graphical Lasso on the residuals with the plug-in estimates $\widehat{B}^{(k)}$ and $\widehat{C}^{(k)}$;

– For fixed $\widehat{\Omega}_v^{(k)}$, $(\widehat{B}^{(k+1)}, \widehat{C}^{(k+1)})$ solves
$$\min_{B,C} \big\{ \tfrac{1}{T} \mathrm{tr}\big[\widehat{\Omega}_v^{(k)} (\mathcal{Z}^T - \mathcal{X} B' - \mathcal{Z} C')'(\mathcal{Z}^T - \mathcal{X} B' - \mathcal{Z} C')\big] + \lambda_B \mathcal{R}(B) + \lambda_C \|C\|_1 \big\}.$$

· Fix $\widehat{C}^{[s]}$, update $\widehat{B}^{[s+1]}$ by Lasso or singular value thresholding, which solves
$$\min_B \big\{ \tfrac{1}{T} \mathrm{tr}\big[\widehat{\Omega}_v^{(k)} (\mathcal{Z}^T - \mathcal{X} B' - \mathcal{Z} \widehat{C}^{[s]'})'(\mathcal{Z}^T - \mathcal{X} B' - \mathcal{Z} \widehat{C}^{[s]'})\big] + \lambda_B \mathcal{R}(B) \big\};$$

· Fix $\widehat{B}^{[s]}$, update $\widehat{C}^{[s]}$ by Lasso, which solves
$$\min_C \big\{ \tfrac{1}{T} \mathrm{tr}\big[\widehat{\Omega}_v^{(k)} (\mathcal{Z}^T - \mathcal{X} \widehat{B}^{[s]'} - \mathcal{Z} C')'(\mathcal{Z}^T - \mathcal{X} \widehat{B}^{[s]'} - \mathcal{Z} C')\big] + \lambda_C \|C\|_1 \big\}.$$

Output: Estimated transition matrices $\widehat{B}$, $\widehat{C}$ and sparse $\widehat{\Omega}_v$.

and $\widehat{\Omega}_v$. Note that $\widehat{B}^{(k)}$ and $\widehat{C}^{(k)}$ need to be treated as a "joint block" in the outer update and convergence of the "joint block" is required before moving on to updating Ω_v.

Note that the objective function in (6) is not *jointly convex* in both parameters, but *biconvex*. Similarly in (8), the objective function is biconvex in $[(B, C), \Omega_v]$. Consequently, convergence to a stationary point is guaranteed, as long as estimates from all iterations lie within a ball around the true value of the parameters, with the radius of the ball upper bounded by a universal constant that only depends on model dimensions and sample size (Lin et al., 2016, Theorem 4.1). This condition is satisfied upon the establishment of consistency properties of the estimates.

To establish consistency properties of the estimates requires the existence of good initial values for the model parameters (A, Ω_u), and (B, C, Ω_v), respectively, in the sense that they are sufficiently close to the true parameters. For the (A, Ω_u) parameters, the results in Basu and Michailidis (2015) guarantee that for random realizations of $\{X_t, E_t\}$, with sufficiently large sample size, the errors of $\widehat{A}^{(0)}$ and $\widehat{\Omega}_u^{(0)}$ are bounded with high probability, which provides us with good initialization values. Yet, additional handling of the bounds is required to ensure that estimates from subsequent iterations are also uniformly close to the true value (see Section 3.2 Theorem 5). A similar property for $(\widehat{B}^{(0)}, \widehat{C}^{(0)}, \widehat{\Omega}_v^{(0)})$ and subsequent iterations is established in Section 3.2 Theorem 6 (see also Theorem 15 in Appendix A).

3. Theoretical Properties.

In this section, we investigate the theoretical properties of the penalized maximum likelihood estimation procedure proposed in Section 2, with an emphasis on the error bounds for the obtained

estimates. We focus on the model specification in which the inter-block transition matrix B is *low rank*, which is of interest in many applied settings. Specifically, we consider the consistency properties of $\widehat{A}$ and $(\widehat{B}, \widehat{C})$ that are solutions to the following two optimization problems:

$$(\widehat{A}, \widehat{\Omega}_u) = \underset{A,\Omega_u}{\arg\min} \Big\{ \mathrm{tr}\left[\Omega_u(\mathcal{X}^T - \mathcal{X}A')'(\mathcal{X}^T - \mathcal{X}A')/T\right] - \log\det\Omega_u + \lambda_A\|A\|_1 + \rho_u\|\Omega_u\|_{1,\mathrm{off}} \Big\}, \quad (9)$$

and

$$(\widehat{B}, \widehat{C}, \widehat{\Omega}_v) = \underset{B,C,\Omega_v}{\arg\min} \Big\{ \mathrm{tr}\left[\Omega_v(\mathcal{Z}^T - \mathcal{X}B' - \mathcal{Z}C')'(\mathcal{Z}^T - \mathcal{X}B' - \mathcal{Z}C')/T\right] - \log\det\Omega_v + \lambda_B|||B|||_* + \lambda_C\|C\|_1 + \rho_v\|\Omega_v\|_{1,\mathrm{off}} \Big\}. \quad (10)$$

The case of a sparse B can be handled similarly to that of A and/or C with minor modifications (details shown in Supplementary Material **??**).

3.1 A road map for establishing the consistency results.

Next, we outline the main steps followed in establishing the theoretical properties for the model parameters. Throughout, we denote with a superscript "$\star$" the true value of the corresponding parameters.

The following key concepts, widely used in high-dimensional regularized estimation problems, are needed in subsequent developments.

Definition 2 (Restricted Strong Convexity (RSC)) *For some generic operator $\mathfrak{X} : \mathbb{R}^{m_1\times m_2} \mapsto \mathbb{R}^{T\times m_1}$, it satisfies the RSC condition with respect to norm Φ with curvature $\alpha_{RSC} > 0$ and tolerance $\tau > 0$ if*

$$\frac{1}{2T}|||\mathfrak{X}(\Delta)|||_F^2 \geq \alpha_{RSC}|||\Delta|||_F^2 - \tau\Phi^2(\Delta), \qquad \textit{for some } \Delta \in \mathbb{R}^{m_1\times m_2}.$$

Note that the choice of the norm Φ is context specific. For example, in sparse regression problems, $\Phi(\Delta) = \|\Delta\|_1$ corresponds to the element-wise ℓ_1 norm of the matrix (or the usual vector ℓ_1 norm for the vectorized version). The RSC condition becomes equivalent to the *restricted eigenvalue (RE) condition* (see Loh and Wainwright, 2012; Basu and Michailidis, 2015, and references therein). This is the case for the problem of estimating transition matrix A. For estimating B and C, define $\mathcal{Q}$ to be the weighted regularizer $\mathcal{Q}(B, C) := |||B|||_* + \frac{\lambda_C}{\lambda_B}\|C\|_1$, and the associated norm Φ in this setting is defined as

$$\Phi(\Delta) := \inf_{B_{\mathrm{aug}}+C_{\mathrm{aug}}=\Delta} \mathcal{Q}(B, C).$$

Definition 3 (Diagonal dominance) *A matrix $\Omega \in \mathbb{R}^{p\times p}$ is strictly diagonally dominant if*

$$|\Omega_{ii}| > \sum_{j\neq i}|\Omega_{ij}|, \quad \forall \;\; i = 1, \cdots, p.$$

Definition 4 (Incoherence condition (Ravikumar et al., 2011)) *A matrix $\Omega \in \mathbb{R}^{p\times p}$ satisfies the incoherence condition if:*

$$\max_{e\in(S_\Omega)^c} \|H_{eS_\Omega}(H_{S_\Omega S_\Omega})^{-1}\|_1 \leq 1 - \xi, \quad \textit{for some } \xi \in (0, 1),$$

where $H_{S_\Omega S_\Omega}$ denotes the Hessian of the log-determinant barrier $\log\det\Omega$ restricted to the true edge set of Ω denoted by S_Ω, and H_{eS} is similarly defined.

The above two conditions are associated with the inverse covariance matrices Ω_u and Ω_v. Specifically, the diagonal dominance condition is required for $\Omega_u^\star$ and $\Omega_v^\star$ as we build the consistency properties for $\widehat{A}$ and $(\widehat{B}, \widehat{C})$ with the penalized maximum likelihood formulation. The incoherence condition is primarily required for establishing the consistency of $\widehat{\Omega}_u$ and $\widehat{\Omega}_v$.

We additionally introduce the upper and lower extremes of the spectrum, defined as

$$\mathcal{M}(f_X) := \operatorname*{esssup}_{\theta\in[-\pi,\pi]} \Lambda_{\max}(f_X(\theta)) \qquad \text{and} \qquad \mathfrak{m}(f_X) := \operatorname*{essinf}_{\theta\in[-\pi,\pi]} \Lambda_{\min}(f_X(\theta)).$$

Analogously, the upper extreme for the cross-spectrum is given by:

$$\mathcal{M}(f_{X,Z}) := \operatorname*{esssup}_{\theta\in[-\pi,\pi]} \sqrt{\Lambda_{\max}(f^*_{X,Z}(\theta)f_{X,Z}(\theta))},$$

with $f^*_{X,Z}(\theta)$ being the conjugate transpose of $f_{X,Z}(\theta)$. With this definition,

$$\mathcal{M}(f_{X,Z}) = \mathcal{M}(f_{Z,X}).$$

Next, consider the solution to (9) that is obtained by the alternate update between A and Ω_u. If Ω_u is held fixed, then A solves (11), and we denote the solution by $\bar{A}$ and its corresponding vectorized version as $\bar{\beta}_A := \mathrm{vec}(\bar{A})$:

$$\bar{\beta}_A := \underset{\beta\in\mathbb{R}^{p_1^2}}{\arg\min} \big\{ -2\beta'\gamma_X + \beta'\Gamma_X\beta + \lambda_A\|\beta\|_1 \big\}, \quad (11)$$

where

$$\Gamma_X = \Omega_u \otimes \tfrac{\mathcal{X}'\mathcal{X}}{T}, \qquad \gamma_X = \tfrac{1}{T}\left(\Omega_u \otimes \mathcal{X}'\right)\mathrm{vec}(\mathcal{X}^T). \quad (12)$$

Using a similar notation, if A is held fixed, then Ω_u solves (13) with the solution being $\bar{\Omega}_u$:

$$\bar{\Omega}_u := \underset{\Theta\in\mathbb{S}^{++}_{p_1\times p_1}}{\arg\min} \big\{ \log\det\Omega_u - \mathrm{trace}\,(S_u\Omega_u) + \rho_u\|\Omega_u\|_{1,\mathrm{off}} \big\}, \quad (13)$$

where

$$S_u = \tfrac{1}{T}(\mathcal{X}^T - \mathcal{X}A')'(\mathcal{X}^T - \mathcal{X}A'). \quad (14)$$

For *fixed* realizations of $\mathcal{X}$ and $\mathcal{U}$, by Basu and Michailidis (2015), the error bound of $\bar{\beta}_A$ relies on (1) Γ_X satisfying the RSC condition defined above; and (2) the tuning parameter λ_A is chosen in accordance with a deviation bound condition associated with $\|\mathcal{X}'\mathcal{U}\Omega_u/T\|_\infty$. By Ravikumar et al. (2011), the error bound of $\bar{\Omega}_u$ relies on how well S_u concentrates around $\Sigma_u^\star$, that is, $\|S_u - \Sigma_u^\star\|_\infty$. Specifically, for (12) and (14), with $\Omega_u^\star$ and $A^\star$ plugged in respectively, for *random* realizations of $\mathcal{X}$ and $\mathcal{U}$, these conditions hold with high probability. In the actual implementation of the algorithm, however, quantities in (12) and (14) are substituted by estimates so that at iteration k, $\widehat{\beta}_A^{(k)}$ and $\widehat{\Omega}_u^{(k)}$ solve

$$\widehat{\beta}_A^{(k)} := \underset{\beta\in\mathbb{R}^{p_1^2}}{\arg\min} \big\{ -2\beta'\widehat{\gamma}_X^{(k)} + \beta'\widehat{\Gamma}_X^{(k)}\beta + \lambda_A\|\beta\|_1 \big\},$$

$$\widehat{\Omega}_u^{(k)} := \underset{\Omega_u\in\mathbb{S}^{++}_{p_1\times p_1}}{\arg\min} \big\{ \log\det\Omega_u - \mathrm{trace}\big(\widehat{S}_u^{(k)}\Omega_u\big) + \rho_u\|\Omega_u\|_{1,\mathrm{off}} \big\},$$

where

$$\widehat{\Gamma}_X^{(k)} = \widehat{\Omega}_u^{(k-1)} \otimes \tfrac{\mathcal{X}'\mathcal{X}}{T}, \quad \widehat{\gamma}_X^{(k)} = \tfrac{1}{T}\big(\widehat{\Omega}_u^{(k-1)} \otimes \mathcal{X}\big), \quad \widehat{S}_u^{(k)} = \tfrac{1}{T}\big[\mathcal{X}^T - \mathcal{X}(\widehat{A}^{(k)})'\big]'\big[\mathcal{X}^T - \mathcal{X}(\widehat{A}^{(k)})'\big].$$

As a consequence, to establish the finite-sample bounds of $\widehat{A}$ and $\widehat{\Omega}_u$ given in (9), we need $\widehat{\Gamma}_X^{(k)}$ to satisfy the RSC condition, a bound on $\|\mathcal{X}'\mathcal{U}\widehat{\Omega}_u^{(k-1)}\|_\infty$ and a bound on $\|\widehat{S}_u^{(k)} - \Sigma_u^\star\|_\infty$ for all k. Toward this end, we prove that for random realizations of $\mathcal{X}$ and $\mathcal{U}$, with high probability, the RSC condition for $\widehat{\Gamma}_X^{(k)}$ and the universal bounds for $\|\mathcal{X}'\mathcal{U}\widehat{\Omega}_u^{(k-1)}\|_\infty$ and $\|\widehat{S}_u^{(k)} - \Sigma_u^\star\|_\infty$ hold for *all iterations* k, albeit the quantities of interest rely on estimates from the previous or current iterations. Consistency results of $\widehat{A}$ and $\widehat{\Omega}_u$ then readily follow.

Next, consider the solution to (10) that alternately updates (B, C) and Ω_v. As the regularization term involves both the nuclear norm penalty and the ℓ_1 norm penalty, additional handling of the norms is required which leverages the idea of decomposable regularizers (Agarwal et al., 2012). Specifically, if Ω_v and (B, C) are respectively held fixed, then

$$(\bar{B}, \bar{C}) := \arg\min_{B,C} \Big\{ \tfrac{1}{T}\mathrm{tr}\big[\Omega_v(\mathcal{Z}^T - \mathcal{X}B' - \mathcal{Z}C')'(\mathcal{Z}^T - \mathcal{X}B' - \mathcal{Z}C')\big] + \lambda_B|||B|||_* + \lambda_C\|C\|_1 \Big\},$$
$$\bar{\Omega}_v := \arg\min_{\Omega_v} \big\{ \log\det\Omega_v - \mathrm{trace}\big(S_v\Omega_v\big) + \rho_v\|\Omega_v\|_{1,\mathrm{off}} \big\},$$

where $S_v = \frac{1}{T}(Z^T - \mathcal{X}B' - \mathcal{Z}C')'(Z^T - \mathcal{X}B' - \mathcal{Z}C')$. If we let $\mathcal{W} := [\mathcal{X}, \mathcal{Z}] \in \mathbb{R}^{T\times(p_1+p_2)}$, and define the operator $\mathfrak{W}_{\Omega_v} : \mathbb{R}^{p_2\times(p_1+p_2)} \mapsto \mathbb{R}^{T\times p_2}$ induced jointly by $\mathcal{W}$ and Ω_v as

$$\mathfrak{W}_{\Omega_v}(\Delta) := \mathcal{W}\Delta'\Omega_v^{1/2} \quad \text{for } \Delta \in \mathbb{R}^{p_2\times(p_1+p_2)}, \tag{15}$$

then $\bar{B}_{\mathrm{aug}} := [\bar{B}, O_{p_2\times p_1}]$ and $\bar{C}_{\mathrm{aug}} := [O_{p_2\times p_1}, \bar{C}]$ are equivalently given by

$$(\bar{B}_{\mathrm{aug}}, \bar{C}_{\mathrm{aug}}) = \arg\min_{B,C} \Big\{ \tfrac{1}{T}\Big|\Big|\Big|\mathcal{Z}^T\Omega_v^{1/2} - \mathfrak{W}_{\Omega_v}(B_{\mathrm{aug}} + C_{\mathrm{aug}})\Big|\Big|\Big|_F^2 + \lambda_B|||B|||_* + \lambda_C\|C\|_1 \Big\}, \tag{16}$$

where $B_{\mathrm{aug}} := [B, O_{p_2\times p_2}], C_{\mathrm{aug}} := [O_{p_2\times p_1}, C] \in \mathbb{R}^{p_2\times(p_1+p_2)}$. Then, for fixed realizations of $\mathcal{Z}$, $\mathcal{X}$ and $\mathcal{V}$, with an extension of Agarwal et al. (2012) the error bound of $(\bar{B}_{\mathrm{aug}}, \bar{C}_{\mathrm{aug}})$ relies on (1) the operator $\mathfrak{W}_{\Omega_v}$ satisfying the RSC condition; and (2) tuning parameters λ_B and λ_C are respectively chosen in accordance with the deviation bound conditions associated with

$$|||\mathcal{W}'\mathcal{V}\Omega_v/T|||_{op} \qquad \text{and} \qquad |||\mathcal{W}'\mathcal{V}\Omega_v/T|||_\infty. \tag{17}$$

The error bound of $\bar{\Omega}_v$ again relies on $\|S_v - \Sigma_v^\star\|_\infty$. In an analogous way, for the actual alternate update,

$$(\widehat{B}_{\mathrm{aug}}^{(k)}, \widehat{C}_{\mathrm{aug}}^{(k)}) = \arg\min_{B,C} \Big\{ \tfrac{1}{T}\Big|\Big|\Big|\mathcal{Z}^T\big[\widehat{\Omega}_v^{(k-1)}\big]^{1/2} - \mathfrak{W}_{\widehat{\Omega}_v^{(k-1)}}(B_{\mathrm{aug}} + C_{\mathrm{aug}})\Big|\Big|\Big|_F^2 + \lambda_B|||B|||_* + \lambda_C\|C\|_1 \Big\},$$
$$\widehat{\Omega}_v^{(k)} := \arg\min_{\Omega_v} \big\{ \log\det\Omega_v - \mathrm{trace}\big(\widehat{S}_v^{(k)}\Omega_v\big) + \rho_v\|\Omega_v\|_{1,\mathrm{off}} \big\},$$

and the error bound of $(\widehat{B}, \widehat{C}, \widehat{\Omega}_v)$ defined in (10) depends on the properties of $\mathfrak{W}_{\widehat{\Omega}_v^{(k)}}$, $|||\mathcal{W}'\mathcal{V}\Omega_v^{(k)}/T|||_{op}$ and $|||\mathcal{W}'\mathcal{V}\Omega_v^{(k)}/T|||_\infty$ for all k. Specifically, when Ω_v and (B, C) (in (15) and (17), resp.) are substituted by estimated quantities, we prove that the RSC condition and bounds hold with high probability for random realizations of $\mathcal{Z}$, $\mathcal{X}$ and $\mathcal{V}$, for *all iterations* k, which then establishes the consistency properties of $(\widehat{B}, \widehat{C})$ and $\widehat{\Omega}_v$.

3.2 Consistency results for the Maximum Likelihood estimators.

Theorems 5 and 6 below give the error bounds for the estimators in (9) and (10) obtained through Algorithms 1 and 2, using random realizations coming from the stable VAR system defined in (1). As previously mentioned, to establish error bounds for both the transition matrices and the inverse covariance matrix obtained from alternating updates, we need to take into account that the quantities associated with the RSC condition and the deviation bound condition are based on *estimated quantities* obtained from the previous iteration. On the other hand, the sources of randomness contained in the observed data are fixed, hence errors from observed data stop accumulating once all sources of randomness are considered after a few iterations, which govern both the leading term of the error bounds and the probability for the bounds to hold.

Specifically, using the same notation as defined in Section 3.1, we obtain the error bounds of the estimated transition matrices and inverse covariance matrices iteratively, building upon that for all iterations k:

(1) $\widehat{\Gamma}_X^{(k)}$ or the operator $\mathfrak{W}_{\widehat{\Omega}_v^{(k)}}$ satisfies the RSC condition;
(2) deviation bounds hold for $\|\mathcal{X}'\mathcal{U}\widehat{\Omega}_u^{(k)}/T\|_\infty$, $\|\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(k)}/T\|_\infty$, and $|||\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(k)}/T|||_{\mathrm{op}}$;
(3) a good concentration given by $\|\widehat{S}_u^{(k)} - \Sigma_u^\star\|_\infty$ and $\|\widehat{S}_v^{(k)} - \Sigma_v^\star\|_\infty$.

We keep track of how the bounds change in each iteration until convergence, by properly controlling the norms and track the rate of the error bound that depends on p_1, p_2 and T, and reach the conclusion that the error bounds *hold uniformly for all iterations*, for the estimates of both the transition matrices A, B and C and the inverse covariance matrices Ω_u and Ω_v.

Theorem 5 *Consider the stable Gaussian VAR process defined in (1) in which $A^\star$ is assumed to be $s_A^\star$-sparse. Further, assume the following:*

C.1 The incoherence condition holds for $\Omega_u^\star$.

C.2 $\Omega_u^\star$ is diagonally dominant.

C.3 The maximum node degree of $\Omega_u^\star$ satisfies $d^{\max}_{\Omega_u^\star} = o(p_1)$.

Then, for random realizations of $\{X_t\}$ and $\{U_t\}$, and the sequence $\{\widehat{A}^{(k)}, \widehat{\Omega}_u^{(k)}\}_k$ returned by Algorithm 1 outlined in Section 2.1, there exist constants $c_1, c_2, \tilde{c}_1, \tilde{c}_2 > 0, \tau > 0$ such that for sample size $T \gtrsim \max\{(d^{\max}_{\Omega_u^\star})^2, s_A^\star\}\log p_1$, with probability at least

$$1 - c_1\exp(-c_2 T) - \tilde{c}_1\exp(-\tilde{c}_2\log p_1) - \exp(-\tau\log p_1),$$

the following hold for all $k \geq 1$ for some $C_0, C_0' > 0$ that are functions of the upper and lower extremes $\mathcal{M}(f_X), \mathfrak{m}(f_X)$ of the spectrum $f_X(\theta)$ and do not depend on p_1, T or k:

(i) $\widehat{\Gamma}_X^{(k)}$ satisfies the RSC condition;

(ii) $\|\mathcal{X}'\mathcal{U}\widehat{\Omega}_u^{(k)}/T\|_\infty \leq C_0\sqrt{\frac{\log p_1}{T}}$;

(iii) $\|\widehat{S}_u^{(k)} - \Sigma_u^\star\|_\infty \le C_0' \sqrt{\frac{\log p_1}{T}}$.

As a consequence, the following bounds hold uniformly for all iterations $k \ge 1$*:*

$$\left\|\left|\widehat{A}^{(k)} - A^\star\right|\right\|_F = O\Big(\sqrt{\tfrac{s_A^\star \log p_1}{T}}\Big), \qquad \left\|\left|\widehat{\Omega}_u^{(k)} - \Omega_u^\star\right|\right\|_F = O\Big(\sqrt{\tfrac{(s_{\Omega_u}^\star + p_1)\log p_1}{T}}\Big).$$

It should be noted that the above result establishes the *consistency for the ML estimates* of the model presented in Basu and Michailidis (2015).

Theorem 6 *Consider the stable Gaussian VAR system defined in (1) in which* $B^\star$ *is assumed to be low rank with rank* $r_B^\star$ *and* $C^\star$ *is assumed to be* $s_C^\star$*-sparse. Further, assume the following*

C.1 The incoherence condition holds for $\Omega_v^\star$.

C.2 $\Omega_v^\star$ *is diagonally dominant.*

C.3 The maximum node degree of $\Omega_v^\star$ *satisfies* $d^{\max}_{\Omega_v^\star} = o(p_2)$.

Then, for random realizations of $\{X_t\}$, $\{Z_t\}$ *and* $\{V_t\}$*, and the sequence* $\{(\widehat{B}^{(k)}, \widehat{C}^{(k)}), \widehat{\Omega}_v^{(k)}\}_k$ *returned by Algorithm 2 outlined in Section 2.1, there exist constants* $\{c_i, \tilde{c}_i\}, i = (0, 1, 2)$ *and* $\tau > 0$ *such that for sample size* $T \gtrsim (d^{\max}_{\Omega_v^\star})^2(p_1 + 2p_2)$*, with probability at least*

$$1 - c_0 \exp\{-\tilde{c}_0(p_1+p_2)\} - c_1 \exp\{-\tilde{c}_1(p_1+2p_2)\} - c_2 \exp\{-\tilde{c}_2 \log[p_2(p_1+p_2)]\} - \exp\{-\tau \log p_2\},$$

the following hold for all $k \ge 1$ *for* $C_0, C_0', C_0'' > 0$ *that are functions of the upper and lower extremes* $\mathcal{M}(f_W), \mathfrak{m}(f_W)$ *of the spectrum* $f_W(\theta)$ *and of the upper extreme* $\mathcal{M}(f_{W,V})$ *of the cross-spectrum* $f_{W,V}(\theta)$ *and do not depend on* p_1, p_2 *or* T*:*

(i) $\widehat{\Gamma}_W^{(k)}$ *satisfies the RSC condition;*

(ii) $\|\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(k)}/T\|_\infty \le C_0 \sqrt{\frac{(p_1+p_2)+p_2}{T}}$ *and* $\|\!|\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(k)}/T|\!\|_{op} \le C_0' \sqrt{\frac{(p_1+p_2)+p_2}{T}}$;

(iii) $\|\widehat{S}_v^{(k)} - \Sigma_v^\star\|_\infty \le C_0'' \sqrt{\frac{(p_1+p_2)+p_2}{T}}$.

As a consequence, the following bounds hold uniformly for all iterations $k \ge 1$*:*

$$\left\|\left|\widehat{B}^{(k)} - B^\star\right|\right\|_F^2 + \left\|\left|\widehat{C}^{(k)} - C^\star\right|\right\|_F^2 = O\Big(\tfrac{\max\{r_B^\star, s_C^\star\}(p_1+2p_2)}{T}\Big), \quad \left\|\left|\widehat{\Omega}_v^{(k)} - \Omega_v^\star\right|\right\|_F = O\Big(\sqrt{\tfrac{(s_{\Omega_v}^\star + p_2)(p_1+2p_2)}{T}}\Big).$$

Remark 7 *It is worth pointing out that the initializers* $\widehat{A}^{(0)}$ *and* $(\widehat{B}^{(0)}, \widehat{C}^{(0)})$ *are slightly different from those obtained in successive iterations, as they come from the penalized least square formulation where the inverse covariance matrices are temporarily assumed diagonal. Consistency results for these initializers under deterministic realizations are established in Theorems 14 and 15 (see Appendix A), and the corresponding conditions are later verified for random realizations in Lemmas 16 to 19 (see Appendix B). These theorems and lemmas serve as the stepping stone toward the proofs of Theorems 5 and 6.*

Further, the constants C_0, C_0', C_0'' *reflect both the temporal dependence among* X_t *and* Z_t *blocks, as well as the cross-sectional dependence within and across the two blocks.*

3.3 The effect of temporal and cross-dependence on the established bounds.

We conclude this section with a discussion on the error bounds of the estimators that provides additional insight into the impact of temporal and cross dependence within and between the blocks; specifically, how the exact bounds depend on the underlying processes through their spectra when explicitly taking into consideration the triangular structure of the joint transition matrix.

First, we introduce additional notations needed in subsequent technical developments. The definition of the spectral densities and the cross-spectrum are the same as previously defined in Section 2 and their upper and lower extremes are defined in Section 3.1. For $\{X_t\}$ defined in (1), let $\mathcal{A}(\theta) = \mathrm{I}_{p_1} - A\theta$ denote the characteristic matrix-valued polynomial of $\{X_t\}$ and $\mathcal{A}^*(\theta)$ denote its conjugate. We further define its upper and lower extremes by:

$$\mu_{\max}(\mathcal{A}) = \max_{|\theta|=1} \Lambda_{\max}\left(\mathcal{A}^*(\theta)\mathcal{A}(\theta)\right), \quad \mu_{\min}(\mathcal{A}) = \min_{|\theta|=1} \Lambda_{\min}\left(\mathcal{A}^*(\theta)\mathcal{A}(\theta)\right).$$

The same set of quantities for the joint process $\{W_t = (X_t', Z_t')'\}$ are analogously defined, that is,

$$\mathcal{G}(\theta) = \mathrm{I}_{p_1+p_2} - G\theta, \quad \mu_{\max}(\mathcal{G}) = \max_{|\theta|=1} \Lambda_{\max}\left(\mathcal{G}^*(\theta)\mathcal{G}(\theta)\right), \quad \mu_{\min}(\mathcal{G}) = \min_{|\theta|=1} \Lambda_{\min}\left(\mathcal{G}^*(\theta)\mathcal{G}(\theta)\right).$$

Using the result in Theorem 6 as an example, we show how the error bound depends on the underlying processes $\{(X_t', Z_t')'\}$. Specifically, we note that the bounds for $(\widehat{B}^{(k)}, \widehat{C}^{(k)})$ can be equivalently written as

$$\|\widehat{B}^{(k)} - B^\star\|_F^2 + \|\widehat{C}^{(k)} - C^\star\|_F^2 \le \bar{C}\Big(\tfrac{\max\{r_B^\star, s_C^\star\}(p_1+2p_2)}{T}\Big),$$

which holds for all k and some constant $\bar{C}$ that does not depend on p_1, p_2 or T. Specifically, by Theorem 15, Lemmas 18 and 19,

$$C_0 \propto \left[\mathcal{M}(f_W) + \tfrac{1}{2\pi}\Lambda_{\max}(\Sigma_v) + \mathcal{M}(f_{W,V})\right] / \mathfrak{m}(f_W).$$

This indicates that the exact error bound depends on $\mathfrak{m}(f_W), \mathcal{M}(f_W)$ and $\mathcal{M}(f_{W,V})$. Next, we provide bounds on these quantities. The joint process W_t as we have noted in (2), is a VAR(1) process with characteristic polynomial $\mathcal{G}(\theta)$ and spectral density $f_W(\theta)$. The bounds for $\mathfrak{m}(f_W)$ and $\mathcal{M}(f_W)$ are given by Basu and Michailidis (2015, Proposition 2.1), that is,

$$\mathfrak{m}(f_W) \ge \frac{\min\{\Lambda_{\min}(\Sigma_u), \Lambda_{\min}(\Sigma_v)\}}{(2\pi)\mu_{\max}(\mathcal{G})} \quad \text{and} \quad \mathcal{M}(f_W) \le \frac{\max\{\Lambda_{\max}(\Sigma_u), \Lambda_{\max}(\Sigma_v)\}}{(2\pi)\mu_{\min}(\mathcal{G})}. \quad (18)$$

Consider the bound for $\mathcal{M}(f_{W,V})$. First, we note that $\{V_t\}$ is a sub-process of the joint error process $\{\varepsilon_t\}$, where $\varepsilon_t = (U_t', V_t')'$. Then, by Lemma 24,

$$\mathcal{M}(f_{W,V}) \le \mathcal{M}(f_{W,\varepsilon}) \le \mathcal{M}(f_W)\mu_{\max}(\mathcal{G}),$$

where the second inequality follows from Basu and Michailidis (2015, Proof of Proposition 2.4).

What are left to be bounded are $\mu_{\min}(\mathcal{G})$ and $\mu_{\max}(\mathcal{G})$. By Proposition 2.2 in Basu and Michailidis (2015), these two quantities are bounded by:

$$\mu_{\max}(\mathcal{G}) \leq \left[1 + \frac{|||G|||_\infty + |||G|||_1}{2}\right]^2 \tag{19}$$

and

$$\mu_{\min}(\mathcal{G}) \geq (1 - \rho(G))^2 \cdot |||P|||_{op}^{-2} \cdot |||P^{-1}|||_{op}^{-2},$$

where $G = P\Lambda_G P^{-1}$ with Λ_G being a diagonal matrix consisting of the eigenvalues of G. Since $|||P^{-1}|||_{op} \geq |||P|||_{op}^{-1}$, it follows that

$$|||P|||_{op}^{-2} \cdot |||P^{-1}|||_{op}^{-2} \geq |||P^{-1}|||_{op}^{2} \cdot |||P^{-1}|||_{op}^{-2} = 1,$$

and therefore

$$\mu_{\min}(\mathcal{G}) \geq (1 - \max\{\rho(A), \rho(C)\})^2 . \tag{20}$$

Remark 8 The impact of the system's lower-triangular structure on the established bounds. *Consider the bounds in (19) and (20). An upper bound of $\mu_{\max}(\mathcal{G})$ depends on $|||G|||_\infty$ and $|||G|||_1$, whereas a lower bound of $\mu_{\min}(\mathcal{G})$ involves only the spectral radius of G. Combined with (18), this suggests that the lower extreme of the spectral density is associated with the average of the maximum weighted in-degree and out-degree of the system, whereas the upper extreme is associated with the stability condition: the* less the system is intra- and inter-connected, *the* tighter the bound *for the lower extreme will be; similarly, the* more stable *(exhibits smaller temporal dependence) the system is, the* tighter the bound *for the upper extreme will be. Finally, we note that an upper bound for $(|||G|||_\infty + |||G|||_1)$ is given by*

$$\max\{|||A|||_\infty + |||B|||_\infty, |||C|||_\infty\} + \max\{|||A|||_1, |||B|||_1 + |||C|||_1\}.$$

The presence of $|||B|||_\infty$ and $|||B|||_1$ depicts the role of the inter-connectedness between $\{X_t\}$ and $\{Z_t\}$ on the lower extreme of the spectrum, which is associated with the overall curvature of the joint process.

The impact of the system's lower-triangular structure on the system capacity. *With G being a lower-triangular matrix, we only require $\rho(A) < 1$ and $\rho(C) < 1$ to ensure the stability of the system. This enables the system to have "larger capacity" (can accommodate more cross-dependence within each block), since the two sparse components A and C can exhibit larger average weighted in- and out-degrees compared with a system where G does not possess such triangular structure. In the case where G is a complete matrix, one deals with a $(p_1 + p_2)$-dimensional VAR system and $\rho(G) < 1$ is required to ensure its stability. As a consequence, the average weighted in- and out-degree requirements for each time series become more restrictive.*

4. Testing Group Granger-Causality.

In this section, we develop a procedure for testing the hypothesis $H_0 : B = 0$. Note that without the presence of B, the blocks X_t and Z_t in the model become *decoupled* and can be treated as two separate VAR models, whereas with a nonzero B, the group of variables in

Z_t is collectively "Granger-caused" by those in X_t. Moreover, since we are testing whether or not the entire block of B is zero, we do not need to rely on the exact distribution of its individual entries, but rather on the properly measured correlation between the responses and the covariates. To facilitate presentation of the testing procedure, we illustrate the proposed framework via a simpler model setting with $Y_t = \Pi X_t + \epsilon_t$ and testing whether $\Pi = 0$; subsequently, we translate the results to the actual setting of interest, namely, whether or not $B = 0$ in the model $Z_t = BX_{t-1} + CZ_{t-1} + V_t$.

The testing procedure focuses on the following sequence of tests for the rank of B:

$$H_0 : \text{rank}(B) \leq r, \qquad \text{for an arbitrary} \quad r < \min(p_1, p_2). \tag{21}$$

Note that the hypothesis of interest, $B = 0$ corresponds to the special case with $r = 0$. To test for it, we develop a procedure associated with *canonical correlations*, which leverages ideas present in the literature (see Anderson, 2002).

As mentioned above, we consider a simpler setting similar to that in Anderson (2002), given by

$$Y_t = \Pi X_t + \epsilon_t,$$

where $Y_t \in \mathbb{R}^{p_2}$, $X \in \mathbb{R}^{p_1}$ and ϵ_t is independent of X_t. At the population level, let

$$\mathbb{E}Y_t Y_t' = \Sigma_Y, \qquad \mathbb{E}X_t X_t' = \Sigma_X, \qquad \mathbb{E}Y_t X_t' = \Sigma_{YX} = \Sigma_{XY}'.$$

The population canonical correlations between Y_t and X_t are the roots of

$$\begin{vmatrix} -\rho\Sigma_Y & \Sigma_{YX} \\ \Sigma_{XY} & -\rho\Sigma_X \end{vmatrix} = 0,$$

i.e., the nonnegative solutions to

$$|\Sigma_{YX}\Sigma_X^{-1}\Sigma_{XY} - \rho^2\Sigma_Y| = 0, \tag{22}$$

with ρ being the unknown. By the results in Anderson (2002), the number of positive solutions to (22) is equal to the rank of Π, and indicates the "degree of dependency" between processes Y_t and X_t. This suggests that if $\text{rank}(\Pi) \leq r < p$, we would expect $\sum_{k=r+1}^{p} \lambda_k$ to be small, where the λ's solve the eigen-equation

$$|S_{YX}S_X^{-1}S_{XY} - \lambda S_Y| = 0, \qquad \text{with} \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p,$$

and S_X, S_{XY} and S_Y are the sample counterparts corresponding to Σ_X, Σ_{XY} and Σ_Y, respectively.

With this background, we switch to our model setting given by

$$Z_t = BX_{t-1} + CZ_{t-1} + V_t, \tag{23}$$

where V_t is assumed to be independent of X_{t-1} and Z_{t-1}, B encodes the canonical correlation between Z_t and X_{t-1}, conditional on Z_{t-1}. We use the same notation as in Section 3; that

is, let $\Gamma_X(h) = \mathbb{E}X_tX_{t+h}'$, $\Gamma_Z(h) = \mathbb{E}Z_tZ_{t+h}'$, and $\Gamma_{X,Z}(h) = \mathbb{E}X_tZ_{t+h}'$, with (h) omitted whenever $h = 0$. At the population level, under the Gaussian assumption,

$$\begin{bmatrix} Z_t \\ X_{t-1} \\ Z_{t-1} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \Gamma_Z & \Gamma_{X,Z}'(1) & \Gamma_Z(1) \\ \Gamma_{X,Z}(1) & \Gamma_X & \Gamma_{X,Z} \\ \Gamma_Z'(1) & \Gamma_{X,Z}' & \Gamma_Z \end{bmatrix} \right),$$

which suggests that conditionally,

$$Z_t \big| Z_{t-1} \sim \mathcal{N}\left(\Gamma_Z(1)\Gamma_Z^{-1}Z_{t-1}, \Sigma_{00}\right) \qquad \text{and} \qquad X_{t-1}\big|Z_{t-1} \sim \mathcal{N}\left(\Gamma_{X,Z}\Gamma_Z^{-1}Z_{t-1}, \Sigma_{11}\right),$$

where

$$\Sigma_{00} := \Gamma_Z - \Gamma_Z(1)\Gamma_Z^{-1}\Gamma_Z'(1) \qquad \text{and} \qquad \Sigma_{11} := \Gamma_X - \Gamma_{X,Z}\Gamma_Z^{-1}\Gamma_{X,Z}'. \tag{24}$$

Then, we have that jointly

$$\begin{bmatrix} Z_t \\ X_{t-1} \end{bmatrix} \Big| Z_{t-1} \sim \mathcal{N}\left(\begin{bmatrix} \Gamma_Z(1) \\ \Gamma_{X,Z} \end{bmatrix} \Gamma_Z^{-1}Z_{t-1}, \begin{bmatrix} \Gamma_Z & \Gamma_{XZ}'(1) \\ \Gamma_{XZ}(1) & \Gamma_Z \end{bmatrix} - \begin{bmatrix} \Gamma_Z(1) \\ \Gamma_{XZ} \end{bmatrix} \Gamma_Z^{-1} \begin{bmatrix} \Gamma_Z'(1) & \Gamma_{ZX} \end{bmatrix} \right),$$

so the partial covariance matrix between Z_t and X_{t-1} conditional on Z_{t-1} is given by

$$\Sigma_{10} = \Sigma_{01}' := \Gamma_{X,Z}(1) - \Gamma_Z(1)\Gamma_Z^{-1}\Gamma_{X,Z}. \tag{25}$$

The population canonical correlations between Z_t and X_{t-1} conditional on Z_{t-1} are the non-negative roots of

$$\left|\Sigma_{01}\Sigma_{11}^{-1}\Sigma_{10} - \rho^2\Sigma_{00}\right| = 0,$$

and the number of positive solutions corresponds to the rank of B; see Anderson (1951) for a discussion in which the author is interested in estimating and testing linear restrictions on regression coefficients. Therefore, to test $\text{rank}(B) \leq r$, it is appropriate to examine the behavior of $\Psi_r := \sum_{k=r+1}^{\min(p_1,p_2)} \phi_k$, where ϕ's are ordered non-increasing solutions to

$$|S_{01}S_{11}^{-1}S_{10} - \phi S_{00}| = 0, \tag{26}$$

and S_{01}, S_{11} and S_{00} are the empirical surrogates for the population quantities Σ_{01}, Σ_{11} and Σ_{00}. For subsequent developments, we make the very mild assumption that $p_1 < T$ and $p_2 < T$ so that $\mathcal{Z}'\mathcal{Z}$ is invertible.

Proposition 9 gives the tail behavior of the eigenvalues and Corollary 11 gives the testing procedure for block "Granger-causality" as a direct consequence.

Proposition 9 *Consider the model setup given in (23), where $B \in \mathbb{R}^{p_2 \times p_1}$. Further, assume all positive eigenvalues μ of the following eigen-equation are of algebraic multiplicity one:*

$$\left|\Sigma_{01}\Sigma_{11}^{-1}\Sigma_{10} - \mu\Sigma_{00}\right| = 0, \tag{27}$$

where Σ_{00}, Σ_{11} and Σ_{01} are given in (24) and (25). The test statistic for testing

$$H_0 : \text{rank}(B) \leq r, \qquad \textit{for an arbitrary} \quad r < \min(p_1, p_2),$$

is given by

$$\Psi_r := \sum_{k=r+1}^{\min(p_1,p_2)} \phi_k,$$

where ϕ_k's are ordered decreasing solutions to the eigen-equation $|S_{01}S_{11}^{-1}S_{10} - \phi S_{00}| = 0$ where

$$S_{11} = \tfrac{1}{T}\mathcal{X}'(I - P_z)\mathcal{X}, \quad S_{00} = \tfrac{1}{T}\left(\mathcal{Z}^T\right)'(I - P_z)\left(\mathcal{Z}^T\right), \quad S_{10} = S_{01}' = \tfrac{1}{T}\mathcal{X}'(I - P_z)\left(\mathcal{Z}^T\right),$$

and $P_z = \mathcal{Z}(\mathcal{Z}'\mathcal{Z})^{-1}\mathcal{Z}'$. Moreover, the limiting behavior of Ψ_r is given by

$$T\Psi_r \sim \chi^2_{(p_1-r)(p_2-r)}.$$

Remark 10 *We provide a short comment on the assumption that the positive solutions to (27) have algebraic multiplicity one in Proposition 9. This assumption is imposed on the eigen-equation associated with population quantities, to exclude the case where a positive root has algebraic multiplicity greater than one and its geometric multiplicity does not match the algebraic one, and hence we would fail to obtain r mutually independent canonical variates and the rank-r structure becomes degenerate. With the imposed assumption which is common in the canonical correlation analysis literature (e.g. Anderson, 2002), such a scenario is automatically excluded. Specifically, this condition is not stringent, as for ϕ's that are solutions to the eigen-equation associated with sample quantities, the distinctiveness amongst roots is satisfied with probability 1 (see Hsu, 1941b, Proof of Lemma 3).*

Corollary 11 (Testing group Granger-causality) *Under the model setup in (23), the test statistic for testing $B = 0$ is given by*

$$\Psi_0 := \sum_{k=1}^{\min(p_1,p_2)} \phi_k,$$

with ϕ_k being the ordered decreasing solutions of

$$\left|S_{01}\left[\text{diag}(S_{11})\right]^{-1}S_{10} - \phi S_{00}\right| = 0.$$

Asymptotically, $T\Psi_0 \sim \chi^2_{p_1p_2}$. To conduct a level α test, we reject the null hypothesis $H_0 : B = 0$ if

$$\Psi_0 > \frac{1}{T}\chi^2_{p_1p_2}(\alpha),$$

where $\chi^2_d(\alpha)$ is the upper α quantile of the χ^2 distribution with d degrees of freedom.

Remark 12 *Corollary 11 is a special case of Proposition 9 with the null hypothesis being $H_0 : r = 0$, which corresponds to the Granger-causality test. Under this particular setting, we are able to take the inverse with respect to $diag(S_{11})$, yet maintain the same asymptotic distribution due to the fact that $S_{01} = S_{10} = 0$ under the null hypothesis $B = 0$. This enables us to perform the test even with $p_1 > T$.*

The above testing procedure takes advantage of the fact that when $B = 0$, the canonical correlations among the partial regression residuals after removing the effect of Z_{t-1} are very close to zero. However, the test may not be as powerful under a sparse alternative, i.e., $H_A : B$ is sparse. In Supplementary Material **??**, we present a testing procedure that specifically takes into consideration the fact that the alternative hypothesis is sparse, and the corresponding performance evaluation is shown in Section 5.2 under this setting.

5. Performance Evaluation.

Next, we present the results of numerical studies to evaluate the performance of the developed ML estimates (Section 2.1) of the model parameters, as well as that of the testing procedure (Section 4).

5.1 Simulation results for the estimation procedure.

A number of factors may potentially influence the performance of the estimation procedure; in particular, the model dimension p_1 and p_2, the sample size T, the rank of $B^\star$ and the sparsity level of $A^\star$ and $C^\star$, as well as the spectral radius of $A^\star$ and $C^\star$. Hence, we consider several settings where these parameters vary.

For all settings, the data $\{x_t\}_1^T$ and $\{z_t\}_1^T$ are generated according to the model

$$x_t = A^\star x_{t-1} + u_t,$$
$$z_t = B^\star x_{t-1} + C^\star z_{t-1} + v_t.$$

For the sparse components, each entry in $A^\star$ and $C^\star$ is nonzero with probability $2/p_1$ and $1/p_2$ respectively, and the nonzero entries are generated from $\mathsf{Unif}([-2.5,-1.5]\cup[1.5,2.5])$, then scaled down so that the spectral radii $\rho(A)$ and $\rho(C)$ satisfy the stability condition. For the low rank component, each entry in $B^\star$ is generated from $\mathsf{Unif}(-10,10)$, followed by singular value thresholding, so that $\mathrm{rank}(B^\star)$ conforms with the model setup. For the contemporaneous dependence encoded by $\Omega_u^\star$ and $\Omega_v^\star$, both matrices are generated according to an Erdös-Rényi random graph, with sparsity being 0.05 and condition number being 3.

Table 1 depicts the values of model parameters under different model settings. Specifically, we consider three major settings in which the size of the system, the rank of the cross-dependence component, and the stability of the system vary. The sample size is fixed at $T = 200$ unless otherwise specified. Additional settings examined (not reported due to space considerations) are consistent with the main conclusions presented next.

We use sensitivity (SEN), specificity (SPC) and relative error in Frobenius norm (Error) as criteria to evaluate the performance of the estimates of transition matrices A, B and C. Tuning parameters are chosen based on BIC. Since the exact contemporaneous dependence is not of primary concern, we omit the numerical results for $\widehat{\Omega}_u$ and $\widehat{\Omega}_v$.

$$\mathrm{SEN} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \quad \mathrm{SPE} = \frac{\mathrm{TN}}{\mathrm{FP}+\mathrm{TN}}, \quad \mathrm{Error} = \frac{|||\mathrm{Est.} - \mathrm{Truth}|||_F}{|||\mathrm{Truth}|||_F}.$$

Table 2 illustrates the performance for each of the parameters under different simulation settings considered. The results are based on an average of 100 replications and their

Table 1: Model parameters under different model settings

		model parameters				
		p_1	p_2	rank(B^*)	ρ_A	ρ_C
model dimension	A.1	50	20	5	0.5	0.5
	A.2	100	50	5	0.5	0.5
	A.3	200	50	5	0.5	0.5
	A.4	50	100	5	0.5	0.5
rank	B.1	100	50	10	0.5	0.5
	B.2	100	50	20	0.5	0.5
spectral radius	C.1	50	20	5	0.8	0.5
	C.2	50	20	5	0.5	0.8
	C.3	50	20	5	0.8	0.8

Table 2: Performance evaluation of $\widehat{A}$, $\widehat{B}$ and $\widehat{C}$ under different model settings.

	performance of $\widehat{A}$			performance of $\widehat{B}$		performance of $\widehat{C}$		
	SEN	SPC	Error	rank($\widehat{B}$)	Error	SEN	SPC	Error
A.1	0.98(0.014)	0.99(0.004)	0.34(0.032)	5.2(0.42)	0.11(0.008)	1.00(0.000)	0.97(0.008)	0.15(0.074)
A.2	0.97(0.014)	0.99(0.001)	0.38(0.015)	5.2(0.42)	0.31(0.011)	0.97(0.008)	0.97(0.004)	0.28(0.033)
A.3	0.99(0.005)	0.96(0.002)	0.87(0.011)	5.8(0.92)	0.54(0.022)	0.98(0.000)	0.92(0.009)	0.28(0.028)
A.4	0.96(0.0261)	0.99(0.002)	0.36(0.034)	5.2(0.42)	0.32(0.012)	0.95(0.009)	0.98(0.001)	0.37(0.010)
B.1	0.97(0.008)	0.99(0.001)	0.37(0.017)	11.4(1.17)	0.15(0.008)	1.00(0.000)	0.99(0.001)	0.09(0.021)
B.2	0.98(0.008)	0.99(0.001)	0.38(0.016)	21.2(0.91)	0.12(0.006)	1.00(0.000)	0.99(0.001)	0.08(0.018)
C.1	1.00(0.000)	0.97(0.005)	0.25(0.015)	5.6(0.52)	0.23(0.006)	1.00(0.000)	0.92(0.021)	0.11(0.072)
C.2	0.99(0.007)	0.95(0.004)	0.45(0.022)	5.0(0.00)	0.31(0.014)	1.00(0.000)	0.92(0.019)	0.04(0.011)
C.3	1.00(0.000)	0.96(0.004)	0.18(0.013)	6.7(1.16)	0.19(0.011)	1.00(0.000)	0.87(0.029)	0.14(0.067)
C.3'	1.00(0.000)	0.99(0.002)	0.13(0.016)	5.2(0.42)	0.23(0.005)	1.00(0.000)	0.90(0.021)	0.06(0.023)

standard deviations are given in parentheses. Overall, the results are highly satisfactory and all the parameters are estimated with a high degree of accuracy. Further, all estimates were obtained in less than 20 iterations, thus indicating that the estimation procedure is numerically stable. As expected, when the the spectral radii of A and C increase thus leading to less stable $\{X_t\}$ and $\{Z_t\}$ processes, a larger sample size is required for the estimation procedure to match the performance of the setting with same parameters but smaller $\rho(A)$ and $\rho(C)$. This is illustrated in row C.3' of Table 2, where the sample size is increased to $T = 500$, which outperforms the results in row C.3 in which $T = 200$ and broadly matches that of row A.1.

Since in some application settings the data may deviate from the posited Gaussian assumption, we further investigate the robustness of the algorithm in the presence of heavier than Gaussian distributions in the Supplementary Material **??**. Further, a brief comparison of the ML estimates with their two-step (non-iterative) counterparts is given in the Supplementary Material **??** , that illustrates the numerical benefits of the proposed procedure.

Lastly, we examine the performance with respect to one-step-ahead forecasting. Recall that VAR models are widely used for forecasting purposes in many application areas

(Lütkepohl, 2005). The performance metric is given by the relative error as measured by the ℓ_2 norm of the out-of-sample points x_{T+1} and z_{T+1}, where the predicted values are given by $\widehat{x}_{T+1} = \widehat{A}x_T$ and $\widehat{z}_{T+1} = \widehat{B}x_T + \widehat{C}z_T$, respectively. It is worth noting that both $\{X_t\}$ and $\{Z_t\}$ are mean-zero processes. However, since the transition matrix of $\{X_t\}$ is subject to the spectral radius constraints to ensure the stability of the corresponding process, the magnitude of the realized value $\{x_t\}$ is small; whereas for $\{Z_t\}$, since no constraints are imposed on the B coefficient matrix that encodes the inter-dependence, z_t's has the capacity of having relative large values in magnitude. Consequently, the relative error of $\widehat{x}_{T+1}$ is significantly larger than that of $\widehat{z}_{T+1}$, partially due to the small total magnitude of the denominator.

The results show that an increase in the spectral radius (keeping the other structural parameters fixed) leads to a decrease of the relative error, since future observations become more strongly correlated over time. On the other hand, an increase in dimension leads to a deterioration in forecasting, since the available sample size impacts the quality of the parameter estimates. Finally, an increase in the rank of the B matrix is beneficial for forecasting, since it plays a stronger role in the system's temporal evolution.

Table 3: One-step-ahead relative forecasting error.

		$\frac{\|\widehat{x}_{T+1}-x_{T+1}\|_2}{\|x_{T+1}\|_2}$	$\frac{\|\widehat{z}_{T+1}-z_{T+1}\|_2}{\|z_{T+1}\|_2}$
baseline	A.1	0.89(0.066)	0.23(0.075)
spectral radius	C.1	0.62(0.100)	0.10(0.035)
	C.2	0.93(0.062)	0.17(0.059)
	C.3	0.68(0.096)	0.10(0.045)
rank	B.1	0.92(0.044)	0.14(0.038)
	B.2	0.94(0.042)	0.14(0.025)
dimension	A.2	0.87(0.051)	0.24(0.073)
	A.3	0.96(0.040)	0.44(0.139)
	A.4	0.89(0.059)	0.274(0.068)

5.2 Simulation results for the block Granger-causality test.

Next, we illustrate the empirical performance of the testing procedure introduced in Section 4, together with the one specifically tailored to a sparse alternative (described in detail in the Supplementary Material **??**) with the null hypothesis being $B^\star = 0$ and the alternative being $B^\star \neq 0$, either low rank or sparse. Specifically, when the alternative hypothesis is true and has a low-rank structure, we use the general testing procedure proposed in Section 4, whereas when the alternative is true and sparse, we use the testing procedure presented in Supplementary Material **??**. We focus on evaluating the type I error (empirical false rejection rate) when $B^\star = 0$, as well as the power of the test when $B^\star$ has nonzero entries.

For both testing procedures, the transition matrix $A^\star$ is generated with each entry being nonzero with probability $2/p_1$, and the nonzeros are generated from $\mathsf{Unif}([-2.5,-1.5]\cup[1.5,2.5])$, then further scaled down so that $\rho(A^\star) = 0.5$. For transition matrix $C^\star$, each entry is nonzero with probability $1/p_2$, and the nonzeros are generated from $\mathsf{Unif}([-2.5,-1.5]\cup[1.5,2.5])$,

then further scaled down so that $\rho(C^\star) = 0.5$ or 0.8, depending on the simulation setting. Finally, we only consider the case where v_t and u_t have diagonal covariance matrices.

We use sub-sampling as in Politis et al. (1999) with the number of subsamples set to 3,000; an alternative would have been a block bootstrap procedure (e.g., Hall, 1985). Note that the length of the subsamples varies across simulation settings in order to gain insight on how sample size impacts the type I error or the power of the test.

Low-rank testing. To evaluate the type I error control and the power of the test, we primarily consider the case where $\mathrm{rank}(B^\star) = 0$, with the data alternatively generated based on $\mathrm{rank}(B^\star) = 1$. We test the hypothesis $H_0 : \mathrm{rank}(B) = 0$ and tabulate the empirical proportion of falsely rejecting H_0 when $\mathrm{rank}(B^*) = 0$ (type I error) and the probability that we reject H_0 when $\mathrm{rank}(B^*) = 1$ (power). In addition, we also show how the testing procedure performs when the underlying $B^\star$ has rank $r \geq 0$. In particular, when $\mathrm{rank}(B^\star) = r^\star$, the type I error of the test corresponds to the empirical proportion of rejections of the null hypothesis $H_0 : r \leq r^\star$, while the power of the test to the empirical proportion of rejections of the null hypothesis set to $H_0 : r \leq (r^\star - 1)$. The latter resembles the sequential test in Johansen (1988).

Empirically, we expect that when $B^\star = 0$, the value of the proposed test statistic mostly falls below the cut-off value (upper α quantile), while when $\mathrm{rank}(B^\star) = 1$, the value of the proposed test statistic mostly falls beyond the critical value $\chi^2(\alpha)_{p_1p_2}/T$ with T being the sample size, hence leading to a detection. Table 4 gives the type I error of the test when setting $\alpha = 0.1, 0.05, 0.1$, and the power of the test using the upper 0.01 quantile of the reference distribution as the cut-off, for different combinations of model dimensions (p_1, p_2) and sample size.

Based on the results shown in Table 4, it can be concluded that the proposed low-rank testing procedure accurately detects the presence of "Granger causality" across the two blocks, when the data have been generated based on a truly multi-layer VAR system. Further, when $B^\star = 0$, the type I error is close to the nominal α level for sufficiently large sample sizes, but deteriorates for increased model dimensions. In particular, relatively large values of p_2 and larger spectral radius $\rho(C^\star)$ negatively impact the empirical false rejection proportion, which deviates from the desired control level of the type I error. In the case where $\mathrm{rank}(B^\star) = r > 0$, the testing procedure provides satisfactory type I error control for larger sample sizes and excellent power.

Sparse testing. Since the rejection rule of the HC-statistic is based on empirical process theory (Shorack and Wellner, 2009) and its dependence on α is not explicit, we focus on illustrating how the empirical proportion of false rejections (type I error) varies with the sample size T, the model dimensions (p_1, p_2) and the spectral radius of $C^\star$. To show the power of the test, each entry in $B^\star$ is nonzero with probability $q \in (0,1)$ such that $q(p_1p_2) = (p_1p_2)^\theta$ with $\theta = 0.6$, to ensure the overall sparsity of $B^\star$ satisfies the sparsity requirement posited in Proposition **??**. The magnitude is set such that the signal-to-noise ratio is 1.2. Note that the actual number of parameters is p_1p_2, while the total number of subsamples used is 3000 with the length of subsamples varying according to different simulation settings to demonstrate the dependence of type I error and power on sample sizes.

Table 4: Empirical type I error and power for low-rank testing

			type I error ($B^\star = 0$)			power (rank($B^\star$) = 1)
	(p_1, p_2)	sample size	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.1$	cut-off $\chi^2(0.01)_{p_1p_2}/T$
$\rho(C^\star) = 0.5$	$(20, 20)$	$T = 500$	0.028	0.123	0.227	1
		$T = 1000$	0.015	0.073	0.137	1
		$T = 2000$	0.011	0.059	0.118	1
	$(50, 20)$	$T = 500$	0.070	0.228	0.355	1
		$T = 1000$	0.026	0.125	0.226	1
		$T = 2000$	0.013	0.094	0.163	1
	$(20, 50)$	$T = 500$	0.484	0.751	0.857	1
		$T = 1000$	0.089	0.246	0.375	1
		$T = 2000$	0.020	0.088	0.164	1
	$(100, 50)$	$T = 500$	0.997	0.999	1	1
		$T = 1000$	0.608	0.828	0.908	1
		$T = 2000$	0.166	0.374	0.511	1
$\rho(C^\star) = 0.8$	$(20, 50)$	$T = 500$	0.533	0.789	0.880	1
		$T = 1000$	0.130	0.306	0.452	1
		$T = 2000$	0.045	0.145	0.252	1
	$(50, 20)$	$T = 500$	0.083	0.250	0.382	1
		$T = 1000$	0.039	0.133	0.234	1
		$T = 2000$	0.019	0.096	0.174	1
			type I error ($H_0 : r \leq 5$)			power ($H_0 : r \leq 4$)
			$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.1$	cut-off $\chi^2(0.01)_{(p_1-4)(p_2-4)}/T$
$\rho(C^\star) = 0.5$ rank($B^\star$) = 5	$(20, 50)$	$T = 500$	0.092	0.274	0.400	1
		$T = 1000$	0.034	0.140	0.236	1
		$T = 2000$	0.022	0.096	0.178	1
	$(50, 20)$	$T = 500$	0.454	0.722	0.829	1
		$T = 1000$	0.126	0.313	0.452	1
		$T = 2000$	0.062	0.184	0.284	1

Table 5: Empirical type I error and power for sparse testing

		type I error ($B^\star = 0$)				power (SNR($B^\star$) = 0.8)			
	(p_1, p_2)	200	500	1000	2000	200	500	1000	2000
$\rho(C^\star) = 0.5$	$(20, 20)$	0.244	0.097	0.074	0.055	1	1	1	1
	$(50, 20)$	0.393	0.131	0.108	0.074	1	1	1	1
	$(20, 50)$	0.996	0.351	0.153	0.093	1	1	1	1
	$(100, 50)$	1.000	0.963	0.270	0.115	1	1	1	1
$\rho(C^\star) = 0.8$	$(50, 20)$	0.402	0.158	0.112	0.075	0.829	0.996	1	1
	$(20, 50)$	0.999	0.430	0.166	0.111	1	1	1	1

Based on the results shown in Table 5, when $B^\star = 0$, the proposed testing procedure can effectively detect the absence of block "Granger causality", provided that the sample size is moderately large compared to the total number of parameters being tested. However, if the model dimension is large, whereas the sample size is small, the test procedure becomes problematic and fails to provide legitimate type I error control as desired. When $B^\star$ is

nonzero, empirically the test is always able detect its presence, as long as the effective signal-to-noise ratio is beyond the detection threshold.

6. Real Data Analysis Illustration.

We employ the developed framework and associated testing procedures to address one of the motivating applications. Specifically, we analyze the temporal dynamics of the log-returns of stocks with large market capitalization and key macroeconomic variables, as well as their cross-dependence. Specifically, using the notation in (1), we assume that the X_t block consists of the stock log-returns, while the macroeconomic variables form the Z_t block. With this specification, we assume that the macroeconomic variables are "Granger-caused" by the stock market, but not vice versa. Note that our framework allows us to pose and test a more general question than previous work in the economics literature considered. For example, Farmer (2015) building on previous work by Fitoussi et al. (2000); Phelps (1999) tests only the relationship between the employment index and the composite stock index, using a bivariate VAR model. On the other hand, our framework enables us to consider the components of the S&P 100 index and the "medium" list of macroeconomic variables considered in the work of Stock and Watson (2005).

Next, we provide a brief description of the data used. The stock data consist of monthly observations of 71 stocks corresponding to a stable set of historical components comprising the S&P 100 index for the 2001-2016 period. The macroeconomic variables are chosen from the "medium" list in Stock and Watson (2005). The complete lists of stocks and macroeconomic variables, together with the preprocessing to ensure stationarity used in this study are given in Supplementary Material **??**.

We start the analysis by using the VAR model for the stock log-returns to study their evolution over the 2001-2016 period. Analogously to the strategy employed by Billio et al. (2012), we consider 36-month-long rolling-windows for fitting the model $X_t = AX_{t-1} + U_t$, for a total of 143 estimates of the transition matrix A. VAR models involving more than 1 lag were also fitted to the data, but did not indicate temporal dependence beyond lag 1. To obtain the final estimates across all 143 subsamples, we employ *stability selection* (Meinshausen and Bühlmann, 2010), with the threshold set at 0.6 for including an edge in A.[1] Figure 2 depicts the global clustering coefficient (Luce and Perry, 1949) of the skeleton of the estimated A over all 143 rolling windows, with the time stamps on the horizontal axis specifying the starting time of the corresponding window. The results clearly indicate strong connectivity in lead-lag stock relationships during the financial crisis period March 2007-June 2009. Next, we present the analysis based on the VAR-X component of our model, given by $Z_t = BX_{t-1} + CZ_{t-1} + V_t$ with the stock log-returns corresponding to the X_t block and the (stationary) macroeconomic variables to the Z_t block. As before, we fit the data within each rolling window, with the tuning parameters based on a search over a 10×10 lattice (with $(\lambda_B, \lambda_C) \in [0.5, 4] \times [0.2, 2]$, equal-spaced) using the BIC. It should be noted that for the majority of the rolling windows, the rank of B is 1 (data not shown). The sparsity level of the estimated C over the 143 rolling windows is depicted in Figure 3. The connectivity patterns in C show more complex and nuanced patterns than

1. The threshold is set at a relatively low level to compensate for the relative small rolling window size.

Figure 2: Global clustering coefficient of estimated A over different periods

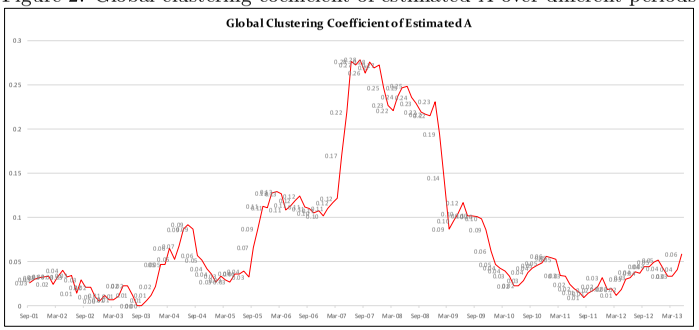
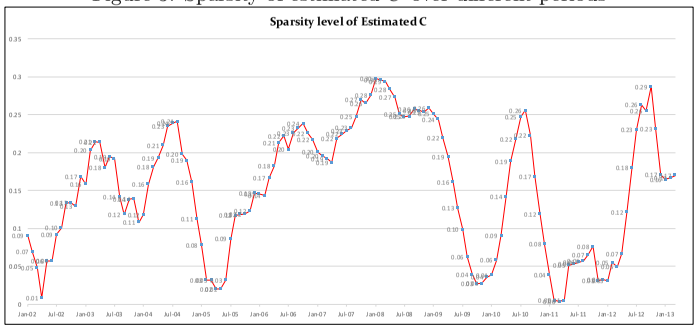


Figure 3: Sparsity of estimated C over different periods



for stocks. A more detailed analysis of the various peaks identified through our model is given in Supplementary Material **??**.

Based on the previous findings, we partition the time frame spanning 2001-2016 into the following periods: pre- (2001/07–2007/03), during- (2007/01–2009/12) and post-crisis (2010/01-2016/06) one. We estimate the model parameters using the data within the entire sub-period(s).

The estimation procedure of the transition matrix A for different periods is identical to that described above using subsamples over rolling-windows. For the pre- and post- crisis periods, since we have 76 and 77 samples respectively, the stability selection threshold is set at 0.75, whereas for the during-crisis period, at 0.6 to compensate for the small sample size (36). Table 6 shows the average R-square for all 71 stocks, as well as its standard deviation, which is calculated based on in-sample fit; i.e.,the proportion of variation explained by using the VAR(1) model to fit the data. The overall sparsity level and the spectral radius of the estimated transition matrices A are also presented. The results are consistent with the previous finding of increased connectivity during the crisis. Further, for all periods the estimate of the spectral radius is fairly large, indicating strong temporal dependence of the log-returns.

Table 6: Summary for estimated A within different periods.

	2001/07–2007/03	2007/01–2009/12	2010/01–2016/06
Averaged R sq	0.31	0.72	0.28
Sd of R sq	0.103	0.105	0.094
Sparsity level of $\widehat{A}$	0.17	0.23	0.19
Spectral radius of $\widehat{A}$	0.67	0.90	0.75

Next, we focus on the key motivation for developing the proposed modeling framework, namely the inter-dependence of stocks and macroeconomic variables over the specified three sub-periods. The p-value for testing the hypothesis of lack of block "Granger causality" $H_0 : B = 0$, together with the spectral radius and the sparsity level for the estimated C transition matrices are listed in Table 7. Specifically, for all three periods, the rank of estimated B is 1, indicating that the stock market as captured by its leading stocks, "Granger-causes" the temporal evolution of the macroeconomic variables. The fact that the rank of B is 1, indicates that the inter-block influence can be captured as a single portfolio acting in unison. To investigate the relative importance of each sector in the portfolio, we group the stocks by sectors. The proportion of each sector (up to normalization) is obtained by summing up the loadings (first right singular vector of the estimated B) of the stocks within this sector, weighted by their market capitalization (results shown in Supplementary Material **??**).

Next, we discuss some key relationships emerging from the model. We start with total employment (ETTL), whose dynamics are only influenced by its own past values as seen by the lack of an incoming arrow in Figure **??**. Further, an examination of the left singular vector (see Table 8) of B strongly indicates the impact of the stock market on total employment. This finding is consistent with the analysis in Farmer (2015), which argues that the crash of the stock market provides a plausible explanation for the great recession. However,

the analysis in Farmer (2015) is based on bivariate VAR models involving only employment and the stock index. Therefore, there is a possibility that the stock market is reacting to some other information captured by other macroeconomic variables, such as GDP, capital spending, inflation, interest rates, etc. However, our high-dimensional VAR model simultaneously analyzes a key set of macroeconomic variables and also accounts for the influence of the largest stocks in the market. Hence, it automatically overcomes the criticism leveraged by Sims (1992) about misinterpretations of findings from small scale VAR models due to the omission of important variables, and further echoed in the discussion in Bernanke et al. (2005). Other interesting findings are presented in Supplementary Material **??**.

Table 7: Summary for estimated B and C within different periods.

	2001/07–2007/03	2007/01–2009/12	2010/01–2016/16
p-value for testing $H_0 : B = 0$	0.075	0.009	0.044
Sparsity level of $\widehat{C}$	0.06	0.25	0.06
Spectral radius of $\widehat{C}$	0.35	0.76	0.40

Table 8: Left Singular Vectors of Estimated B for different periods

	Pre-Crisis	During-Crisis	Post-Crisis
FFR	-0.24	-0.26	-0.23
T10yr	-0.09	0.14	0.16
UNEMPL	-0.07	0.01	-0.07
IPI	-0.43	0.34	0.26
ETTL	0.33	0.24	0.13
M1	0.23	-0.12	-0.47
AHES	-0.01	0.30	0.17
CU	-0.49	0.32	0.27
M2	0.10	-0.04	-0.32
HS	0.51	-0.02	-0.02
EX	-0.18	0.41	0.06
PCEQI	-0.07	-0.18	0.41
GDP	0.10	-0.02	0.05
PCEPI	0.00	0.14	-0.01
PPI	-0.15	0.00	0.06
CPI	0.01	0.15	-0.31
SP.IND	-0.06	-0.53	0.38

Remark 13 *We also applied our multi-block model with the first block X_t corresponding to the macro-economic variables and the second block Z_t the stocks variables (results not shown). The key question is whether there is also "Granger causality" from the broader economy to the stock market. The results are inconclusive due to sample size issues that do not allow us to properly test for the key hypothesis whether $B = 0$ or not. Specifically, the length of the sub-periods is short compared to the dimensionality required for the test procedure. A similar issue arises, which is related to the detection boundary for the sparse testing procedure during the crisis period. Further, for a sparse B, an examination of its entries shows that Employment Total did not impact the stock market, which is in line with the conclusion reached at the aggregate level by Farmer (2015). On the other hand, GDP negatively impacts stock log-returns, which may act as a leading indicator for suppressed investment and business growth and hence future stock returns.*

7. Discussion.

We briefly discuss generalizations of the model to the case of more than two blocks, as mentioned in the introductory section. For the sake of concreteness, consider a triangular recursive linear dynamical system given by:

$$\begin{aligned} X_t^{(1)} &= A_{11}X_{t-1}^{(1)} + \epsilon_t^{(1)}, \\ X_t^{(2)} &= A_{12}X_{t-1}^{(1)} + A_{22}X_{t-1}^{(2)} + \epsilon_t^{(2)}, \\ X_t^{(3)} &= A_{13}X_{t-1}^{(1)} + A_{23}X_{t-1}^{(2)} + A_{33}X_{t-1}^{(3)} + \epsilon_t^{(3)}, \\ &\vdots \end{aligned} \tag{28}$$

where $X^{(j)} \in \mathbb{R}^{p_j}$ denotes the variables in group j, A_{ij} $(i < j)$ encodes the dependency of $X^{(j)}$ on the past values of variables in group i, and A_{jj} encodes the dependency on its own past values. Further, $\{\epsilon_t^{(j)}\}$ is the innovation process that is neither temporally, nor cross-sectionally correlated, i.e.,

$$\mathrm{Cov}(\epsilon_t^{(j)}, \epsilon_s^{(j)}) = 0 \ (s \neq t), \quad \mathrm{Cov}(\epsilon_t^{(i)}, \epsilon_s^{(j)}) = 0 \ (i \neq j, \ \forall \ (s,t)), \quad \mathrm{Cov}(\epsilon_t^{(j)}, \epsilon_t^{(j)}) = \big(\Omega^{(j)}\big)^{-1},$$

with $\Omega^{(j)}$ capturing the conditional contemporaneous dependency of variables within group j. The model in (28) can also be viewed from a multi-layered time-varying network perspective: nodes in each layer are "Granger-caused" by nodes from its previous layers, and are also dependent on its own past values. As previously mentioned, in various real applications, it is of interest to obtain estimates of the transition matrices, and/or test if "Granger-causality" is present between interacting blocks; i.e., to test $A_{ij} = 0$ for some $i \neq j$.

The triangular structure of the system decouples the estimation of the transition matrices from each equation, and hence a straightforward extension of the estimation procedure presented in Section 2.1 becomes applicable. Specifically, to obtain estimates of the transition matrices A_{ij}'s for fixed j and $1 \leq i \leq j$, and the inverse covariance $\Omega^{(j)}$, the optimization problem is formulated as follows:

$$\begin{aligned} (\{\widehat{A}_{ij}\}_{i\leq j}, \widehat{\Omega}^{(j)}) = \underset{A_{ij},\Omega^{(j)}}{\arg\min} \Big\{ &- \log\det\Omega^{(j)} + \frac{1}{T}\sum_{t=1}^{T} \big(x_t^{(j)} - \sum_{i=1}^{j} A_{ij}x_{t-1}^{(i)}\big)'\Omega^{(j)}\big(x_t^{(j)} - \sum_{1\leq i\leq j} A_{ij}x_{t-1}^{(i)}\big) \\ &+ \sum_{i=1}^{j} \mathcal{R}(A_{ij}) + \rho^{(j)}\|\Omega^{(j)}\|_{1,\mathrm{off}}\Big\}, \end{aligned} \tag{29}$$

where the exact expression for the $\mathcal{R}(A_{ij})$ adapts to the structural assumption imposed on the corresponding transition matrix (sparse/low-rank). Solving (29) again requires an iterative algorithm involving the alternate update between transition matrices and the inverse

covariance matrices. Further, for updating the values of the transition matrices, a cyclic block-coordinate updating procedure is used.

Consistency results can be established analogously to those provided in Section 3, under the posited conditions of restricted strong convexity (RSC) and a deviation bound. With a larger number of interacting blocks of variables, lower bounds for the lower extremes of the spectra involve all corresponding transition matrices. The error rates that can be obtained are as follows: (i) if equation k only involves sparse transition matrices, then the finite-sample bounds of the transition matrices in this layer in Frobenius norm are of the order $O\big(\sqrt{\frac{\log p_k + \log \sum_{i \le k} p_k}{T}}\big)$, while (ii) if some of the transition matrices are assumed low rank, then the corresponding finite sample bounds are of the order $O\big(\sqrt{\frac{p_k + \sum_{i<k} p_k}{T}}\big)$.

Another generalization that can be handled algorithmically with the same estimation procedure discussed above is the presence of d-lags in the specification of the linear dynamical system. Based on the consistency results developed in this work, together with the theoretical findings for VAR(d) models presented in Basu and Michailidis (2015), we expect all the established theoretical properties of the transition matrices estimates to go through under appropriate RSC and deviation bound conditions.

Acknowledgments

The authors would like to thank Action Editor Jie Peng and two anonymous referees for many constructive comments and suggestions. The work of GM was supported in part by NSF grants CCF-1540093, DMS-1545277 and IIS-1632730 and by NIH grant 5-R01-GM11402902.

Appendix A. Additional Theorems and Proofs for Theorems.

In this section, we introduce two additional theorems that respectively establish the consistency properties for the initializers $\widehat{A}^{(0)}$ and $(\widehat{B}^{(0)}, \widehat{C}^{(0)})$, for *fixed* realizations of the processes $\{X_t\}$ and $\{Z_t\}$. Specifically, $\widehat{A}^{(0)}$ and $(\widehat{B}^{(0)}, \widehat{C}^{(0)})$ are solutions to the following optimization problems:

$$\widehat{A}^{(0)} := \arg\min_{A} \big\{ \tfrac{1}{T} \big\| \mathcal{X}^T - \mathcal{X} A' \big\|_F + \lambda_A \|A\|_1 \big\}, \tag{30}$$

$$(\widehat{B}^{(0)}, \widehat{C}^{(0)}) := \arg\min_{B,C} \big\{ \tfrac{1}{T} \big\| \mathcal{Z}^T - \mathcal{X} B' - \mathcal{Z} C' \big\|_F + \lambda_B \|B\|_* + \lambda_C \|C\|_1 \big\}. \tag{31}$$

Note that they also correspond to estimators of the setting where there is no contemporaneous dependence among the idiosyncratic error processes. If we additionally introduce operators $\mathfrak{X}_0$ and $\mathfrak{W}_0$ defined as

$$\begin{aligned} \mathfrak{X}_0 : &\quad \mathfrak{X}_0(\Delta) = \mathcal{X}'\Delta, \quad \text{for } \Delta \in \mathbb{R}^{p_1 \times p_1}, \\ \mathfrak{W}_0 : &\quad \mathfrak{W}_0(\Delta) = \mathcal{W}'\Delta, \quad \text{for } \Delta \in \mathbb{R}^{p_2 \times (p_1+p_2)} \quad \text{where } \mathcal{W} := [\mathcal{X}, \mathcal{Z}], \end{aligned}$$

then (30) and (31) can be equivalently written as

$$\widehat{A}^{(0)} := \arg\min_{A} \big\{ \tfrac{1}{T} \big\| \mathcal{X}^T - \mathfrak{X}_0(A) \big\|_F + \lambda_A \|A\|_1 \big\},$$

$$(\widehat{B}^{(0)}, \widehat{C}^{(0)}) := \arg\min_{B,C} \big\{ \tfrac{1}{T} \big\| \mathcal{Z}^T - \mathfrak{W}_0(B_{\text{aug}}, C_{\text{aug}}) \big\|_F + \lambda_B \|B\|_* + \lambda_C \|C\|_1 \big\},$$

where $B_{\text{aug}} := [B, O_{p_2 \times p_2}], C_{\text{aug}} := [O_{p_2 \times p_1}, C]$.

Theorem 14 (Error bounds for $\widehat{A}^{(0)}$) *Suppose the operator $\mathfrak{X}_0$ satisfies the RSC condition with norm $\Phi(\Delta) = \|\Delta\|_1$, curvature $\alpha_{RSC} > 0$ and tolerance $\tau > 0$, so that*

$$s_A^\star \tau \le \alpha_{RSC}/32.$$

Then, with regularization parameter λ_A satisfying $\lambda_A \ge 4\|\mathcal{X}'\mathcal{U}/T\|_\infty$, the solution to (30) satisfies the following bounds:

$$\|\widehat{A}^{(0)} - A^\star\|_F \le 12\sqrt{s_A^\star}\lambda_A/\alpha_{RSC} \qquad \text{and} \qquad \|\widehat{A} - A^\star\|_1 \le 48 s_A^\star \lambda_A/\alpha_{RSC}.$$

Theorem 15 (Error bound for $(\widehat{B}^{(0)}, \widehat{C}^{(0)})$) *Let $\mathcal{J}_{C^\star}$ be the support set of $C^\star$ and $s_C^\star$ denote its cardinality. Let $r_B^\star$ be the rank of $B^\star$. Assume that $\mathfrak{W}_0$ satisfies the RSC condition with norm*

$$\Phi(\Delta) := \inf_{B_{aug} + C_{aug} = \Delta} \mathcal{Q}(B, C), \qquad \text{where} \quad \mathcal{Q}(B, C) := \|B\|_* + \tfrac{\lambda_C}{\lambda_B}\|C\|_1,$$

curvature α_{RSC} and tolerance τ such that

$$128\tau r_B^\star < \alpha_{RSC}/4 \qquad \text{and} \qquad 64\tau s_C^\star (\lambda_C/\lambda_B)^2 < \alpha_{RSC}/4.$$

Then, with regularization parameters λ_B and λ_C satisfying

$$\lambda_B \ge 4\big\|\mathcal{W}'\mathcal{V}/T\big\|_{op} \quad \text{and} \quad \lambda_C \ge 4\big\|\mathcal{W}'\mathcal{V}/T\big\|_\infty,$$

the solution to (31) satisfies the following bounds:

$$\|\widehat{B}^{(0)} - B^\star\|_F^2 + \|\widehat{C}^{(0)} - C^\star\|_F^2 \le 4\left(2 r_B^\star \lambda_B^2 + s_C^\star \lambda_C^2\right)/\alpha_{RSC}^2.$$

In the rest of this subsection, we first prove Theorem 14 and 15, then prove Theorem 5 and 6, whose statements are given in Section 3.2.

Proof [**Proof of Theorem 14**]

For the ease of notation, in this proof, we use $\widehat{A}$ to refer to $\widehat{A}^{(0)}$ whenever there is no ambiguity. Let $\beta_A^\star = \text{vec}(A^\star)$ and denote the residual matrix and its vectorized version by $\Delta_A = \widehat{A} - A^\star$ and $\Delta_{\beta_A} = \widehat{\beta}_A - \beta_A^\star$, respectively. By the optimality of $\widehat{A}$ and the feasibility of $A^\star$, the following *basic inequality* holds:

$$\tfrac{1}{T}\|\mathfrak{X}_0(\Delta_A)\|_F^2 \le \tfrac{2}{T}\langle\!\langle \Delta_A, \mathcal{X}'\mathcal{U}\rangle\!\rangle + \lambda_A\left\{\|A^\star\|_1 - \|A^\star + \Delta_A\|_1\right\},$$

which is equivalent to:

$$\Delta_{\beta_A}'\widehat{\Gamma}_X^{(0)}\Delta_{\beta_A} \le \tfrac{2}{T}\langle\Delta_{\beta_A}, \text{vec}(\mathcal{X}'\mathcal{U})\rangle + \lambda_A\left\{\|\beta_A^\star\|_1 - \|\beta_A^\star + \Delta_{\beta_A}\|_1\right\}, \tag{32}$$

where $\widehat{\Gamma}_X^{(0)} = \mathrm{I}_{p_1} \otimes \frac{\mathcal{X}'\mathcal{X}}{T}$. By Hölder's inequality and the triangle inequality, an upper bound for the right-hand-side of (32) is given by

$$\tfrac{2}{T}\|\Delta_{\beta_A}\|_1\left\|\mathcal{X}'\mathcal{U}\right\|_\infty + \lambda_A\|\Delta_{\beta_A}\|_1. \tag{33}$$

Now with the specified choice of λ_A, by Lemma **??**, $\|\Delta_{\beta_A|\mathcal{J}_{A^\star}}\|_1 \le 3\|\Delta_{\beta_A|\mathcal{J}_{A^\star}^c}\|_1$ i.e., $\Delta_{\beta_A} \in \mathcal{C}(\mathcal{J}_{A^\star}, 3)$, hence $\|\Delta_{\beta_A}\|_1 \le 4\|\Delta_{\beta_A|\mathcal{J}_{A^\star}}\|_1 \le 4\sqrt{s_A^\star}\|\Delta_{\beta_A}\|$. By choosing $\lambda_A \ge 4\|\mathcal{X}'\mathcal{U}/T\|_\infty$, (33) is further upper bounded by

$$\frac{3}{2}\lambda_A\|\Delta_{\beta_A}\|_1 \le 6\sqrt{s_A^\star}\lambda_A\|\Delta_{\beta_A}\|. \tag{34}$$

Combined with the RSC condition and the upper bound given in (34), we have

$$\frac{\alpha_{\text{RSC}}}{2}\|\Delta_{\beta_A}\|^2 - \frac{\tau}{2}\|\Delta_{\beta_A}\|_1^2 \le \frac{1}{2}\Delta_{\beta_A}'\widehat{\Gamma}_X^{(0)}\Delta_{\beta_A} \le 3\sqrt{s_A^\star}\lambda_A\|\Delta_{\beta_A}\|,$$
$$\frac{\alpha_{\text{RSC}}}{4}\|\Delta_{\beta_A}\|^2 \le \left(\frac{\alpha_{\text{RSC}}}{2} - \frac{16 s_A^\star\tau}{4}\right)\|\Delta_{\beta_A}\|^2 \le 3\sqrt{s_A^\star}\lambda_A\|\Delta_{\beta_A}\|,$$

which implies

$$\|\Delta_{\beta_A}\| \le 12\sqrt{s_A^\star}\lambda_A/\alpha_{\text{RSC}} \qquad \text{and} \qquad \|\Delta_{\beta_A}\|_1 \le 48 s_A^\star\lambda_A/\alpha_{\text{RSC}}.$$

It is easy to see that these bounds also hold for $\|\Delta_A\|_F$ and $\|\Delta_A\|_1$, respectively. ■

Next, to prove Theorem 15, we introduce the following two sets of subspaces $\{\mathcal{S}_\Theta, \mathcal{S}_\Theta^\perp\}$ and $\{\mathcal{R}_\Theta, \mathcal{R}_\Theta^c\}$ associated with some generic matrix $\Theta \in \mathbb{R}^{m_1\times m_2}$, in which the nuclear norm and the ℓ_1-norm are decomposable, respectively (see Negahban et al., 2012). Specifically, let the singular value decomposition of Θ be $\Theta = U\Sigma V'$ with U and V being orthogonal matrices. Let $r = \text{rank}(\Theta)$, and we use U^r and V^r to denote the first r columns of U and V associated with the r singular values of Θ. Further, define

$$\begin{aligned}\mathcal{S}_\Theta &:= \left\{\Delta \in \mathbb{R}^{m_1\times m_2} | \text{row}(\Delta) \subseteq V^r \quad \text{and} \quad \text{col}(\Delta) \subseteq U^r\right\},\\ \mathcal{S}_\Theta^\perp &:= \left\{\Delta \in \mathbb{R}^{m_1\times m_2} | \text{row}(\Delta) \perp V^r \quad \text{and} \quad \text{col}(\Delta) \perp U^r\right\}.\end{aligned} \tag{35}$$

Then, for an arbitrary (generic) matrix $M \in \mathbb{R}^{m_1\times m_2}$, its restriction on the subspace $\mathcal{S}(\Theta)$ and $\mathcal{S}^\perp(\Theta)$, denoted by $M_{\mathcal{S}(\Theta)}$ and $M_{\mathcal{S}^\perp(\Theta)}$ respectively, are given by:

$$M_{\mathcal{S}_\Theta} = U\begin{bmatrix}\widetilde{M}_{11} & \widetilde{M}_{12}\\ \widetilde{M}_{21} & O\end{bmatrix}V' \qquad \text{and} \qquad M_{\mathcal{S}_\Theta^\perp} = U\begin{bmatrix}O & O\\ O & \widetilde{M}_{22}\end{bmatrix}V',$$

where $\Theta = U\Sigma V'$ and $\widetilde{M}$ is defined and partitioned as

$$\widetilde{M} = U'MV = \begin{bmatrix}\widetilde{M}_{11} & \widetilde{M}_{12}\\ \widetilde{M}_{21} & \widetilde{M}_{22}\end{bmatrix}, \quad \text{where } \widetilde{M}_{11} \in \mathbb{R}^{r\times r}.$$

Note that by Lemma **??**, $M_{\mathcal{S}_\Theta} + M_{\mathcal{S}_\Theta^\perp} = M$. Moreover, when Θ is restricted to the subspace induced by itself $\Theta_{\mathcal{S}_\Theta}$ (and we write $\Theta_{\mathcal{S}}$ for short for this specific case), the following decomposition for the nuclear norm holds:

$$\|\Theta\|_* = \|\Theta_{\mathcal{S}} + \Theta_{\mathcal{S}^\perp}\|_* = \|\Theta_{\mathcal{S}}\|_* + \|\Theta_{\mathcal{S}^\perp}\|_*.$$

Let $\mathcal{J}(\Theta)$ be the set of indexes in which Θ is nonzero. Analogously, we define

$$\begin{aligned}\mathcal{R}_\Theta &:= \left\{\Delta \in \mathbb{R}^{m_1\times m_2} | \Delta_{ij} = 0 \ \text{ for } \ (i,j) \notin \mathcal{J}(\Theta)\right\},\\ \mathcal{R}_\Theta^c &:= \left\{\Delta \in \mathbb{R}^{m_1\times m_2} | \Delta_{ij} = 0 \ \text{ for } \ (i,j) \in \mathcal{J}(\Theta)\right\}.\end{aligned} \tag{36}$$

Then, for an arbitrary matrix M, $M_{\mathcal{J}_\Theta} \in \mathcal{R}_\Theta$ is obtained by setting the entries of M whose indexes are not in $\mathcal{J}(\Theta)$ to 0, and $M_{\mathcal{J}_\Theta^c} \in \mathcal{R}_\Theta^c$ is obtained by setting the entries of M whose indexes are in $\mathcal{J}(\Theta)$ to 0. Then, the following decomposition holds:

$$\left\|M_{\mathcal{J}_\Theta} + M_{\mathcal{J}_\Theta^c}\right\|_1 = \|M_{\mathcal{J}_\Theta}\|_1 + \left\|M_{\mathcal{J}_\Theta^c}\right\|_1.$$

Proof [**Proof of Theorem 15**] Again for the ease of notation, in this proof, we drop the superscript and use $(\widehat{B}^{(0)}, \widehat{C}^{(0)})$ to denote $(\widehat{B}, \widehat{C})$ whenever there is no ambiguity. Define $\mathcal{Q}$ to be the weighted regularizer:

$$\mathcal{Q}(B, C) = \|B\|_* + \tfrac{\lambda_C}{\lambda_B}\|C\|_1.$$

Note that $(B^\star, C^\star)$ is always feasible, and by the optimality of $(\widehat{B}, \widehat{C})$, the following inequality holds:

$$\frac{1}{T}\|\mathcal{Z}^T - \mathfrak{W}_0(\widehat{B}_{\text{aug}} + \widehat{C}_{\text{aug}})\|_F^2 + \lambda_B\|\widehat{B}\|_* + \lambda_C\|\widehat{C}\|_1 \le \frac{1}{T}\|\mathcal{Z}^T - \mathfrak{W}_0(B^\star + C^\star)\|_F^2 + \lambda_B\|B^\star\|_* + \lambda_C\|C^\star\|_1,$$

By defining $\Delta_{\text{aug}}^B = \widehat{B}_{\text{aug}} - B_{\text{aug}}^\star = [\Delta^B, O]$, $\Delta_{\text{aug}}^C = \widehat{C}_{\text{aug}} - C_{\text{aug}}^\star = [O, \Delta^C]$, we obtain the following *basic inequality*:

$$\tfrac{1}{T}\left\|\mathfrak{W}_0(\Delta_{\text{aug}}^B + \Delta_{\text{aug}}^C)\right\|_F^2 \le \tfrac{2}{T}\langle\!\langle \Delta_{\text{aug}}^B + \Delta_{\text{aug}}^C, \mathcal{W}'\mathcal{V}\rangle\!\rangle + \lambda_B\mathcal{Q}(B^\star, C^\star) - \lambda_B\mathcal{Q}(\widehat{B}, \widehat{C}). \tag{37}$$

By Hölder's inequality and Lemma 21, we have

$$\begin{aligned}\tfrac{1}{T}\left\|\mathfrak{W}_0(\Delta_{\text{aug}}^B + \Delta_{\text{aug}}^C)\right\|_F^2 \le{}& \tfrac{2}{T}\left(\left\|\Delta_{\mathcal{S}_{B^\star}}^B\right\|_* + \|\Delta_{\mathcal{S}_{B^\star}^\perp}^B\|_*\right)\|\mathcal{W}'\mathcal{V}\|_{op} + \tfrac{2}{T}\left(\|\Delta_{\mathcal{J}_{C^\star}^c}^C\|_1 + \|\Delta_{\mathcal{J}_{C^\star}^c}^C\|_1\right)\|\mathcal{W}'\mathcal{V}\|_\infty\\ &+ \lambda_B\mathcal{Q}(\Delta_{\mathcal{S}_{B^\star}}^B, \Delta_{\mathcal{J}_{C^\star}}^C) - \lambda_B\mathcal{Q}(\Delta_{\mathcal{S}_{B^\star}^\perp}^B, \Delta_{\mathcal{J}_{C^\star}^c}^C).\end{aligned} \tag{38}$$

With the specified choice of λ_B and λ_C, after some algebra, (38) is further bounded by

$$\frac{3\lambda_B}{2}\mathcal{Q}(\Delta^B_{\mathcal{S}_{B^\star}}, \Delta^C_{\mathcal{J}_{C^\star}}) - \frac{\lambda_B}{2}\mathcal{Q}(\Delta^B_{\mathcal{S}^\perp_{B^\star}}, \Delta^C_{\mathcal{J}^c_{C^\star}}).$$

By Lemma 22, and using this upper bound, we obtain

$$\frac{\alpha_{\text{RSC}}}{2}(|||\Delta^B|||_F^2 + |||\Delta^C|||_F^2) - \frac{\lambda_B}{2}\mathcal{Q}(\Delta^B, \Delta^C) \le \frac{3\lambda_B}{2}\mathcal{Q}(\Delta^B_{\mathcal{S}_{B^\star}}, \Delta^C_{\mathcal{J}_{C^\star}}) - \frac{\lambda_B}{2}\mathcal{Q}(\Delta^B_{\mathcal{S}^\perp_{B^\star}}, \Delta^C_{\mathcal{J}^c_{C^\star}}).$$

By the triangle inequality, $\mathcal{Q}(\Delta^B, \Delta^C) \le \mathcal{Q}(\Delta^B_{\mathcal{S}_{B^\star}}, \Delta^C_{\mathcal{J}_{C^\star}}) + \mathcal{Q}(\Delta^B_{\mathcal{S}^\perp_{B^\star}}, \Delta^C_{\mathcal{J}^c_{C^\star}})$, rearranging gives

$$\frac{\alpha_{\text{RSC}}}{2}(|||\Delta^B|||_F^2 + |||\Delta^C|||_F^2) \le 2\lambda_B\mathcal{Q}(\Delta^B_{\mathcal{S}_{B^\star}}, \Delta^C_{\mathcal{J}_{C^\star}}). \tag{39}$$

By Lemma **??**, with $N = B^\star$, $M_1 = \Delta^B_{\mathcal{S}_{B^\star}}$, $M_2 = \Delta^B_{\mathcal{S}^\perp_{B^\star}}$, we get

$$\text{rank}(\Delta^B_{\mathcal{S}_{B^\star}}) \le 2r^\star_B \qquad \text{and} \qquad \langle\!\langle \Delta^B_{\mathcal{S}_{B^\star}}, \Delta^B_{\mathcal{S}^\perp_{B^\star}} \rangle\!\rangle = 0,$$

which implies $|||\Delta^B_{\mathcal{S}_{B^\star}}|||_* \le (\sqrt{2r^\star_B})|||\Delta^B_{\mathcal{S}_{B^\star}}|||_F \le (\sqrt{2r^\star_B})|||\Delta^B|||_F$. Since $\Delta^C_{\mathcal{J}_{C^\star}}$ has at most $s^\star_C$ nonzero entries, it follows that $\|\Delta^C_{\mathcal{J}_{C^\star}}\|_1 \le \sqrt{s^\star_C}|||\Delta^C_{\mathcal{J}_{C^\star}}|||_F \le \sqrt{s^\star_C}|||\Delta^C|||_F$. Therefore,

$$\mathcal{Q}(\Delta^B_{\mathcal{S}_{B^\star}}, \Delta^C_{\mathcal{J}_{C^\star}}) = \lambda_B|||\Delta^B_{\mathcal{S}_{B^\star}}|||_* + \lambda_C\|\Delta^C_{\mathcal{J}_{C^\star}}\|_1 \le \lambda_B(\sqrt{2r^\star_B})|||\Delta^B|||_F + \lambda_C(\sqrt{s^\star_C})|||\Delta^C|||_F$$

With an application of the Cauchy-Schwartz inequality, (39) yields:

$$\frac{\alpha_{\text{RSC}}}{2}(|||\Delta^B|||_F^2 + |||\Delta^C|||_F^2) \le \sqrt{2r^\star_B\lambda_B^2 + s^\star_C\lambda_C^2} * \sqrt{|||\Delta^B|||_F^2 + |||\Delta^C|||_F^2}$$

and we obtain the following bound:

$$|||\Delta^B|||_F^2 + |||\Delta^C|||_F^2 \le 4\left(2r^\star_B\lambda_B^2 + s^\star_C\lambda_C^2\right)/\alpha^2_{\text{RSC}}.$$

■

Proof [**Proof of Theorem 5**] At iteration 0, $\widehat{A}^{(0)}$ solves the following optimization problem:

$$\widehat{A}^{(0)} = \underset{A\in\mathbb{R}^{p_1\times p_1}}{\arg\min}\left\{\tfrac{1}{T}|||\mathcal{X}^T - \mathcal{X}A'|||_F^2 + \lambda_A|||A|||_*\right\}.$$

By Theorem 14, its error bound is given by

$$\|\widehat{A}^{(0)} - A^\star\|_1 \le 48s^\star_A\lambda_A/\alpha_{\text{RSC}},$$

provided that $\widehat{\Gamma}^{(0)}_X = \mathrm{I}_{p_1}\otimes\mathcal{X}'\mathcal{X}/T$ satisfies the RSC condition, and the regularization parameter λ_A satisfies $\lambda_A \ge 4\|\mathcal{X}'\mathcal{U}/T\|_\infty$. For random realizations $\mathcal{X}$ and $\mathcal{U}$, by Lemma 16 and Lemma 17, there exist constants c_i and c_i' such that for sample size $T \gtrsim s^\star_A\log p_1$, with probability at least $1 - c_1\exp(-c_2T\min\{1,\omega^{-2}\})$, where $\omega = c_3\mu_{\max}(\mathcal{A})/\mu_{\min}(\mathcal{A})$

$$(\mathbf{E_1}) \qquad \widehat{\Gamma}^{(0)}_X \text{ satisfies RSC condition with } \alpha_{\text{RSC}} = \Lambda_{\min}(\Sigma^\star_u)/(2\mu_{\max}(\mathcal{A}))\ ,$$

and with probability at least $1 - c_1'\exp(-c_2'\log p_1)$,

$$(\mathbf{E_2}) \qquad \|\mathcal{X}'\mathcal{U}/T\|_\infty \le C_0\sqrt{\frac{\log p_1}{T}}, \qquad \text{for some constant } C_0.$$

Hence with probability at least $1 - c_1\exp(-c_2T) - c_1'\exp(-c_2'\log p_1)$,

$$\|\widehat{A}^{(0)} - A^\star\|_1 = O\Big(s^\star_A\sqrt{\tfrac{\log p_1}{T}}\Big).$$

Moving onto $\widehat{\Omega}^{(0)}_u$, which is given by

$$\widehat{\Omega}^{(0)}_u = \underset{\Omega_u\in\mathbb{S}^{p_1\times p_1}_{++}}{\arg\min}\left\{\log\det\Omega_u - \text{trace}\big(\widehat{S}^{(0)}_u\Omega_u\big) + \rho_u\|\Omega_u\|_{1,\text{off}}\right\},$$

where $\widehat{S}^{(0)}_u = \frac{1}{T}(\mathcal{X}^T - \mathcal{X}\widehat{A}^{(0)\prime})'(\mathcal{X}^T - \mathcal{X}\widehat{A}^{(0)\prime})$. By Theorem 1 in Ravikumar et al. (2011), the error bound for $\widehat{\Omega}^{(0)}_u$ relies on how well $\widehat{S}^{(0)}_u$ concentrates around $\Sigma^\star_u$, more specifically, $\|\widehat{S}^{(0)}_u - \Sigma^\star_u\|_\infty$. Note that

$$\|\widehat{S}^{(0)}_u - \Sigma^\star_u\|_\infty \le \|S_u - \Sigma^\star_u\|_\infty + \|\widehat{S}^{(0)}_u - S_u\|_\infty,$$

where $S_u = \mathcal{U}'\mathcal{U}/T$ is the sample covariance based on true errors. For the first term, by Ravikumar et al. (2011), there exists constant $\tau_0 > 2$ such that with probability at least $1 - 1/p_1^{\tau_0-2} = 1 - \exp(-\tau\log p_1)$ $(\tau > 0)$, the following bound holds:

$$(\mathbf{E_3}) \qquad \|S_u - \Sigma^\star_u\|_\infty \le C_1\sqrt{\frac{\log p_1}{T}}, \qquad \text{for some constant } C_1.$$

For the second term,

$$\widehat{S}^{(0)}_u - S_u = \tfrac{2}{T}\mathcal{U}'\mathcal{X}(A^\star - \widehat{A}^{(0)})' + (A^\star - \widehat{A}^{(0)})\left(\tfrac{\mathcal{X}'\mathcal{X}}{T}\right)(A^\star - \widehat{A}^{(0)})' := I_1 + I_2,$$

For I_1, based on the analysis of $\|A^\star - \widehat{A}^{(0)}\|_1$ and $\|\mathcal{X}'\mathcal{U}/T\|_\infty$,

$$\|I_1\|_\infty \le 2|||A^\star - \widehat{A}^{(0)}|||_\infty\|\tfrac{1}{T}\mathcal{X}'\mathcal{U}\|_\infty \le 2\|A^\star - \widehat{A}^{(0)}\|_1\|\tfrac{1}{T}\mathcal{X}'\mathcal{U}\|_\infty = O\Big(\tfrac{s^\star_A\log p_1}{T}\Big)$$

For I_2,

$$\begin{aligned}\|(A^\star - \widehat{A}^{(0)})\left(\tfrac{\mathcal{X}'\mathcal{X}}{T}\right)(A^\star - \widehat{A}^{(0)})'\|_\infty &\le |||A^\star - \widehat{A}^{(0)}|||_\infty|||A^\star - \widehat{A}^{(0)}|||_1\|\tfrac{\mathcal{X}'\mathcal{X}}{T}\|_\infty \\ &\le \|A^\star - \widehat{A}^{(0)}\|_1^2\|\tfrac{\mathcal{X}'\mathcal{X}}{T}\|_\infty,\end{aligned}$$

where by Proposition 2.4 in Basu and Michailidis (2015) and then taking a union bound, with probability at least $1 - c_1''\exp(-c_2''\log p_1)$ $(c_1'', c_2'' > 0)$,

$$(\mathbf{E_4}) \qquad \|\tfrac{\mathcal{X}'\mathcal{X}}{T}\|_\infty \le C_2\sqrt{\frac{\log p_1}{T}} + \Lambda_{\max}(\Gamma_X), \qquad \text{for some constant } C_2.$$

Hence,

$$\|I_2\|_\infty = O\Big((s_A^\star)^2\big(\tfrac{\log p_1}{T}\big)^{3/2}\Big) + O\Big((s_A^\star)^2\tfrac{\log p_1}{T}\Big)$$

Combining all terms, and since we assume that $T^{-1}\log p_1$ is small, $O(\sqrt{T^{-1}\log p_1})$ becomes the leading term, and the following bound holds with probability at least $1-c_1\exp(-c_2T)-c_1'\exp(-c_2'\log p_1)-c_1''\exp(-c_2''\log p_1)-\exp(-\tau\log p_1)$:

$$\|\widehat{S}_u^{(0)} - \Sigma_u^\star\|_\infty = O\Big(\sqrt{\tfrac{\log p_1}{T}}\Big).$$

Consequently,

$$\|\widehat{\Omega}_u^{(0)} - \Omega_u^\star\|_\infty = O\Big(\sqrt{\tfrac{\log p_1}{T}}\Big).$$

At iteration 1, the vectorized $\widehat{A}^{(1)}$ solves

$$\widehat{\beta}_A^{(1)} = \underset{\beta\in\mathbb{R}^{p_1^2}}{\arg\min}\big\{-2\beta'\widehat{\gamma}_X^{(1)} + \beta'\widehat{\Gamma}_X^{(1)}\beta + \lambda_A\|\beta\|_1\big\},$$

where

$$\widehat{\gamma}_X^{(1)} = \tfrac{1}{T}(\widehat{\Omega}_u^{(0)}\otimes\mathcal{X}')\mathrm{vec}(\mathcal{X}^T), \qquad \widehat{\Gamma}_X^{(1)} = \widehat{\Omega}_u^{(0)}\otimes\tfrac{\mathcal{X}'\mathcal{X}}{T}.$$

The error bound for $\widehat{\beta}_A^{(1)}$ relies on (1) $\widehat{\Gamma}_X^{(1)}$ satisfying the RSC condition, which holds for sample size $T \gtrsim (d_{\Omega_u^\star}^{\max})^2\log p_1$ upon $\|\widehat{\Omega}_u^{(0)} - \Omega_u^\star\|_\infty = O(\sqrt{T^{-1}\log p_1})$; and (2) a bound for $\|\mathcal{X}'\mathcal{U}\widehat{\Omega}_u^{(0)}/T\|_\infty$. For $\|\mathcal{X}'\mathcal{U}\widehat{\Omega}_u^{(0)}/T\|_\infty$,

$$\tfrac{1}{T}\mathcal{X}'\mathcal{U}\widehat{\Omega}_u^{(0)} = \tfrac{1}{T}\mathcal{X}'\mathcal{U}\Omega_u^\star + \tfrac{1}{T}\mathcal{X}'\mathcal{U}(\widehat{\Omega}_u^{(0)} - \Omega_u^\star) := I_3 + I_4.$$

For I_3, by Lemma 3 in Lin et al. (2016) and with the aid of Proposition 2.4 in Basu and Michailidis (2015), again with probability at least $1-c_1'''\exp(-c_2'''\log p_1)$ we get

$$(\mathbf{E_5}) \qquad \big\|\tfrac{1}{T}\mathcal{X}'\mathcal{U}\Omega_u^\star\big\|_\infty \le C_3\sqrt{\frac{\log p_1}{T}}, \qquad \text{for some constant } C_3.$$

For I_4, by Corollary 3 in Ravikumar et al. (2011), we get

$$\Big\|\tfrac{1}{T}\mathcal{X}'\mathcal{U}(\widehat{\Omega}_u^{(0)} - \Omega_u^\star)\Big\|_\infty \le d_{\Omega_u^\star}^{\max}\|\tfrac{1}{T}\mathcal{X}'\mathcal{U}\|_\infty\|\widehat{\Omega}_u^{(0)} - \Omega_u^\star\|_\infty = O\Big(\tfrac{\log p_1}{T}\Big).$$

Combining all terms and taking the leading one, once again we have

$$\|\widehat{A}^{(1)} - A^\star\|_1 = O\Big(s_A^\star\sqrt{\tfrac{\log p_1}{T}}\Big),$$

which holds with probability at least $1-c_1\exp(-c_2T)-\tilde{c}_1\exp(-\tilde{c}_2\log p_1)-\exp(-\tau\log p_1)$, by letting $\tilde{c}_1 = \max\{c_1', c_1'', c_1'''\}$ and $\tilde{c}_1 = \min\{c_2', c_2'', c_2'''\}$. It should be noted that up to this step, all sources of randomness from the random realizations have been captured by events from $\mathbf{E_1}$ to $\mathbf{E_5}$; thus, for $\widehat{\Omega}_u^{(1)}$ and iterations thereafter, the probability for which the bounds hold will no longer change, and the same holds for the error bounds for $\widehat{A}^{(k)}$ and $\widehat{\Omega}_u^{(k)}$ in

terms of the relative order with respect to the dimension p_1 and sample size T. Therefore, we conclude that with high probability, for all iterations k,

$$\|\mathcal{X}'\mathcal{U}\widehat{\Omega}_u^{(k)}/T\|_\infty = O\Big(\sqrt{\tfrac{\log p_1}{T}}\Big), \qquad \|\widehat{S}_u^{(k)} - \Sigma_u^\star\|_\infty = O\Big(\sqrt{\tfrac{\log p_1}{T}}\Big).$$

With the aid of Theorem 14, it then follows that

$$\|\widehat{A}^{(k)} - A^\star\|_F = O\Big(\sqrt{\tfrac{s_A^\star\log p_1}{T}}\Big), \qquad \|\widehat{\Omega}_u^{(k)} - \Omega_u^\star\|_F = O\Big(\sqrt{\tfrac{(s_{\Omega_u^\star}+p_1)\log p_1}{T}}\Big).$$

■

Proof [**Proof of Theorem 6**] At iteration 0, $(\widehat{B}^{(0)}, \widehat{C}^{(0)})$ solves the following optimization:

$$(\widehat{B}^{(0)}, \widehat{C}^{(0)}) = \underset{(B,C)}{\arg\min}\Big\{\tfrac{1}{T}\|\mathcal{Z}^T - \mathcal{X}B' - \mathcal{Z}C'\|_F^2 + \lambda_B\|B\|_* + \lambda_C\|C\|_1\Big\}.$$

Let $W_t = (X_t', Z_t')' \in \mathbb{R}^{p_1+p_2}$ be the joint process and $\mathcal{W}$ be the realizations, with operators $\mathfrak{W}_0$ identically defined to that in Theorem 15 . By Theorem 15,

$$\|\widehat{B}^{(0)} - B^\star\|_F^2 + \|\widehat{C}^{(0)} - C^\star\|_F^2 \le 4(2r_B^\star\lambda_B^2 + s_C^\star\lambda_C^2)/\alpha_{\mathrm{RSC}}^2,$$

provided that $\mathfrak{W}$ satisfies the RSC condition and λ_B, λ_C respectively satisfy

$$\lambda_B \ge 4\big\|\mathcal{W}'\mathcal{V}/T\big\|_{op} \quad \text{and} \quad \lambda_C \ge 4\big\|\mathcal{W}'\mathcal{V}/T\big\|_\infty.$$

In particular, by Lemma 18 for random realizations of $\mathcal{X}$, $\mathcal{Z}$ and $\mathcal{V}$, for sample size $T \gtrsim c_0(p_1+2p_2)$, with probability at least $1-c_1\exp\{-c_2(p_1+p_2)\}$,

$$(\mathbf{E_1'}) \qquad \mathfrak{W}_0 \text{ satisfies the RSC condition.}$$

By Lemma 19, for sample size $T \gtrsim (p_1+2p_2)$ and some constant $C_1, C_2 > 0$,

$$(\mathbf{E_2'}) \qquad \big\|\mathcal{W}'\mathcal{V}/T\big\|_{op} \le C_1\sqrt{\frac{p_1+2p_2}{T}} \quad \text{and} \quad \big\|\mathcal{W}'\mathcal{V}/T\big\|_\infty \le C_2\sqrt{\frac{\log(p_1+p_2)+\log p_2}{T}},$$

with probability at least $1-c_1'\exp\{-c_2'(p_1+2p_2)\}$ and $1-c_1''\exp\{-c_2''\log[p_2(p_1+p_2)]\}$, respectively. Hence, with probability at least

$$1-c_1\exp\{-c_2(p_1+p_2)\}-c_1'\exp\{-c_2'(p_1+2p_2)\}-c_1''\exp\{-c_2''\log[p_2(p_1+p_2)]\},$$

the following bound holds for the initializers as long as sample size $T \gtrsim (p_1+2p_2)$:

$$\|\widehat{B}^{(0)} - B^\star\|_F^2 + \|\widehat{C}^{(0)} - C^\star\|_F^2 = O\Big(\tfrac{p_1+2p_2}{T}\Big) + O\Big(\tfrac{\log(p_1+p_2)+\log p_2}{T}\Big). \tag{40}$$

Considering the estimation of $\widehat{\Omega}_v^{(0)}$, it solves a graphical Lasso problem:

$$\widehat{\Omega}_v^{(0)} = \underset{\Omega_v\in\mathbb{S}_{++}^{p_2\times p_2}}{\arg\min}\Big\{\log\det\Omega_v - \mathrm{trace}\big(\widehat{S}_u^{(0)}\Omega_v\big) + \rho_v\|\Omega_v\|_{1,\mathrm{off}}\Big\},$$

where $\widehat{S}_v^{(0)} = \frac{1}{T}(\mathcal{Z}^T - \mathcal{X}\widehat{B}^{(0)\prime} - \mathcal{Z}\widehat{C}^{(0)\prime})'(\mathcal{Z}^T - \mathcal{X}\widehat{B}^{(0)\prime} - \mathcal{Z}\widehat{C}^{(0)\prime})$. Similar to the proof of Theorem 5, the error bound for $\widehat{\Omega}_v^{(0)}$ depends on $\|\widehat{S}_v^{(0)} - \Sigma_v^\star\|_\infty$, which can be decomposed as

$$\|\widehat{S}_v^{(0)} - \Sigma_v^\star\|_\infty \leq \|S_v - \Sigma_v^\star\|_\infty + \|\widehat{S}_v^{(0)} - S_v\|_\infty,$$

where $S_v = \mathcal{V}'\mathcal{V}/T$ is the sample covariance based on the true errors. For the first term, by Lemma 1 in Ravikumar et al. (2011), there exists constant $\tau_0 > 2$ such that with probability at least $1 - 1/p_2^{\tau_0 - 2} = 1 - \exp(-\tau \log p_2)$ $(\tau > 0)$, the following bound holds:

$$(\mathbf{E}_\mathbf{3}') \qquad \|S_v - \Sigma_v^\star\|_\infty \leq C_3\sqrt{\frac{\log p_1}{T}}, \qquad \text{for some constant } C_3.$$

For the second term, let $\Pi = [B, C] \in \mathbb{R}^{p_2 \times (p_1+p_2)}$, then

$$\widehat{S}_v^{(0)} - S_v = \frac{2}{T}\mathcal{V}'\mathcal{W}(\Pi^\star - \widehat{\Pi}^{(0)})' + (\Pi^\star - \widehat{\Pi}^{(0)})\left(\frac{\mathcal{W}'\mathcal{W}}{T}\right)(\Pi^\star - \widehat{\Pi}^{(0)})' := I_1 + I_2,$$

For I_1, we have

$$\|\tfrac{2}{T}\mathcal{V}'\mathcal{W}(\Pi^\star - \widehat{\Pi}^{(0)})'\|_\infty \leq \|\tfrac{2}{T}\mathcal{V}'\mathcal{W}(\Pi^\star - \widehat{\Pi}^{(0)})'\|_F \leq 2\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\|_{op}\|\Pi^\star - \widehat{\Pi}^{(0)}\|_F.$$

Consider the leading term of $\|\Pi^\star - \widehat{\Pi}^{(0)}\|_F$ as in (40), whose rate is $O(\sqrt{T^{-1}(p_1+2p_2)})$. We therefore obtain

$$\|I_1\|_\infty \leq \|I_1\|_F = O\Big(\tfrac{p_1+2p_2}{T}\Big).$$

Similarly for I_2,

$$\|I_2\|_\infty \leq \|I_2\|_F \leq \|\Pi^\star - \widehat{\Pi}^{(0)}\|_F^2\|\tfrac{\mathcal{W}'\mathcal{W}}{T}\|_{op},$$

where with a similar derivation to that in Lemma 23, for sample size $T \gtrsim (p_1+p_2)$, with probability at least $1 - c_1'''\exp\{-c_2'''(p_1+p_2)\}$, we get

$$(\mathbf{E}_\mathbf{4}') \qquad \|\tfrac{\mathcal{W}'\mathcal{W}}{T}\|_{op} \leq C_4\sqrt{\frac{p_1+2p_2}{T}} + \Lambda_{\max}(\Gamma_X), \qquad \text{for some constant } C_4.$$

Hence,

$$\|I_2\|_\infty \leq \|I_2\|_F \leq = O\Big((\frac{p_1+2p_2}{T})^{3/2}\Big).$$

Combining all terms and then taking the leading one, with probability at least

$$1 - c_1\exp\{-c_2(p_1+p_2)\} - c_1'\exp\{-c_2'(p_1+2p_2)\} - c_1''\exp\{-c_2''\log[p_2(p_1+p_2)]\}$$
$$-c_1'''\exp\{-c_2'''(p_1+p_2)\} - \exp(-\tau\log p_2),$$

we obtain

$$\|\widehat{S}_v^{(0)} - \Sigma_v^\star\|_\infty = O\Big(\sqrt{\tfrac{p_1+2p_2}{T}}\Big).$$

Note that here with the required sample size, $(p_1+2p_2)/T$ is a small quantity, and therefore

$$O\Big((\tfrac{p_1+2p_2}{T})^{3/2}\Big) \leq O\Big(\tfrac{p_1+2p_2}{T}\Big) \leq O\Big(\sqrt{\tfrac{p_1+2p_2}{T}}\Big).$$

At iteration 1, the bound of $\|\widehat{B}^{(1)} - B^\star\|_F^2 + \|\widehat{C}^{(1)} - C^\star\|_F^2$ relies on the following two quantities:

$$\Big\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(0)}\Big\|_{op} \qquad \text{and} \qquad \Big\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(0)}\Big\|_\infty.$$

Using a similar derivation to that in the proof of Theorem 5,

$$\Big\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(0)}\Big\|_\infty \leq \Big\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}(\widehat{\Omega}_v^{(0)} - \Omega_v^\star)\Big\|_\infty + \|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\Omega_v^\star\|_\infty, \tag{41}$$

where by viewing $\mathcal{V}\Omega_v^\star$ as some random realization coming from a certain sub-Gaussian process, with probability at least $1 - \bar{c}_1''\exp\{-\bar{c}_2''\log[p_2(p_1+p_2)]\}$, we get

$$(\mathbf{E}_\mathbf{5}') \qquad \|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\Omega_v^\star\|_\infty \leq C_5\sqrt{\frac{\log(p_1+p_2) + \log p_2}{T}}, \qquad \text{for some constant } C_5,$$

and

$$\Big\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}(\widehat{\Omega}_v^{(0)} - \Omega_v^\star)\Big\|_\infty \leq d_{\max}^{\Omega_v^\star}\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\|_\infty\|\widehat{\Omega}_v^{(0)} - \Omega_v^\star\|_\infty$$
$$= O\Big(\sqrt{\tfrac{\log(p_1+p_2)+\log p_2}{T}}\Big)\cdot O\Big(\sqrt{\tfrac{p_1+2p_2}{T}}\Big).$$

For $\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(0)}\|_{op}$, similarly we have

$$\Big\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(0)}\Big\|_{op} \leq \|\tfrac{1}{T}\mathcal{W}'\mathcal{V}(\widehat{\Omega}_v^{(0)} - \Omega_v^\star)\|_{\text{op}} + \|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\Omega_v^\star\|_{\text{op}}, \tag{42}$$

where with probability at least $1 - \bar{c}_1'\exp\{-\bar{c}_2'(p_1+p_2)\}$,

$$(\mathbf{E}_\mathbf{6}') \qquad \|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\Omega_v^\star\|_{op} \leq C_6\sqrt{\frac{p_1+2p_2}{T}} \qquad \text{for some constant } C_6,$$

and

$$\Big\|\tfrac{1}{T}\mathcal{W}'\mathcal{V}(\widehat{\Omega}_v^{(0)} - \Omega_v^\star)\Big\|_{op} \leq \|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\|_{\text{op}}\|\widehat{\Omega}_v^{(0)} - \Omega_v^\star\|_{\text{op}}$$
$$\leq \|\tfrac{1}{T}\mathcal{W}'\mathcal{V}\|_{op}\Big[d_{\max}^{\Omega_v^\star}\|\widehat{\Omega}_v^{(0)} - \Omega_v^\star\|_\infty\Big] = O\Big(\tfrac{p_1+2p_2}{T}\Big),$$

where the second inequality follows from Corollary 3 of Ravikumar et al. (2011). Combining all terms from (41) and (42), the leading term gives the following bound:

$$\|\widehat{B}^{(1)} - B^\star\|_F^2 + \|\widehat{C}^{(1)} - C^\star\|_F^2 \leq C_7\Big(\frac{p_1+2p_2}{T}\Big) \qquad \text{for some constant } C_7,$$

and this error rate coincides with that in the bound of $\|\widehat{B}^{(0)} - B^\star\|_F^2 + \|\widehat{C}^{(0)} - C^\star\|_F^2$. This implies that for $\widehat{\Omega}_v^{(1)}$ and iterations thereafter, the error rate remains unchanged. Moreover, all sources of randomness have been captured up to this step in events $\mathbf{E}_\mathbf{1}'$ to $\mathbf{E}_\mathbf{6}'$, and therefore the probability for the bounds to hold no longer changes. Consequently, the following bounds hold for all iterations k:

$$\|\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(k)}/T\|_\infty = \|\mathcal{W}'\mathcal{V}\widehat{\Omega}_v^{(k)}/T\|_{\text{op}} = O\Big(\sqrt{\tfrac{p_1+2p_2}{T}}\Big)$$

and

$$\|\widehat{S}_v^{(k)} - \Sigma_v^\star\|_\infty = O\Big(\sqrt{\tfrac{p_1+2p_2}{T}}\Big),$$

with probability at least

$$1-c_0\exp\{-\tilde{c}_0(p_1+p_2)\}-c_1\exp\{-\tilde{c}_1(p_1+2p_2)\}-c_2\exp\{-\tilde{c}_2\log[p_2(p_1+p_2)]\}-\exp\{-\tau\log p_2\}.$$

for some new positive constants $c_i, \tilde{c}_i$ $(i=0,1,2)$ and τ.[2] The above bounds directly imply the bound in the statement in Theorem 6, with the aid of Theorem 15. ■

Appendix B. Key Lemmas and Their Proofs.

In this section, we verify the conditions that are required for establishing the consistency results in Theorem 14 and 15, under random realizations of $\mathcal{X}$, $\mathcal{Z}$, $\mathcal{U}$ and $\mathcal{V}$.

The following two lemmas verify the conditions for establishing the consistency properties for $\widehat{A}^{(0)}$. Specifically, Lemma 16 establishes that with high probability, $\mathfrak{X}_0$ satisfies the RSC condition. Further, Lemma 17 gives a high probability upper bound for $\|\mathcal{X}'\mathcal{U}/T\|_\infty$ for random $\mathcal{X}$ and $\mathcal{U}$.

Lemma 16 (Verification of the RSC condition) *For the* VAR(1) *model* $\{X_t\}$ *posited in* (1)*, there exist* $c_i > 0$ $(i=1,2,3)$ *such that for sample size* $T \gtrsim \max\{\omega^2,1\}s_A^\star \log p_1$*, with probability at least*

$$1-c_1\exp\left[-c_2 T\min\{1,\omega^{-2}\}\right], \qquad \omega = c_3\frac{\Lambda_{\max}(\Sigma_u)\mu_{\max}(\mathcal{A})}{\Lambda_{\min}(\Sigma_u)\mu_{\min}(\mathcal{A})},$$

the following inequality holds

$$\frac{1}{2T}|||\mathfrak{X}_0(\Delta)|||_F^2 \geq \alpha_{RSC}|||\Delta|||_F^2 - \tau\|\Delta\|_1^2, \qquad \text{for } \Delta\in\mathbb{R}^{p_1\times p_1},$$

where $\alpha_{RSC} = \frac{\Lambda_{\min}(\Sigma_u)}{\mu_{\max}(\mathcal{A})}$, $\tau = 4\alpha_{RSC}\max\{\omega^2,1\}\log p_1/T$.

Proof [**Proof of Lemma 16**] For the specific VAR(1) process $\{X_t\}$ given in (1), using Proposition 4.2 in Basu and Michailidis (2015) with $d=1$ directly gives the result. Specifically, we note that by letting $\theta = \mathrm{vec}(\Delta)$,

$$\tfrac{1}{T}|||\mathfrak{X}_0(\Delta)|||_F^2 = \theta'\widehat{\Gamma}_X^{(0)}\theta,$$

where $\widehat{\Gamma}_X^{(0)} = \mathrm{I}_{p_1}\otimes(\mathcal{X}'\mathcal{X}/T)$, and $\|\theta\|_2^2 = |||\Delta|||_F^2$, $\|\theta\|_1 = \|\Delta\|_1$. ■

Lemma 17 (Verification of the deviation bound) *For the model in* (1)*, there exist constants* $c_i > 0$, $i=0,1,2$ *such that for* $T \gtrsim 2\log p_1$*, with probability at least* $1 - c_1\exp(-2c_2\log p_1)$*, the following bound holds:*

$$\|\mathcal{X}'\mathcal{U}/T\|_\infty \leq c_0\Lambda_{\max}(\Sigma_u)\left[1+\frac{1}{\mu_{\min}(\mathcal{A})}+\frac{\mu_{\max}(\mathcal{A})}{\mu_{\min}(\mathcal{A})}\right]\sqrt{\frac{2\log p_1}{T}}. \tag{43}$$

2. Here we slightly abuse the notations and redefine $c_0 := \max\{c_1, c_1''\}$, $c_1 := \max\{c_1', \tilde{c}_1'\}$, $\tilde{c}_1 := \min\{c_2', \tilde{c}_2'\}$, $c_2 = \max\{c_1'', \tilde{c}_1''\}$, $\tilde{c}_2 := \min\{c_2'', \tilde{c}_2''\}$.

Proof [**Proof of Lemma 17**] First, we note that,

$$\|\mathcal{X}'\mathcal{U}/T\|_\infty = \max_{\substack{1\leq i\leq p_1\\ 1\leq j\leq p_1}}\left|e_i'\left(\mathcal{X}'\mathcal{U}/T\right)e_j\right|.$$

Applying Proposition 2.4(b) in Basu and Michailidis (2015) for an arbitrary pair of (i,j) gives:

$$\mathbb{P}\left(\left|e_i'\left(\mathcal{X}'\mathcal{U}/T\right)e_j\right| > \eta\left[\Lambda_{\max}(\Sigma_u)\left(1+\frac{1}{\mu_{\min}(\mathcal{A})}+\frac{\mu_{\max}(\mathcal{A})}{\mu_{\min}(\mathcal{A})}\right)\right]\right) \leq 6\exp[-cT\min\{\eta,\eta^2\}].$$

Setting $\eta = c_0\sqrt{2\log p_1/T}$ and taking a union bound over all $1\leq i\leq p_1, 1\leq j\leq p_1$, we get that for some $c_1, c_2 > 0$, with probability at least $1-c_1\exp[-2c_2\log p_1]$,

$$\max_{\substack{1\leq i\leq p_1\\ 1\leq j\leq p_1}}\left|e_i'\left(\mathcal{X}'\mathcal{U}/T\right)e_j\right| \leq c_0\Lambda_{\max}(\Sigma_u)\left[1+\frac{1}{\mu_{\min}(\mathcal{A})}+\frac{\mu_{\max}(\mathcal{A})}{\mu_{\min}(\mathcal{A})}\right]\sqrt{\frac{2\log p_1}{T}}.$$

■

In the next two lemmas, Lemma 18 gives an RSC curvature that holds with high probability for $\mathfrak{W}$ induced by a random $\mathcal{W}$, and Lemma 19 gives a high probability upper bound for $|||W'\mathcal{V}/T|||_{\mathrm{op}}$ and $\|W'\mathcal{V}/T\|_\infty$.

Lemma 18 (Verification of the RSC condition) *Consider the covariance stationary process* $W_t = (X_t', Z_t')' \in \mathbb{R}^{p_1+p_2}$ *whose spectral density exists. Suppose* $\mathfrak{m}(f_W) > 0$*. There exist constants* $c_i > 0, i=1,2,3$ *such that with probability at least* $1-2c_1\exp(-c_2(p_1+p_2))$*, the RSC condition for* $\mathfrak{W}$ *induced by a random* $\mathcal{W}$ *holds for* α_{RSC} *and tolerance 0, where*

$$\alpha_{RSC} = \pi\mathfrak{m}(f_W)/4,$$

whenever $T \gtrsim c_3(p_1+p_2)$*.*

Proof [**Proof of Lemma 18**] First ,we note that the following inequality holds, for any $\mathcal{W}$:

$$\frac{1}{2T}|||\mathfrak{W}_0(\Delta)|||_F^2 = \frac{1}{2T}|||\mathcal{W}'\Delta|||_F^2 = \frac{1}{2T}\sum_{j=1}^{p_2}\left\|[\mathcal{W}'\Delta]_j\right\|_2^2 \geq \frac{1}{2}\Lambda_{\min}\left(\widehat{\Gamma}_W^{(0)}\right)|||\Delta|||_F^2. \tag{44}$$

where $\widehat{\Gamma}_W^{(0)} = \mathcal{W}'\mathcal{W}/T$. Applying Lemma 4 in Negahban and Wainwright (2011) on $\mathcal{W}$ together with Proposition 2.3 in Basu and Michailidis (2015), the following bound holds with probability at least $1-2c_1\exp[-c_2(p_1+p_2)]$, as long as $T\gtrsim c_3(p_1+p_2)$:

$$\Lambda_{\min}\left(\widehat{\Gamma}_W^{(0)}\right) \geq \frac{\Lambda_{\min}(\Gamma_W(0))}{4} \geq \frac{\pi}{2}\mathfrak{m}(f_W),$$

where $\Gamma_W(0) = \mathbb{E}W_tW_t'$. Combining with (44), the RSC condition holds with $\kappa(\mathfrak{W}) = \pi\mathfrak{m}(f_W)/4$. ■

Lemma 19 (Verification of the deviation bound) *There exist constants $c_i > 0$ and $c_i' > 0, i = 1, 2, 3$ such that the following statements hold:*

(a) With probability at least $1 - c_1 \exp[-c_2(p_1 + 2p_2)]$, as long as $T \gtrsim c_3(p_1 + 2p_2)$,

$$\left\|\left|\mathcal{W}'\mathcal{V}/T\right|\right\|_{op} \leq c_0 \left[\mathcal{M}(f_W) + \tfrac{1}{2\pi}\Lambda_{\max}(\Sigma_v) + \mathcal{M}(f_{W,V})\right] \sqrt{\frac{p_1 + 2p_2}{T}}. \tag{45}$$

(b) With probability at least $1 - c_1' \exp(-c_2' \log(p_1+p_2) - c_2' \log p_2)$, as long as $T \gtrsim c_3' \log[(p_1 + p_2)p_2]$,

$$\left\|\mathcal{W}'\mathcal{V}/T\right\|_{\infty} \leq c_0' \left[\mathcal{M}(f_W) + \tfrac{1}{2\pi}\Lambda_{\max}(\Sigma_v) + \mathcal{M}(f_{W,V})\right] \sqrt{\frac{\log(p_1 + p_2) + \log p_2}{T}}. \tag{46}$$

Proof [**Proof of Lemma 19**] (a) is a direct application of Lemma 23 on processes $\{W_t\} \in \mathbb{R}^{(p_1+p_2)}$ and $\{V_t\} \in \mathbb{R}^{p_2}$, and (b) is a direct application of Lemma 17. ■

Appendix C. Auxiliary Lemmas and Their Proofs.

Lemma 20 *Consider two centered stationary Gaussian processes $\{X_t\}$ and $\{Z_t\}$. Further, assume that the spectral density of the joint process $\{(X_t', Z_t')'\}$ exists. Denote their cross-covariance by $\Gamma_{X,Z}(\ell) := \mathrm{Cov}(X_t, Z_{t+\ell})$, and their cross-spectral density is defined as*

$$f_{X,Z}(\theta) := \frac{1}{2\pi} \sum_{\ell=-\infty}^{\infty} \Gamma_{X,Z}(\ell) e^{-i\ell\theta}, \qquad \theta \in [-\pi, \pi],$$

whose upper extreme is given by:

$$\mathcal{M}(f_{X,Z}) = \mathrm{esssup}_{\theta\in[-\pi,\pi]} \sqrt{\Lambda_{\max}\left(f_{X,Z}^*(\theta) f_{X,Z}(\theta)\right)}.$$

Let $\mathcal{X}$ and $\mathcal{Z}$ be data matrices with sample size n. Then, there exists a constant $c > 0$, such that for any $u, v \in \mathbb{R}^p$ with $\|u\| \leq 1$, $\|v\| \leq 1$, we have

$$\mathbb{P}\left[\left|u'\left(\frac{\mathcal{X}'Z}{T} - \mathrm{Cov}(X_t, Z_t)\right)v\right| > 2\pi\left(\mathcal{M}(f_X) + \mathcal{M}(f_Z) + \mathcal{M}(f_{X,Z})\right)\eta\right] \leq 6\exp\left(-cT\min\{\eta, \eta^2\}\right).$$

Proof Let $\xi_t = \langle u, X_t\rangle$, $\eta_t = \langle v, Z_t\rangle$. Let $f_X(\theta), f_Z(\theta)$ denote the spectral density of $\{X_t\}$ and $\{Z_t\}$, respectively. Then, the spectral density of $\{\xi_t\}$ and $\{\eta_t\}$, respectively, is $f_\xi(\theta) = u' f_X(\theta) u$, $f_\eta(\theta) = v' f_Z(\theta) v$. Also, we note that $\mathcal{M}(f_\xi) \leq \mathcal{M}(f_X)$, $\mathcal{M}(f_\eta) \leq \mathcal{M}(f_Z)$. Then,

$$\begin{aligned}\frac{2}{T}\left[\sum_{t=0}^{T} \xi_t\eta_t - \mathrm{Cov}(\xi_t, \eta_t)\right] &= \left[\frac{1}{T}\sum_{t=0}^{T}(\xi_t + \eta_t)^2 - \mathrm{Var}(\xi_t + \eta_t)\right] \\ &\quad - \left[\frac{1}{T}\sum_{t=0}^{T}(\xi_t)^2 - \mathrm{Var}(\xi_t)\right] - \left[\frac{1}{T}\sum_{t=0}^{T}(\eta_t)^2 - \mathrm{Var}(\eta_t)\right].\end{aligned} \tag{47}$$

By Proposition 2.7 in Basu and Michailidis (2015),

$$\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=0}^{T}(\xi_t)^2 - \mathrm{Var}(\xi_t)\right| > 2\pi\mathcal{M}(f_X)\eta\right) \geq 2\exp\left[-cn\min(\eta, \eta^2)\right],$$

and

$$\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=0}^{T}(\eta_t)^2 - \mathrm{Var}(\eta_t)\right| > 2\pi\mathcal{M}(f_Z)\eta\right) \geq 2\exp\left[-cn\min(\eta, \eta^2)\right].$$

What remains to be considered is the first term in (47), whose spectral density is given by

$$f_{\xi+\eta}(\theta) = u' f_X(\theta) u + v' f_Z(\theta) z + u' f_{X,Z}(\theta) v + v' f_{X,Z}^*(\theta) u,$$

and its upper extreme satisfies

$$\mathcal{M}(f_{\xi+\eta}) \leq \mathcal{M}(f_X) + \mathcal{M}(f_Z) + 2\mathcal{M}(f_{X,Z}).$$

Hence, we get:

$$\mathbb{P}\left(\left|\frac{1}{T}\sum_{t=0}^{T}(\xi_t + \eta_t)^2 - \mathrm{Var}(\xi_t + \eta_t)\right| > 2\pi[\mathcal{M}(f_X) + \mathcal{M}(f_Z) + 2\mathcal{M}(f_{X,Z})]\eta\right) \geq 2\exp[-cn\min(\eta, \eta^2)].$$

Combining all three terms yields the desired result. ■

Lemma 21 *Define the error matrix by $\Delta^B = \widehat{B} - B^\star$ and $\Delta^C = \widehat{C} - C^\star$, and let the weighted regularizer $\mathcal{Q}$ be defined as*

$$\mathcal{Q}(B, C) = \left\|\left|B\right|\right\|_* + \frac{\lambda_C}{\lambda_B}\|C\|_1.$$

With the subspaces defined in (35) and (36), the following inequality holds:

$$\mathcal{Q}(B^\star, C^\star) - \mathcal{Q}(\widehat{B}, \widehat{C}) \leq \mathcal{Q}(\Delta^B_{\mathcal{S}_{B^\star}}, \Delta^C_{\mathcal{J}_{C^\star}}) - \mathcal{Q}(\Delta^B_{\mathcal{S}^\perp_{B^\star}}, \Delta^C_{\mathcal{J}^c_{C^\star}}).$$

Proof First, from definitions (35) and (36), we know that $B^\star_{\mathcal{S}^\perp} = 0$ and $C^\star_{\mathcal{J}^c_{C^\star}} = 0$. Using the definition of $\mathcal{Q}$, we obtain

$$\mathcal{Q}(B^\star, C^\star) = \left\|\left|B^\star_{\mathcal{S}} + B^\star_{\mathcal{S}^\perp}\right|\right\|_* + \frac{\lambda_C}{\lambda_B}\left\|C^\star_{\mathcal{J}_{C^\star}} + C^\star_{\mathcal{J}^c_{C^\star}}\right\|_1 = \left\|\left|B^\star_{\mathcal{S}}\right|\right\|_* + \frac{\lambda_C}{\lambda_B}\left\|C^\star_{\mathcal{J}_{C^\star}}\right\|_1,$$

and

$$\begin{aligned}\mathcal{Q}(\widehat{B}, \widehat{C}) &= \mathcal{Q}(B^\star + \Delta^B, C^\star + \Delta^C) \\ &= \left\|\left|B^\star_{\mathcal{S}} + \Delta^B_{\mathcal{S}^\perp_{B^\star}} + \Delta^B_{\mathcal{S}_{B^\star}} + B^\star_{\mathcal{S}^\perp}\right|\right\|_* + \frac{\lambda_C}{\lambda_B}\left\|C^\star_{\mathcal{J}_{C^\star}} + \Delta^C_{\mathcal{J}_{C^\star}} + C^\star_{\mathcal{J}^c_{C^\star}} + \Delta^C_{\mathcal{J}^c_{C^\star}}\right\|_1 \\ &\geq \left\|\left|B^\star_{\mathcal{S}} + \Delta^B_{\mathcal{S}^\perp_{B^\star}}\right|\right\|_* - \left\|\left|\Delta^B_{\mathcal{S}_{B^\star}}\right|\right\|_* + \frac{\lambda_C}{\lambda_B}\left(\left\|C^\star_{\mathcal{J}_{C^\star}} + \Delta^C_{\mathcal{J}_{C^\star}}\right\|_1 + \left\|\Delta^C_{\mathcal{J}^c_{C^\star}}\right\|_1\right) \\ &\geq \left\|\left|B^\star_{\mathcal{S}}\right|\right\|_* + \left\|\left|\Delta^B_{\mathcal{S}^\perp_{B^\star}}\right|\right\|_* - \left\|\left|\Delta^B_{\mathcal{S}_{B^\star}}\right|\right\|_* + \frac{\lambda_C}{\lambda_B}\left(\left\|C^\star_{\mathcal{J}_{C^\star}}\right\|_1 + \|\Delta^C_{\mathcal{J}_{C^\star}}\|_1 - \left\|\Delta^C_{\mathcal{J}^c_{C^\star}}\right\|_1\right).\end{aligned}$$

The decomposition of the first term comes from the construction of $\Delta^B_{\mathcal{S}^\perp_{B^\star}}$. It then follows that

$$\begin{aligned}\mathcal{Q}(B^\star, C^\star) - \mathcal{Q}(\widehat{B}, \widehat{C}) &\le \frac{\lambda_C}{\lambda_B}\left\|C^\star_{\mathcal{J}^\star_C}\right\|_1 + |||\Delta^B_{\mathcal{S}_{B^\star}}|||_* - \left|||\Delta^B_{\mathcal{S}^\perp_{B^\star}}\right|||_* + \frac{\lambda_C}{\lambda_B}\left(\|\Delta^C_{\mathcal{J}_{C^\star}}\|_1 - \left\|\Delta^C_{\mathcal{J}^c_{C^\star}}\right\|_1 - \left\|C^\star_{\mathcal{J}^\star_C}\right\|_1\right) \\ &= |||\Delta^B_{\mathcal{S}_{B^\star}}|||_* + \frac{\lambda_C}{\lambda_B}\|\Delta^C_{\mathcal{J}_{C^\star}}\|_1 - \left(\left|||\Delta^B_{\mathcal{S}^\perp_{B^\star}}\right|||_* + \frac{\lambda_C}{\lambda_B}\left\|\Delta^C_{\mathcal{J}^c_{C^\star}}\right\|_1\right) \\ &= \mathcal{Q}(\Delta^B_{\mathcal{S}_{B^\star}}, \Delta^C_{\mathcal{J}_{C^\star}}) - \mathcal{Q}(\Delta^B_{\mathcal{S}^\perp_{B^\star}}, \Delta^C_{\mathcal{J}^c_{C^\star}}).\end{aligned}$$

■

Lemma 22 *Under the conditions of Theorem 15, the following bound holds:*

$$\frac{1}{T}|||\mathfrak{W}_0(\Delta^B_{aug} + \Delta^C_{aug})|||^2_F \ge \frac{\alpha_{RSC}}{2}(|||\Delta^B|||^2_F + |||\Delta^C|||^2_F) - \frac{\lambda_B}{2}\mathcal{Q}(\Delta^B, \Delta^C).$$

Proof This lemma directly follows from Lemma 2 in Agarwal et al. (2012), by setting $\Theta^\star = B^\star$, $\Gamma^\star = C^\star$, with the regularizer $\mathcal{R}(\cdot)$ being the element-wise ℓ_1 norm. Note that $\sigma_j(B^\star) = 0$ for $j = r+1, \cdots, \min\{p_1, p_2\}$ since $\mathrm{rank}(B) = r$. For our problem, it suffices to set $\mathbb{M}^\perp$ as $\mathcal{J}^c_{C^\star}$, and therefore $\|C^\star_{\mathcal{J}^c_{C^\star}}\|_1 = 0$. ■

Lemma 23 *Consider the two centered Gaussian processes $\{X_t\} \in \mathbb{R}^{p_1}$ and $\{Z_t\} \in \mathbb{R}^{p_2}$, and denote their cross covariance matrix by $\Gamma_{X,Z}(h) = (X_t, Z_{t+h}) = \mathbb{E}(X_t Z'_{t+h})$. Let $\mathcal{X}$ and $\mathcal{Z}$ denote the data matrix. There exist positive constants $c_i > 0$ such that whenever $T \gtrsim c_3(p_1 + p_2)$, with probability at least*

$$1 - c_1 \exp[-c_2(p_1 + p_2)],$$

the following bound holds:

$$\frac{1}{T}\left|||\mathcal{X}'\mathcal{Z}|||_{op} \le \mathbb{Q}_{X,Z}\sqrt{\frac{p_1 + p_2}{T}} + 4|||\Gamma_{X,Z}(0)|||_{op},\right.$$

where

$$\mathbb{Q}_{X,Z} = c_0\left[\mathcal{M}(f_X) + \mathcal{M}(f_Z) + \mathcal{M}(f_{X,Z})\right].$$

Proof The main structure of this proof follows from that of Lemma 3 in Negahban and Wainwright (2011), and here we focus on how to handle the temporal dependency present in our problem. Let $S^p = \{u \in \mathbb{R}^p | \|u\| = 1\}$ denote the p-dimensional unit sphere. The operator norm has the following variational representation form:

$$\frac{1}{T}\left|||\mathcal{X}'\mathcal{Z}|||_{op} = \frac{1}{n}\sup_{u \in S^{p_1}}\sup_{v \in S^{p_2}} u'\mathcal{X}'\mathcal{Z}v.\right.$$

For positive scalars s_1 and s_2, define

$$\Psi(s_1, s_2) = \sup_{u \in s_1 S^{p_1}}\sup_{v \in s_2 S^{p_2}}\langle \mathcal{X}u, \mathcal{Z}v\rangle,$$

and the goal is to establish an upper bound for $\Psi(1,1)/T$. Let $\mathcal{A} = \{u^1, \cdots, u^A\}$ and $\mathcal{B} = \{v^1, \cdots, v^B\}$ denote the 1/4 coverings of S^{p_1} and S^{p_2}, respectively. Negahban and Wainwright (2011) showed that

$$\Psi(1,1) \le 4\max_{u^a \in \mathcal{A}, v^b \in \mathcal{B}}\langle \mathcal{X}u^a, \mathcal{Z}v^b\rangle,$$

and by Anderson et al. (1998) and Anderson (2011), there exists a 1/4 covering of S^{p_1} and S^{p_2} with at most $A \le 8^{p_1}$ and $B \le 8^{p_2}$ elements, respectively. Consequently,

$$\mathbb{P}\left[\left|\frac{1}{T}\Psi(1,1)\right| \ge 4\delta\right] \le 8^{p_1+p_2}\max_{u^a, v^b}\mathbb{P}\left[\frac{|(u^a)'\mathcal{X}\mathcal{Z}(v^b)|}{T} \ge \delta\right].$$

What remains to be bounded is

$$\frac{1}{T}u'\mathcal{X}'\mathcal{Z}v, \qquad \text{for an arbitrary fixed pair of } (u, v) \in S^{p_1} \times S^{p_2}.$$

By Lemma 20, we have

$$\mathbb{P}\left[\left|u'\left(\frac{\mathcal{X}'Z}{T}\right)v\right| > 2\pi\left(\mathcal{M}(f_X) + \mathcal{M}(f_Z) + \mathcal{M}(f_{X,Z})\right)\eta + |||\Gamma_{X,Z}(0)|||_{op}\right] \le 6\exp\left(-cT\min\{\eta, \eta^2\}\right).$$

Therefore, we have

$$\mathbb{P}\left[\left|\frac{1}{T}\Psi(1,1)\right| \ge 8\pi\left(\mathcal{M}(f_X) + \mathcal{M}(f_Z) + \mathcal{M}(f_{X,Z})\right)\eta + 4|||\Gamma_{X,Z}(0)|||_{op}\right] \le 6\exp\left[(p_1 + p_2)\log 8 - cT\min\{\eta, \eta^2\}\right].$$

With the specified choice of sample size T, the probability vanishes by choosing $\eta = c_0\sqrt{\frac{p_1+p_2}{T}}$, for c_0 large enough, and we yield the conclusion in Lemma 23. ■

Lemma 24 *Let $\{X_t\}$ and $\{\varepsilon_t\}$ be two generic processes, where $\varepsilon_t = (U'_t, V'_t)'$. Suppose the spectral density of the joint process (X'_t, ε'_t) exists. Then, the following inequalities hold*

$$\mathfrak{m}(f_{X,V}) \ge \mathfrak{m}(f_{X,\varepsilon}), \qquad \mathcal{M}(f_{X,V}) \le \mathcal{M}(f_{X,\varepsilon}).$$

Proof By definition, the spectral density $f_{X,\varepsilon}(\theta)$ can be written as

$$\begin{aligned}f_{X,\varepsilon}(\theta) &= \left(\frac{1}{2\pi}\right)\sum_{\ell=-\infty}^{\infty}\Gamma_{X,\varepsilon}(\ell)e^{-i\ell\theta}, \qquad \theta \in [-\pi, \pi] \\ &= \left(\frac{1}{2\pi}\right)\sum_{\ell=-\infty}^{\infty}\left(\mathbb{E}X_t U_{t+\ell}, \;\; \mathbb{E}X_t V_{t+\ell}\right)e^{-i\ell\theta} \\ &= \left(f_{X,U}(\theta), \;\; f_{X,V}(\theta)\right).\end{aligned}$$

It follows that

$$\mathcal{M}(f_{X,\varepsilon}) = \operatorname*{ess\,sup}_{\theta \in [-\pi,\pi]}\sqrt{\Lambda_{\max}(H(\theta))},$$

where

$$H(\theta) = \begin{bmatrix} f^*_{X,U}(\theta) \\ f^*_{X,V}(\theta) \end{bmatrix} \begin{bmatrix} f_{X,U}(\theta) & f_{X,V}(\theta) \end{bmatrix} = \begin{bmatrix} f^*_{X,U}(\theta) f_{X,U}(\theta) & f^*_{X,U}(\theta) f_{X,V}(\theta) \\ f^*_{X,V}(\theta) f_{X,U}(\theta) & f^*_{X,V}(\theta) f_{X,V}(\theta) \end{bmatrix}.$$

Note that

$$\mathcal{M}(f_{X,V}) = \operatorname*{ess\,sup}_{\theta \in [-\pi,\pi]} \sqrt{\Lambda_{\max}(f^*_{X,V}(\theta) f_{X,V}(\theta))}.$$

By Lemma **??**, $\forall \theta$, $\Lambda_{\min}(f^*_{X,V}(\theta) f_{X,V}(\theta)) \geq \Lambda_{\min}(H(\theta))$ and $\Lambda_{\max}(f^*_{X,V}(\theta) f_{X,V}(\theta)) \leq \Lambda_{\max}(H(\theta))$, hence

$$\mathfrak{m}(f_{X,V}) \geq \mathfrak{m}(f_{X,\varepsilon}), \qquad \mathcal{M}(f_{X,V}) \leq \mathcal{M}(f_{X,\varepsilon}).$$

■

Appendix D. Proof of Proposition(s).

Proof [**Proof of Proposition 9**] The joint process $W_t = \{(X_t', Z_t')'\}$ is a stationary VAR(1) process, and it follows that

$$S_w(h) := \begin{bmatrix} S_x(h) & S_{x,z}(h) \\ S_{z,x}(h) & S_z(h) \end{bmatrix} = \frac{1}{T} \sum_{t=1}^{T} w_t w_{t+h}' \xrightarrow{p} \Gamma_W(h) := \mathbb{E} W_t W_{t+h}', \quad \text{as} \quad T \to \infty,$$

which implies

$$S_x \xrightarrow{p} \Gamma_X, \quad S_z \xrightarrow{p} \Gamma_Z, \quad S_{x,z} \xrightarrow{p} \Gamma_{X,Z}, \quad S_{x,z}(1) \xrightarrow{p} \Gamma_{X,Z}(1).$$

Note that sample partial regression residual covariances can be obtained by

$$S_{00} = S_z - S_z(1) S_z^{-1} S_z'(1), \qquad S_{11} = S_x - S_{x,z} S_z^{-1} S_{x,z}', \qquad S_{10} = S_{x,z}(1) - S_z(1) S_z^{-1} S_{x,z}'.$$

An application of the Continuous Mapping Theorem yields

$$S_{00} \xrightarrow{p} \Sigma_{00}, \qquad S_{10} \xrightarrow{p} \Sigma_{10}, \qquad S_{11} \xrightarrow{p} \Sigma_{11}.$$

By Hsu (1941a,b), the limiting behavior of $T\Psi_r$ is given by

$$T\Psi_r \sim \chi^2_{(p_1-r)(p_2-r)}, \qquad \text{as} \quad T \to \infty.$$

Note that since μ is of multiplicity one and the ordered eigenvalues are continuous functions of the matrices, the following holds:

$$\phi_k \xrightarrow{p} \mu_k, \qquad \forall\, k = 1, \ldots, \min(p_1, p_2).$$

■

References

Tilak Abeysinghe. Estimation of direct and indirect impact of oil price on growth. *Economics Letters*, 73(2):147–153, 2001.

Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Noisy matrix decomposition via convex relaxation: optimal rates in high dimensions. *The Annals of Statistics*, 40(2): 1171–1197, 2012.

Charles W Anderson, Erik A Stolz, and Sanyogita Shamsunder. Multivariate autoregressive models for classification of spontaneous electroencephalographic signals during mental tasks. *IEEE Transactions on Biomedical Engineering*, 45(3):277–286, 1998.

Theodore Wilbur Anderson. Estimating linear restrictions on regression coefficients for multivariate normal distributions. *The Annals of Mathematical Statistics*, 22(3):327–351, 1951.

Theodore Wilbur Anderson. Canonical correlation analysis and reduced rank regression in autoregressive models. *The Annals of Statistics*, 30(4):1134–1154, 2002.

Theodore Wilbur Anderson. *The Statistical Analysis of Time Series*, volume 19. John Wiley & Sons, 2011.

Ery Arias-Castro, Emmanuel J Candès, and Yaniv Plan. Global testing under sparse alternatives: ANOVA, multiple comparisons and the higher criticism. *The Annals of Statistics*, 39(5):2533–2556, 2011.

Sumanta Basu and George Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535–1567, 2015.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Ben S Bernanke, Jean Boivin, and Piotr Eliasz. Measuring the effects of monetary policy: a factor-augmented vector autoregressive (FAVAR) approach. *The Quarterly Journal of Economics*, 120(1):387–422, 2005.

Carluccio Bianchi, Alessandro Carta, Dean Fantazzini, Maria Elena De Giuli, and Mario A Maggi. A copula-VAR-X approach for industrial production modeling and forecasting. *Applied Economics*, 42(25):3267–3277, 2010.

Monica Billio, Mila Getmansky, Andrew W Lo, and Loriana Pelizzon. Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *Journal of Financial Economics*, 104(3):535–559, 2012.

Celso Brunetti, Jeffrey H Harris, Shawn Mankad, and George Michailidis. Interconnectedness in the interbank market. *Social Sciences Research Network*, 2015.

David O Cushman and Tao Zha. Identifying monetary policy in a small open economy under flexible exchange rates. *Journal of Monetary Economics*, 39(3):433–448, 1997.

David Donoho and Jiashun Jin. Higher criticism for detecting sparse heterogeneous mixtures. *The Annals of Statistics*, 32(3):962–994, 2004.

Roger EA Farmer. The stock market crash really did cause the great recession. *Oxford Bulletin of Economics and Statistics*, 77(5):617–633, 2015.

US Financial Crisis Inquiry Commission. *The financial crisis inquiry report: Final report of the national commission on the causes of the financial and economic crisis in the United States*. Public Affairs, 2011.

Jean-Paul Fitoussi, David Jestaz, Edmund S Phelps, Gylfi Zoega, Olivier Blanchard, and Christopher A Sims. Roots of the recent recoveries: labor reforms or private sector forces? *Brookings Papers on Economic Activity*, 2000(1):237–311, 2000.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Christian Gourieroux and Joann Jasiak. *Financial Econometrics: Problems, Models, and Methods*. Princeton University Press Princeton, NJ, 2001.

Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, 3(3):424–438, 1969.

Peter Hall. Resampling a coverage pattern. *Stochastic Processes and Their Applications*, 20(2):231–246, 1985.

Edward James Hannan. *Multiple Time Series*. John Wiley & Sons, 1970.

Lars Peter Hansen and Thomas J Sargent. *Recursive Models of Dynamic Linear Economies*. Princeton University Press, 2013.

PL Hsu. On the limiting distribution of roots of a determinantal equation. *Journal of the London Mathematical Society*, 1(3):183–194, 1941a.

PL Hsu. On the problem of rank and the limiting distribution of fisher's test function. *The Annals of Eugenics*, 11(1):39–41, 1941b.

Søren Johansen. Statistical analysis of cointegration vectors. *Journal of Economic Dynamics and Control*, 12(2):231–254, 1988.

P.R. Kumar and Pravin Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentice Hall, 1986.

Jiahe Lin, Sumanta Basu, Moulinath Banerjee, and George Michailidis. Penalized maximum likelihood estimation of multi-layered gaussian graphical models. *Journal of Machine Learning Research*, 17(146):1–51, 2016.

Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: provable guarantees with nonconvexity. *The Annals of Statistics*, 40(3):1637–1664, 2012.

R Duncan Luce and Albert D Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, 1949.

Helmut Lütkepohl. *New Introduction to Multiple Time series Analysis*. Springer Science & Business Media, 2005.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Sahand Negahban and Martin J Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.

Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Yurii Nesterov. On an approach to the construction of optimal methods of minimization of smooth convex functions. *Ekonomika i Mateaticheskie Metody*, 24(3):509–517, 1988.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

Yurii Nesterov. Gradient methods for minimizing composite objective function. Technical report, UCL, 2007.

William Nicolson, David Matteson, and Jacob Bien. VARX-L: Structured regularization for large vector autoregressions with exogenous variables. *arXiv preprint 1508.07497*, 2016.

Vincent R Nijs, Marnik G Dekimpe, Jan-Benedict EM Steenkamps, and Dominique M Hanssens. The category-demand effects of price promotions. *Marketing Science*, 20(1): 1–22, 2001.

Koen Pauwels and Allen Weiss. Moving from free to fee: How online firms market to change their business model successfully. *Journal of Marketing*, 72(3):14–31, 2008.

M Hashem Pesaran, Til Schuermann, and Scott M Weiner. Modeling regional interdependencies using a global error-correcting macroeconometric model. *Journal of Business & Economic Statistics*, 22(2):129–162, 2004.

Edmund S Phelps. Behind this structural boom: the role of asset valuations. *The American Economic Review*, 89(2):63–68, 1999.

Dimitris N Politis, Joseph P Romano, and Michael Wolf. *Subsampling*. Springer Series in Statistics, 1999.

Huitong Qiu, Sheng Xu, Fang Han, Han Liu, and Brian Caffo. Robust estimation of transition matrices in high dimensional heavy-tailed vector autoregressive processes. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1843–1851, 2015.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Bruno Rémillard, Nicolas Papageorgiou, and Frédéric Soustra. Copula-based semiparametric models for multivariate time series. *Journal of Multivariate Analysis*, 110:30–42, 2012.

Mark Rudelson and Roman Vershynin. Hanson-Wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 18(82):1–9, 2013.

Anil K Seth. Interoceptive inference, emotion, and the embodied self. *Trends in Cognitive Sciences*, 17(11):565–573, 2013.

Ali Shojaie, Sumanta Basu, and George Michailidis. Adaptive thresholding for reconstructing regulatory networks from time-course gene expression data. *Statistics in Biosciences*, 4(1):66–83, 2012.

Galen R Shorack and Jon A Wellner. *Empirical Processes with Applications to Statistics*. SIAM, 2009.

Christopher A Sims. Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, 48(1):1–48, 1980.

Christopher A Sims. Interpreting the macroeconomic time series facts: The effects of monetary policy. *European Economic Review*, 36(5):975–1000, 1992.

James H Stock and Mark W Watson. An empirical comparison of methods for forecasting using many predictors. *Manuscript, Princeton University*, 2005.

Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report, University of Washington, Seattle, 2008.

Journal of Machine Learning Research 18 (2017) 1-25 Submitted 7/17; Revised 10/17; Published 10/17

Learning Theory of Distributed Regression with Bias Corrected Regularization Kernel Network

Zheng-Chu Guo GUOZHENGCHU@ZJU.EDU.CN
School of Mathematical Sciences, Zhejiang University,
Hangzhou 310027, P. R. China
Lei Shi LEISHI@FUDAN.EDU.CN
Shanghai Key Laboratory for Contemporary Applied Mathematics,
School of Mathematical Sciences, Fudan University,
Shanghai 200433, P. R. China
Qiang Wu QWU@MTSU.EDU
Department of Mathematical Sciences, Middle Tennessee State University,
Murfreesboro, TN 37132, USA

Editor: John Shawe-Taylor

Abstract

Distributed learning is an effective way to analyze big data. In distributed regression, a typical approach is to divide the big data into multiple blocks, apply a base regression algorithm on each of them, and then simply average the output functions learnt from these blocks. Since the average process will decrease the variance, not the bias, bias correction is expected to improve the learning performance if the base regression algorithm is a biased one. Regularization kernel network is an effective and widely used method for nonlinear regression analysis. In this paper we will investigate a bias corrected version of regularization kernel network. We derive the error bounds when it is applied to a single data set and when it is applied as a base algorithm in distributed regression. We show that, under certain appropriate conditions, the optimal learning rates can be reached in both situations.

Keywords: Distributed learning, kernel method, regularization, bias correction, error bound

1. Introduction

Data acquisition become much faster and easier as the development of technology. In this big data era, distributed learning has received considerable attention and is shown to be an effective way to analyze data that is so big and cannot be handled by a single machine. Among various distributed learning paradigms, a simple one is to divide the whole data set into multiple blocks, apply a base learning algorithm to each block, and then average the results from different blocks (Rosenblatt and Nadler, 2016; Zhang et al., 2015). This process, though simple, has some advantages. First, it is computationally efficient because the second stage can be easily parallelized. Second, because no mutual communication is required, the data security or confidentiality can be well protected. Last, recent research shows this method is consistent and sometimes reaches optimal learning rate (Zhang et al., 2015; Lin et al., 2017). Thus its asymptotic effectiveness is theoretically guaranteed.

©2017 Zheng-Chu Guo, Lei Shi and Qiang Wu.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-423.html.

In distributed learning the performance highly depends on the selection of the base algorithm in the second stage. Assume a big data set D of N observations is randomly divided into m blocks, $D_1, D_2, \ldots, D_m$, which are assumed to be of the same size at the moment so that D_i are independent and identically distributed if the entire sample set D is independently drawn from some unknown distribution ρ. Let $\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_m$ be the estimators obtained by applying a base algorithm on these data blocks. Assume each estimator $\hat{f}_i$ has bias b and variance v . Then the mean squared error of $\hat{f}_i$ is $\mathrm{mse}(\hat{f}_i) = b^2+v$ while the average estimator

$$\bar{f} = \frac{1}{m}\sum_{i=1}^{m} \hat{f}_i$$

has $\mathrm{mse}(\bar{f}) = b^2 + \frac{v}{m}$. On a single data block the algorithm usually trades off the bias and variance well to achieve the optimal performance. In distributed learning, however, the variance shrinks fast when m is large but the bias keeps unchanging during the average process. In this case, the bias may dominate the learning performance. An algorithm (or a model selection strategy) that is optimal for a single block is not necessarily still optimal for distributed learning. Instead, distributed learning prefers algorithms of small bias as the base learning algorithm on each block. Therefore, when a base learning algorithm is biased, bias correction is expected to play a role to improve the performance. The purpose of this paper is to investigate the application of biased corrected regularization kernel network for distributed regression analysis.

In regression analysis, the data $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{|D|}, y_{|D|})\}$ is a set of observations collected for input variable X of predictors and a scalar response variable Y, where $|D|$ is the sample size of the data set D. Assume they are linked by

$$y_i = f^*(x_i) + \epsilon_i, \qquad i = 1, 2, \ldots, |D|,$$

where x_i comes from a compact metric space (e.g., a bounded subset in $\mathbb{R}^p$), $y_i \in \mathbb{R}$, and ϵ_i is a zero-mean noise. The target is to recover the unknown true model f^* as accurate as possible to understand the impact of predictors and predict the response on unobserved data. Numerous regression methods have been developed in the literature, e.g. ridge regression, LASSO, and regularization kernel network (RKN). Among them, the regularization kernel network is a popular kernel method for nonlinear regression analysis. Its predictive consistency has been extensively studied in a vast literature; see e.g. Evgeniou et al. (2000); Bousquet and Elisseeff (2002); Zhang (2003); De Vito et al. (2005); Wu et al. (2006); Bauer et al. (2007); Caponnetto and De Vito (2007); Smale and Zhou (2007); Sun and Wu (2009); Steinwart et al. (2009); Guo et al. (2017b) and the references therein. Its applications were also extensively explored and shown successful in many problem domains. More recently, a bias corrected version for RKN, or BCRKN for short, was proposed in Wu (2017) to improve the performance of block wise data processing. In Wu (2017), the asymptotic bias and variance of BCRKN on a single data set was characterized, which indicates BCRKN has smaller bias than RKN and thus implies its efficiency in learning with block wise data intuitively. Empirical study also confirmed this. However, without rigorous analysis of the error bounds, there is lack of theoretical guarantee. In this paper, we will derive the error bounds and learning rates of BCKRN both for a single data set and for distributed

regression. This will provide a theoretical guarantee for the use of BCRKN from a learning theory perspective.

The rest of this paper will be arranged as follows. In Section 2 we will describe the BCRKN algorithm and state our main results. In particular, we show that BCRKN can achieve the minimax optimal rates in both single data learning and distributed learning. Moreover, BCRKN relaxes the saturation effect of RKN. In Section 3 we discuss the relations of our results with existing work and conduct some comparisons. The proofs of our results are given in Sections 4-8.

2. Main results

Let $\mathcal{X}$ denote the input space which is assumed to be a compact metric space. A Mercer kernel on $\mathcal{X}$ is a continuous, symmetric, and positive-semidefinite function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The function class spanned by $\{K_x = K(x,\cdot) : x \in \mathcal{X}\}$ and equipped with the inner product satisfying $\langle K_x, K_t\rangle_K = K(x,t)$ forms a pre-Hilbert space. Its completion is called a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$ associated to the kernel K, with the name coming after the reproducing property $f(x) = \langle f, K(x,\cdot)\rangle_K$, $\forall f \in \mathcal{H}_K$. Note that $|f(x)| \le \sqrt{K(x,x)}\|f\|_K$ for all $f \in \mathcal{H}_K$. Consequently, with $\kappa = \sup_{x\in\mathcal{X}}\sqrt{K(x,x)} < \infty$, $\mathcal{H}_K$ can be embedded into $C(\mathcal{X})$ and $\|f\|_\infty \le \kappa\|f\|_K$. More other properties of RKHS that will not be used in this paper can be found in Aronszajn (1950).

Given the data D and the RKHS $\mathcal{H}_K$, RKN estimates the true model by

$$f_{D,\lambda} = \arg\min_{f\in\mathcal{H}_K} \frac{1}{|D|}\sum_{i=1}^{|D|}(y_i - f(x_i))^2 + \lambda\|f\|_K^2, \tag{1}$$

where $\lambda > 0$ is a regularization parameter that trades off the fitting error and model complexity. The well known representer theorem (Wahba, 1990) tells that

$$f_{D,\lambda}(x) = \sum_{i=1}^{|D|} c_i K(x_i, x)$$

with the coefficients $\mathbf{c} = (c_1, \ldots, c_{|D|})^\top$ satisfying $(\lambda|D|I + \mathbf{K})\mathbf{c} = \mathbf{y}_D$ where I represents an identity matrix (or operator), $\mathbf{K} = (K(x_i, x_j))_{i,j=1}^{|D|}$ is the kernel matrix on the input data $\mathbf{x}_D = \{x_1, \ldots, x_{|D|}\}$ and $\mathbf{y}_D = (y_1, \cdots, y_{|D|})^\top$ is the vector of the response data. Let $S_D : \mathcal{H}_K \to \mathbb{R}^{|D|}$ be the sampling operator defined by

$$S_D f = (f(x_1), \ldots, f(x_{|D|}))^\top, \qquad \forall\ f \in \mathcal{H}_K.$$

Its dual operator S_D^* is given by

$$S_D^*\mathbf{c} = \sum_{i=1}^{|D|} c_i K_{x_i} \in \mathcal{H}_K, \qquad \forall\ \mathbf{c} \in \mathbb{R}^{|D|}.$$

Then $f_{D,\lambda}$ has the following operator representation (Smale and Zhou, 2007)

$$f_{D,\lambda} = \tfrac{1}{|D|}\left(\lambda I + \tfrac{1}{|D|}S_D^* S_D\right)^{-1} S_D^*\mathbf{y}_D. \tag{2}$$

Note that the operator $\frac{1}{|D|}S_D^* S_D$ is a sample version of the integral operator

$$L_K f(x) = \mathbf{E}_t\left[K(x,t)f(t)\right] = \int_{\mathcal{X}} K(x,t)f(t)\mathrm{d}\rho_{\mathcal{X}}(t)$$

where $\rho_{\mathcal{X}}$ is the marginal distribution of ρ on $\mathcal{X}$. Recall that L_K defines a compact, symmetric, and positive operator on $\mathcal{H}_K$. In the sequel we also use the notation $L_{K,D} = \frac{1}{|D|}S_D^* S_D$ and write

$$f_{D,\lambda} = (\lambda I + L_{K,D})^{-1}\left(\tfrac{1}{|D|}S_D^*\mathbf{y}_D\right).$$

By the aid of operator representation (2), the asymptotic bias of RKN can be characterized as $-\lambda(\lambda I + L_K)^{-1}f^*$. The bias corrected regularization kernel network (BCRKN) is defined by subtracting a plug-in estimator of the bias (Wu, 2017)

$$f_{D,\lambda}^\sharp = f_{D,\lambda} + \lambda\left(\lambda I + L_{K,D}\right)^{-1} f_{D,\lambda}. \tag{3}$$

It is also verified in Wu (2017) that

$$f_{D,\lambda}^\sharp(x) = \sum_{i=1}^{n} c_i^\sharp K(x_i, x)$$

with $\mathbf{c}^\sharp = \mathbf{c} + \lambda\left(\lambda I + \frac{1}{n}\mathbf{K}\right)^{-1}\mathbf{c}$. The effectiveness of BCRKN has been tested empirically by a variety of simulations and real applications in Wu (2017). The main purpose of this paper is to verify its effectiveness in distributed regression from a learning theory perspective.

To perform rigorous error analysis and present our main results, we need some notations and assumptions that are used throughout the paper. Firstly, we assume $|y| \le M$ almost surely for some constant $M > 0$. This implies the true regression function f^* satisfies $\|f^*\|_\infty \le M$.

Note that we can extend the domain of L_K to $L^2_{\rho_{\mathcal{X}}}$ and obtain a compact, symmetric, and positive operator on $L^2_{\rho_{\mathcal{X}}}$, which will be denoted by L. We can in turn say L_K is the restriction of L on $\mathcal{H}_K$. So $Lf = L_K f$ for $f \in \mathcal{H}_K$ and we do not need to differentiate them when operating on functions in $\mathcal{H}_K$. Our second assumption is a regularity condition on the true model:

$$f^* = L^r(u^*) \ \text{ for some } r > 0 \text{ and } u^* \in L^2_{\rho_{\mathcal{X}}}. \tag{4}$$

This assumption has been widely used in the literature of learning theory to characterize the approximation ability of $\mathcal{H}_K$; see e.g. De Vito et al. (2005); Smale and Zhou (2007); Bauer et al. (2007); Zhang et al. (2015) and many references therein. Recall that $L^{\frac{1}{2}}$ is an isomorphism from $\overline{\mathcal{H}_K}$ onto $\mathcal{H}_K$, i.e.

$$\|f\|_{L^2_{\rho_{\mathcal{X}}}} = \|L^{\frac{1}{2}}f\|_K, \qquad \text{for } f \in \overline{\mathcal{H}_K}, \tag{5}$$

where $\overline{\mathcal{H}_K}$ is the closure of $\mathcal{H}_K$ in $L^2_{\rho_{\mathcal{X}}}$. So if $r \ge \frac{1}{2}$, the condition (4) implies $f^* \in \mathcal{H}_K$.

We shall use the effective dimension $\mathcal{N}(\lambda) = \mathrm{Tr}((L_K + \lambda I)^{-1}L_K)$, that is, the trace of $(L_K + \lambda I)^{-1}L_K$, to measure the complexity of $\mathcal{H}_K$ with respect to $\rho_{\mathcal{X}}$. We assume that there exist a constant $C_0 > 0$ and some $0 < \beta \le 1$ such that for all $\lambda > 0$

$$\mathcal{N}(\lambda) \le C_0\lambda^{-\beta}. \tag{6}$$

Again this is a natural and widely used assumption in the literature; see e.g. De Vito et al. (2005); Caponnetto and De Vito (2007); Zhang et al. (2013); Lin et al. (2017).

Assume $\kappa \geq 1$ without loss of generality for otherwise all our statements and proofs are still valid by defining $\kappa = \max\{1, \sup_{x\in\mathcal{X}} \sqrt{K(x,x)}\}$. Denote

$$\mathcal{B}_{|D|,\lambda} = \frac{2\kappa}{\sqrt{|D|}} \left\{ \frac{\kappa}{\sqrt{|D|\lambda}} + \sqrt{\mathcal{N}(\lambda)} \right\}.$$

The consistency of RKN as well as BCRKN generally requires the regularization parameter λ to be chosen according to the sample size and satisfy $\lambda \to 0$ and $\lambda|D| \to \infty$ as $|D| \to \infty$. This implies λ is upper bounded by an absolute constant. So, in the sequel, we will assume $\lambda \leq 1$ without loss of generality to simplify our notations and presentations.

As the performance of distributed learning highly depends on the base algorithm, we will conduct a thorough error analysis of BCRKN for a single data set first and then turn to the distributed regression.

2.1 Error bound for learning with a single data set

We derive the following error bounds and learning rates for BCRKN when it is applied on a single data set.

Theorem 1 *If the regularity condition* (4) *holds with* $0 < r \leq 2$ *and* $0 < \lambda \leq 1$*, then for any* $0 < \delta < 1$*, with confidence at least* $1 - \delta$*,*

$$\|f_{D,\lambda}^\sharp - f^*\|_{L^2_{\rho_\mathcal{X}}} \leq C \left(\frac{\mathcal{B}_{|D|,\lambda}}{\sqrt{\lambda}} + 1 \right)^3 \left(\mathcal{B}_{|D|,\lambda} + \lambda^r \right) \left(\log \frac{4}{\delta} \right)^4, \tag{7}$$

where C *is a constant independent of* $|D|$ *or* δ*. Consequently, we have*

$$\mathbf{E}\left[\|f_{D,\lambda}^\sharp - f^*\|^2_{L^2_{\rho_\mathcal{X}}} \right] \leq 4\Gamma(9) C^2 \left(\frac{\mathcal{B}_{|D|,\lambda}}{\sqrt{\lambda}} + 1 \right)^6 \left(\mathcal{B}_{|D|,\lambda} + \lambda^r \right)^2. \tag{8}$$

Corollary 2 *Assume the regularity condition* (4) *holds with* $0 < r \leq 2$ *and* (6) *holds with* $0 < \beta \leq 1$*.*

(i) If $0 < r < \frac{1}{2}$*, choose* $\lambda = |D|^{-\frac{1}{1+\beta}}$*. Then for any* $0 < \delta < 1$*, with confidence at least* $1 - \delta$*, we have*

$$\|f_{D,\lambda}^\sharp - f^*\|_{L^2_{\rho_\mathcal{X}}} \leq C_1 |D|^{-\frac{r}{1+\beta}} \left(\log \frac{4}{\delta} \right)^4,$$

where C_1 *is a constant independent of* $|D|$ *or* δ*. Consequently,*

$$\mathbf{E}\left[\|f_{D,\lambda}^\sharp - f^*\|^2_{L^2_{\rho_\mathcal{X}}} \right] = \mathcal{O}\left(|D|^{-\frac{2r}{1+\beta}} \right).$$

(ii) If $\frac{1}{2} \leq r \leq 2$*, choose* $\lambda = |D|^{-\frac{1}{2r+\beta}}$*. Then for any* $0 < \delta < 1$*, with confidence at least* $1 - \delta$*, we have*

$$\|f_{D,\lambda}^\sharp - f^*\|_{L^2_{\rho_\mathcal{X}}} \leq C_2 |D|^{-\frac{r}{2r+\beta}} \left(\log \frac{4}{\delta} \right)^4,$$

where C_2 *is a constant independent of* $|D|$ *or* δ*. Consequently,*

$$\mathbf{E}\left[\|f_{D,\lambda}^\sharp - f^*\|^2_{L^2_{\rho_\mathcal{X}}} \right] = \mathcal{O}\left(|D|^{-\frac{2r}{2r+\beta}} \right).$$

Recall that the minimax optimal learning rate under the assumptions (4) and (6) is $\mathcal{O}\left(|D|^{-\frac{2r}{2r+\beta}} \right)$. Theorem 1 tells that, when $r \geq \frac{1}{2}$, BCRKN achieves the minimax optimal learning rate on a single data set.

Since $f^* \in \mathcal{H}_K$ when $r \geq \frac{1}{2}$, we can also measure the convergence of $f_{D,\lambda}^\sharp$ to f^* in $\mathcal{H}_K$. As pointed out in Smale and Zhou (2007), the convergence in $\mathcal{H}_K$ implies the convergence in $C^s(\mathcal{X})$ if $K \in C^{2s}(\mathcal{X}\times\mathcal{X})$, here $C^s(\mathcal{X})$ is the space of all functions on $\mathcal{X} \subset \mathbb{R}^p$ whose partial derivatives up to order s are continuous with $\|f\|_{C^s(\mathcal{X})} = \sum_{|\alpha|\leq s} \|D^\alpha f\|_\infty$. So the convergence in $\mathcal{H}_K$ is much stronger. It is not only for the target function itself, but also for its derivatives.

Theorem 3 *If the regularity condition* (4) *holds with* $\frac{1}{2} < r \leq 2$*, then for any* $0 < \delta < 1$ *with confidence at least* $1 - \delta$*,*

$$\|f_{D,\lambda}^\sharp - f^*\|_K \leq C_K \left(\frac{\mathcal{B}_{|D|,\lambda}}{\sqrt{\lambda}} + 1 \right)^2 \left(\lambda^{-\frac{1}{2}} \mathcal{B}_{|D|,\lambda} + \lambda^{r-\frac{1}{2}} \right) \left(\log \frac{4}{\delta} \right)^3, \tag{9}$$

where C_K *is a constant independent of* $|D|$ *or* δ*. If* (6) *holds with* $0 < \beta \leq 1$ *and* $\lambda = |D|^{-\frac{1}{2r+\beta}}$*, then for any* $0 < \delta < 1$*, with confidence at least* $1 - \delta$*, we have*

$$\|f_{D,\lambda}^\sharp - f^*\|_K \leq \tilde{C}_K |D|^{-\frac{r-\frac{1}{2}}{2r+\beta}} \left(\log \frac{4}{\delta} \right)^3, \tag{10}$$

where $\tilde{C}_K$ *is a constant independent of* $|D|$ *or* δ*. Moreover,*

$$\mathbf{E}\left[\|f_{D,\lambda}^\sharp - f^*\|^2_K \right] = \mathcal{O}\left(|D|^{-\frac{2r-1}{2r+\beta}} \right). \tag{11}$$

Under the assumptions (4) with $r > \frac{1}{2}$ and (6) with $0 < \beta \leq 1$, the minimax optimality of the bound $\mathcal{O}\left(|D|^{-\frac{2r-1}{2r+\beta}} \right)$ in the $\mathcal{H}_K$-metric has been proved in Guo et al. (2016). Theorem 3 indicates that the stronger convergence of BCRKN is also rate optimal in the minimax sense.

When $0 < r < \frac{1}{2}$, we are unfortunately not able to obtain the minimax rate by the integral operator technique under the assumption (6). Note that if $\mathcal{H}_K$ is finite dimensional the range of L^r is exactly $\mathcal{H}_K$ for all $r > 0$. The assumption (4) always implies $f^* \in \mathcal{H}_K$. So the situation $0 < r < \frac{1}{2}$ makes sense only when $\mathcal{H}_K$ is infinite dimensional. In this case, L_K has infinite positive eigenvalues which converge to 0. This imposes the main difficulty of error analysis via integral technique – although $L_{K,D}$ converges well to L_K at a rate $\mathcal{O}(|D|^{-1/2})$, the difference of $(\lambda I + L_{K,D})^{-1}$ and $(\lambda I + L_K)^{-1}$ cannot be well bounded when $\lambda \to 0$. Actually, even for RKN which has been exhaustedly studied in the literature, it is an open problem to obtain the minimax rate under the assumptions (4) and (6). However, if there is sufficient amount of unlabeled data which helps to improve the estimate of the integral operator, minimax rate can be achieved, as has been verified in

Blanchard and Krämer (2010). For this purpose we propose the following semi-supervised approach. Assume, in addition to the labeled data D, we have a sequence of unlabelled data $x_{|D|+1}, \ldots, x_{|D'|}$. We create a fully labeled data set

$$D' = \{(x_1, y_1'), \cdots, (x_{|D|}, y_{|D|}'), (x_{|D|+1}, 0), \cdots, (x_{|D'|}, 0)\},$$

where $y_i' = \frac{|D'|}{|D|} y_i$ for $1 \le i \le |D|$. We can apply RKN and BCRKN on D' to obtain semi-supervised estimators $f_{D',\lambda}$ and $f^\sharp_{D',\lambda}$. Note that $D = D'$ when $|D'| = |D|$. So the semi-supervised method can be regarded as an extension of the supervised method while the supervised method is a special case of the semi-supervised method with no unlabeled data. The next theorem confirms that BCRKN can achieve the minimax rate for $0 < r < \frac{1}{2}$ when there are enough unlabeled data.

Theorem 4 *Assume the regularity condition* (4) *with* $0 < r < \frac{1}{2}$. *For any* $0 < \delta < 1$, *we have with confidence at least* $1 - \delta$,

$$\|f^\sharp_{D',\lambda} - f^*\|^2_{L^2_{\rho_\mathcal{X}}} \le \left(\frac{2M}{\kappa} + 4\|u^*\|^2_{L^2_{\rho_\mathcal{X}}}\right) \left(\frac{\mathcal{B}_{|D'|,\lambda}}{\sqrt{\lambda}} + 1\right)^3 \left(\mathcal{B}_{|D|,\lambda} + \lambda^r\right) \left(\log \frac{4}{\delta}\right)^3. \tag{12}$$

If in addition (6) *holds with* $0 < \beta \le 1$ *and* $r + \beta \ge \frac{1}{2}$, $\lambda = |D|^{-\frac{1}{2r+\beta}}$, $|D'| \ge |D|^{\frac{1+\beta}{2r+\beta}}$, *then for any* $\delta \in (0,1)$, *with confidence at least* $1 - \delta$, *there holds*

$$\|f^\sharp_{D',\lambda} - f^*\|^2_{L^2_{\rho_\mathcal{X}}} \le C' |D|^{-\frac{r}{2r+\beta}} \left(\log \frac{4}{\delta}\right)^3 \tag{13}$$

where the constant C' *is independent of* δ, $|D|$ *or* $|D'|$ *and will be given explicitly in the proof.*

2.2 Error bound of distributed regression with BCRKN

When BCRKN is used as a base algorithm for distributed regression, a big data set D is split into m blocks $D_1, D_2, \ldots, D_m$. On each block D_j, BCRKN is applied to produce an estimator $f^\sharp_{D_j,\lambda}$, and the weighted average of $f^\sharp_{D_j,\lambda}$,

$$\overline{f}^\sharp_{D,\lambda} = \sum_{j=1}^m \frac{|D_j|}{|D|} f^\sharp_{D_j,\lambda}, \tag{14}$$

is used for the purposes of prediction and inference. For this divide-and-conquer approach, we first give a general error bound for an arbitrary m. Here we do not require each block to have the same sample size.

Theorem 5 *If the regularity condition* (4) *holds with* $\frac{1}{2} \le r \le 2$ *and* $\lambda \le 1$, *then there exists a constant* $\bar{C}$ *independent of* m *or* $|D_j|$ *such that*

$$\mathbf{E}\left[\|\overline{f}^\sharp_{D,\lambda} - f^*\|^2_{L^2_{\rho_\mathcal{X}}}\right] \le \bar{C} \sum_{j=1}^m \frac{|D_j|}{|D|} \left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}} + 1\right)^6 \left(\frac{|D_j|}{|D|} \mathcal{B}^2_{|D_j|,\lambda} + \frac{\lambda^2 \mathcal{N}(\lambda)}{|D|} + \lambda^{2r}\right).$$

We next show that the distributed BCRKN (14) can achieve the optimal learning rate provided that m is not too large.

Theorem 6 *Assume the regularity condition* (4) *with* $\frac{1}{2} \le r \le 2$. *If* (6) *holds with* $0 < \beta \le 1$, $|D_1| = |D_2| = \cdots = |D_m|$, $\lambda = |D|^{-\frac{1}{2r+\beta}}$, *and the number of the local machines satisfies*

$$m \le |D|^{\min\left\{\frac{2}{2r+\beta}, \frac{2r-1}{2r+\beta}\right\}}, \tag{15}$$

then

$$\mathbf{E}\left[\|\overline{f}^\sharp_{D,\lambda} - f^*\|^2_{L^2_{\rho_\mathcal{X}}}\right] = \mathcal{O}\left(|D|^{-\frac{2r}{2r+\beta}}\right).$$

3. Relations to existing work and discussions

The minimax analysis of regularized least square algorithm has received attention in statistics and learning theory literature (DeVore et al., 2004; Györfi et al., 2006; Temlyakov, 2008; Caponnetto and De Vito, 2007; Steinwart et al., 2009). In particular, assume L_K admits an eigendecomposition $L_K = \sum_{i=1}^\infty \tau_i \phi_i \otimes \phi_i$, where $\tau_i \ge 0$ and ϕ_i are the eigenvalues and eigenfunctions of L_K, respectively. It is proved in Caponnetto and De Vito (2007) that, if the regularity condition (4) holds with some $r \ge \frac{1}{2}$ and the eigenvalues satisfy $\tau_i \sim i^{-2\alpha}$ for some $\alpha > \frac{1}{2}$, then the minimax optimal learning rate of regularized least square algorithm is $\mathcal{O}(|D|^{-\frac{2\alpha}{4\alpha r+1}})$. It is also proved that RKN can achieve minimax rate if $\frac{1}{2} < r \le 1$. When $r = \frac{1}{2}$, they obtained a suboptimal rate $\mathcal{O}\left(\left(\frac{\log |D|}{|D|}\right)^{-\frac{2\alpha}{2\alpha+1}}\right)$. In Steinwart et al. (2009), under the additional restriction

$$\|f\|_\infty \le C \|f\|_K^{\frac{1}{2\alpha}} \|f\|^{1-\frac{1}{2\alpha}}_{L^2_{\rho_\mathcal{X}}}, \qquad \forall\, f \in \mathcal{H}_K,$$

it is proved that the projected (or clipped) RKN estimator can achieve the minimax learning rate. More recently, Lin et al. (2017) proved that RKN can achieve minimax learning rate for r in the whole range of $[\frac{1}{2}, 1]$ without any restrictions except for the conditions (4) and $\tau_i \sim i^{-2\alpha}$. It improves the results in Caponnetto and De Vito (2007); Steinwart et al. (2009). When $r \ge 1$, RKN suffers the saturation effect and the learning rate will not improve. Note our condition (6) on the effective dimension is nearly equivalent to $\tau_i \sim i^{-2\alpha}$ with $\beta = \frac{1}{2\alpha}$. The result in Corollary 2 tells that BCRKN can achieve the minimax learning rate for $r \in [\frac{1}{2}, 2]$ and thus relaxes the saturation effect of RKN.

For distributed regression problem, assume all data blocks D_i, $i = 1, \ldots, m$, are of equal size. If RKN is used as the base algorithm, under the assumptions that $\mathbf{E}[|\phi_i(x)|^{2k}] \le A^{2k}$ for some $k > 2$ and constant $A < \infty$, $\lambda_i \le a i^{-2\alpha}$, and $f^* \in \mathcal{H}_K$ (i.e. $r = \frac{1}{2}$), it is proved in Zhang et al. (2015) that the optimal learning rate of $\mathcal{O}(n^{-\frac{2\alpha}{2\alpha+1}})$ can be achieved by choosing $\lambda = |D|^{-\frac{2\alpha}{2\alpha+1}}$ and restricting the number of local processors

$$m \le c_\alpha \left(\frac{|D|^{\frac{2(k-4)\alpha-k}{2\alpha+1}}}{A^{4k} \log^k |D|}\right)^{\frac{1}{k-2}}.$$

Later in Lin et al. (2017) the regularity condition (4) was taken into consideration and it is proved that the distributed regression can achieve the minimax optimal rate for all $r \in [\frac{1}{2}, 1]$ if

$$m \leq |D|^{\min\{\frac{6\alpha(2r-1)+1}{5(4\alpha r+1)}, \frac{2\alpha(2r-1)}{4\alpha r+1}\}}. \tag{16}$$

The method suffers from the saturation effect inherited from RKN. So the learning rate cannot improve with $r > 1$. When BCRKN is applied as the base algorithm for distributed regression, the saturation effect is relaxed and the minimax optimal learning rate can be achieved for the whole range $r \in [\frac{1}{2}, 2]$ as in the single data learning case. Compare (15) with (16) and we see our analysis also relaxes the restriction on the number m of local processors.

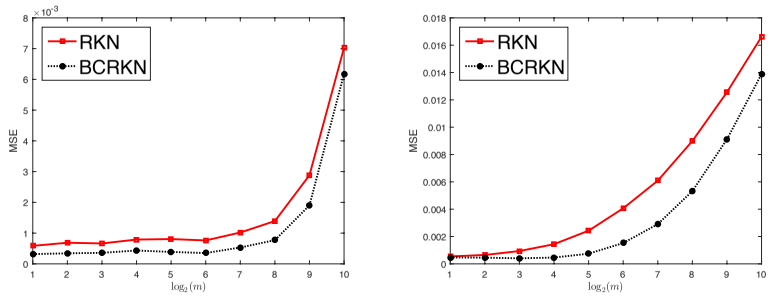
When $r < 1$ we notice that distributed regression with RKN and BCRKN both reach the optimal rates by underregularization, that is, selecting the regularization parameter according to the number of all observations $|D|$, not the number of observations in each block $|D_i|$. But due to the reduced bias the parameter selection of BCRKN is less sensitive and thus could be advantageous in practice. We show this by an illustrative example used in Zhang et al. (2015). Consider the model $f^*(x) = \min\{x, 1-x\}$ with $x \sim \text{Uniform}[0,1]$ and the noise $\epsilon \sim N(0, \sigma^2)$ with $\sigma^2 = \frac{1}{5}$. Let $K(x,t) = 1 + \min\{x,t\}$. Then $f^* \in \mathcal{H}_K$ and $\|f^*\|_K = 1$. We first compare the distributed RKN and the distributed BCRKN when $\lambda = |D|^{-2/3}$, a theoretically optimal choice. We generate $|D| = 4098$ sample points and use number of partitions $m \in \{2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$. The mean squared errors of two methods are plotted in Figure 1 (a). We see BCRKN slightly outperforms RKN for all m.

Recall that the analyses in Zhang et al. (2015); Lin et al. (2017) and this paper indicate the optimal choice of the regularization parameter is $\lambda = |D|^{-\theta}$ with θ an index depending on the regularity of the true target function f^* and the effective dimension of the integral operator L_K. Clearly both are unknown in practice and thus a theoretical optimal choice of the regularization parameter is actually not available. At the same time, in a big data setting where distributed regression is necessary globally tuning the optimal parameter is either impossible or too time consuming. A reasonable way is to tune the parameter locally to get optimal choice $\lambda_i = |D_i|^{-\theta}$ on D_i and then underregularize it using

$$\lambda = \lambda_i^{\frac{\log|D|}{\log|D_i|}} = |D|^{-\theta}.$$

So we next compare the use of RKN and BCRKN in distributed regression when this parameter selection strategy is used. The results are shown in Figure 1 (b). We see the requirement on the number of local processors becomes more restrictive for both methods, indicating that underregularizing locally optimal parameter does not lead to globally optimal parameter. BCRKN significantly outperforms RKN as m increases, indicating it is less sensitive to the parameter selection when a globally optimal parameter is not available.

Note that the upper bounds on the number of local processors are constrained by the analysis techniques, not necessarily reflect the true limit on the number of local processors allowed in practice. Also, since underregularization is necessary in distributed regression but globally tuning the optimal parameter is impractical, the impact of parameter selection strategy on the number of local processors is unknown. Further investigation on these issues is of great interest in future research.



(a) (b)

Figure 1: MSE of distributed RKN and distributed BCRKN. (a) $\lambda = |D|^{-2/3}$ is used. (b) λ is first tuned locally and the underregularized.

Finally, it is worth mentioning that debiasing techniques have also been developed to improve the accuracy of statistical inference of distributed lasso and received considerable attention recently (Bühlmann, 2013; Zhang and Zhang, 2014; Javanmard and Montanari, 2014; Lee et al., 2017; Battey et al., 2015).

4. Preliminary lemmas

The following lemmas can be found in Lin et al. (2017); Guo et al. (2017a).

Lemma 7 *Let D be a sample drawn independently according to ρ and g be a measurable bounded function on $\mathcal{Z}$ and ξ_g be a random variable with values on $\mathcal{H}_K$ given by $\xi_g(z) = g(z)K_x$ for $z = (x,y) \in \mathcal{Z}$. For any $0 < \delta < 1$, with confidence at least $1-\delta$, there holds*

$$\left\| (\lambda I + L_K)^{-\frac{1}{2}} \left(\frac{1}{|D|} \sum_{z \in D} \xi_g(z) - \mathbf{E}[\xi_g] \right) \right\|_K \leq \frac{\|g\|_\infty \log\frac{2}{\delta}}{\kappa} \mathcal{B}_{|D|,\lambda}.$$

Lemma 8 *Let D be a sample drawn independently according to ρ. If $|y| \leq M$ almost surely, then with confidence at least $1-\delta$, there holds*

$$\left\| (\lambda I + L_K)^{-\frac{1}{2}} (\tfrac{1}{|D|} S_D^* \mathbf{y}_D - L_{K,D} f^*) \right\|_K \leq 2M \mathcal{B}_{|D|,\lambda} \log\frac{2}{\delta}.$$

Lemma 9 *Let D be a sample drawn independently according to ρ. If $|y| \leq M$ almost surely, then for any $0 < \delta < 1$, with confidence at least $1-\delta$, there holds*

$$\Xi_D := \|(\lambda I + L_K)^{-\frac{1}{2}} (L_K - L_{K,D})\| \leq \mathcal{B}_{|D|,\lambda} \log\frac{2}{\delta}, \tag{17}$$

If A and B are invertible operators on a Banach space, then by the second order operator decomposition proposed in Lin et al. (2017), we have

$$A^{-1} - B^{-1} = B^{-1}(B-A)B^{-1}(B-A)A^{-1} + B^{-1}(B-A)B^{-1}. \tag{18}$$

This implies the following decomposition of the operator product

$$BA^{-1} = (B-A)B^{-1}(B-A)A^{-1} + (B-A)B^{-1} + I. \tag{19}$$

With $A = L_{K,D}+\lambda I$ and $B = L_K+\lambda I$ in (19), and applying Lemma 9, we have the following bound for $\|(L_K+\lambda I)(L_{K,D}+\lambda I)^{-1}\|$; for the detailed proof see Guo et al. (2017a).

Proposition 10 *For any $0<\delta<1$, with confidence at least $1-\delta$, there holds*

$$\Omega_D := \|(L_K+\lambda I)(L_{K,D}+\lambda I)^{-1}\| \le \left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac{2}{\delta}}{\sqrt{\lambda}}+1\right)^2.$$

Moreover, the confidence set is the same as that in Lemma 9.

Lemma 11 *Let Q be positive random variable. If there are constants $a>0, b>0, \tau>0$ such that for any $0<\delta\le 1$, with confident at least $1-\delta$, there holds $Q \le a(\log\frac{b}{\delta})^\tau$, then for any $s>0$ we have $\mathbf{E}[Q^s] \le a^s b\Gamma(\tau s+1)$.*

Proof Note the condition implies that for all $t>0$ there is

$$\Pr\left[Q^{\frac{1}{\tau}} > t\right] \le b\exp\left(-\frac{t}{a^{1/\tau}}\right).$$

So we have

$$\begin{aligned}\mathbf{E}[Q^s] &= \mathbf{E}\left[\left(Q^{1/\tau}\right)^{\tau s}\right] = \tau s\int_0^\infty t^{\tau s-1}\Pr\left[Q^{\frac{1}{\tau}}>t\right]dt\\ &\le \tau s b\int_0^\infty t^{\tau s-1}\exp\left(-\frac{t}{a^{1/\tau}}\right)dt\\ &= b\tau s\Gamma(\tau s)a^s = a^s b\Gamma(\tau s+1).\end{aligned}$$

This proves the lemma. ∎

5. Error analysis of BCRKN in $L^2_{\rho_{\mathcal{X}}}$ when $r \ge \frac{1}{2}$

We will split the proof of Theorem 1 into three cases: $0<r<\frac{1}{2}$, $\frac{1}{2}\le r\le\frac{3}{2}$, and $\frac{3}{2}<r\le 2$. In this section we prove it for the second and third cases while leave the first case to Section 7. Denote $\Delta_D = \frac{1}{|D|}S_D^*(\mathbf{y}_D - S_D f^*)$.

Proposition 12 *If $\frac{1}{2}\le r\le\frac{3}{2}$, we have*

$$\|f^\sharp_{D,\lambda} - f^*\|_{L^2_{\rho_{\mathcal{X}}}} \le 2\Omega_D\left\|(\lambda I+L_K)^{-\frac{1}{2}}\Delta_D\right\|_K + \lambda^r(\Omega_D)^r\|u^*\|_{L^2_{\rho_{\mathcal{X}}}}.$$

Proof By the triangle inequality, we have

$$\left\|f^\sharp_{D,\lambda} - f^*\right\|_{L^2_{\rho_{\mathcal{X}}}} \le \left\|f^\sharp_{D,\lambda} - \mathbf{E}^*[f^\sharp_{D,\lambda}]\right\|_{L^2_{\rho_{\mathcal{X}}}} + \left\|\mathbf{E}^*[f^\sharp_{D,\lambda}] - f^*\right\|_{L^2_{\rho_{\mathcal{X}}}}, \tag{20}$$

where $\mathbf{E}^*[f^\sharp_{D,\lambda}] = (2\lambda I+L_{K,D})(\lambda I+L_{K,D})^{-2}L_{K,D}f^*$ is the conditional expectation with respect to $\mathbf{y}_D$ given $\mathbf{x}_D$.

For the first term $\left\|f^\sharp_{D,\lambda} - \mathbf{E}^*[f^\sharp_{D,\lambda}]\right\|_{L^2_{\rho_{\mathcal{X}}}}$, noting that $2\lambda I+L_{K,D}$ and $(\lambda I+L_{K,D})^{-1}$ commute, we have

$$\begin{aligned}&\left\|f^\sharp_{D,\lambda} - \mathbf{E}^*[f^\sharp_{D,\lambda}]\right\|_{L^2_{\rho_{\mathcal{X}}}} = \left\|L_K^{\frac{1}{2}}(2\lambda I+L_{K,D})(\lambda I+L_{K,D})^{-2}\Delta_D\right\|_K\\ \le\ &\left\|(\lambda I+L_K)^{\frac{1}{2}}(\lambda I+L_{K,D})^{-\frac{1}{2}}\right\|\left\|(\lambda I+L_{K,D})^{-\frac{1}{2}}(2\lambda I+L_{K,D})(\lambda I+L_{K,D})^{-\frac{1}{2}}\right\|\\ &\times\left\|(\lambda I+L_{K,D})^{-\frac{1}{2}}(\lambda I+L_K)^{\frac{1}{2}}\right\|\left\|(\lambda I+L_K)^{-\frac{1}{2}}\Delta\right\|_K\\ \le\ &2\Omega_D\left\|(\lambda I+L_K)^{-\frac{1}{2}}\Delta_D\right\|_K,\end{aligned} \tag{21}$$

here we have used the fact (Blanchard and Krämer, 2010) that

$$\|A^sB^s\| \le \|AB\|^s, \qquad 0\le s\le 1,$$

for positive operators A and B on Hilbert spaces.

For the second term, we have

$$\mathbf{E}^*[f^\sharp_{D,\lambda}] - f^* = [(2\lambda I+L_{K,D})(\lambda I+L_{K,D})^{-2}L_{K,D} - I]f^* = \lambda^2(\lambda I+L_{K,D})^{-2}f^*.$$

By the regularity condition (4),

$$\begin{aligned}\left\|\mathbf{E}^*[f^\sharp_{D,\lambda}] - f^*\right\|_{L^2_{\rho_{\mathcal{X}}}} &= \lambda^2\left\|(\lambda I+L_{K,D})^{-2}L^r u^*\right\|_{L^2_{\rho_{\mathcal{X}}}}\\ &\le \lambda^2\left\|(\lambda I+L_K)^{\frac{1}{2}}(\lambda I+L_{K,D})^{-2}L_K^{r-\frac{1}{2}}L^{\frac{1}{2}}u^*\right\|_K\\ &\le \lambda^2\left\|(\lambda I+L_K)^{\frac{1}{2}}(\lambda I+L_{K,D})^{-\frac{1}{2}}\right\|\left\|(\lambda I+L_{K,D})^{-\frac{3}{2}}L_K^{r-\frac{1}{2}}\right\|\left\|L^{\frac{1}{2}}u^*\right\|_K\\ &\le \lambda^2(\Omega_D)^{\frac{1}{2}}\left\|(\lambda I+L_{K,D})^{-\frac{3}{2}}L_K^{r-\frac{1}{2}}\right\|\|u^*\|_{L^2_{\rho_{\mathcal{X}}}}.\end{aligned} \tag{22}$$

Since $\frac{1}{2}\le r\le\frac{3}{2}$, we have

$$\begin{aligned}&\left\|(\lambda I+L_{K,D})^{-\frac{3}{2}}L_K^{r-\frac{1}{2}}\right\|\\ =\ &\left\|(\lambda I+L_{K,D})^{r-2}(\lambda I+L_{K,D})^{-r+\frac{1}{2}}(\lambda I+L_K)^{r-\frac{1}{2}}(\lambda I+L_K)^{-r+\frac{1}{2}}L_K^{r-\frac{1}{2}}\right\|\end{aligned}$$

$$\leq \left\|(\lambda I+L_{K,D})^{r-2}\right\|\left\|(\lambda I+L_{K,D})^{-r+\frac{1}{2}}(\lambda I+L_K)^{r-\frac{1}{2}}\right\|\left\|(\lambda I+L_K)^{-r+\frac{1}{2}}L_K^{r-\frac{1}{2}}\right\|$$
$$\leq \lambda^{r-2}(\Omega_D)^{r-\frac{1}{2}}.$$

Therefore,

$$\left\|\mathbf{E}^*[f^\sharp_{D,\lambda}]-f^*\right\|_{L^2_{\rho_\mathcal{X}}}\leq \lambda^r(\Omega_D)^r\|u^*\|_{L^2_{\rho_\mathcal{X}}}. \tag{23}$$

Then the conclusion follows by combining (21) and (23). ■

Proposition 13 *If $\frac{3}{2}<r\leq 2$, we have*

$$\|f^\sharp_{D,\lambda}-f^*\|_{L^2_{\rho_\mathcal{X}}}\leq 2\Omega_D\left\|(\lambda I+L_K)^{-\frac{1}{2}}\Delta_D\right\|_K+\lambda\Xi_D(\Omega_D)^{\frac{3}{2}}\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}+\lambda^r\Omega_D\|u^*\|_{L^2_{\rho_\mathcal{X}}}.$$

Proof The proof is similar to Proposition 12. First, $\|f^\sharp_{D,\lambda}-f^*\|_{L^2_{\rho_\mathcal{X}}}$ can be divided into two terms by (20). The first term has been estimated in Proposition 12 as (21). We now focus on the second term. To this end, by (22), we only need to estimate $\left\|(\lambda I+L_{K,D})^{-\frac{3}{2}}L_K^{r-\frac{1}{2}}\right\|$. When $\frac{3}{2}\leq r<2$, we have

$$\begin{aligned}
&(\lambda I+L_{K,D})^{-\frac{3}{2}}L_K^{r-\frac{1}{2}}\\
=\;&(\lambda I+L_{K,D})^{-\frac{1}{2}}\left[(\lambda I+L_{K,D})^{-1}-(\lambda I+L_K)^{-1}\right]L_K^{r-\frac{1}{2}}\\
&+(\lambda I+L_{K,D})^{-\frac{1}{2}}(\lambda I+L_K)^{-1}L_K^{r-\frac{1}{2}}\\
=\;&(\lambda I+L_{K,D})^{-\frac{1}{2}}(\lambda I+L_K)^{-1}(L_K-L_{K,D})(\lambda I+L_{K,D})^{-1}(\lambda I+L_K)(\lambda I+L_K)^{-1}L_K^{r-\frac{1}{2}}\\
&+(\lambda I+L_{K,D})^{-\frac{1}{2}}(\lambda I+L_K)^{\frac{1}{2}}(\lambda I+L_K)^{-\frac{3}{2}}L_K^{r-\frac{1}{2}}.
\end{aligned}$$

By the bounds $\left\|(\lambda I+L_{K,D})^{-\frac{1}{2}}\right\|\leq\frac{1}{\sqrt{\lambda}}$, $\left\|(\lambda I+L_K)^{-\frac{1}{2}}\right\|\leq\frac{1}{\sqrt{\lambda}}$, and $\|L_K\|\leq\kappa^2$, we have,

$$\begin{aligned}
&\left\|(\lambda I+L_{K,D})^{-\frac{3}{2}}L_K^{r-\frac{1}{2}}\right\|\\
\leq\;&\frac{1}{\lambda}\left\|(\lambda I+L_K)^{-\frac{1}{2}}(L_K-L_{K,D})\right\|\left\|(\lambda I+L_{K,D})^{-1}(\lambda I+L_K)\right\|\left\|(\lambda I+L_K)^{-1}L_K^{r-\frac{1}{2}}\right\|\\
&+\left\|(\lambda I+L_{K,D})^{-\frac{1}{2}}(\lambda I+L_K)^{\frac{1}{2}}\right\|\left\|(\lambda I+L_K)^{-\frac{3}{2}}L_K^{r-\frac{1}{2}}\right\|\\
\leq\;&\lambda^{-1}\Omega_D\left\|(\lambda I+L_K)^{-\frac{1}{2}}(L_K-L_{K,D})\right\|\kappa^{2r-3}+\lambda^{r-2}(\Omega_D)^{\frac{1}{2}}.
\end{aligned}$$

Therefore, putting the above bound back into (22) yields

$$\begin{aligned}
\left\|\mathbf{E}^*[f^\sharp_{D,\lambda}]-f^*\right\|_{L^2_{\rho_\mathcal{X}}}\leq\;&\lambda\left\|(\lambda I+L_K)^{-\frac{1}{2}}(L_K-L_{K,D})\right\|(\Omega_D)^{\frac{3}{2}}\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}\\
&+\lambda^r\Omega_D\|u^*\|_{L^2_{\rho_\mathcal{X}}}.
\end{aligned}\tag{24}$$

Now the conclusion follows by plugging (21) and (24) into (20). ■

Now we are ready to prove Theorem 1 and Corollary 2 for $r\geq\frac{1}{2}$.

Proof of Theorem 1: Case $\frac{1}{2}\leq r\leq 2$. By Lemma 8, we have with confidence at least $1-\frac{\delta}{2}$,

$$\left\|(\lambda I+L_K)^{-\frac{1}{2}}\Delta_D\right\|_K\leq 2M\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}. \tag{25}$$

By Lemma 9 and Proposition 10, we obtain that, with confidence at least $1-\frac{\delta}{2}$,

$$\Xi_D=\|(\lambda I+L_K)^{-\frac{1}{2}}(L_K-L_{K,D})\|\leq\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta} \tag{26}$$

and

$$\Omega_D=\|(L_K+\lambda I)(L_{K,D}+\lambda I)^{-1}\|\leq\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}}{\sqrt{\lambda}}+1\right)^2 \tag{27}$$

hold simultaneously.

When $\frac{1}{2}\leq r\leq\frac{3}{2}$, we apply (25) and (27) to Proposition 12 and obtain

$$\begin{aligned}
\|f^\sharp_{D,\lambda}-f^*\|_{L^2_{\rho_\mathcal{X}}}\leq\;&4M\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}}{\sqrt{\lambda}}+1\right)^2\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}+\lambda^r\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}}{\sqrt{\lambda}}+1\right)^{2r}\|u^*\|_{L^2_{\rho_\mathcal{X}}}\\
\leq\;&(4M+\|u^*\|_{L^2_{\rho_\mathcal{X}}})\left(\frac{\mathcal{B}_{|D|,\lambda}}{\sqrt{\lambda}}+1\right)^3(\mathcal{B}_{|D|,\lambda}+\lambda^r)\left(\log\frac{4}{\delta}\right)^4
\end{aligned}$$

When $\frac{3}{2}<r\leq 2$, we apply (25), (26) and (27) to Proposition 13 and obtain

$$\begin{aligned}
\|f^\sharp_{D,\lambda}-f^*\|_{L^2_{\rho_\mathcal{X}}}\leq\;&4M\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}}{\sqrt{\lambda}}+1\right)^2\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}\\
&+\lambda\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}}{\sqrt{\lambda}}+1\right)^3\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}\\
&+\lambda^r\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac{4}{\delta}}{\sqrt{\lambda}}+1\right)^2\|u^*\|_{L^2_{\rho_\mathcal{X}}}\\
\leq\;&(4M+2\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}})\left(\frac{\mathcal{B}_{|D|,\lambda}}{\sqrt{\lambda}}+1\right)^3(\mathcal{B}_{|D|,\lambda}+\lambda^r)\left(\log\frac{4}{\delta}\right)^4.
\end{aligned}$$

So (7) is proved for all $\frac{1}{2}\leq r\leq 2$. Applying Lemma 11 with $b=4$, $\tau=4$, and $s=2$, we obtain the desired estimation in (8). ■

Proof of Corollary 2 (ii). With $\frac{1}{2}\leq r\leq 2$ and the choice of $\lambda=|D|^{-\frac{1}{2r+\beta}}$, we have

$$B_{|D|,\lambda}\leq\frac{2\kappa}{\sqrt{|D|}}\left\{\frac{\kappa|D|^{\frac{1}{2(2r+\beta)}}}{\sqrt{|D|}}+\sqrt{C_0}|D|^{\frac{\beta}{2(2r+\beta)}}\right\}\leq 2\kappa\left(\kappa+\sqrt{C_0}\right)|D|^{-\frac{r}{2r+\beta}} \tag{28}$$

and

$$\frac{B_{|D|,\lambda}}{\sqrt{\lambda}}+1\le 2\kappa(\kappa+\sqrt{C_0})|D|^{-\frac{r}{2r+\beta}}|D|^{\frac{1}{2r+\beta}}+1\le 2\kappa\left(\kappa+\sqrt{C_0}\right)+1. \tag{29}$$

Then the conclusions follow from Theorem 1. ■

6. Error analysis in $\mathcal{H}_K$

In this section, we derive the error bound for $\|f^\sharp_{D,\lambda}-f^*\|_K$ and prove the convergence of BCRKN in $\mathcal{H}_K$. It is similar to the error analysis in $L^2_{\rho_\mathcal{X}}$.

Proposition 14 *If $r\in[\frac12,\frac32]$, we have*

$$\|f^\sharp_{D,\lambda}-f^*\|_K\le 2\lambda^{-\frac12}(\Omega_D)^{\frac12}\left\|(\lambda I+L_K)^{-\frac12}\Delta_D\right\|_K+\lambda^{r-\frac12}(\Omega_D)^{r-\frac12}\|u^*\|_{L^2_{\rho_\mathcal{X}}}.$$

If $r\in(\frac32,2]$, we have

$$\begin{aligned}\|f^\sharp_{D,\lambda}-f^*\|_K &\le 2\lambda^{-\frac12}(\Omega_D)^{\frac12}\left\|(\lambda I+L_K)^{-\frac12}\Delta_D\right\|_K+\lambda^{\frac12}\Xi_D\Omega_D\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}\\ &\quad+\lambda^{r-\frac12}\Omega_D\|u^*\|_{L^2_{\rho_\mathcal{X}}}.\end{aligned}$$

Proof By the triangle inequality in $\mathcal{H}_K$, we have

$$\|f^\sharp_{D,\lambda}-f^*\|_K\le\|f^\sharp_{D,\lambda}-\mathbf{E}^*[f^\sharp_{D,\lambda}]\|_K+\|\mathbf{E}^*[f^\sharp_{D,\lambda}]-f^*\|_K.$$

To estimate the first term, we see that

$$f^\sharp_{D,\lambda}-\mathbf{E}^*[f^\sharp_{D,\lambda}]=(2\lambda I+L_{K,D})(\lambda I+L_{K,D})^{-2}\Delta_D.$$

Then

$$\begin{aligned}&\left\|f^\sharp_{D,\lambda}-\mathbf{E}^*[f^\sharp_{D,\lambda}]\right\|_K=\|(2\lambda I+L_{K,D})(\lambda I+L_{K,D})^{-2}\Delta_D\|_K\\ \le\ &\left\|(2\lambda I+L_{K,D})(\lambda I+L_{K,D})^{-\frac32}\right\|\left\|(\lambda I+L_{K,D})^{-\frac12}(\lambda I+L_K)^{\frac12}\right\|\left\|(\lambda I+L_K)^{-\frac12}\Delta_D\right\|_K\\ \le\ &2\lambda^{-\frac12}(\Omega_D)^{\frac12}\left\|(\lambda I+L_K)^{-\frac12}\Delta_D\right\|_K.\end{aligned}$$

For the second term, we have

$$\left\|\mathbf{E}^*[f^\sharp_{D,\lambda}]-f^*\right\|_K=\left\|\lambda^2(\lambda I+L_{K,D})^{-2}L^ru^*\right\|_K\le\lambda^2\|(\lambda I+L_{K,D})^{-2}L_K^{r-\frac12}\|\|u^*\|_{L^2_{\rho_\mathcal{X}}}.$$

Following the same idea as in the proof of Proposition 12, we obtain for $r\in[\frac12,\frac32]$,

$$\left\|\mathbf{E}^*[f^\sharp_{D,\lambda}]-f^*\right\|_K\le\lambda^{r-\frac12}(\Omega_D)^{r-\frac12}\|u^*\|_{L^2_{\rho_\mathcal{X}}}$$

and following the idea in the proof of Proposition 13, we obtain for $r\in(\frac32,2]$,

$$\left\|\mathbf{E}^*[f^\sharp_{D,\lambda}]-f^*\right\|_K\le\lambda^{\frac12}\Xi_D\Omega_D\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}+\lambda^{r-\frac12}\Omega_D\|u^*\|_{L^2_{\rho_\mathcal{X}}}.$$

The desired error bounds now follow by combining the estimates for both terms. ■

Proof of Theorem 3. Note that (25), (26) and (27) hold simultaneously with probability at least $1-\delta$. Therefore, when $\frac12\le r\le\frac32$, we have with confidence at least $1-\delta$

$$\begin{aligned}\|f^\sharp_{D,\lambda}-f^*\|_K &\le 2\lambda^{-\frac12}\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac4\delta}{\sqrt{\lambda}}+1\right)2M\mathcal{B}_{|D|,\lambda}\log\frac4\delta\\ &\quad+\lambda^{r-\frac12}\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac4\delta}{\sqrt{\lambda}}+1\right)^{2r-1}\|u^*\|_{L^2_{\rho_\mathcal{X}}}\\ &\le (4M+\|u^*\|_{L^2_{\rho_\mathcal{X}}})\left(\frac{\mathcal{B}_{|D|,\lambda}}{\sqrt{\lambda}}+1\right)^2(\lambda^{-\frac12}\mathcal{B}_{|D|,\lambda}+\lambda^{r-\frac12})\left(\log\frac4\delta\right)^3\end{aligned}$$

and, when $\frac32<r\le2$,

$$\begin{aligned}\|f^\sharp_{D,\lambda}-f^*\|_K &\le 2\lambda^{-\frac12}\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac4\delta}{\sqrt{\lambda}}+1\right)2M\mathcal{B}_{|D|,\lambda}\log\frac4\delta\\ &\quad+\lambda^{\frac12}\mathcal{B}_{|D|,\lambda}\log\frac4\delta\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac4\delta}{\sqrt{\lambda}}+1\right)^2\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}\\ &\quad+\lambda^{r-\frac12}\left(\frac{\mathcal{B}_{|D|,\lambda}\log\frac4\delta}{\sqrt{\lambda}}+1\right)^2\|u^*\|_{L^2_{\rho_\mathcal{X}}}\\ &\le \left(4M+2\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}\right)\left(\frac{\mathcal{B}_{|D|,\lambda}}{\sqrt{\lambda}}+1\right)^2(\lambda^{-\frac12}\mathcal{B}_{|D|,\lambda}+\lambda^{r-\frac12})\left(\log\frac4\delta\right)^3.\end{aligned}$$

This proves the error bound (9). Then (10) follows from estimates (28) and (29), and (11) follows by applying Lemma 11. ■

7. Improve the error analysis by unlabelled data

The error analysis for the semi-supervised approach is more involved. Before we move on, notice that Theorem 1 with $0<r<\frac12$ is a special case of Theorem 4 with $D'=D$ when there is no unlabeled data. So upon finishing Theorem 4, we also obtain Theorem 1 with $0<r<\frac12$.

We need to introduce an intermediate function. Recall L is a compact operator on $L^2_{\rho_\mathcal{X}}$. Let $\{\tau_i\}_{i=1}^\infty$ and $\{\psi_i\}_{i=1}^\infty$ be the eigenvalues and eigenfunctions of L. Then $\{\psi_i\}_{i=1}^\infty$ form an orthonormal basis of $L^2_{\rho_\mathcal{X}}$. Let P_λ be the projection operator on $L^2_{\rho_\mathcal{X}}$ that projects each $f\in L^2_{\rho_\mathcal{X}}$ onto the subspace spanned by $\{\psi_i:\tau_i\ge\lambda\}$, i.e.

$$P_\lambda f=\sum_{\{i:\tau_i\ge\lambda\}}\langle\psi_i,f\rangle_{L^2_{\rho_\mathcal{X}}}\psi_i,\qquad\forall\ f\in L^2_{\rho_\mathcal{X}}.$$

By the isomorphism property (5) of $L^{\frac{1}{2}}$, $\{\phi_i = \sqrt{\tau_i}\psi_i : \sigma_i > 0\}$ form an orthonormal basis of $\mathcal{H}_K$. Since $\{i : \tau_i \geq \lambda\}$ is a finite set, it is obvious $P_\lambda f \in \mathcal{H}_K$ for all $f \in L^2_{\rho_\mathcal{X}}$. Define $f^{tr}_\lambda = P_\lambda f^*$. We can bound $\|f^\sharp_{D',\lambda} - f^*\|_{L^2_{\rho_\mathcal{X}}}$ as follows.

Proposition 15 *We have*

$$\|f^\sharp_{D',\lambda} - f^*\|_{L^2_{\rho_\mathcal{X}}} \leq I_1 + I_2 + I_3,$$

where

$$\begin{aligned} I_1 &= \left\|(\lambda I + L_K)^{\frac{1}{2}}(2\lambda I + L_{K,D'})(\lambda I + L_{K,D'})^{-2}\left(\tfrac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_{K,D'}f^{tr}_\lambda\right)\right\|_K \\ I_2 &= \left\|\lambda^2(\lambda I + L_K)^{\frac{1}{2}}(\lambda I + L_{K,D'})^{-2}f^{tr}_\lambda\right\|_K, \\ I_3 &= \|f^{tr}_\lambda - f^*\|_{L^2_{\rho_\mathcal{X}}}. \end{aligned}$$

Proof Note that

$$\|f^\sharp_{D',\lambda} - f^*\|_{L^2_{\rho_\mathcal{X}}} \leq \|f^\sharp_{D',\lambda} - f^{tr}_\lambda\|_{L^2_{\rho_\mathcal{X}}} + \|f^{tr}_\lambda - f^*\|_{L^2_{\rho_\mathcal{X}}}. \tag{30}$$

Since $f^\sharp_{D',\lambda} - f^{tr}_\lambda \in \mathcal{H}_K$, by the isometry property (5) of $L^{\frac{1}{2}} = L_K^{\frac{1}{2}}$, we have

$$\|f^\sharp_{D',\lambda} - f^{tr}_\lambda\|_{L^2_{\rho_\mathcal{X}}} = \|L_K^{\frac{1}{2}}(f^\sharp_{D',\lambda} - f^{tr}_\lambda)\|_K \leq \|(\lambda I + L_K)^{\frac{1}{2}}(f^\sharp_{D',\lambda} - f^{tr}_\lambda)\|_K. \tag{31}$$

Recall that

$$f^\sharp_{D',\lambda} = f_{D',\lambda} + \lambda(\lambda I + L_{K,D'})^{-1}f_{D',\lambda} = (2\lambda I + L_{K,D'})(\lambda I + L_{K,D'})^{-2}\frac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'}.$$

It is easy to check that

$$\begin{aligned} f^\sharp_{D',\lambda} - f^{tr}_\lambda &= (2\lambda I + L_{K,D'})(\lambda I + L_{K,D'})^{-2}\left(\frac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_{K,D'}f^{tr}_\lambda\right) \\ &\quad -\lambda^2(\lambda I + L_{K,D'})^{-2}f^{tr}_\lambda. \end{aligned}$$

Putting this in (31) we have $\|f^\sharp_{D',\lambda} - f^{tr}_\lambda\|_{L^2_{\rho_\mathcal{X}}}$ bounded by $I_1 + I_2$. Together with (30), we obtain the desired conclusion. ■

Next we estimate the three terms respectively. The third term I_3 can be easily bounded by the following lemma, which has been proved in Caponnetto (2006).

Lemma 16 *We have* $\|f^{tr}_\lambda - f^*\|_{L^2_{\rho_\mathcal{X}}} \leq \lambda^r\|u^*\|_{L^2_{\rho_\mathcal{X}}}$ *and* $\|f^{tr}_\lambda\|_K \leq \lambda^{-\frac{1}{2}+r}\|u^*\|_{L^2_{\rho_\mathcal{X}}}$.

For the first term I_1, we have the following bound.

Proposition 17 *For any $\delta \in (0,1)$, with confidence at least $1-\delta$, there holds*

$$I_1 \leq \left(\frac{2M}{\kappa} + 2\|u^*\|_{L^2_{\rho_\mathcal{X}}}\right)\left(\frac{\mathcal{B}_{|D'|,\lambda}}{\sqrt{\lambda}} + 1\right)^3\left(\mathcal{B}_{|D|,\lambda} + \lambda^r\right)\left(\log\frac{4}{\delta}\right)^3.$$

Proof Since $2\lambda I + L_{K,D'}$ and $(\lambda I + L_{K,D'})^{-1}$ commute, we have

$$\begin{aligned} I_1 &= \left\|(\lambda I + L_K)^{\frac{1}{2}}(2\lambda I + L_{K,D'})(\lambda I + L_{K,D'})^{-2}\left(\tfrac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_{K,D'}f^{tr}_\lambda\right)\right\|_K \\ &\leq \left\|(\lambda I + L_K)^{\frac{1}{2}}(\lambda I + L_{K,D'})^{-\frac{1}{2}}\right\|\left\|(2\lambda I + L_{K,D'})(\lambda I + L_{K,D'})^{-1}\right\| \\ &\quad \times\left\|(\lambda I + L_{K,D'})^{-\frac{1}{2}}(\lambda I + L_K)^{\frac{1}{2}}\right\|\left\|(\lambda I + L_K)^{-\frac{1}{2}}\left(\tfrac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_{K,D'}f^{tr}_\lambda\right)\right\|_K \\ &\leq 2\Omega_{D'}\left\|(\lambda I + L_K)^{-\frac{1}{2}}\left(\tfrac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_{K,D'}f^{tr}_\lambda\right)\right\|_K. \end{aligned}$$

Proposition 10 ensures that, with confidence at least $1-\frac{\delta}{2}$,

$$\Omega_{D'} \leq \left(\frac{\mathcal{B}_{|D'|,\lambda}}{\sqrt{\lambda}} + 1\right)^2\left(\log\frac{4}{\delta}\right)^2. \tag{32}$$

Now it suffices to consider the term $\left\|(\lambda I + L_K)^{-\frac{1}{2}}\left(\frac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_{K,D'}f^{tr}_\lambda\right)\right\|_K$. We further divide it into three parts as follows

$$\begin{aligned} \left\|(\lambda I + L_K)^{-\frac{1}{2}}\left(\frac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_{K,D'}f^{tr}_\lambda\right)\right\| &\leq \left\|(\lambda I + L_K)^{-\frac{1}{2}}\left(\frac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_Kf^*\right)\right\|_K \\ &\quad + \left\|(\lambda I + L_K)^{-\frac{1}{2}}(L_Kf^* - L_Kf^{tr}_\lambda)\right\|_K \\ &\quad + \left\|(\lambda I + L_K)^{-\frac{1}{2}}(L_K - L_{K,D'})f^{tr}_\lambda\right\|_K. \end{aligned}$$

By the definition of y'_i, it is easy to check that $\frac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} = \frac{1}{|D|}S^*_D\mathbf{y}_D$. Applying Lemma 7 with $\xi_g(z) = yK_x$, we obtain, with confidence at least $1-\frac{\delta}{2}$,

$$\left\|(\lambda I + L_K)^{-\frac{1}{2}}\left(\frac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_Kf^*\right)\right\|_K \leq \frac{M\log\frac{4}{\delta}}{\kappa}\mathcal{B}_{|D|,\lambda}.$$

By Lemma 16, we have

$$\left\|(\lambda I + L_K)^{-\frac{1}{2}}(L_Kf^* - L_Kf^{tr}_\lambda)\right\|_K \leq \|f^* - f^{tr}_\lambda\|_{L^2_{\rho_\mathcal{X}}} \leq \lambda^r\|u^*\|_{L^2_{\rho_\mathcal{X}}}.$$

For $\left\|(\lambda I + L_K)^{-\frac{1}{2}}(L_K - L_{K,D'})f^{tr}_\lambda\right\|_K$, observe that

$$\left\|(\lambda I + L_K)^{-\frac{1}{2}}(L_K - L_{K,D'})f^{tr}_\lambda\right\|_K \leq \left\|(\lambda I + L_K)^{-\frac{1}{2}}(L_K - L_{K,D'})\right\|\|f^{tr}_\lambda\|_K.$$

By Lemma 16, we have $\|f^{tr}_\lambda\|_K \leq \lambda^{-\frac{1}{2}+r}\|u^*\|_{L^2_{\rho_\mathcal{X}}}$. By Lemma 9, we have with confidence at least $1-\frac{\delta}{2}$,

$$\left\|(\lambda I + L_K)^{-\frac{1}{2}}(L_K - L_{K,D'})f^{tr}_\lambda\right\|_K \leq \lambda^r\frac{\mathcal{B}_{|D'|,\lambda}}{\sqrt{\lambda}}\|u^*\|_{L^2_{\rho_\mathcal{X}}}\log\frac{4}{\delta},$$

with the confidence set the same as that for (32). Combining the above estimations together yields

$$\left\|(\lambda I + L_K)^{-\frac{1}{2}}\left(\frac{1}{|D'|}S^*_{D'}\mathbf{y}_{D'} - L_{K,D'}f^{tr}_\lambda\right)\right\|_K$$
$$\leq \left(\frac{M}{\kappa} + \|u^*\|_{L^2_{\rho_\mathcal{X}}}\right)\left(\frac{\mathcal{B}_{|D'|,\lambda}}{\sqrt{\lambda}} + 1\right)\left(\mathcal{B}_{|D|,\lambda} + \lambda^r\right)\log\frac{4}{\delta}. \quad (33)$$

Then our desired result follows by (32) and (33). ■

Proposition 18 *For any $\delta \in (0,1)$, we have, with confidence at least $1 - \frac{\delta}{2}$,*

$$I_2 \leq \left(\frac{\mathcal{B}_{|D'|,\lambda}}{\sqrt{\lambda}} + 1\right)\lambda^r\|u^*\|_{L^2_{\rho_\mathcal{X}}}\log\frac{4}{\delta}$$

with the confidence set the same as that for (32).

Proof We have

$$\begin{aligned} I_2 &= \|\lambda^2(\lambda I + L_K)^{\frac{1}{2}}(\lambda I + L_{K,D'})^{-2}f^{tr}_\lambda\|_K \\ &= \lambda^2\left\|(\lambda I + L_K)^{\frac{1}{2}}(\lambda I + L_{K,D'})^{-\frac{1}{2}}(\lambda I + L_{K,D'})^{-\frac{3}{2}}f^{tr}_\lambda\right\|_K \\ &\leq \lambda^2\|(\lambda I + L_K)^{\frac{1}{2}}(\lambda I + L_{K,D'})^{-\frac{1}{2}}\|\|(\lambda I + L_{K,D'})^{-\frac{3}{2}}\|\|f^{tr}_\lambda\|_K \\ &\leq \lambda^r(\Omega_{D'})^{\frac{1}{2}}\|u^*\|_{L^2_{\rho_\mathcal{X}}}, \end{aligned}$$

where we used the bounds $\|(\lambda I + L_{K,D'})^{-\frac{3}{2}}\| \leq \lambda^{-\frac{3}{2}}$ and $\|f^{tr}_\lambda\|_K \leq \lambda^{r-\frac{1}{2}}\|u^*\|_{L^2_{\rho_\mathcal{X}}}$. By (32), we obtain the desired bound and confidence set. ■

Now we can prove Theorem 4.

Proof of Theorem 4. Plugging the bounds of I_1, I_2, and I_3 into Proposition 15, we obtain the error bound for $\|f^\sharp_{D',\lambda} - f^*\|_{L^2_{\rho_\mathcal{X}}}$ in (12).

If $\lambda = |D|^{-\frac{1}{2r+\beta}}$ with $0 < r \leq \frac{1}{2}$ and $\mathcal{N}(\lambda) \leq C_0\lambda^{-\beta}$, we have

$$B_{|D|,\lambda} \leq \frac{2\kappa}{\sqrt{|D|}}\left\{\kappa|D|^{-\frac{1}{2}}|D|^{\frac{1}{2(2r+\beta)}} + \sqrt{C_0}|D|^{\frac{\beta}{2(2r+\beta)}}\right\} \leq 2\kappa(\kappa + \sqrt{C_0})|D|^{-\frac{r}{2r+\beta}}.$$

Under the condition $|D'| \geq |D|^{\frac{1+\beta}{2r+\beta}}$, we have

$$\begin{aligned} \frac{\mathcal{B}_{|D'|,\lambda}}{\sqrt{\lambda}} &= \frac{2\kappa}{\sqrt{|D'|\lambda}}\left\{\frac{\kappa}{\sqrt{|D'|\lambda}} + \sqrt{\mathcal{N}(\lambda)}\right\} \\ &\leq \frac{2\kappa}{\sqrt{|D'|}}|D|^{\frac{1}{2(2r+\beta)}}\left\{\kappa|D'|^{-\frac{1}{2}}|D|^{\frac{1}{2(2r+\beta)}} + \sqrt{C_0}|D|^{\frac{\beta}{2(2r+\beta)}}\right\} \end{aligned}$$

$$\begin{aligned} &\leq \frac{2\kappa}{\sqrt{|D'|}}|D|^{\frac{1}{2(2r+\beta)}}(\kappa + \sqrt{C_0})|D|^{\frac{\beta}{2(2r+\beta)}} \\ &\leq 2\kappa(\kappa + \sqrt{C_0}). \end{aligned}$$

Applying these two estimates to (12), we have for any $\delta \in (0,1)$, with confidence at least $1 - \delta$,

$$\|f^\sharp_{D',\lambda} - f^*\|_{L^2_{\rho_\mathcal{X}}} \leq C_1|D|^{-\frac{r}{2r+\beta}}\left(\log\frac{4}{\delta}\right)^3$$

where

$$C_1 = \left(\frac{2M}{\kappa} + 4\|u^*\|_{L^2_{\rho_\mathcal{X}}}\right)\left[2\kappa(\kappa + \sqrt{C_0}) + 1\right]^3\left[2\kappa(\kappa + \sqrt{C_0}) + 1\right].$$

This completes the proof of Theorem 4. ■

Note we also proved Theorem 1 with $0 < r < \frac{1}{2}$ because it is a special case of Theorem 4 with $D' = D$. So we are in position to prove Corollary 2 (i).

Proof of Corollary 2 (i). When $0 < r \leq \frac{1}{2}$, take $\lambda = |D|^{-\frac{1}{1+\beta}}$. Then

$$\begin{aligned} B_{|D|,\lambda} &\leq \frac{2\kappa}{\sqrt{|D|}}\left\{\frac{\kappa|D|^{\frac{1}{2(1+\beta)}}}{\sqrt{|D|}} + \sqrt{C_0}|D|^{\frac{\beta}{2(1+\beta)}}\right\} \\ &\leq 2\kappa(\kappa + \sqrt{C_0})|D|^{-\frac{1}{2(1+\beta)}} \leq 2\kappa(\kappa + \sqrt{C_0})|D|^{-\frac{r}{1+\beta}} \end{aligned}$$

and

$$\frac{B_{|D|,\lambda}}{\sqrt{\lambda}} \leq 2\kappa(\kappa + \sqrt{C_0})|D|^{-\frac{1}{2(1+\beta)}}|D|^{\frac{1}{2(1+\beta)}} = 2\kappa(\kappa + \sqrt{C_0}).$$

Plugging them into the estimation (7) we obtain the desired learning rate. ■

8. Error analysis for distributed BCRKN

The following lemma is analogous to Guo et al. (2017a, Proposition 4).

Lemma 19 *Let $\overline{f}^\sharp_{D,\lambda}$ be defined by* (14). *We have*

$$\mathbf{E}\left[\|\overline{f}^\sharp_{D,\lambda} - f^*\|^2_{L^2_{\rho_\mathcal{X}}}\right] \leq \sum_{j=1}^m \frac{|D_j|^2}{|D|^2}\mathbf{E}\left[\|f^\sharp_{D_j,\lambda} - f^*\|^2_{L^2_{\rho_\mathcal{X}}}\right] + \sum_{j=1}^m \frac{|D_j|}{|D|}\left\|\mathbf{E}[f^\sharp_{D_j,\lambda}] - f^*\right\|^2_{L^2_{\rho_\mathcal{X}}}. \quad (34)$$

Proof of Theorem 5. By Theorem 1, for each fixed $j \in \{1, 2, \ldots, m\}$,

$$\mathbf{E}\left[\|f^\sharp_{D_j,\lambda} - f^*\|^2_{L^2_{\rho_\mathcal{X}}}\right] \leq 4\Gamma(9)C^2\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}} + 1\right)^6\left(\mathcal{B}_{|D_j|,\lambda} + \lambda^r\right)^2.$$

Then the first term on the right of (34) can be estimated as

$$\sum_{j=1}^{m}\frac{|D_j|^2}{|D|^2}\mathbf{E}\left[\|f^\sharp_{D_j,\lambda}-f^*\|^2_{L^2_{\rho_\mathcal{X}}}\right]\le 4\Gamma(9)C^2\sum_{j=1}^{m}\frac{|D_j|^2}{|D|^2}\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^6\left(\mathcal{B}_{|D_j|,\lambda}+\lambda^r\right)^2. \quad (35)$$

To estimate the second term on the right of (34), for each fixed $j\in\{1,2,\dots,m\}$, by Jensen's inequality, we have

$$\left\|\mathbf{E}[f^\sharp_{D_j,\lambda}]-f^*\right\|_{L^2_{\rho_\mathcal{X}}}\le\mathbf{E}\left[\|\mathbf{E}^*[f^\sharp_{D_j,\lambda}]-f^*\|_{L^2_{\rho_\mathcal{X}}}\right].$$

We will bound the second term in two different ways according to the range of r. First consider the case when $\frac{1}{2}\le r\le\frac{3}{2}$. The bound (23) in the proof of Proposition 12 tells us that

$$\left\|\mathbf{E}^*[f^\sharp_{D_j,\lambda}]-f^*\right\|_{L^2_{\rho_\mathcal{X}}}\le\lambda^r\Omega^r_{D_j}\|u^*\|_{L^2_{\rho_\mathcal{X}}}.$$

It follows that

$$\|\mathbf{E}[f^\sharp_{D_j,\lambda}]-f^*\|_{L^2_{\rho_\mathcal{X}}}\le\lambda^r\|u^*\|_{L^2_{\rho_\mathcal{X}}}\mathbf{E}\left[\Omega^r_{D_j}\right]. \quad (36)$$

Applying Proposition 10 to each fixed $j\in\{1,\dots,m\}$, with confidence at least $1-\frac{\delta}{2}$, there holds

$$\Omega_{D_j}\le\left(\frac{\mathcal{B}_{|D_j|,\lambda}\log\frac{4}{\delta}}{\sqrt{\lambda}}+1\right)^2\le\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^2\left(\log\frac{4}{\delta}\right)^2.$$

By Lemma 11, this implies that for any $s>0$,

$$\mathbf{E}\left[\Omega^s_{D_j}\right]\le 2\Gamma(2s+1)\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^{2s}. \quad (37)$$

Applying (37) with $s=r$ to (36) yields

$$\|\mathbf{E}[f^\sharp_{D_j,\lambda}]-f^*\|_{L^2_{\rho_\mathcal{X}}}\le 2\Gamma(2r+1)\|u^*\|_{L^2_{\rho_\mathcal{X}}}\lambda^r\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^{2r}. \quad (38)$$

Combining (34), (35) and (38), we have

$$\begin{aligned}\mathbf{E}[\|\overline{f}^\sharp_{D,\lambda}-f^*\|^2_{L^2_{\rho_\mathcal{X}}}] &\le 4\Gamma(9)C^2\sum_{j=1}^{m}\frac{|D_j|^2}{|D|^2}\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^6\left(\mathcal{B}_{|D_j|,\lambda}+\lambda^r\right)^2\\ &\quad+4\Gamma^2(2r+1)\|u^*\|^2_{L^2_{\rho_\mathcal{X}}}\lambda^{2r}\sum_{j=1}^{m}\frac{|D_j|}{|D|}\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^{4r}.\end{aligned}$$

This proves the desired bound for $\frac{1}{2}\le r\le\frac{3}{2}$.

For $\frac{3}{2}<r\le 2$, by the bound (24) in the proof of Proposition 13, we have

$$\|\mathbf{E}^*[f^\sharp_{D_j,\lambda}]-f^*\|_{L^2_{\rho_\mathcal{X}}}\le\lambda\Xi_{D_j}\Omega^{\frac{3}{2}}_{D_j}\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}+\lambda^r\Omega_{D_j}\|u^*\|_{L^2_{\rho_\mathcal{X}}}.$$

So,

$$\begin{aligned}\mathbf{E}[\|\mathbf{E}^*[f^\sharp_{D_j,\lambda}]-f^*\|_{L^2_{\rho_\mathcal{X}}}] &\le \lambda\mathbf{E}\left[\Xi_{D_j}(\Omega_D)^{\frac{3}{2}}\right]\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}+\lambda^r\mathbf{E}[\Omega_{D_j}]\|u^*\|_{L^2_{\rho_\mathcal{X}}}\\ &\le \lambda\left(\mathbf{E}\left[\Xi^2_{D_j}\right]\right)^{\frac{1}{2}}\left(\mathbf{E}\left[\Omega^3_{D_j}\right]\right)^{\frac{1}{2}}\kappa^{2r-3}\|u^*\|_{L^2_{\rho_\mathcal{X}}}+\lambda^r\mathbf{E}[\Omega_{D_j}]\|u^*\|_{L^2_{\rho_\mathcal{X}}}.\end{aligned}$$

From Lin et al. (2017, Lemma 17), we know

$$\mathbf{E}\left[\left\|(\lambda I+L_K)^{-\frac{1}{2}}(L_K-L_{K,D})\right\|^2\right]\le\frac{\kappa^2\mathcal{N}(\lambda)}{|D|}$$

and by (37) with $s=3$, we have

$$\mathbf{E}\left[\Omega^3_{D_j}\right]\le 2\Gamma(7)\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^6.$$

Therefore,

$$\begin{aligned}&\|\mathbf{E}[f^\sharp_{D_j,\lambda}]-f^*\|^2_{L^2_{\rho_\mathcal{X}}}\\ \le\;& 2\lambda^2\mathbf{E}\left[\Xi^2_{D_j}\right]\mathbf{E}\left[\Omega^3_{D_j}\right]\kappa^{4r-6}\|u^*\|^2_{L^2_{\rho_\mathcal{X}}}+2\lambda^{2r}(\mathbf{E}[\Omega_{D_j}])^2\|u^*\|^2_{L^2_{\rho_\mathcal{X}}}\\ \le\;& 2\lambda^2\frac{\kappa^2\mathcal{N}(\lambda)}{|D_j|}2\Gamma(7)\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^6\kappa^{4r-6}\|u^*\|^2_{L^2_{\rho_\mathcal{X}}}\\ &+32\lambda^{2r}\|u^*\|^2_{L^2_{\rho_\mathcal{X}}}\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^4\\ \le\;& \left(4\Gamma(7)\kappa^{4r-4}+32\right)\|u^*\|^2_{L^2_{\rho_\mathcal{X}}}\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^6\left(\frac{\lambda^2\mathcal{N}(\lambda)}{|D_j|}+\lambda^{2r}\right). \quad (39)\end{aligned}$$

Combining (34), (35) and (39), we have

$$\begin{aligned}&\mathbf{E}[\|\overline{f}^\sharp_{D,\lambda}-f^*\|^2_{L^2_{\rho_\mathcal{X}}}]\\ \le\;& 4\Gamma(9)C^2\sum_{j=1}^{m}\frac{|D_j|^2}{|D|^2}\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^6\left(\mathcal{B}_{|D_j|,\lambda}+\lambda^r\right)^2\\ &+\left(4\Gamma(7)\kappa^{4r-4}+32\right)\|u^*\|^2_{L^2_{\rho_\mathcal{X}}}\sum_{j=1}^{m}\frac{|D_j|}{|D|}\left(\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}}+1\right)^6\left(\frac{\lambda^2\mathcal{N}(\lambda)}{|D_j|}+\lambda^{2r}\right).\end{aligned}$$

This proves the conclusion for $\frac{3}{2}<r\le 2$. ■

Next let us turn to the special case that the local machines are assigned the same number of observations, i.e., $|D_1|=|D_2|=\cdots=|D_m|$.

Proof of Theorem 6. For $\frac{1}{2}\le r\le 2$, we choose $\lambda=|D|^{-\frac{1}{2r+\beta}}$. Then by the capacity assumption (6), restriction on the number of local machines $m\le|D|^{\min\{\frac{2r-1}{2r+\beta},\frac{2}{2r+\beta}\}}$, and the

fact $|D_1| = |D_2| = \ldots = |D_m| = \frac{|D|}{m}$, we have

$$\frac{\mathcal{N}(\lambda)}{\lambda |D_j|} \leq C_0 m N^{\frac{1-2r}{2r+\beta}} \leq C_0.$$

It follows that, for each $j = 1, \ldots, m$,

$$\frac{\mathcal{B}_{|D_j|,\lambda}}{\sqrt{\lambda}} = \frac{2\kappa}{\sqrt{\lambda |D_j|}} \left\{ \frac{\kappa}{\sqrt{|D_j|\lambda}} + \sqrt{\mathcal{N}(\lambda)} \right\} \leq 2\kappa(\kappa + \sqrt{C_0})$$

and

$$\frac{|D_j|}{|D|} \mathcal{B}^2_{|D_j|,\lambda} \leq 8\kappa^2 \left(\frac{\kappa^2}{|D||D_j|\lambda} + \frac{\mathcal{N}(\lambda)}{|D|} \right) \leq 8\kappa^2(\kappa^2 + C_0)|D|^{-\frac{2r}{2r+\beta}}$$

and

$$\frac{\lambda^2 \mathcal{N}(\lambda)}{|D_j|} \leq \frac{C_0 |D|^{-\frac{2}{2r+\beta}} |D|^{\frac{\beta}{2r+\beta}} m}{|D|} \leq C_0 |D|^{-\frac{2r}{2r+\beta}}.$$

Then by Theorem 5,

$$\mathbf{E}\left[\|\bar{f}^\sharp_{D,\lambda} - f^*\|^2_{L^2_{\hat{\rho}_\mathcal{X}}} \right] \leq \bar{C} \left(2\kappa(\kappa + \sqrt{C_0}) + 1 \right)^6 \left(16\kappa^2(\kappa^2 + C_0) + C_0 + 1 \right) |D|^{-\frac{2r}{2r+\beta}}.$$

This completes the proof of Theorem 6. ■

Acknowledgments

The work described in this paper is partially supported by the National Natural Science Foundation of China (Grants No. 11401524, 11531013, 11571078, 11631015, 11671171). Lei Shi is also supported by the Joint Research Fund by National Natural Science Foundation of China and Research Grants Council of Hong Kong (Project No. 11461161006 and Project No. CityU 104012) and Zhuo Xue Program of Fudan University. Qiang Wu is also supported in part by the USDA National Institute of Food and Agriculture (Grant Number 2016-70001-24636). Part of the work was carried out while Zheng-Chu Guo was visiting Shanghai Key Laboratory for Contemporary Applied Mathematics. All authors contributed equally to this paper and are listed alphabetically. The corresponding author is Qiang Wu.

References

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

H. Battey, J. Fan, H. Liu, J. Lu, and Z. Zhu. Distributed estimation and inference with statistical guarantees. 2015. arXiv preprint arXiv:1509.05457.

F. Bauer, S. Pereverzev, and L. Rosasco. On regularization algorithms in learning theory. *Journal of complexity*, 23(1):52–72, 2007.

G. Blanchard and N. Krämer. Optimal learning rates for kernel conjugate gradient regression. In *Advances in Neural Information Processing Systems*, pages 226–234, 2010.

O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

P. Bühlmann. Statistical significance in high-dimensional linear models. *Bernoulli*, 19(4): 1212–1242, 2013.

A. Caponnetto. Optimal rates for regularization operators in learning theory. Technical report, MIT, 2006. `http://cbcl.mit.edu/publications/ps/MIT-CSAIL-TR-2006-062.pdf`.

A. Caponnetto and E. De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

E. De Vito, A. Caponnetto, and L. Rosasco. Model selection for regularized least-squares algorithm in learning theory. *Foundations of Computational Mathematics*, 5(1):59–85, 2005.

R. DeVore, G. Kerkyacharian, D. Picard, and V. Temlyakov. Mathematical methods for supervised learning. *IMI Preprints*, 22:1–51, 2004.

T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13:1–50, 2000.

X. Guo, J. Fan, and D.-X. Zhou. Sparsity and error analysis of empirical feature-based regularization schemes. *Journal of Machine Learning Research*, 17(1):3058–3091, 2016.

Z.-C. Guo, S.-B. Lin, and D.-X. Zhou. Learning theory of distributed spectral algorithms. *Inverse Problems*, 33(7):074009, 2017a.

Z.-C. Guo, D.-H. Xiang, X. Guo, and D.-X. Zhou. Thresholded spectral algorithms for sparse approximations. *Analysis and Applications*, 15(3):433–455, 2017b.

L. Györfi, M. Kohler, A. Krzyzak, and H. Walk. *A distribution-free theory of nonparametric regression*. Springer Science & Business Media, 2006.

A. Javanmard and A. Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

J. D. Lee, Q. Liu, Y. Sun, and J. E. Taylor. Communication-efficient sparse regression. *Journal of Machine Learning Research*, 18(5):1–30, 2017.

S.-B. Lin, X. Guo, and D.-X. Zhou. Distributed learning with least square regularization. *Journal of Machine Learning Research*, 18(92):1–31, 2017.

J. D. Rosenblatt and B. Nadler. On the optimality of averaging in distributed statistical learning. *Information and Inference: A Journal of the IMA*, 5(4):379–404, 2016.

S. Smale and D. X. Zhou. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26:153–172, 2007.

I. Steinwart, D. R. Hush, and C. Scovel. Optimal rates for regularized least squares regression. In *Conference on Learning Theory*, 2009.

H. Sun and Q. Wu. A note on application of integral operator in learning theory. *Applied and Computational Harmonic Analysis*, 26(3):416–421, 2009.

V. Temlyakov. Approximation in learning theory. *Constructive Approximation*, 27(1):33–74, 2008.

G. Wahba. *Spline models for observational data*. SIAM, 1990.

Q. Wu. Bias corrected regularization kernel network and its applications. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1072–1079, 2017.

Q. Wu, Y. Ying, and D.-X. Zhou. Learning rates of least-square regularized regression. *Foundations of Computational Mathematics*, 6(2):171–192, 2006.

C.-H. Zhang and S. S. Zhang. Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):217–242, 2014.

T. Zhang. Leave-one-out bounds for kernel methods. *Neural Computation*, 15(6):1397–1437, 2003.

Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression. In *Conference on Learning Theory*, pages 592–617, 2013.

Y. Zhang, J. C. Duchi, and M. J. Wainwright. Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16:3299–3340, 2015.

Journal of Machine Learning Research 18 (2017) 1-59 Submitted 1/17; Revised 6/17; Published 11/17

Probabilistic Line Searches for Stochastic Optimization

Maren Mahsereci mmahsereci@tue.mpg.de
Philipp Hennig phennig@tue.mpg.de
Max Planck Institute for Intelligent Systems
Max-Planck-Ring 4, 72076 Tübingen, Germany

Editor: Mark Schmidt

Abstract

In deterministic optimization, line searches are a standard tool ensuring stability and efficiency. Where only stochastic gradients are available, no direct equivalent has so far been formulated, because uncertain gradients do not allow for a strict sequence of decisions collapsing the search space. We construct a probabilistic line search by combining the structure of existing deterministic methods with notions from Bayesian optimization. Our method retains a Gaussian process surrogate of the univariate optimization objective, and uses a probabilistic belief over the Wolfe conditions to monitor the descent. The algorithm has very low computational cost, and no user-controlled parameters. Experiments show that it effectively removes the need to define a learning rate for stochastic gradient descent.

Keywords: stochastic optimization, learning rates, line searches, Gaussian processes, Bayesian optimization

1. Introduction

This work substantially extends the work of Mahsereci and Hennig (2015) published at NIPS 2015. Stochastic gradient descent (SGD, Robbins and Monro, 1951) is currently the standard in machine learning for the optimization of highly multivariate functions if their gradient is corrupted by noise. This includes the online or mini-batch training of neural networks, logistic regression (Zhang, 2004; Bottou, 2010) and variational models (e.g. Hoffman et al., 2013; Hensman et al., 2012; Broderick et al., 2013). In all these cases, noisy gradients arise because an exchangeable loss-function $\mathcal{L}(x)$ of the optimization parameters $x \in \mathbb{R}^D$, across a large dataset $\{d_i\}_{i=1\ldots,M}$, is evaluated only on a subset $\{d_j\}_{j=1,\ldots,m}$:

$$\mathcal{L}(x) := \frac{1}{M}\sum_{i=1}^{M} \ell(x, d_i) \approx \frac{1}{m}\sum_{j=1}^{m} \ell(x, d_j) =: \hat{\mathcal{L}}(x) \qquad m \ll M. \tag{1}$$

If the indices j are i.i.d. draws from $[1, M]$, by the Central Limit Theorem, the error $\hat{\mathcal{L}}(x) - \mathcal{L}(x)$ is unbiased and approximately normal distributed. Despite its popularity and its low cost per step, SGD has well-known deficiencies that can make it inefficient, or at least tedious to use in practice. Two main issues are that, first, the gradient itself, even without noise, is not the optimal search *direction*; and second, SGD requires a *step size* (learning rate) that has drastic effect on the algorithm's efficiency, is often difficult to choose well, and virtually never optimal for each individual descent step. The former issue, adapting the search direction, has been addressed by many authors (see George and Powell, 2006, for an overview). Existing approaches range from lightweight 'diagonal preconditioning' like Adam (Kingma and Ba, 2014), AdaGrad (Duchi et al., 2011), and 'stochastic meta-descent' (Schraudolph, 1999), to empirical estimates for the natural gradient (Amari et al., 2000) or the Newton direction (Roux and Fitzgibbon, 2010), to problem-specific algorithms (Rajesh et al., 2013), and more elaborate estimates of the Newton direction (Hennig, 2013). Most of these algorithms also include an auxiliary adaptive effect on the learning rate. Schaul et al. (2013) provided an estimation method to explicitly adapt the learning rate from one gradient descent step to another. Several very recent works have proposed the use of reinforcement learning and 'learning-to-learn' approaches for parameter adaption (Andrychowicz et al., 2016; Hansen, 2016; Li and Malik, 2016). Mostly these methods are designed to work well on a specified subset of optimization problems, which they are also trained on; they thus need to be re-learned for differing objectives. The corresponding algorithms are usually orders of magnitude more expensive than the low-level black box proposed here, and often require a classic optimizer (e.g SGD) to tune their internal hyper-parameters.

None of the mentioned algorithms change the size of the *current* descent step. Accumulating statistics across steps in this fashion requires some conservatism: If the step size is initially too large, or grows too fast, SGD can become unstable and 'explode', because individual steps are not checked for robustness at the time they are taken.

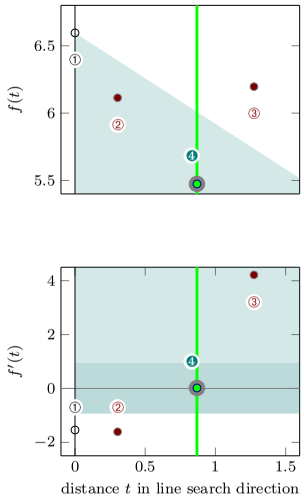


Figure 1: Sketch: The task of a *classic line search* is to tune the step taken by an optimization algorithm along a univariate search direction. The search starts at the endpoint ① of the previous line search, at $t = 0$. The *upper* plot shows function values, the *lower* plot corresponding gradients. A sequence of extrapolation steps ②,③ finds a point of positive gradient at ③. It is followed by interpolation steps until an acceptable point ❹ is found. Points of insufficient decrease, above the line $f(0) + c_1 t f'(0)$ (white area in upper plot) are excluded by the Armijo condition W-I, while points of steep negative gradient (white area in lower plot) are excluded by the curvature condition W-II (the strong extension of the Wolfe conditions also excludes the light green area in the lower plot). Point ❹ is the first to fulfill both conditions, and is thus accepted.

In essence, the same problem exists in deterministic (noise-free) optimization problems. There, providing stability is one of several tasks of the *line search* subroutine. It is a standard constituent of algorithms like the classic nonlinear conjugate gradient (Fletcher and Reeves, 1964) and BFGS (Broyden, 1969; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970)

©2017 Maren Mahsereci and Philipp Hennig.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-049.html.

methods (Nocedal and Wright, 1999, §3).[1] In the noise-free case, line searches are considered a solved problem (Nocedal and Wright, 1999, §3). But the methods used in deterministic optimization are not stable to noise. They are easily fooled by even small disturbances, either becoming overly conservative or failing altogether. The reason for this brittleness is that existing line searches take a sequence of hard decisions to shrink or shift the search space. This yields efficiency, but breaks hard in the presence of noise. Section 3 constructs a probabilistic line search for noisy objectives, stabilizing optimization methods like the works cited above. As line searches only change the length, not the direction of a step, they could be used in combination with the algorithms adapting SGD's direction, cited above. In this paper we focus on parameter tuning of the SGD algorithm and leave other search directions to future work.

2. Connections

2.1 Deterministic Line Searches

There is a host of existing line search variants (Nocedal and Wright, 1999, §3). In essence, though, these methods explore a univariate domain 'to the right' of a starting point, until an 'acceptable' point is reached (Figure 1). More precisely, consider the problem of minimizing $\mathcal{L}(x) : \mathbb{R}^D \to \mathbb{R}$, with access to $\nabla\mathcal{L}(x) : \mathbb{R}^D \to \mathbb{R}^D$. At iteration i, some 'outer loop' chooses, at location x_i, a search direction $s_i \in \mathbb{R}^D$ (e.g. by the BFGS-rule, or simply $s_i = -\nabla\mathcal{L}(x_i)$ for gradient descent). It will *not* be assumed that s_i has unit norm. The line search operates along the univariate domain $x(t) = x_i + ts_i$ for $t \in \mathbb{R}_+$. Along this direction it collects scalar function values and projected gradients that will be denoted $f(t) = \mathcal{L}(x(t))$ and $f'(t) = s_i^\intercal \nabla\mathcal{L}(x(t)) \in \mathbb{R}$. Most line searches involve an initial extrapolation phase to find a point t_r with $f'(t_r) > 0$. This is followed by a search in $[0, t_r]$, by interval nesting or by interpolation of the collected function and gradient values, e.g. with cubic splines.[2]

2.1.1 The Wolfe Conditions for Termination

As the line search is only an auxiliary step within a larger iteration, it need not find an exact root of f'; it suffices to find a point 'sufficiently' close to a minimum. The *Wolfe conditions* (Wolfe, 1969) are a widely accepted formalization of this notion; they consider t acceptable if it fulfills

$$f(t) \leq f(0) + c_1 t f'(0) \quad \text{(W-I)} \qquad \text{and} \qquad f'(t) \geq c_2 f'(0) \quad \text{(W-II)}, \tag{2}$$

using two constants $0 \leq c_1 < c_2 \leq 1$ chosen by the designer of the line search, not the user. W-I is the *Armijo* or *sufficient decrease* condition (Armijo, 1966). It encodes that acceptable functions values should lie below a linear extrapolation line of slope $c_1 f'(0)$. W-II is the *curvature condition*, demanding a decrease in slope. The choice $c_1 = 0$ accepts any value below $f(0)$, while $c_1 = 1$ rejects all points for convex functions. For the curvature condition,

1. In these algorithms, another task of the line search is to guarantee certain properties of the surrounding estimation rule. In BFGS, e.g., it ensures positive definiteness of the estimate. This aspect will not feature here.
2. This is the strategy in `minimize.m` by C. Rasmussen, which provided a model for our implementation. At the time of writing, it can be found at `http://learning.eng.cam.ac.uk/carl/code/minimize/minimize.m`

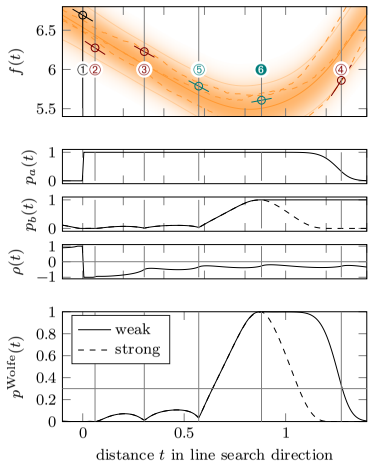


Figure 2: Sketch of a *probabilistic line search*. As in Fig. 1, the algorithm performs extrapolation (②,③,④) and interpolation (⑤,❻), but receives unreliable, noisy function and gradient values. These are used to construct a GP posterior (top. solid posterior mean, thin lines at 2 standard deviations, local pdf marginal as shading, three dashed sample paths). This implies a bivariate Gaussian belief (§3.3) over the validity of the weak Wolfe conditions (middle three plots. $p_a(t)$ is the marginal for W-I, $p_b(t)$ for W-II, $\rho(t)$ their correlation). Points are considered acceptable if their joint probability $p^{\text{Wolfe}}(t)$ (bottom) is above a threshold (gray). An approximation (§3.3.1) to the strong Wolfe conditions is shown dashed.

$c_2 = 0$ only accepts points with $f'(t) \geq 0$; while $c_2 = 1$ accepts any point of greater slope than $f'(0)$. W-I and W-II are known as the *weak* form of the Wolfe conditions. The *strong* form replaces W-II with $|f'(t)| \leq c_2|f'(0)|$. This guards against accepting points of low function value but large positive gradient. Figure 1 shows a conceptual sketch illustrating the typical process of a line search, and the weak and strong Wolfe conditions. The exposition in §3.3 will initially focus on the weak conditions, which can be precisely modeled probabilistically. Section 3.3.1 then adds an approximate treatment of the strong form.

2.2 Bayesian Optimization

A recently blossoming sample-efficient approach to global optimization revolves around modeling the objective f with a probability measure $p(f)$; usually a Gaussian process (GP). Searching for extrema, evaluation points are then chosen by a utility functional $u[p(f)]$. Our line search borrows the idea of a Gaussian process surrogate, and a popular acquisition function, *expected improvement* (Jones et al., 1998). Bayesian optimization (BO) methods are often computationally expensive, thus ill-suited for a cost-sensitive task like a line search. But since line searches are governors more than information extractors, the kind of sample-efficiency expected of a Bayesian optimizer is not needed. The following sections develop a lightweight algorithm which adds only minor computational overhead to stochastic optimization.

3. A Probabilistic Line Search

We now consider minimizing $f(t) = \hat{\mathcal{L}}(x(t))$ from Eq. 1. That is, the algorithm can access only noisy function values and gradients y_t, y'_t at location t, with Gaussian likelihood

$$p(y_t, y'_t \mid f) = \mathcal{N}\left(\begin{bmatrix} y_t \\ y'_t \end{bmatrix}; \begin{bmatrix} f(t) \\ f'(t) \end{bmatrix}, \begin{bmatrix} \sigma_f^2 & 0 \\ 0 & \sigma_{f'}^2 \end{bmatrix}\right). \tag{3}$$

The Gaussian form is supported by the Central Limit argument at Eq. 1. The function value y_t and the gradient y'_t are assumed independent for simplicity; see §3.4 and Appendix A regarding estimation of the variances $\sigma_f^2, \sigma_{f'}^2$, and some further notes on the independence assumption of y and y'. Each evaluation of $f(t)$ uses a newly drawn mini-batch.

Our algorithm is modeled after the classic line search routine `minimize.m`[2] and translates each of its building blocks one-by-one to the language of probability. The following table illustrates these four ingredients of the probabilistic line search and their corresponding classic parts.

building block	**classic**	**probabilistic**
1) 1D surrogate for objective $f(t)$	piecewise cubic splines	GP where the mean are piecewise cubic splines
2) candidate selection	*one* local minimizer of cubic splines *xor* extrapolation	local minimizers of cubic splines *and* extrapolation
3) choice of best candidate	———	BO acquisition function
4) acceptance criterion	classic Wolfe conditions	prob. Wolfe conditions

The table already motivates certain design choices, for example the particular choice of the GP-surrogate for $f(t)$, which strongly resembles the classic design. Probabilistic line searches operate in the same scheme as classic ones: 1) they construct a surrogate for the underlying 1D-function 2) they select candidates for evaluation which can interpolate between datapoints or extrapolate 3) a heuristic chooses among the candidate locations and the function is evaluated there 4) the evaluated points are checked for Wolfe-acceptance. The following sections introduce all of these building blocks with greater detail: A robust yet lightweight Gaussian process surrogate on $f(t)$ facilitating analytic optimization (§ 3.1); a simple Bayesian optimization objective for exploration (§ 3.2); and a probabilistic formulation of the Wolfe conditions as a termination criterion (§ 3.3). Appendix D contains a detailed pseudocode of the probabilistic line search; Algorithm 1 very roughly sketches the structure of the probabilistic line search and highlights its essential ingredients.

3.1 Lightweight Gaussian Process Surrogate

We model information about the objective in a probability measure $p(f)$. There are two requirements on such a measure: First, it must be robust to irregularity (low and high variability) of the objective. And second, it must allow analytic computation of discrete candidate points for evaluation, because a line search should not call yet another optimization subroutine itself. Both requirements are fulfilled by a once-integrated Wiener process, i.e. a zero-mean Gaussian process prior $p(f) = \mathcal{GP}(f; 0, k)$ with covariance function

$$k(t, t') = \theta^2 \left[\tfrac{1}{3} \min^3(\tilde{t}, \tilde{t}') + \tfrac{1}{2} |t - t'| \min^2(\tilde{t}, \tilde{t}') \right]. \tag{4}$$

Algorithm 1 probLineSearchSketch(f, y_0, y'_0, σ_{f_0}, $\sigma_{f'_0}$)

```
GP ⇐ initGP(y_0, y'_0, σ_{f_0}, σ_{f'_0})
T, Y, Y' ⇐ initStorage(0, y_0, y'_0)                    ▷ for observed points
t ⇐ 1                                                   ▷ scaled position of initial candidate

while budget not used and no Wolfe-point found do
    [y, y'] ⇐ f(t)                                      ▷ evaluate objective
    T, Y, Y' ⇐ updateStorage(t, y, y')
    GP ⇐ updateGP(t, y, y')
    P^Wolfe ⇐ probWolfe(T, GP)                          ▷ compute Wolfe probability at points in T

    if any P^Wolfe above Wolfe threshold c_W then
        return Wolfe-point
    else
        T_cand ⇐ computeCandidates(GP)                  ▷ positions of new candidates
        EI ⇐ expectedImprovement(T_cand, GP)
        PW ⇐ probWolfe(T_cand, GP)
        t ⇐ where (PW ⊙ EI) is maximal                  ▷ find best candidate among T_cand
    end if
end while

return observed point in T with lowest GP mean since no Wolfe-point found
```

Here $\tilde{t} := t + \tau$ and $\tilde{t}' := t' + \tau$ denote a shift by a constant $\tau > 0$. This ensures this kernel is positive semi-definite, the precise value τ is irrelevant as the algorithm only considers positive values of t (our implementation uses $\tau = 10$). See §3.4 regarding the scale θ^2. With the likelihood of Eq. 3, this prior gives rise to a GP posterior whose mean function is a cubic spline[3] (Wahba, 1990). We note in passing that regression on f and f' from N observations of pairs (y_t, y'_t) can be formulated as a filter (Särkkä, 2013) and thus performed in $\mathcal{O}(N)$ time. However, since a line search typically collects < 10 data points, generic GP inference, using a Gram matrix, has virtually the same, low cost.

Because Gaussian measures are closed under linear maps (Papoulis, 1991, §10), Eq. 4 implies a Wiener process (linear spline) model on f':

$$p(f; f') = \mathcal{GP}\left(\begin{bmatrix} f \\ f' \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k & k^{\partial} \\ \partial k & \partial k \partial \end{bmatrix}\right), \tag{5}$$

3. Eq. 4 can be generalized to the 'natural spline', removing the need for the constant τ (Rasmussen and Williams, 2006, §6.3.1). However, this notion is ill-defined in the case of a single observation, as in the line search.

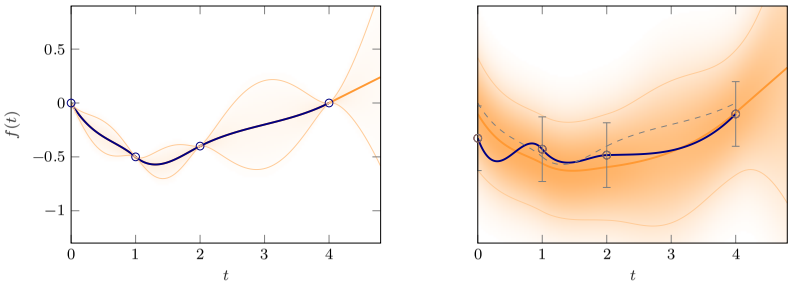


Figure 3: *Integrated Wiener process:* GP marginal posterior of function values; posterior mean in solid orange and, two standard deviations in thinner solid orange, local pdf marginal as shading; function value observations as gray circles (corresponding gradients not shown). Classic interpolation by piecewise cubic spline in dark blue. *Left:* observations are exact; the mean of the GP and the cubic spline interpolator of a classic line search coincide. *Right:* same observations with additive Gaussian noise (error-bars indicate ± 1 standard deviations); noise free interpolator in dashed gray for comparison. The classic interpolator in dark blue, which exactly matches the observations, becomes unreliable; the GP reacts robustly to noisy observations; the GP-mean still consists of piecewise cubic splines.

with (using the indicator function $\mathbb{I}(x) = 1$ if x, else 0)

$$\begin{aligned} k^{\partial}{}_{tt'} &:= \frac{\partial k(t,t')}{\partial t'} &&= \theta^2 \left[\mathbb{I}(t < t')\frac{\tilde{t}^2}{2} + \mathbb{I}(t \geq t')\left(\tilde{t}\tilde{t}' - \frac{\tilde{t}'^2}{2}\right)\right] \\ {}^{\partial}k_{tt'} &:= \frac{\partial k(t,t')}{\partial t} &&= \theta^2 \left[\mathbb{I}(t' < t)\frac{\tilde{t}'^2}{2} + \mathbb{I}(t' \geq t)\left(\tilde{t}\tilde{t}' - \frac{\tilde{t}^2}{2}\right)\right] \\ {}^{\partial}k^{\partial}{}_{tt'} &:= \frac{\partial^2 k(t,t')}{\partial t' \partial t} &&= \theta^2 \min(\tilde{t}, \tilde{t}'). \end{aligned} \tag{6}$$

Given a set of evaluations $(\boldsymbol{t}, \boldsymbol{y}, \boldsymbol{y}')$ (vectors, with elements t_i, y_{t_i}, y'_{t_i}) with independent likelihood 3, the posterior $p(f \mid \boldsymbol{y}, \boldsymbol{y}')$ is a GP with posterior mean function μ and covariance function $\tilde{k}$ as follows:

$$\begin{bmatrix} \mu(t) \\ \mu'(t) \end{bmatrix} = \underbrace{\begin{bmatrix} k_{t\boldsymbol{t}} & k^{\partial}{}_{t\boldsymbol{t}} \\ {}^{\partial}k_{t\boldsymbol{t}} & {}^{\partial}k^{\partial}{}_{t\boldsymbol{t}} \end{bmatrix} \begin{bmatrix} k_{\boldsymbol{tt}} + \sigma_f^2 \boldsymbol{I} & k^{\partial}{}_{\boldsymbol{tt}} \\ {}^{\partial}k_{\boldsymbol{tt}} & {}^{\partial}k^{\partial}{}_{\boldsymbol{tt}} + \sigma_{f'}^2 \boldsymbol{I} \end{bmatrix}^{-1}}_{=:\boldsymbol{g}^{\intercal}(t)} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{y}' \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} \tilde{k}(t,t') & \tilde{k}^{\partial}(t,t') \\ {}^{\partial}\tilde{k}(t',t) & {}^{\partial}\tilde{k}^{\partial}(t,t') \end{bmatrix} = \begin{bmatrix} k_{tt'} & k^{\partial}{}_{tt'} \\ {}^{\partial}k_{t't} & {}^{\partial}k^{\partial}{}_{tt'} \end{bmatrix} - \boldsymbol{g}^{\intercal}(t) \begin{bmatrix} k_{\boldsymbol{t}t'} & k^{\partial}{}_{\boldsymbol{t}t'} \\ {}^{\partial}k_{t'\boldsymbol{t}} & {}^{\partial}k^{\partial}{}_{\boldsymbol{t}t'} \end{bmatrix}$$

The posterior marginal variance will be denoted by $\mathbb{V}(t) = \tilde{k}(t,t)$. To see that μ is indeed piecewise cubic (i.e. a cubic spline), we note that it has at most three non-vanishing

derivatives[4], because

$$\begin{aligned} {}^{\partial^2}k_{tt'} &:= \frac{\partial^2 k(t,t')}{\partial t^2} = \theta^2 \mathbb{I}(t \leq t') & {}^{\partial^3}k_{tt'} &:= \frac{\partial^3 k(t,t')}{\partial t^3} = \theta^2 \mathbb{I}(t \leq t')(t' - t) \\ {}^{\partial^2}k^{\partial}{}_{tt'} &:= \frac{\partial^4 k(t,t')}{\partial t^2 \partial t'} = -\theta^2 \mathbb{I}(t \leq t') & {}^{\partial^3}k^{\partial}{}_{tt'} &:= \frac{\partial^4 k(t,t')}{\partial t^3 \partial t'} = 0. \end{aligned} \tag{8}$$

This piecewise cubic form of μ is crucial for our purposes: having collected N values of f and f', respectively, all local minima of μ can be found analytically in $\mathcal{O}(N)$ time in a single sweep through the 'cells' $t_{i-1} < t < t_i$, $i = 1, \ldots, N$ (here $t_0 = 0$ denotes the start location, where (y_0, y'_0) are 'inherited' from the preceding line search. For typical line searches $N < 10$, c.f. §4. In each cell, $\mu(t)$ is a cubic polynomial with at most one minimum in the cell, found by an inexpensive quadratic computation from the three scalars $\mu'(t_i), \mu''(t_i), \mu'''(t_i)$. This is in contrast to other GP regression models—for example the one arising from a squared exponential kernel—which give more involved posterior means whose local minima can be found only approximately. Another advantage of the cubic spline interpolant is that it does not assume the existence of higher derivatives (in contrast to the Gaussian kernel, for example), and thus reacts robustly to irregularities in the objective. In our algorithm, after each evaluation of (y_N, y'_N), we use this property to compute a short list of *candidates* for the next evaluation, consisting of the $\leq N$ local minimizers of $\mu(t)$ and one additional *extrapolation* node at $t_{\max} + \alpha$, where $t_{\max}$ is the currently largest evaluated t, and α is an extrapolation step size starting at $\alpha = 1$ and doubled after each extrapolation step.[5]

A conceptual (rather than algorithmic) motivation for using the integrated Wiener process as surrogate for the objective, as well as for the described candidate selection, are classic line searches. There, the 1D-objective is modeled by piecewise cubic interpolations between neighboring datapoints. In a sense, this is a non-parametric approach, since a new spline is defined, when a datapoint is added. Classic line searches always only deal with one spline at a time, since they are able to collapse all other parts of the search space. Indeed, for noise free observations, the mean of the posterior GP is identical to the classic cubic interpolations, and thus candidate locations are identical as well; this is illustrated in Figure 3. The non-parametric approach also prevents issues of over-constrained surrogates for more than two datapoints. For example, unless the objective is a perfect cubic function, it is impossible to fit a parametric third order polynomial to it, for more than two noise free observations. All other variability in the objective would need to be explained away by artificially introducing noise on the observations. An integrated Wiener process very naturally extends its complexity with each newly added datapoint without being overly assertive – the encoded assumption is that the objective has *at least* one derivative (which is also observed in this case).

4. There is no well-defined probabilistic belief over f'' and higher derivatives—sample paths of the Wiener process are almost surely non-differentiable almost everywhere (Adler, 1981, §2.2). But $\mu(t)$ is always a member of the reproducing kernel Hilbert space induced by k, thus piecewise cubic (Rasmussen and Williams, 2006, §6.1).
5. For the integrated Wiener process and heteroscedastic noise, the variance *always* attains its maximum exactly at the mid-point between two evaluations; including the variance into the candidate selection biases the existing candidates towards the center (additional candidates might occur between evaluations without local minimizer, even for noise free observations/classic line searches). We did not explore this further since the algorithm showed very good sample efficiency already with the adopted scheme.

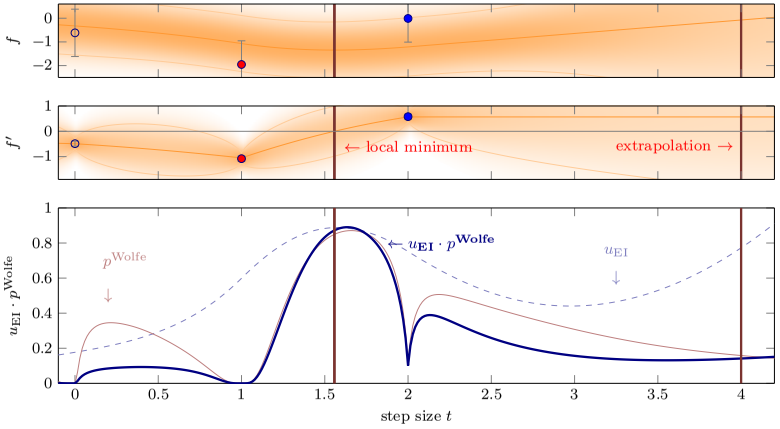


Figure 4: *Candidate selection* by Bayesian optimization. *Top:* GP marginal posterior of function values. Posterior mean in solid orange and, two standard deviations in thinner solid orange, local pdf marginal as shading. The red and the blue point are evaluations of the objective function, collected by the line search. *Middle:* GP marginal posterior of corresponding gradients. Colors same as in top plot. In all three plots the locations of the *two* candidate points (§3.1) are indicated as vertical dark red lines. The left one at about $t_1^{\mathrm{cand}} \approx 1.54$ is a local minimum of the posterior mean in between the red and blue point (the mean of the gradient belief (solid orange, middle plot) crosses through zero here). The right one at $t_2^{\mathrm{cand}} = 4$ is a candidate for extrapolation. *Bottom:* Decision criterion in arbitrary scale: The expected improvement u_{EI} (Eq. 9) is shown in dashed light blue, the Wolfe probability p^{Wolfe} (Eq. 14 and Eq. 16) in light red and their decisive product in solid dark blue. For illustrative purposes all criteria are plotted for the whole t-space. In practice solely the values at t_1^{cand} and t_2^{cand} are computed, compared, and the candidate with the higher value of $u_{\mathrm{EI}} \cdot p^{\mathrm{Wolfe}}$ is chosen for evaluation. In this example this would be the candidate at t_1^{cand}.

3.2 Choosing Among Candidates

The previous section described the construction of $< N+1$ discrete candidate points for the next evaluation. To decide at which of the candidate points to actually call f and f', we make use of a popular acquisition function from Bayesian optimization. *Expected improvement* (Jones et al., 1998) is the expected amount, under the GP surrogate, by which the function $f(t)$ might be smaller than a 'current best' value η (we set $\eta = \min_{i=0,\ldots,N}\{\mu(t_i)\}$, where t_i

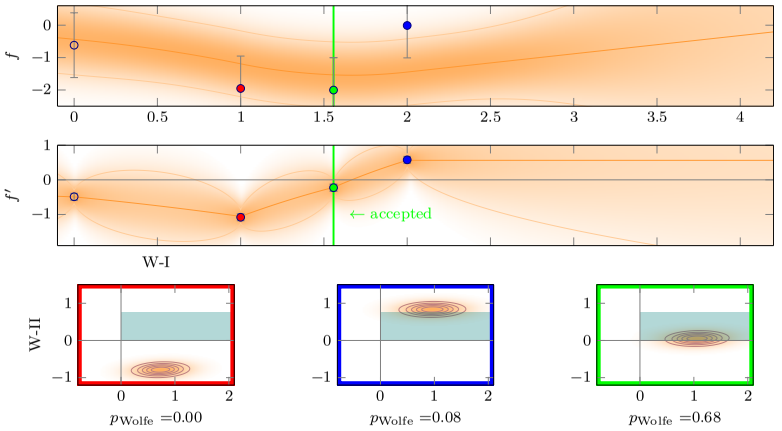


Figure 5: *Acceptance procedure. Top and middle:* plot and colors as in Figure 4 with an additional 'green' observation. *Bottom:* Implied bivariate Gaussian belief over the validity of the Wolfe conditions (Eq. 11) at the red, blue and green point respectively. Points are considered acceptable if their Wolfe probability p_t^{Wolfe} is above a threshold $c_W = 0.3$; this means that at least 30% of the orange 2D Gauss density must cover greenish shaded area. Only the green point fulfills this condition and is therefore accepted.

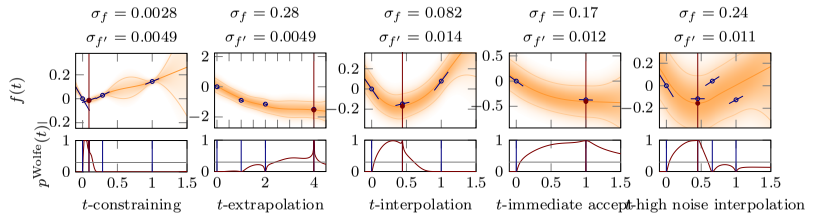


Figure 6: *Curated snapshots* of line searches (from N-I on MNIST), showing variability of the objective's shape and the decision process. *Top row:* GP marginal posterior of function values and evaluations, *bottom row:* approximate p^{Wolfe} over strong Wolfe conditions. Accepted point marked red.

are observed locations),

$$u_{\text{EI}}(t) = \mathbb{E}_{p(f_t \,|\, \boldsymbol{y}, \boldsymbol{y}')}[\min\{0, \eta - f(t)\}] \\ = \frac{\eta - \mu(t)}{2}\left(1 + \operatorname{erf}\frac{\eta - \mu(t)}{\sqrt{2\mathbb{V}(t)}}\right) + \sqrt{\frac{\mathbb{V}(t)}{2\pi}}\exp\left(-\frac{(\eta - \mu(t))^2}{2\mathbb{V}(t)}\right). \tag{9}$$

The next evaluation point is chosen as the candidate maximizing the product of Eq. 9 and Wolfe probability p^{Wolfe}, which is derived in the following section. The intuition is that p^{Wolfe} precisely encodes properties of desired points, but has poor exploration properties; u_{EI} has better exploration properties, but lacks the information that we are seeking a point with low curvature; u_{EI} thus puts weight on (by W-II) clearly ruled out points. An illustration of the candidate proposal and selection is shown in Figure 4.

In principle other acquisition functions (e.g. the upper-confidence bound, GP-UCB (Srinivas et al., 2010)) are possible, which might have a stronger explorative behavior; we opted for u_{EI} since exploration is less crucial for line searches than for general BO and some (e.g. GP-UCB) had one additional parameter to tune. We tracked the sample efficiency of u_{EI} instead and it was very good (low); the experimental Subsection 4.3 contains further comments and experiments on the alternative choices of u_{EI} and p^{Wolfe} as standalone acquisition functions; they performed equally well (in terms of loss and sample efficiency) to their product on the tested setups.

3.3 Probabilistic Wolfe Conditions for Termination

The key observation for a probabilistic extension of the Wolfe conditions W-I and W-II is that they are positivity constraints on two variables a_t, b_t that are both linear projections of the (jointly Gaussian) variables f and f':

$$\begin{bmatrix} a_t \\ b_t \end{bmatrix} = \begin{bmatrix} 1 & c_1 t & -1 & 0 \\ 0 & -c_2 & 0 & 1 \end{bmatrix} \begin{bmatrix} f(0) \\ f'(0) \\ f(t) \\ f'(t) \end{bmatrix} \geq 0. \tag{10}$$

The GP of Eq. (5) on f thus implies, at each value of t, a bivariate Gaussian distribution

$$p(a_t, b_t) = \mathcal{N}\left(\begin{bmatrix} a_t \\ b_t \end{bmatrix}; \begin{bmatrix} m_t^a \\ m_t^b \end{bmatrix}, \begin{bmatrix} C_t^{aa} & C_t^{ab} \\ C_t^{ba} & C_t^{bb} \end{bmatrix}\right), \tag{11}$$

$$\text{with} \quad m_t^a = \mu(0) - \mu(t) + c_1 t \mu'(0)$$

$$m_t^b = \mu'(t) - c_2 \mu'(0) \tag{12}$$

$$\text{and} \quad C_t^{aa} = \tilde{k}_{00} + (c_1 t)^2 {}^{\partial}\tilde{k}_{00}^{\partial} + \tilde{k}_{tt} + 2[c_1 t({}^{\partial}\tilde{k}_{00}^{\partial} - {}^{\partial}\tilde{k}_{0t}) - \tilde{k}_{0t}]$$

$$C_t^{bb} = c_2^2 {}^{\partial}\tilde{k}_{00}^{\partial} - 2c_2 {}^{\partial}\tilde{k}_{0t}^{\partial} + {}^{\partial}\tilde{k}_{tt}^{\partial} \tag{13}$$

$$C_t^{ab} = C_t^{ba} = -c_2({}^{\partial}\tilde{k}_{00}^{\partial} + c_1 t {}^{\partial}\tilde{k}_{00}^{\partial}) + c_2 {}^{\partial}\tilde{k}_{0t} + {}^{\partial}\tilde{k}_{t0} + c_1 t {}^{\partial}\tilde{k}^{\partial}{}_{0t} - \tilde{k}_{tt}^{\partial}.$$

The quadrant probability $p_t^{\text{Wolfe}} = p(a_t > 0 \wedge b_t > 0)$ for the Wolfe conditions to hold, is an integral over a bivariate normal probability,

$$p_t^{\text{Wolfe}} = \int_{-\frac{m_t^a}{\sqrt{C_t^{aa}}}}^{\infty} \int_{-\frac{m_t^b}{\sqrt{C_t^{bb}}}}^{\infty} \mathcal{N}\left(\begin{bmatrix} a \\ b \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho_t \\ \rho_t & 1 \end{bmatrix}\right) da\, db, \tag{14}$$

with correlation coefficient $\rho_t = C_t^{ab}/\sqrt{C_t^{aa} C_t^{bb}}$. It can be computed efficiently (Drezner and Wesolowsky, 1990), using readily available code.[6] The line search computes this probability for all evaluation nodes, after each evaluation. If any of the nodes fulfills the Wolfe conditions with $p_t^{\text{Wolfe}} > c_W$, greater than some threshold $0 < c_W \leq 1$, it is accepted and returned. If several nodes simultaneously fulfill this requirement, the most recently evaluated node is returned; there are additional safeguards for cases where e.g. no Wolfe-point can be found, which can be deduced from the pseudo-code in Appendix D; they are similar to standard safeguards of classic line search routines (e.g. returning the node of lowest mean). Section 3.4.1 below motivates fixing $c_W = 0.3$. The acceptance procedure is illustrated in Figure 5.

3.3.1 Approximation for Strong Conditions:

As noted in Section 2.1.1, deterministic optimizers tend to use the strong Wolfe conditions, which use $|f'(0)|$ and $|f'(t)|$. A precise extension of these conditions to the probabilistic setting is numerically taxing, because the distribution over $|f'|$ is a non-central χ-distribution, requiring customized computations. However, a straightforward variation to 14 captures the spirit of the strong Wolfe conditions that large positive derivatives should not be accepted: Assuming $f'(0) < 0$ (i.e. that the search direction is a descent direction), the strong second Wolfe condition can be written exactly as

$$0 \leq b_t = f'(t) - c_2 f'(0) \leq -2c_2 f'(0). \tag{15}$$

The value $-2c_2 f'(0)$ is bounded to 95% confidence by

$$-2c_2 f'(0) \lesssim 2c_2(|\mu'(0)| + 2\sqrt{\mathbb{V}'(0)}) =: \bar{b}. \tag{16}$$

Hence, an approximation to the strong Wolfe conditions can be reached by replacing the infinite upper integration limit on b in Eq. 14 with $(\bar{b} - m_t^b)/\sqrt{C_t^{bb}}$. The effect of this adaptation, which adds no overhead to the computation, is shown in Figure 2 as a dashed line.

3.4 Eliminating Hyper-parameters

As a black-box inner loop, the line search should not require any tuning by the user. The preceding section introduced six so-far undefined parameters: $c_1, c_2, c_W, \theta, \sigma_f, \sigma_{f'}$. We will now show that c_1, c_2, c_W, can be fixed by hard design decisions: θ can be eliminated by standardizing the optimization objective within the line search; and the noise levels can be estimated at runtime with low overhead for finite-sum objectives of the form in Eq. 1. The result is a parameter-free algorithm that effectively *removes* the one most problematic parameter from SGD—the learning rate.

3.4.1 Design Parameters c_1, c_2, c_W

Our algorithm inherits the Wolfe thresholds c_1 and c_2 from its deterministic sibling. We set $c_1 = 0.05$ and $c_2 = 0.5$. This is a standard setting that yields a 'lenient' line search, i.e. one

6. e.g. `http://www.math.wsu.edu/faculty/genz/software/matlab/bvn.m`

that accepts most descent points. The rationale is that the stochastic aspect of SGD is not always problematic, but can also be helpful through a kind of 'annealing' effect.

The acceptance threshold c_W is a new design parameter arising only in the probabilistic setting. We fix it to $c_W = 0.3$. To motivate this value, first note that in the noise-free limit, all values $0 < c_W < 1$ are equivalent, because p^{Wolfe} then switches discretely between 0 and 1 upon observation of the function. A back-of-the-envelope computation, assuming only two evaluations at $t = 0$ and $t = t_1$ and the same fixed noise level on f and f' (which then cancels out), shows that function values barely fulfilling the conditions, i.e. $a_{t_1} = b_{t_1} = 0$, can have $p^{\text{Wolfe}} \sim 0.2$ while function values at $a_{t_1} = b_{t_1} = -\epsilon$ for $\epsilon \rightarrowtail 0$ with 'unlucky' evaluations (both function and gradient values one standard-deviation from true value) can achieve $p^{\text{Wolfe}} \sim 0.4$. The choice $c_W = 0.3$ balances the two competing desiderata for precision and recall. Empirically (Fig. 6), we rarely observed values of p^{Wolfe} close to this threshold. Even at high evaluation noise, a function evaluation typically either clearly rules out the Wolfe conditions, or lifts p^{Wolfe} well above the threshold. A more in-depth analysis of c_1, c_2, and c_W is done in the experimental Section 4.2.1.

3.4.2 Scale θ

The parameter θ of Eq. 4 simply scales the prior variance. It can be eliminated by scaling the optimization objective: We set $\theta = 1$ and scale $y_i \leftarrowtail (y_i - y_0)/|y_0'|, y_i' \leftarrowtail y_i'/|y_0'|$ within the code of the line search. This gives $y(0) = 0$ and $y'(0) = -1$, and typically ensures the objective ranges in the single digits across $0 < t < 10$, where most line searches take place. The division by $|y_0'|$ causes a non-Gaussian disturbance, but this does not seem to have notable empirical effect.

3.4.3 Noise Scales $\sigma_f, \sigma_{f'}$

The likelihood 3 requires standard deviations for the noise on both function values (σ_f) and gradients ($\sigma_{f'}$). One could attempt to learn these across several line searches; but the resulting estimator would be biased. In exchangeable models as captured by Eq. 1, however, the variance of the loss and its gradient can be estimated locally and unbiased, directly for the mini-batch, at low computational overhead—an approach already advocated by Schaul et al. (2013). We collect the empirical statistics

$$\hat{S}(x) := \frac{1}{m}\sum_j^m \ell^2(x, y_j), \quad \text{and} \quad \hat{\nabla S}(x) := \frac{1}{m}\sum_j^m \nabla\ell(x, y_j)^{\odot 2} \tag{17}$$

(where $^{\odot 2}$ denotes the element-wise square) and estimate, at the beginning of a line search from x_k,

$$\sigma_f^2 = \frac{1}{m-1}\left(\hat{S}(x_k) - \hat{\mathcal{L}}(x_k)^2\right) \text{ and } \sigma_{f'}^2 = s_i^{\odot 2\intercal}\left[\frac{1}{m-1}\left(\hat{\nabla S}(x_k) - (\nabla\hat{\mathcal{L}}(x_k))^{\odot 2}\right)\right]. \tag{18}$$

This amounts to the assumption that noise on the gradient is independent (see also Appendix A). We finally scale the two empirical estimates as described in Section §3.4.2: $\sigma_f \leftarrowtail \sigma_f/|y'(0)|$, and ditto for $\sigma_{f'}$. The overhead of this estimation is small if the computation of $\ell(x, y_j)$ itself is more expensive than the summation over j. In the neural network examples N-I and N-II of the experimental Section 4, the additional steps added only $\sim 1\%$ cost overhead to the evaluation of the loss. This is rather at the lower end for these models; A more general statement about memory and time requirements for neural networks can be found in Sections 3.6 and 3.7. Estimating noise separately for each input dimension captures the often inhomogeneous structure among gradient elements, and its effect on the noise along the projected direction. For example, in multi-layer models, gradient noise is typically higher on weights between the input and first hidden layer, hence line searches along the corresponding directions are noisier than those along directions affecting higher-level weights.

3.4.4 Propagating Step Sizes Between Line Searches

As will be demonstrated in §4, the line search can find good step sizes even if the length of the direction s_i is mis-scaled. Since such scale issues typically persist over time, it would be wasteful to have the algorithm re-fit a good scale in each line search. Instead, we propagate step lengths from one iteration of the search to another: We set the initial search direction to $s_0 = -\alpha_0\nabla\hat{\mathcal{L}}(x_0)$ with some initial learning rate α_0. Then, after each line search ending at $x_i = x_{i-1} + t_* s_i$, the next search direction is set to $s_{i+1} = -\alpha_{\text{ext}} \cdot t_* \alpha_0 \nabla\hat{\mathcal{L}}(x_i)$ (with $\alpha_{\text{ext}} = 1.3$). Thus, the next line search starts its extrapolation at 1.3 times the step size of its predecessor (Section 4.2.2 for details).

3.5 Relation to Bayesian Optimization and Noise-Free Limit

The probabilistic line search algorithm is closely related to Bayesian optimization (BO) since it approximately minimizes a 1D-objective under potentially noisy function evaluations. It thus uses notions of BO (e.g. a GP-surrogate for the objective, and an acquisition function to discriminate locations for the next evaluation of the loss), but there are some differences concerning the aim, requirements on computational efficiency, and termination condition, which are shortly discussed here: (i) *Performance measure:* The final performance measure in BO is usually the lowest found value of the objective function. Line searches are subroutines inside of a greedy, iterative optimization machine, which usually performs several thousand steps (and line searches); many, very approximate steps often performs better than taking less, but preciser steps. (ii) *Termination:* The termination condition of a line search is imposed from the outside in the form of the Wolfe conditions. Stricter Wolfe conditions do not usually improve the performance of the outer optimizer, thus, no matter if a better (lower) minimum could be found, any Wolfe-point is acceptable at all times. (iii) *Sample efficiency:* Since the last evaluation from the previous line search can be re-used in the current line search, only one additional value and gradient evaluation is enough to terminate the procedure. This 'immediate-accept' is the desired behavior if the learning rate is currently well calibrated. (iv) *Locations for evaluation:* BO, usually calls an optimizer to maximize some acquisition function, and the preciseness of this optimization is crucial for performance. Line searches just need to find a Wolfe-acceptable point; classic line searches suggest that it is enough to look at plausible locations, like minimizer of a local interpolator, or some rough extrapolation point; this inexpensive heuristic usually works rather well. (v) *Exploration:* BO needs to solve an intricate trade-off problem in between exploring enough of the parameters space for possible locations of minima, and exploiting locations around them further. Since line searches are only concerned with finding

a Wolfe-point, they do not need to explore the parameter space of possible step sizes to that extend; crucial features are rather the possibility to explore somewhat larger steps than previous ones (which is done by extrapolation-candidates), and likewise to shorted steps (which is done by interpolation-candidates).

In the limit of noise free observed gradients and function values ($\sigma_f = \sigma_{f'} = 0$) the probabilistic line search behaves like its classic parent, except for very slight variations in the candidate choice (building block 3): The GP-mean reverts to the classic interpolator; all candidate locations are thus identical, but the probabilistic line search might propose a second option, since (even if there is a local minimizer) it *always* also proposes an extrapolation candidate. This is illustrated in the following table.

building block	**classic**	**probabilistic (noise free)**
1) 1D surrogate for objective $f(t)$	piecewise cubic splines	GP-mean identical to classic interpolator
2) candidate selection	local minimizer of cubic splines *xor* extrapolation	local minimizer of cubic splines *or* extrapolation
3) choice of best candidate	———	BO acquisition function
4) acceptance criterion	classic Wolfe conditions	p^{Wolfe} identical to classic Wolfe conditions

3.6 Computational Time Overhead

The line search routine itself has little memory and time overhead; most importantly it is independent of the dimensionality of the optimization problem. After every call of the objective function, the GP (§3.1) needs to be updated which, at most, is at the cost of inverting a $2N \times 2N$-matrix, where N usually is equal to $1, 2$, or 3 but never > 10. In addition, the bivariate normal integral p_t^{Wolfe} of Eq. 14 needs to be computed at most N times. On a laptop, one evaluation of p_t^{Wolfe} costs about 100 microseconds. For the choice among proposed candidates (§3.2), again at most N, for each, we need to evaluate p_t^{Wolfe} and $u_{\text{EI}}(t)$ (Eq. 9) where the latter comes at the expense of evaluating two error functions. Since all of these computations have a fixed cost (in total some milliseconds on a laptop), the relative overhead becomes less the more expensive the evaluation of $\nabla\hat{\mathcal{L}}(x)$.

The largest overhead actually lies outside of the actual line search routine. In case the noise levels σ_f and $\sigma_{f'}$ are not known, we need to estimate them. The approach we took is described in Section 3.4.3 where the variance of $\nabla\hat{\mathcal{L}}$ is estimated using the sample variance of the mini-batch, each time the objective function is called. Since in this formulation the variance estimation is about half as expensive as one backward pass of the net, the time overhead depends on the relative cost of the feed forward and backward passes (Balles et al., 2017). If forward and backward pass are the same cost, the most straightforward implementation of the variance estimation would make each function call < 1.3 times as expensive. This is an *upper bound* and the actual cost is usually lower.[7] At the same time though, *all* exploratory experiments which very considerably increase the time spend when

7. It is desirable to decrease this value in the future reusing computation results or by approximation but this is beyond this discussion.

using SGD with a hand tuned learning rate schedule need not be performed anymore. In Section 4.1 we will also see that SGD using the probabilistic line search often needs less function evaluations to converge, which might lead to overall faster convergence in wall clock time than classic SGD in a single run.

3.7 Memory Requirement

Vanilla SGD, at all times, keeps around the current optimization parameters $x \in \mathbb{R}^D$ and the gradient vector $\nabla\hat{\mathcal{L}}(x) \in \mathbb{R}^D$. In addition to this, the probabilistic line search needs to store the estimated gradient variances $\Sigma'(x) = (1-m)^{-1}(\hat{\nabla}S(x) - \nabla\hat{\mathcal{L}}(x)^{\odot 2})$ (Eq. 18) of same size. The memory requirement of SGD+probLS is thus comparable to AdaGrad or Adam. If combined with a search direction other than SGD always one additional vector of size D needs to be stored.

4. Experiments

This section reports on an extensive set of experiments to characterise and test the line search. The overall evidence from these tests is that the line search performs well and is relatively insensitive to the choice of its internal hyper-parameters as well the mini-batch size. We performed experiments on two multi-layer perceptrons N-I and N-II; both were trained on two well known datasets MNIST and CIFAR-10.

- N-I: fully connected net with 1 hidden layer and 800 hidden units + biases, and 10 output units, sigmoidal activation functions and a cross entropy loss. Structure without biases: 784-800-10. Many authors used similar nets and reported performances.[8]
- N-II: fully connected net with 3 hidden layers and 10 output units, tanh-activation functions and a squared loss. Structure without biases: 784-1000-500-250-10. Similar nets were also used for example in Martens (2010) and Sutskever et al. (2013).
- MNIST (LeCun et al., 1998): multi-class classification task with 10 classes: hand-written digits in gray-scale of size 28×28 (numbers '0' to '9'); training set size 60 000, test set size 10 000.
- CIFAR-10 (Krizhevsky and Hinton, 2009): multi-class classification task with 10 classes: color images of natural objects (horse, dog, frog,...) of size 32×32; training set size 50 000, test set size 10 000; like other authors, we only used the "batch 1" sub-set of CIFAR-10 containing 10 000 training examples.

In addition we train logistic regressors with sigmoidal output (N-III) on the following binary classification tasks:

- Wisconsin Breast Cancer Dataset (WDBC) (Wolberg et al., 2011): binary classification of tumors as either 'malignant' or 'benign'. The set consist of 569 examples of which we used 169 to monitor generalization performing; thus 400 remain for the training set; 30 features describe for example radius, area, symmetry, et cetera. In comparison

8. `http://yann.lecun.com/exdb/mnist/`

to the other datasets and networks, this yields a very low dimensional optimization problem with only 30 (+1 bias) input parameters as well as just a small number of datapoints.

- GISETTE (Guyon et al., 2005): binary classification of the handwritten digits '4' and '9'. The original 28×28 images are taken from the MNIST datset; then the feature set was expanded and consists of the original normalized pixels, plus a randomly selected subset of products of pairs of features, which are slightly biased towards the upper part of the image; in total there are 5000 features, instead of 784 as in the original MNIST. The size of the training set and test set is 6000 and 1000 respectively.
- EPSILON: synthetic dataset from the PASCAL Challenge 2008 for binary classification. It consists of 400 000 training set datapoint and 100 000 test set datapoints, each having 2000 features.

In the text and figures, SGD using the probabilistic line search will occasionally be denoted as SGD+PROBLS. Section 4.1 contains experiments on the sensitivity to varying gradient noise levels (mini-batch sizes) performed on both multi-layer perceptrons N-I and N-II, as well as on the logistic regressor N-III. Section 4.2 discusses sensitivity to the hyper-parameters choices introduced in Section 3.4 and Section 4.3 contains additional diagnostics on step size statistics. Each single experiment was performed 10 times with different random seeds that determined the starting weights and the mini-batch selection and seeds were shared across all experiments. We report all results of the 10 instances as well as means and standard deviations.

4.1 Varying Mini-batch Sizes

The noise level of the gradient estimate $\nabla\hat{\mathcal{L}}(x)$ and the loss $\hat{\mathcal{L}}(x)$ is determined by the mini-batch size m and ultimately there should exist an optimal m that maximizes the optimizer's performance in wall-clock-time. In practice of course the cost of computing $\nabla\hat{\mathcal{L}}(x)$ and $\hat{\mathcal{L}}(x)$ is not necessarily linear in m since it is upper bounded by the memory capacity of the hardware used. We assume here that the mini-batch size is chosen by the user; thus we test the line search with the default hyper-parameter setting (see Sections 3.4 and 4.2) on four different mini-batch sizes:

- $m = 10, 100, 200$ and 1000 (for MNIST, CIFAR-10, and EPSILON)
- $m = 10, 50, 100$, and 400 (for WDBC and GISETTE)

which correspond to increasing signal-to-noise ratios. Since the training set of WDBC only consists of 400 datapoints, the run with the larges mini-batch size of 400 in fact runs full-batch gradient descent on WDBC; this is not a problem, since—as discussed above—the probabilistic line search can also handle noise free observations.[9] We compare to SGD-runs using a fixed step size (which is typical for these architectures) and an annealed step size with annealing schedule $\alpha_t = \alpha_0/t$. Because annealed step sizes performed much worse than

9. Since the dataset size M of WDBC is very small, we used the factor $(M-m)/(mM)$ instead of $1/m$ to scale the sample variances of Eq. 17. The former encodes sampling mini-batches $\mathcal{B}$ with replacement, the latter without replacement; for $m \ll M$ both factors are nearly identical.

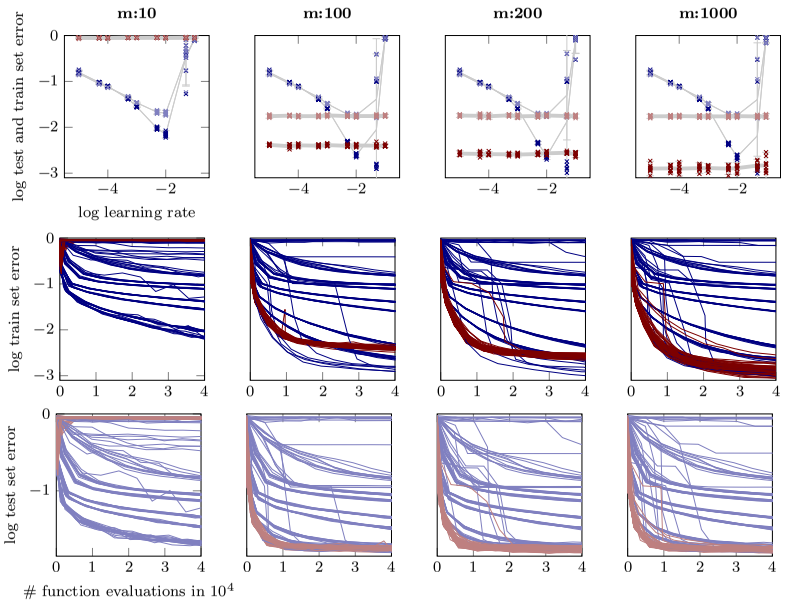


Figure 7: Performance of N-II on MNIST for *varying mini-batch sizes*. *Top:* final logarithmic test set and train set error after 40 000 function evaluations of training versus a large range of learning rates each for 10 different initializations. SGD-runs with fixed learning rates are shown in light blue (test set) and dark blue (train set); SGD+PROBLS-runs in light red (test set) and dark red (train set); means and two standard deviations for each of the 10 runs in gray. *Columns* from left to right refer to different mini-batch sizes m of 10, 100, 200 and 1000 which correspond to decreasing relative noise in the gradient observations. Not surprisingly the performance of SGD-runs with a fixed step size are very sensitive to the choice of this step size. SGD using the probabilistic line search adapts initially mis-scaled step sizes and performs well across the whole range of initial learning rates. *Middle and bottom:* Evolution of the logarithmic test and train set error respectively for all SGD-runs and SGD+PROBLS-runs versus # function evaluations (colors as in top plot). For mini-batch sizes of $m = 100, 200$ and 1000 *all* instances of SGD using the probabilistic line search reach the same best test set error. Similarly a good train set error is reached very fast by SGD+PROBLS. Only very few instances of SGD with a fixed learning rate reach a better train set error (and this advantage usually does not translate to test set error). For very small mini-batch sizes ($m = 10$, first column) the line search performs poorly on this architecture, most likely because of the variance estimation becoming too inaccurate.

SGD+fixed step size, we will only report on the latter results in the plots.[10] Since classic SGD without the line search needs a hand crafted learning rate we search on exhaustive logarithmic grids of

$$\alpha_{\text{SGD}}^{\text{N-I}} = [10^{-5}, 5\cdot 10^{-5}, 10^{-4}, 5\cdot 10^{-4}, 10^{-3}, 5\cdot 10^{-3}, 10^{-2}, 5\cdot 10^{-2}, 10^{-1}, 5\cdot 10^{-1}]$$
$$\alpha_{\text{SGD}}^{\text{N-II}} = [\alpha_{\text{SGD}}^{\text{N-I}},\ 1.0,\ 1.5,\ 2.0,\ 2.5,\ 3.0,\ 3.5,\ 4.0]$$
$$\alpha_{\text{SGD}}^{\text{N-III}} = [10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}].$$

We run 10 different initialization for each learning rate, each mini-batch size and each net and dataset combination ($10\cdot 4\cdot(2\cdot 10+2\cdot 17+3\cdot 11)=3480$ runs in total) for a large enough budget to reach convergence; and report all numbers. Then we perform the same experiments using the same seeds and setups with SGD using the probabilistic line search and compare the results. For SGD+probLS, α_{SGD} is the initial learning rate which is used in the very first step. After that, the line search automatically adapts the learning rate, and shows no significant sensitivity to its initialization.

Results of N-I and N-II on both, MNIST and CIFAR-10 are shown in Figures 7, 14, 15, and 16; results of N-III on WDBC, GISETTE and EPSILON are shown in Figures 18, 17, and 19 respectively. All instances (SGD and SGD+probLS) get the same computational budget (number of mini-batch evaluations) and not the same number of optimization steps. The latter would favour the probabilistic line search since, on average, a bit more than one mini-batch is evaluated per step. Likewise, all plots show performance measure versus the number of mini-batch evaluations, which is proportional to the computational cost.

All plots show similar results: While classic SGD is sensitive to the learning rate choice, the line search-controlled SGD performs as good, close to, or sometimes even better than the (in practice unknown) optimal classic SGD instance. In Figure 7, for example, SGD+probLS converges much faster to a good test set error than the best classic SGD instance. In all experiments, across a reasonable range of mini-batch sizes m and of initial α_{SGD} values, the line search quickly identified good step sizes α_t, stabilized the training, and progressed efficiently, reaching test set errors similar to those reported in the literature for tuned versions of these kind of architectures and datasets. The probabilistic line search thus effectively removes the need for exploratory experiments and learning-rate tuning.

Overfitting and training error curves: The training error of SGD+probLS often plateaus earlier than the one of vanilla SGD, especially for smaller mini-batch sizes. This does not seem to impair the performance of the optimizer on the test set. We did not investigate this further, since it seemed like a nice natural annealing effect; the exact causes are unclear for now. One explanation might be that the line search does indeed improve overfitting, since it tries to measure descent (by Wolfe conditions which rely on the *noise-informed* GP). This means that, if—close to a minimum—successive acceptance decisions can not identify a descent direction anymore, diffusion might set in.

4.2 Sensitivity to Design Parameters

Most, if not all, numerical methods make implicit or explicit choices about their hyper-parameters. Most of these are never seen by the user since they are either estimated at run

10. An example of annealed step size performance can be found in Mahsereci and Hennig (2015).

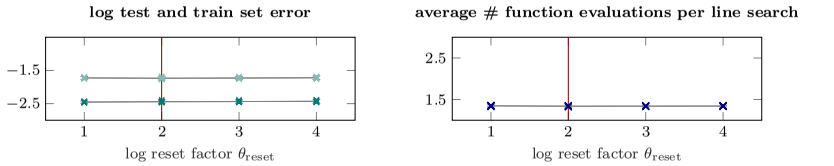


Figure 8: *Sensitivity to varying hyper-parameters* θ_{reset}. Plot and color coding as in Figure 9. Adopted parameter in dark red at $\theta_{\text{reset}}=100$. Resetting the GP scale occurs very rarely. For example for $\theta_{\text{reset}}=100$ the reset occurred in 0.02% of all line searches.

time, or set by design to a fixed, approximately insensitive value. Well known examples are the discount factor in ordinary differential equation solvers (Hairer et al., 1987, §2.4), or the Wolfe parameters c_1 and c_2 of classic line searches (§3.4.1). The probabilistic line search inherits the Wolfe parameters c_1 and c_2 from its classical counterpart as well as introducing two more: The Wolfe threshold c_W and the extrapolation factor α_{ext}. c_W does not appear in the classical formulation since the objective function can be evaluated exactly and the Wolfe probability is binary (either fulfilled or not). While c_W is thus a natural consequence of allowing the line search to model noise explicitly, the extrapolation factor α_{ext} is the result of the line search favoring shorter steps, which we will discuss below in more detail, but most prominently because of bias in the line search's first gradient observation.

In the following sections we will give an intuition about the task of the most influential design parameters c_2, c_W, and α_{ext}, discuss how they affect the probabilistic line search, and validate good design choices through exploring the parameter space and showing insensitivity to most of them. All experiments on hyper-parameter sensitivity were performed training N-II on MNIST with mini-batch size $m=200$. For a full search of the parameter space c_W-c_2-α_{ext} we performed 4950 runs in total with 495 different parameter combinations. All results are reported.

4.2.1 Wolfe II Parameter c_2 and Wolfe Threshold c_W

As described in Section 3.4, c_2 encodes the strictness of the curvature condition W-II. Pictorially speaking, a larger c_2 extends the range of acceptable gradients (green shaded are in the lower part of Figure 5) and leads to a lenient line search while a smaller value of c_2 shrinks this area, leading to a stricter line search. c_W controls how certain we want to be that the Wolfe conditions are actually fulfilled (pictorially, how much mass of the 2D-Gauss need to lie in the green shaded area). In the extreme case of complete uncertainty about the collected gradients and function values (roughly $\det[\text{cov}[a,b]]\to\infty$) p^{Wolfe} will always be ≤ 0.25, if the strong Wolfe conditions are imposed. In the limit of certain observations ($\sigma_f, \sigma_{f'}\to 0$) p^{Wolfe} is binary and reverts to the classic Wolfe criteria. An overly strict line search, therefore (e.g. $c_W=0.99$ and/ or $c_2=0.1$), will still be able to optimize the objective function well, but will waste evaluations at the expense of efficiency. Figure 10 explores the c_2-c_W parameter space (while keeping α_{ext} fixed at 1.3). The left column shows final test and train set error, the right column the average number of function evaluations

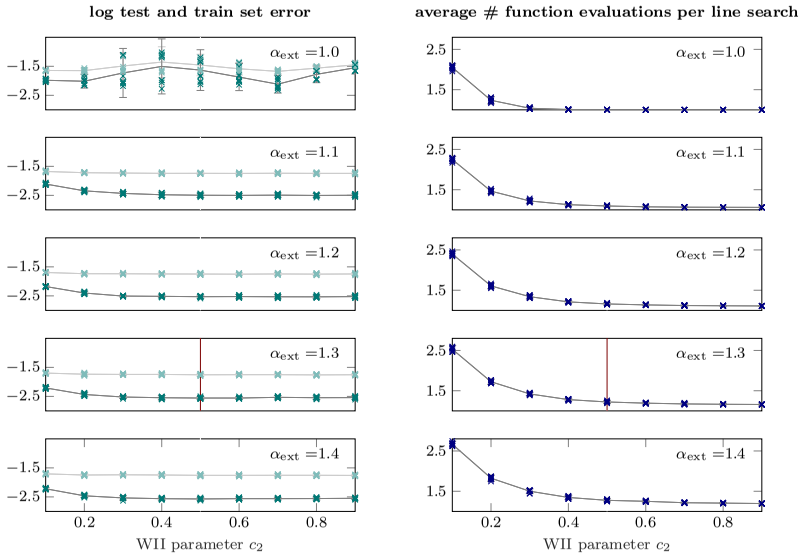


Figure 9: *Sensitivity to varying hyper-parameters* c_2, and α_{ext}. Runs were performed training N-II on MNIST with mini-batch size $m = 200$. For each parameter setting 10 runs with different initializations were performed. *Left column:* logarithmic test set error (light green) and train set error (dark green) after 40 000 function evaluations; mean and ± two standard deviations of the 10 runs in gray. *Right Column:* average number of function evaluations per line search. A low number indicates an efficient line search procedure (perfect efficiency at 1). For most parameter combinations this lies around $\approx 1.3 - 1.5$. Only at extreme parameter values, for example $\alpha_{\text{ext}} = 1.0$, which amounts to no extrapolation at all in between successive line searches, the line search performs poorer. The hyper-parameters adopted in the line search implementation are indicated as vertical dark red line at $\alpha_{\text{ext}} = 1.3$ and $c_2 = 0.5$.

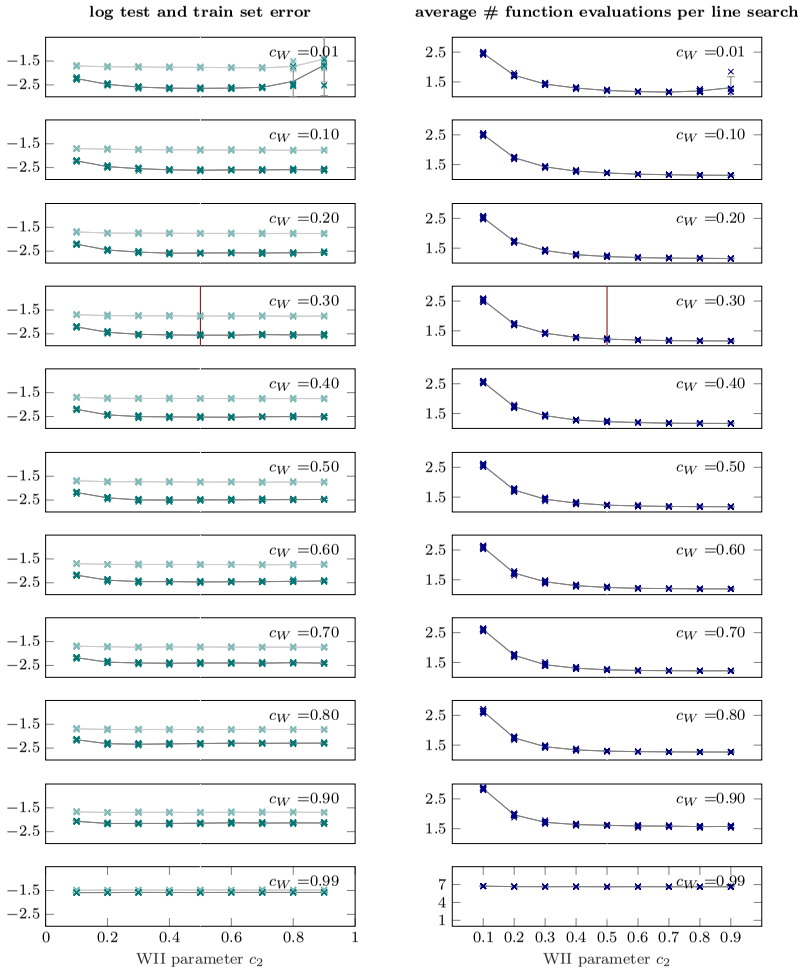


Figure 10: *Sensitivity to varying hyper-parameters* c_2, and c_W. Plot and color coding as in Figure 9 but this time for varying c_W instead of α_{ext}. *Right Column:* Again a low number indicates an efficient line search procedure (perfect efficiency at 1). For most parameter combinations this lies around $\approx 1.3 - 1.5$. Only at extreme parameter values for example $c_W = 0.99$, which amounts to imposing nearly absolute certainty about the Wolfe conditions, the line search becomes less efficient. Adopted parameters again in dark red at $c_W = 0.3$ and $c_2 = 0.5$

per line search, both versus different choices of Wolfe parameter c_2. The left column thus shows the overall performance of the optimizer, while the right column is representative for the computational efficiency of the line search. Intuitively, a line search which is minimally invasive (only corrects the learning rate, when it is really necessary) is preferred. Rows in Figure 10 show the same plot for different choices of the Wolfe threshold c_W.

The effect of strict c_2 can be observed clearly in Figure 10 where for smaller values of $c_2 <\approx 0.2$ the average number of function evaluations spend in one line search goes up slightly in comparison to looser restrictions on c_2, while still a very good perfomace is reached in terms of train and test set error. Likewise, the last row of Figure 10 for the extreme value of $c_W = 0.99$ (demanding 99% certainty about the validity if the Wolfe conditions), shows significant loss in computational efficiency having an average number of 7 function evaluations per line search. Besides loosing efficiency, it is still optimizing the objective well. Lowering this threshold a bit to 90% increases the computational efficiency of the line search to be nearly optimal again.

Ideally, we want to trade off the desiderata of being strict enough to reject too small and too large steps that prevent the optimizer to converge, but being lenient enough to allow all other reasonable steps, thus increasing computational efficiency. The values $c_W = 0.3$ and $c_2 = 0.5$, which are adopted in our current implementation are marked as dark red vertical lines in Figure 10.

4.2.2 Extrapolation Factor α_{EXT}

The extrapolation parameter α_{ext}, introduced in Section 3.4.4, pushes the line search to try a larger learning rate first, than the one which was accepted in the previous step. Figure 9 is structured like Figure 10, but this time explores the line search sensitivity in the c_2-α_{ext} parameter space (abscissa and rows respectively) while keeping c_W fixed at 0.3. Unless we choose $\alpha_{\text{ext}} = 1.0$ (no step size increase between steps) in combination with a lenient choice of c_2 the line search performs well. For now we adopt $\alpha_{\text{ext}} = 1.3$ as default value which again is shown as dark red vertical line in Figure 9.

The introduction of α_{ext} might seem arbitrary at first, but is a necessity and well-working fix because of a few shortcomings of the current design. First, the curvature condition W-II is the single condition that prevents too small steps and pushes optimization progress. On the other hand both W-I and W-II simultaneously penalize too large steps (see Figure 1 for a sketch). This is not a problem in case of deterministic observation ($\sigma_f, \sigma_{f'} \to 0$), where W-II undoubtedly decides if a gradient is still too negative. Unless W-II is chosen very tightly (small c_2) or c_W unnecessarily large (both choices, as discussed above, are undesirable), in the presence of noise, p^{Wolfe} will thus be more reliable in preventing overshooting than pushing progress. The first row of Figure 9 illustrates this behavior, where the performance drops somewhat if no extrapolation is done ($\alpha_{\text{ext}} = 1.0$) in combination with a looser version of W-II (larger c_2).

Another factor that contributes towards accepting small rather than larger learning rates is a bias introduced in the first observation of the line search at $t = 0$. Observations $y'(t)$ that the GP gets to see are projections of the gradient sample $\nabla\hat{\mathcal{L}}(t)$ onto the search direction $s = -\nabla\hat{\mathcal{L}}(0)$. Since the first observations $y'(0)$ is computed from the same mini-batch as the search direction (not doing this would double the optimizer's computational cost) an

inevitable bias is introduced of approximate size of $\cos^{-1}(\gamma)$ (where γ is the expected angle between gradient evaluations from two independent mini-batches at $t = 0$). Since the scale parameter θ of the Wiener process is implicitly set by $y'(0)$ (§3.4.2), the GP becomes more uncertain at unobserved points than it needs to be; or alternatively expects the 1D-gradient to cross zero at smaller steps, and thus underestimates a potential learning rate. The posterior *at* observed positions is little affected. The over-estimation of θ rather pushes the posterior towards the likelihood (since there is less model to trust) and thus still gives a reliable measure for $f(t)$ and $f'(t)$. The effect on the Wolfe conditions is similar. With $y'(0)$ biased towards larger values, the Wolfe conditions, which measure the drop in projected gradient norm, are thus prone to accept larger gradients combined with smaller function values, which again is met by making small steps. Ultimately though, since candidate points at $t^{\text{cand}} > 0$ that are currently queried for acceptance, are always observed and unbiased, this can be controlled by an appropriate design of the Wolfe factor c_2 (§3.4.1 and §4.2.1) and of course α_{ext}.

4.2.3 Full Hyper-Parameter Search: c_W-c_2-α_{EXT}

An exhaustive performance evaluation on the whole c_W-c_2-α_{ext}-grid is shown in Appendix C in Figures 20-24 and Figures 25-35. As discussed above, it shows the necessity of introducing the extrapolation parameter α_{ext} and shows slightly less efficient performance for obviously undesirable parameter combinations. In a large volume of the parameter space, and most importantly in the vicinity of the chosen design parameters, the line search performance is stable and comparable to carefully hand tuned learning rates.

4.2.4 Safeguarding Mis-scaled GPs: θ_{RESET}

For completeness, we performed an additional experiment on the threshold parameter which is denoted by θ_{reset} in the pseudo-code (Appendix D) and safeguards against GP mis-scaling. The introduction of noisy observations necessitates to model the variability of the 1D-function, which is described by the kernel scale parameter θ. Setting this hyper-parameter is implicitly done by scaling the observation input, assuming a similar scale than in the previous line search (§3.4.2). If, for some reason, the previous line search accepted an unexpectedly large or small step (what this means is encoded in θ_{reset}) the GP scale θ for the next line search is reset to an exponential running average of previous scales (α_{stats} in the pseudo-code). This occurs very rarely (for the default value $\theta_{\text{reset}} = 100$ the reset occurred in 0.02% of all line searches), but is necessary to safeguard against extremely mis-scaled GP's. θ_{reset} therefore is not part of the probabilistic line search model as such, but prevents mis-scaled GPs due to some unlucky observation or sudden extreme change in the learning rate. Figure 8 shows performance of the line search for $\theta_{\text{reset}} = 10, 100, 1000$ and 10 000 showing no significant performance change. We adopted $\theta_{\text{reset}} = 100$ in our implementation since this is the expected and desired multiplicative (inverse) factor to maximally vary the learning rate in one single step.

4.3 Candidate Selection and Learning Rate Traces

In the current implementation of the probabilistic line search, the choice among candidates for evaluation is done by evaluating an acquisition function $u_{\text{EI}}(t_i^{\text{cand}}) \cdot p^{\text{Wolfe}}(t_i^{\text{cand}})$ at every

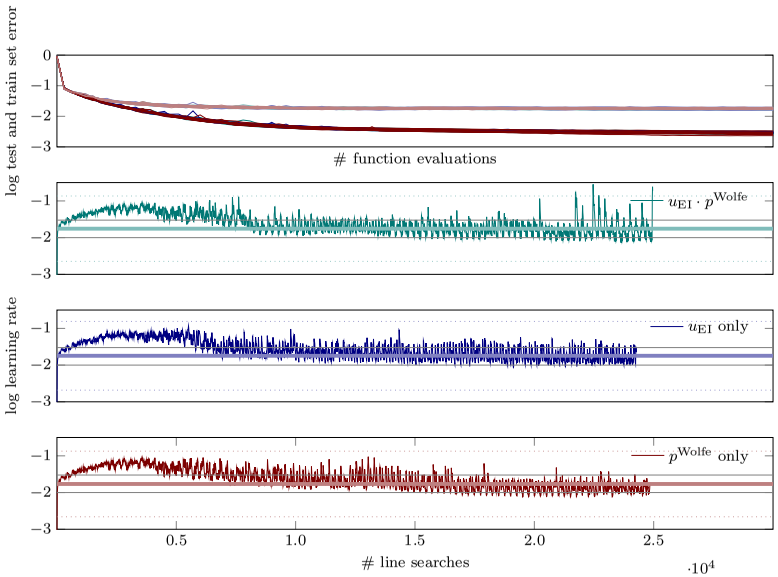


Figure 11: *Different choices of acquisition function.* We compare between using expected improvement u_{EI} (blue), the Wolfe probability p^{Wolfe} (red) and their product $u_{\text{EI}} \cdot p^{\text{Wolfe}}$ (green) which is the default in our code. *Top:* evolution of logarithmic test and train set error. Different lines of the same color correspond to different seeds. *Rows 2-4:* learning rate traces of a single seed (colors same as in top plot). For plotting purposes the curves were smoothed and thinned out. The thick light green, light red and light blue horizontal lines show the mean of the raw (non-smoothed) values of accepted learning rates across the whole optimization process, the dotted lines show ± two standard deviations and the gray solid lines mark a range of well performing constant learning rates.

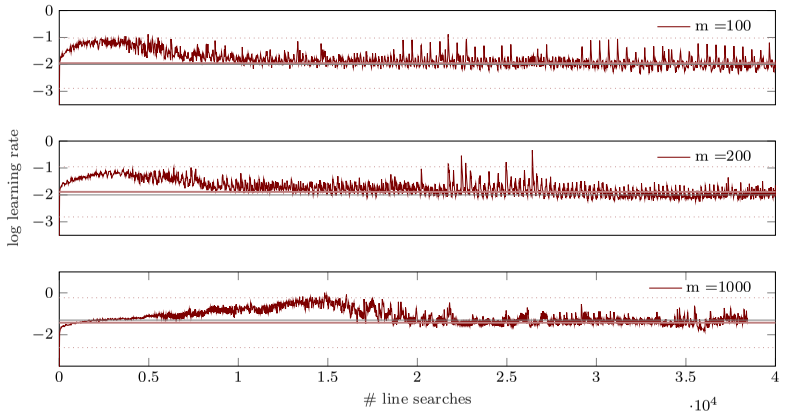


Figure 12: Traces of *accepted logarithmic learning rates*. All runs are performed with default design parameters. Different rows show the same plot for different mini-batch sizes of $m = 100, 200$ and 1000; plots and smoothing as in rows 2-4 of Figure 11 (details in text).

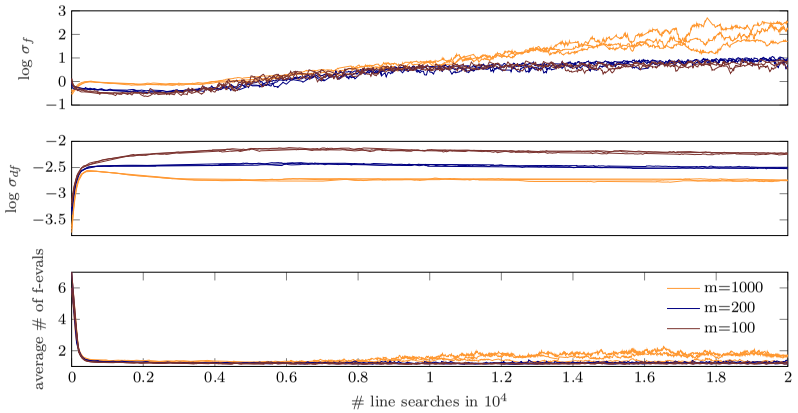


Figure 13: Traces of *logarithmic noise levels* σ_f (top), $\sigma_{f'}$ (middle) and average number of function evaluations per line search (bottom). Setup and smoothing as in Figure 12. Different colors correspond to different mini-batch sizes (see legend). Curves of the same color correspond to different seeds (3 shown).

candidate point t_i^{cand}; then choosing the one with the highest value for evaluation of the objective (§3.2). The Wolfe probability p^{Wolfe} actually encodes precisely what kind of point we want to find and incorporates both (W-I and W-II) conditions about the function value *and* to the gradient (§3.3). However p^{Wolfe} does not have very desirable exploration properties. Since the uncertainty of the GP grows to 'the right' of the last observation, the Wolfe probability quickly drops to a low, approximately constant value there (Figure 4). Also p^{Wolfe} is partially allowing for undesirably short steps (§4.2.2). The expected improvement u_{EI}, on the other hand, is a well studied acquisition function of Bayesian optimization trading off exploration and exploitation. It aims to globally find a point with a function value lower than a current best guess. Though this is a desirable property also for the probabilistic line search, it is lacking the information that we are seeking a point that also fulfills the W-II curvature condition. This is evident in Figure 4 where p^{Wolfe} significantly drops at points where the objective function is already evaluated but u_{EI} does not. In addition, we do not need to explore the positive t space to an extend, the expected improvement suggests, since the aim of a line search is just to find *a* good, acceptable point at positive t and not the globally best one. The product of both acquisition function $u_{\text{EI}} \cdot p^{\text{Wolfe}}$ is thus a trade-off between exploring enough, but still preventing too much exploitation in obviously undesirable regions. In practice, though, we found that all three choices ((i) $u_{\text{EI}} \cdot p^{\text{Wolfe}}$, (ii) u_{EI} only, (iii) p^{Wolfe} only) perform comparable. The following experiments were all performed training N-II on MNIST; only the mini-batch size might vary as indicated.

Figure 11 compares all three choices for mini-batch size $m = 200$ and default design parameters. The top plot shows the evolution of the logarithmic test and train set error (for plot and color description see Figure caption). All test and train set error curves respectively bundle up (only lastly plotted clearly visible). The choice of acquisition function thus does not change the performance here. Rows 2-4 of Figure 11 show learning rate traces of a single seed. All three curves show very similar global behavior. First the learning rate grows, then drops again, and finally settles around the best found constant learning rate. This is intriguing since *on average* a larger learning rate seems to be better at the beginning of the optimization process, then later dropping again to a smaller one. This might also explain why SGD+probLS in the first part of the optimization progress outperforms vanilla SGD (Figure 7). Runs that use just slightly larger constant learning rates than the best performing constant one (above the gray horizontal lines in Figure 11) were failing after a few steps. This shows that there is some non-trivial adaptation going on, not just globally, but locally at every step.

Figure 12 shows traces of accepted learning rates for different mini-batch sizes $m = 100, 200, 1000$. Again the global behavior is qualitatively similar for all three mini-batch sizes on the given architecture. For the largest mini-batch size $m = 1000$ (last row of Figure 12) the probabilistic line search accepts a larger learning rate (on average and in absolute value) than for the smaller mini-batch sizes $m = 100$ and 200, which is in agreement with practical experience and theoretical findings (Hinton (2012, §4 and 7), Goodfellow et al. (2016, §9.1.3), Balles et al. (2016)).

Figure 13 shows traces of the (scaled) noise levels σ_f and $\sigma_{f'}$ and the average number of function evaluations per line search for different noise levels ($m = 100, 200, 1000$; same colors show the same setup but different seeds). The average number of function evaluations rises very slightly to $\approx 1.5 - 2$ for mini-batch size $m = 1000$ towards the end of the optimization

process, in comparison to ≈ 1.5 for $m = 100, 200$. This seems counter intuitive in a way, but since larger mini-batch sizes also observe smaller value and gradients (especially towards the end of the optimization process), the relative noise levels might actually be larger. (Although the curves for varying m are shown versus the same abscissa, the corresponding optimizers might be in different regions of the loss surface, especially $m = 1000$ probably reaches regions of smaller absolute gradients). At the start of the optimization the average number of function evaluations is high, because the initial default learning rate is small (10^{-4}) and the line search extends each step multiple times.

5. Conclusion

The line search paradigm widely accepted in deterministic optimization can be extended to noisy settings. Our design combines existing principles from the noise-free case with ideas from Bayesian optimization, adapted for efficiency. We arrived at a lightweight "black-box" algorithm that exposes no parameters to the user. Empirical evaluations so far show compatibility with the SGD search direction and viability for logistic regression and multi-layer perceptrons. The line search effectively frees users from worries about the choice of a learning rate: Any reasonable initial choice will be quickly adapted and lead to close to optimal performance. Our matlab implementation can be found at `http://tinyurl.com/probLineSearch`.

Acknowledgments

Thanks to Jonas Jaszkowic who prepared the base of the pseudo-code.

Appendix A. – Noise Estimation

Section 3.4.3 introduced the statistical variance estimators

$$\begin{aligned}\Sigma'(x) &= (1-m)^{-1}(\hat{\nabla S}(x) - \nabla\hat{\mathcal{L}}(x)^{\odot 2}) \\ \Sigma(x) &= (1-m)^{-1}(\hat{S}(x) - \hat{\mathcal{L}}(x)^2)\end{aligned} \tag{19}$$

of the function and gradient estimate $\hat{\mathcal{L}}(x)$ and $\nabla\hat{\mathcal{L}}(x)$ at position x. The underlying assumption is that $\hat{\mathcal{L}}(x)$ and $\nabla\hat{\mathcal{L}}(x)$ are distributed according to

$$\begin{bmatrix}\hat{\mathcal{L}}(x) \\ \nabla\hat{\mathcal{L}}(x)\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\hat{\mathcal{L}}(x) \\ \nabla\hat{\mathcal{L}}(x)\end{bmatrix}; \begin{bmatrix}\mathcal{L}(x) \\ \nabla\mathcal{L}(x)\end{bmatrix}, \begin{bmatrix}\Sigma(x) & 0_{D\times 1} \\ 0_{1\times D} & \operatorname{diag}\Sigma'(x)\end{bmatrix}\right) \tag{20}$$

which implies Eq 3

$$\begin{bmatrix}\hat{\mathcal{L}}(x) \\ s(x)'\cdot\nabla\hat{\mathcal{L}}(x)\end{bmatrix} = \begin{bmatrix}y(x) \\ y'(x)\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}f(x) \\ f'(x)\end{bmatrix}, \begin{bmatrix}\sigma_f(x) & 0 \\ 0 & \sigma_{f'}(x)\end{bmatrix}\right). \tag{21}$$

where $s(x)$ is the possibly new search direction at x. This is an approximation since the true covariance matrix is in general not diagonal. A better estimator for the projected gradient noise would be (dropping x from the notation)

$$\begin{aligned}\eta_{f'} &= s^\intercal\left[\frac{1}{m-1}\frac{1}{m}\sum_{k=1}^{m}(\nabla\ell^k - \nabla\hat{\mathcal{L}})(\nabla\ell^k - \nabla\hat{\mathcal{L}})^\intercal\right]s \\ &= \sum_{i,j=1}^{D} s_i s_j \frac{1}{m-1}\frac{1}{m}\sum_{k=1}^{m}\left(\nabla\ell_i^k - \nabla\hat{\mathcal{L}}_i\right)\left(\nabla\ell_j^k - \nabla\hat{\mathcal{L}}_j\right) \\ &= \frac{1}{m-1}\sum_{i,j=1}^{D} s_i s_j\left(\frac{1}{m}\sum_{k=1}^{m}\nabla\ell_i^k\nabla\ell_j^k - \nabla\hat{\mathcal{L}}_i\nabla\hat{\mathcal{L}}_j - \nabla\hat{\mathcal{L}}_j\nabla\hat{\mathcal{L}}_i + \nabla\hat{\mathcal{L}}_i\nabla\hat{\mathcal{L}}_j\right) \\ &= \frac{1}{m-1}\left(\frac{1}{m}\sum_{k=1}^{m}\sum_{i,j=1}^{D} s_i\nabla\ell_i^k s_j\nabla\ell_j^k - \sum_{i,j=1}^{D} s_j\nabla\hat{\mathcal{L}}_j s_i\nabla\hat{\mathcal{L}}_i\right) \\ &= \frac{1}{m-1}\left(\frac{1}{m}\sum_{k=1}^{m}(s'\cdot\nabla\ell^k)^2 - (s'\cdot\nabla\hat{\mathcal{L}})^2\right).\end{aligned} \tag{22}$$

Comparing to $\sigma_{f'}$ yields

$$\begin{aligned}\eta_{f'} &= \frac{1}{m-1}\sum_{i,j=1}^{D} s_i s_j\left(\frac{1}{m}\sum_{k=1}^{m}\nabla\ell_i^k\nabla\ell_j^k - \nabla\hat{\mathcal{L}}_j\nabla\hat{\mathcal{L}}_i\right) \\ &= \frac{1}{m-1}\sum_{i=1}^{D} s_i^2\left(\frac{1}{m}\sum_{k=1}^{m}(\nabla\ell_i^k)^2 - \nabla\hat{\mathcal{L}}_i^2\right) \\ &\quad + \frac{1}{m-1}\sum_{i\neq j=1}^{D} s_i s_j\left(\frac{1}{m}\sum_{k=1}^{m}\nabla\ell_i^k\nabla\ell_j^k - \nabla\hat{\mathcal{L}}_j\nabla\hat{\mathcal{L}}_i\right) \\ \eta_{f'} &= \sigma_{f'} + \frac{1}{m-1}\sum_{i\neq j=1}^{D} s_i s_j\left(\frac{1}{m}\sum_{k=1}^{m}\nabla\ell_i^k\nabla\ell_j^k - \nabla\hat{\mathcal{L}}_j\nabla\hat{\mathcal{L}}_i\right).\end{aligned} \tag{23}$$

From Eq 22 we see that, in order to effectively compute $\eta_{f'}$, we need an efficient way of computing the inner product $(s'\cdot\nabla\ell^k)$ for all k. In addition, we need to know the search direction $s(x)$ of the potential next step (if x was accepted) at the time of computing $\eta_{f'}$. This is possible e.g. for the SGD search direction where $s(x) = -\frac{1}{m}\sum_{k=1}^{m}\nabla\ell^k(x)$ but potentially not possible or practical for arbitrary search directions. For all experiments in this paper we used the approximate variance estimator $\sigma_{f'}$.

The following paragraph is concerned with the independence assumption of gradient and function value y and y' (in contrast to independence among gradient elements). In general y and y' are not independent since the algorithm draws them from the same mini-batch; the likelihood including the correlation factor ρ reads

$$p(y_t, y_t' \mid f) = \mathcal{N}\left(\begin{bmatrix}y_t \\ y_t'\end{bmatrix}; \begin{bmatrix}f(t) \\ f'(t)\end{bmatrix}, \begin{bmatrix}\sigma_f^2 & \rho \\ \rho & \sigma_{f'}^2\end{bmatrix}\right). \tag{24}$$

The noise covariance matrix enters the GP only in the inverse of the sum with the kernel matrix of the observations. We can compute it analytically for one datapoint at position t, since it is only a 2×2 matrix. For $\rho = 0$, define:

$$\begin{aligned}det_{\rho=0} &:= [k_{tt}+\sigma_f^2][{}^\partial k^\partial{}_{tt} + \sigma_{f'}^2] - k^\partial{}_{tt}{}^\partial k_{tt} \\ G_{\rho=0}^{-1} &:= \begin{bmatrix}k_{tt}+\sigma_f^2 & k^\partial{}_{tt} \\ {}^\partial k_{tt} & {}^\partial k^\partial{}_{tt}+\sigma_{f'}^2\end{bmatrix}^{-1} = \frac{1}{det_{\rho=0}}\begin{bmatrix}{}^\partial k^\partial{}_{tt}+\sigma_{f'}^2 & -k^\partial{}_{tt} \\ -{}^\partial k_{tt} & k_{tt}+\sigma_f^2\end{bmatrix}.\end{aligned} \tag{25}$$

For $\rho \neq 0$ we thus get:

$$\begin{aligned}det_{\rho\neq 0} &:= [k_{tt}+\sigma_f^2][{}^\partial k^\partial{}_{tt}+\sigma_{f'}^2] - [k^\partial{}_{tt}+\rho][{}^\partial k_{tt}+\rho] \\ &= det_{\rho=0} - \rho({}^\partial k_{tt} + k^\partial{}_{tt}) - \rho^2 \\ G_{\rho\neq 0}^{-1} &:= \begin{bmatrix}k_{tt}+\sigma_f^2 & k^\partial{}_{tt}+\rho \\ {}^\partial k_{tt}+\rho & {}^\partial k^\partial{}_{tt}+\sigma_{f'}^2\end{bmatrix}^{-1} = \frac{1}{det_{\rho\neq 0}}\begin{bmatrix}{}^\partial k^\partial{}_{tt}+\sigma_{f'}^2 & -(k^\partial{}_{tt}+\rho) \\ -({}^\partial k_{tt}+\rho) & k_{tt}+\sigma_f^2\end{bmatrix} \\ &= \frac{det_{\rho=0}}{det_{\rho\neq 0}}G_{\rho=0}^{-1} - \frac{\rho}{det_{\rho\neq 0}}\begin{bmatrix}0 & 1 \\ 1 & 0\end{bmatrix}\end{aligned} \tag{26}$$

The fraction ${}^{det_{\rho=0}}/_{det_{\rho\neq 0}}$ in the first term of the last row, is a positive scalar that scales all element of $G_{\rho=0}^{-1}$ equally (since $G_{\rho=0}$ and $G_{\rho\neq 0}$ are positive definite matrices, we know that $det_{\rho=0} > 0$, $det_{\rho\neq 0} > 0$). If $|\rho|$ is small in comparison to the determinant $det_{\rho=0}$, then $det_{\rho\neq 0} \approx det_{\rho=0}$ and the scaling factor is approximately one. The second term corrects off-diagonal elements in $G_{\rho\neq 0}$ and is proportional to ρ; if $|\rho| \ll det_{\rho=0}$ this term is small as well.

In might be possible to estimate ρ as well from the mini-batch in a similar style to the estimation of σ_f and $\sigma_{f'}$; it is not clear, however, if the additional computational cost would justify the improvements in the GP-inference.

Appendix B. – Noise Sensitivity

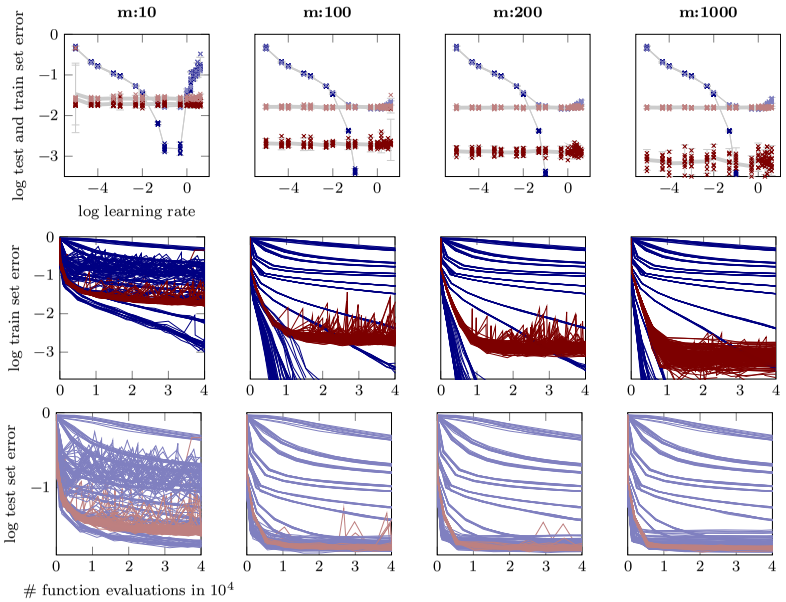


Figure 14: Performance of N-I on MNIST for *varying mini-batch sizes*; plots and colors same as in Figure 7 (middle plots cropped for readability).

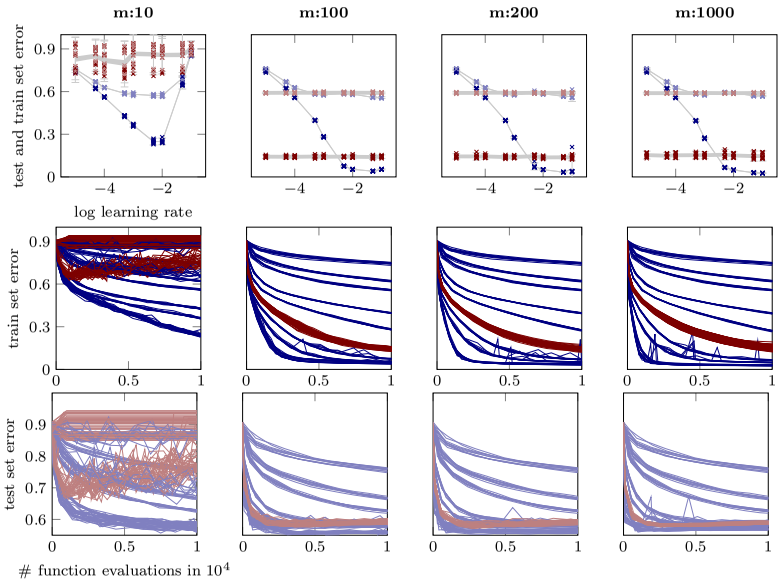


Figure 15: Performance of N-II on CIFAR-10 for *varying mini-batch sizes*; plots and colors same as in Figure 7, except the scaling of the y-axis which is not logarithmic here.

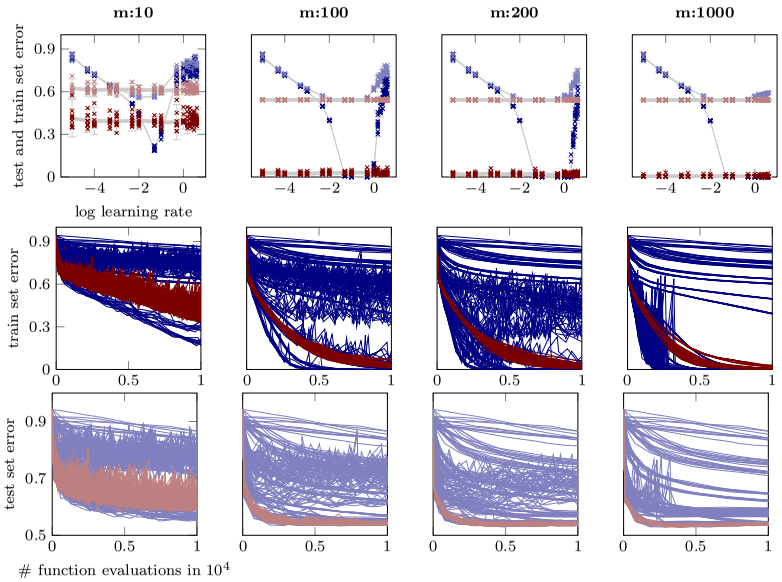


Figure 16: Performance of N-I on CIFAR-10 for *varying mini-batch sizes*; plots and colors same as in Figure 15.

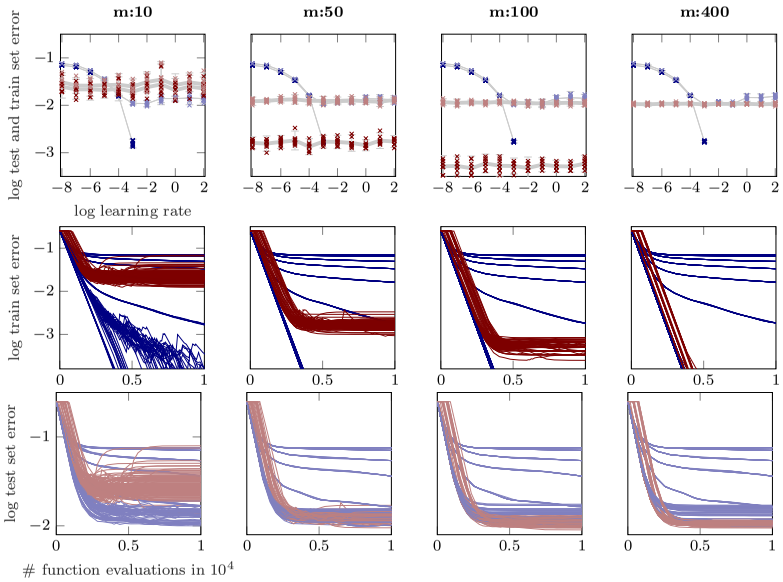


Figure 17: Performance of N-III on GISETTE for *varying mini-batch sizes*; plots and colors same as in Figure 15 (middle plots cropped for readability).

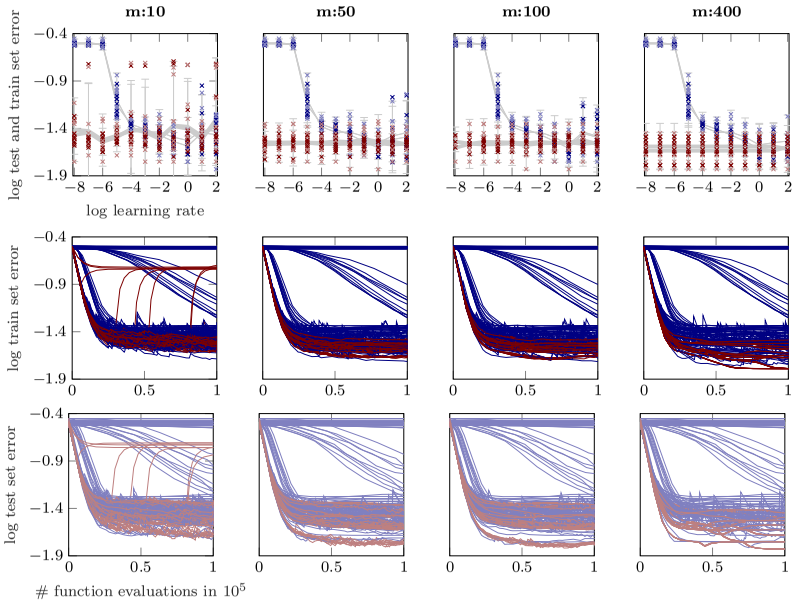


Figure 18: Performance of N-III on WDBC for *varying mini-batch sizes*; plots, colors, and description same as in Figure 17. *Remark:* since the training set is of size 400, the most right column ($m = 400$) in fact runs full-batch gradient descent; this is not a problem, since the probabilistic line search can handle noise free observations as well.

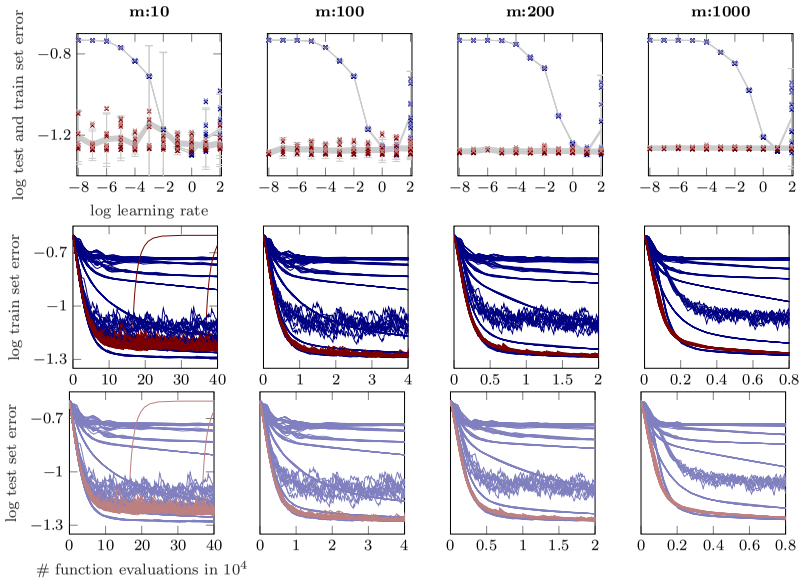


Figure 19: Performance of N-III on EPSILON for *varying mini-batch sizes*; plots, colors, and description same as in Figure 17. EPSILON is the largest dataset that was used in the experiments (400k samples); this did not seem to impair the performance of the line search or variance estimator.

Appendix C. – Parameter Sensitivity

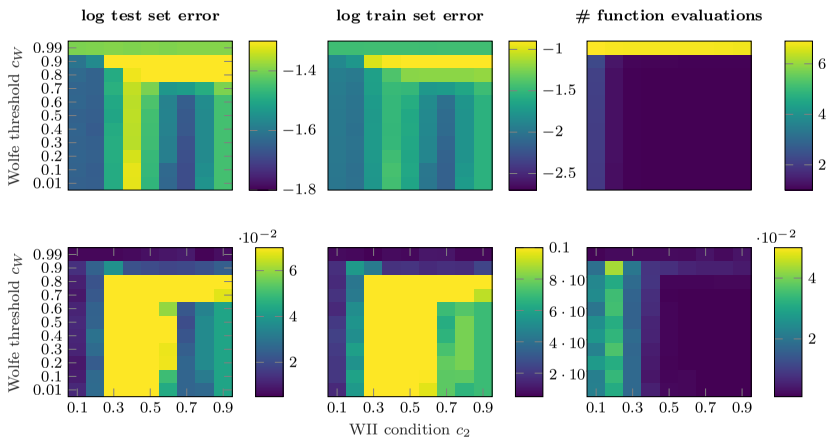


Figure 20: *Sensitivity to varying hyper-parameters* c_2, and c_W and *fixed* $\alpha_{\text{ext}} = 1.0$. Experimental setup as in Figures 9 and 10. *Top row* from left to right: logarithmic test set error, train set error, and average number of function evaluations per line search averaged over 10 different initializations. *Bottom row:* corresponding relative standard deviations. In all plots darker colors are better. For extrapolation parameters $\alpha_{\text{ext}} > 1$ (see Figures 21, 22, 23, and 24) the different parameter combinations all result in similar good performance. Only at extreme choices, for example $\alpha_{\text{ext}} = 1.0$ (this figure), which amounts to no extrapolation at all in between successive line searches, the line search performs poorer. At the extreme value of $c_W = 0.99$, which amounts to imposing nearly absolute certainty about the Wolfe conditions, the line search becomes less efficient. In Figure 23 the default values adopted in the line search implementation ($c_W = 0.3$, $c_2 = 0.5$, and $\alpha_{\text{ext}} = 1.3$) are indicated as red dots.

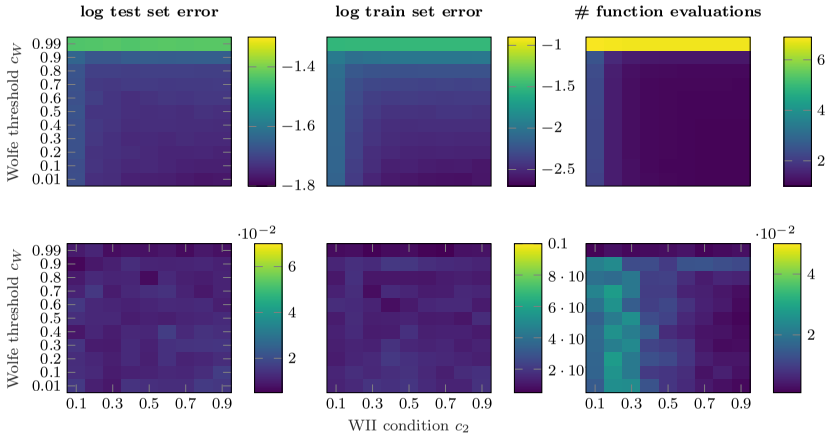


Figure 21: Same as Figure 20 but for *fixed* $\alpha_{\text{ext}} = 1.1$

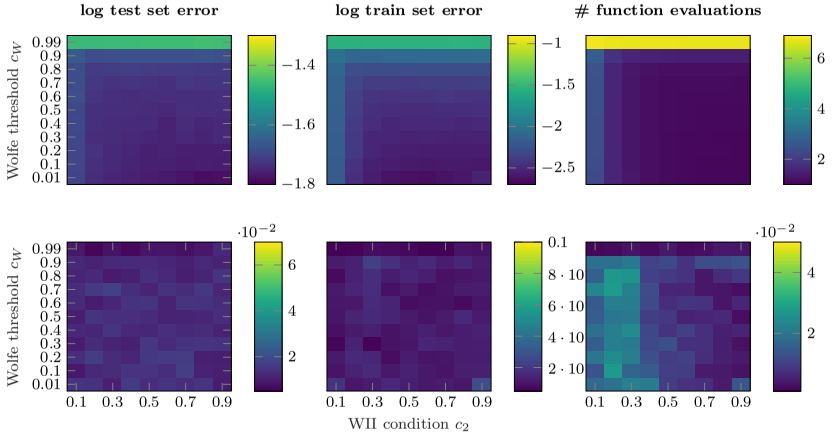


Figure 22: Same as Figure 20 but for *fixed* $\alpha_{\text{ext}} = 1.2$

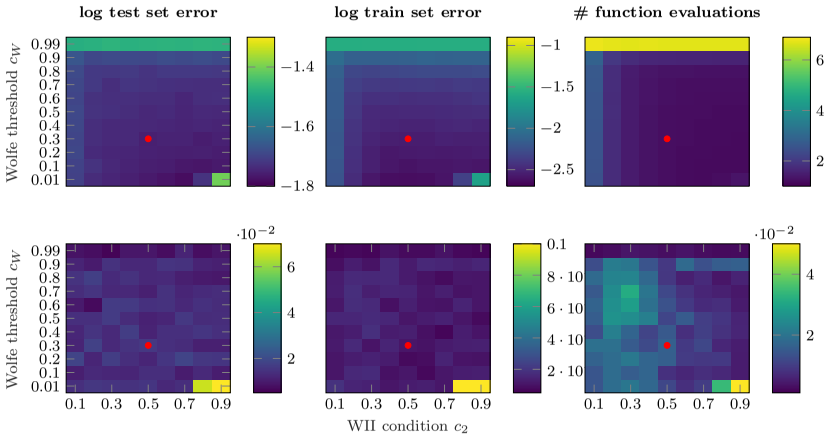


Figure 23: Same as Figure 20 but for *fixed* $\alpha_{\text{ext}} = 1.3$. The default values adopted in the line search implementation ($c_W = 0.3$, $c_2 = 0.5$, and $\alpha_{\text{ext}} = 1.3$) are indicated as red dots.

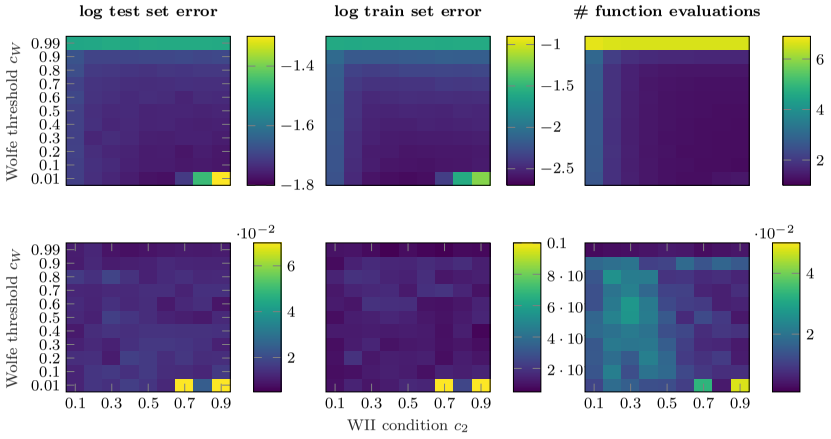


Figure 24: Same as Figure 20 but for *fixed* $\alpha_{\text{ext}} = 1.4$

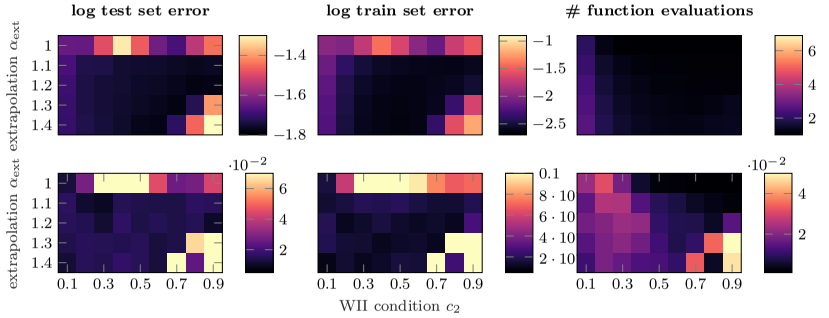


Figure 25: *Sensitivity to varying hyper-parameters* c_2, and α_{ext} and *fixed* $c_W = 0.01$. Experimental setup as in Figures 9 and 10 and plots like in Figure 20. In all plots darker colors are better. All choices of c_W result in good performance though very tight choices of $c_W = 0.99$ (Figure 35), which amounts to imposing nearly absolute certainty about the Wolfe conditions, are less efficient. As described in Figure 20, for a dropped extrapolation factor ($\alpha_{\text{ext}} \to 1$) in combination with a loose curvature condition (large c_2) the line searches performs poorer (top row, right half of columns in Figures 25–34). In Figure 28 the default values adopted in the line search implementation ($c_W = 0.3$, $c_2 = 0.5$, and $\alpha_{\text{ext}} = 1.3$) are indicated as red dots.

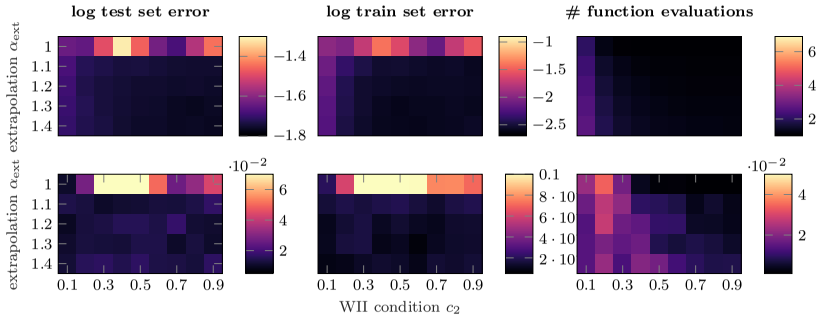


Figure 26: Same as Figure 25 but for *fixed* $c_W = 0.10$.

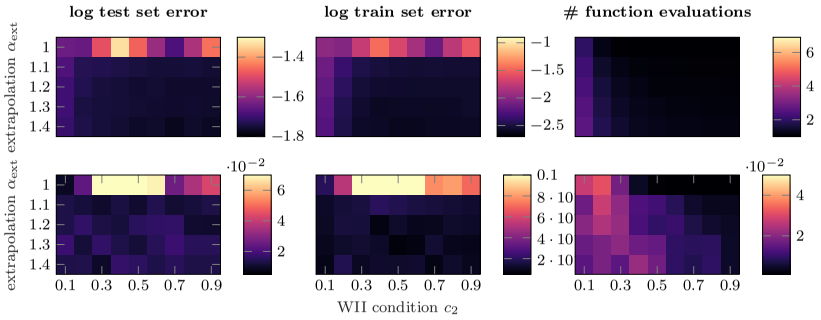


Figure 27: Same as Figure 25 but for *fixed* $c_W = 0.20$.

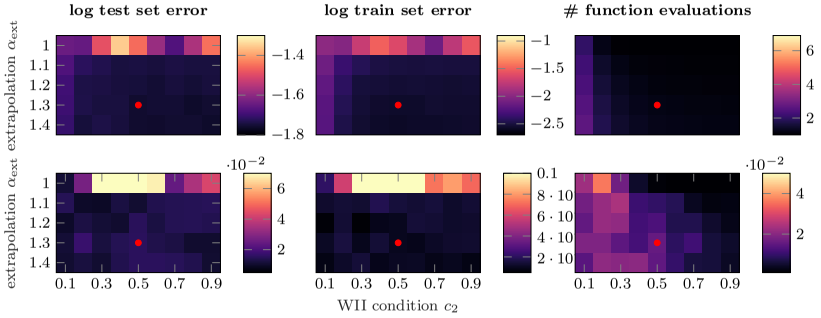


Figure 28: Same as Figure 25 but for *fixed* $c_W = 0.30$. The default values as red dots.

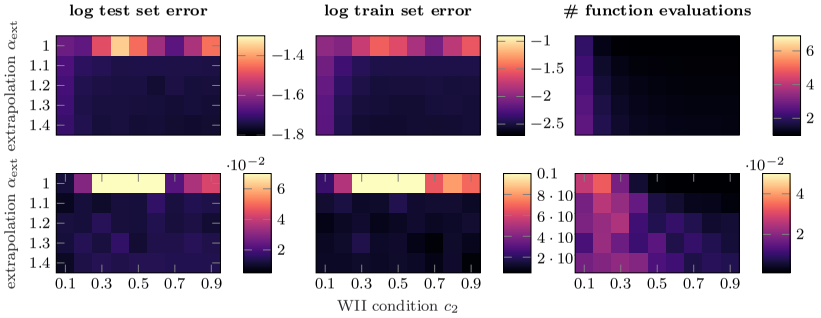


Figure 29: Same as Figure 25 but for *fixed* $c_W = 0.40$.

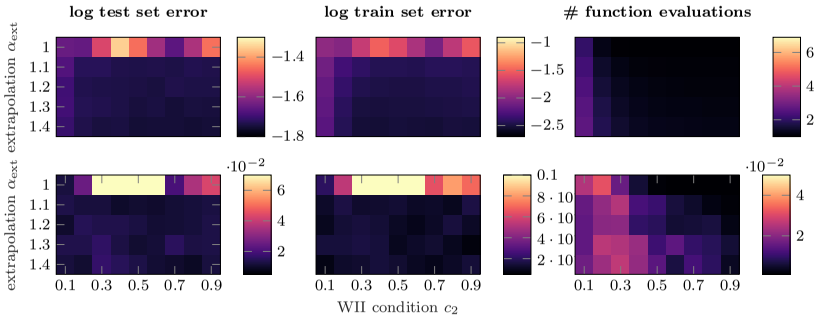


Figure 30: Same as Figure 25 but for *fixed* $c_W = 0.50$.

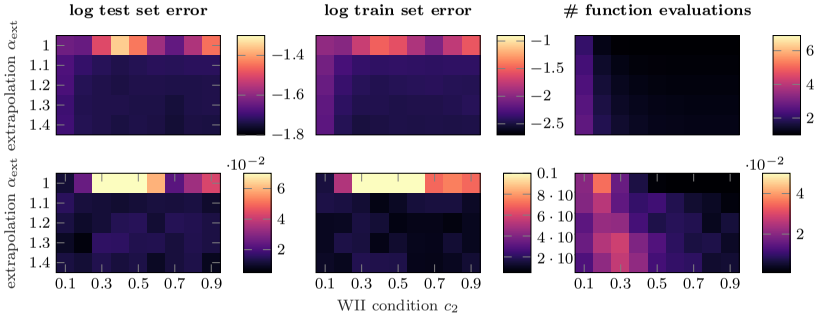


Figure 31: Same as Figure 25 but for *fixed* $c_W = 0.60$.

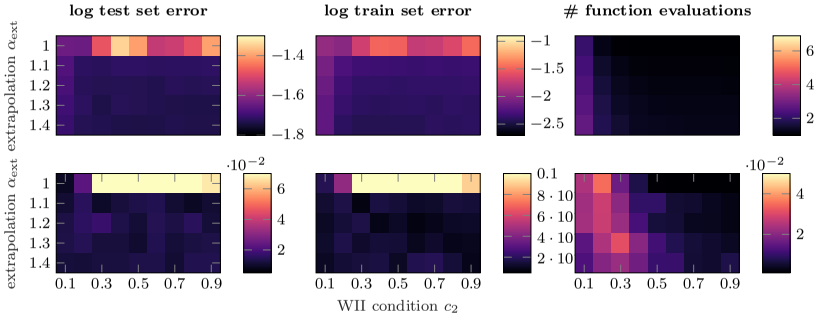


Figure 32: Same as Figure 25 but for *fixed* $c_W = 0.70$.

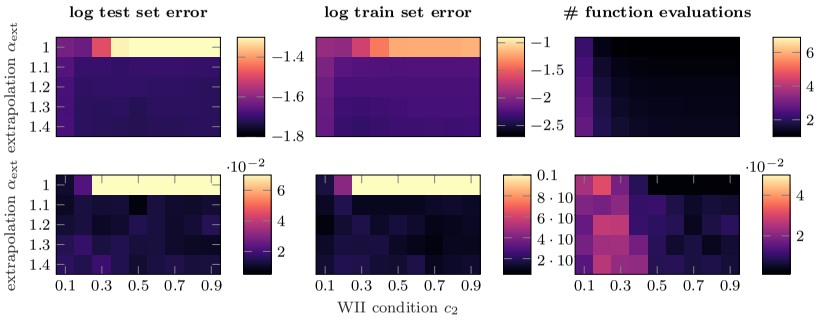


Figure 33: Same as Figure 25 but for *fixed* $c_W = 0.80$.

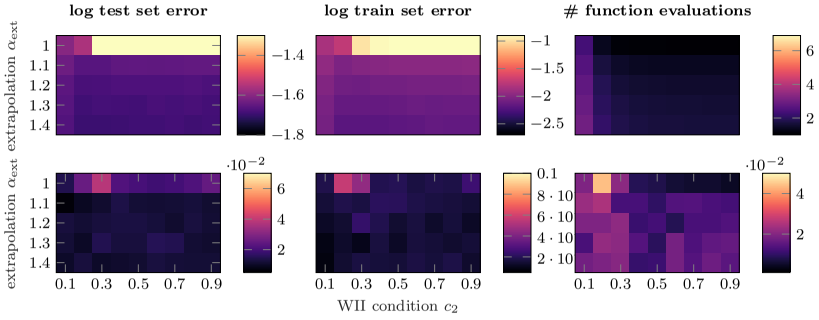


Figure 34: Same as Figure 25 but for *fixed* $c_W = 0.90$.

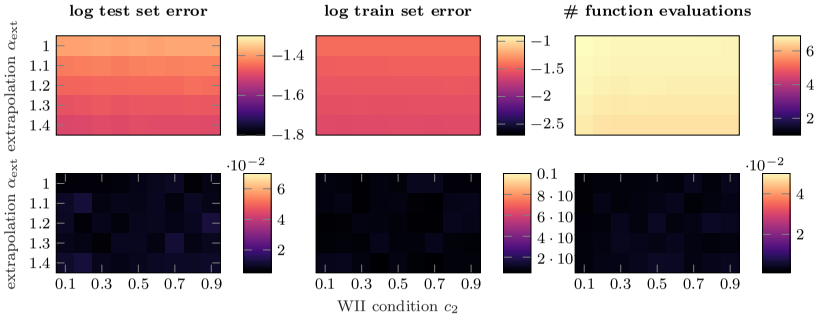


Figure 35: Same as Figure 25 but for *fixed* $c_W = 0.99$.

Appendix D. – Pseudocode

Algorithm 1 of Section 3 roughly sketches the structure of the probabilistic line search and its main ingredients. This section provides a detailed pseudocode which can be used for re-implementation. It is based on the code which was used for the experiments in this paper. A matlab implementation including a minimal example can be found at `http://tinyurl.com/probLineSearch`. The actual line search routine is called probLineSearch below and is quite short. Most of the pseudocode is occupied with comments, helper function that define the kernel of the GP, the GP-update or Gauss cdf and pdf which we printed here for completeness such that a detailed re-implementation is possible. For better readability of the pseudocode we use the following color coding:

- blue: comments
- green: variables of the integrated Wiener process.
- red: most recently evaluated observation (noisy loss and gradient). If the line search terminates, these will be returned as 'accepted'.
- orange: inputs from the main solver procedure and unchanged during each line search.

Notation and operators:

operator or function	definition
$A \odot B$	elementwise multiplication
$A \oslash B$	elementwise division
$A^{\odot b}$	elementwise power of b
A'	transpose of A
$A \cdot B$	scalar-scalar, scalar-matrix or matrix-matrix multiplication
A/B	right matrix division, the same as $A \cdot B^{-1}$
$A\backslash B$	left matrix division, the same as $A^{-1} \cdot B$
$\text{sign}(a)$	sign of scalar a
$\text{erf}(x)$	error function $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$
$\max(A)$	maximum element in A
$\min(A)$	minimum element in A
$\lvert a \rvert$	absolute value of scalar a
$A < B$	elementwise 'less' comparison
$A \leq B$	elementwise 'less-or-equal' comparison
$A > B$	elementwise 'greater' comparison
$A \geq B$	elementwise 'greater-or-equal' comparison
$[a, b, c] \leftarrow f(x)$	function f called at x returns the values a, b and c

For better readability and to avoid confusion with transposes, we denote derivatives for example as dy and df (instead of y' and f' as in the main text).

function SGDSolver(f)
  ▶ f – function handle to objective. Usage: $[y, dy, \Sigma_f, \Sigma_{df}] \leftarrow f(x)$.

  ▶ initial weights
  $x \leftarrow$initial weights

  ▶ initial step size (rather small to avoid over-shooting in very first step)
  $\alpha \leftarrow e.g. \approx 10^{-4}$
  $\alpha_{\text{stats}} \leftarrow \alpha$

  ▶ initial function evaluation at x
  $[y, dy, \Sigma_f, \Sigma_{df}] \leftarrow f(x)$

  ▶ initial search direction
  $d \leftarrow -dy$

  ▶ loop over line searches
  while budget not used do

    ▶ line search finds step size
    $[\alpha, \alpha_{\text{stats}}, x, y, dy, \Sigma_f, \Sigma_{df}] \leftarrow$probLineSearch$(x, d, y, dy, \Sigma_f, \Sigma_{df}, \alpha, \alpha_{\text{stats}}, f)$

    ▶ set new search direction
    $d \leftarrow -dy$
  end while

  return x
end function

function probLineSearch$(x_0, d, f_0, df_0, \Sigma_{f_0}, \Sigma_{df_0}, \alpha_0, \alpha_{stats}, f)$
  ▶ x_0 – current weights $[D \times 1]$
  ▶ f – function handle to objective.
  ▶
  ▶ d – search direction $[D \times 1]$ (does not need to be normalized)
  ▶ f_0 – function value at start, $f_0 = f(x_0)$
  ▶ df_0 – gradient at start, $df_0 = \nabla f(x_0)$ $[D \times 1]$
  ▶ Σ_{f_0} – sample variance of f_0
  ▶ Σ_{df_0} – sample variances of df_0, $[D \times 1]$
  ▶ α_0 – initial step size

  ▶ set maximum # of f evaluations per line search
  $L \leftarrow 6$

  ▶ scaling and noise level of GP
  $\beta \leftarrow |d' \cdot \Sigma_{df_0}|$ ▷ scale factor
  $\sigma_f \leftarrow \sqrt{\Sigma_{f_0}/(\alpha_0 \cdot \beta)}$ ▷ scaled sample variance of f_0
  $\sigma_{df} \leftarrow \sqrt{((d^{\odot 2})' \cdot \Sigma_{df_0})/\beta}$ ▷ scaled and projected sample variances of df_0

  ▶ initialize counter and non-fixed parameters
  $N \leftarrow 1$ ▷ size of GP$= 2 \cdot N$
  $t_{\text{ext}} \leftarrow 1$ ▷ scaled step size for extrapolation
  $tt \leftarrow 1$ ▷ scaled position of first function evaluation

  ▶ initialize storage for GP. Dynamic arrays of maximum size $[L + 1 \times 1]$
  $T \leftarrow [0]$ ▷ scaled positions along search direction
  $Y \leftarrow [0]$ ▷ scaled function values at T
  $dY \leftarrow [(df_0' \cdot d)/\beta]$ ▷ scaled projected gradients at T

  ▶ initialize GP with observation at start
  $[G, A] \leftarrow$updateGP$(T, Y, dY, N, \sigma_f, \sigma_{df})$

  ▶ loop until budged is used or acceptable point is found
  for N **from** 2 **to** $L + 1$ **do**

    ▶ evaluate objective function at tt.
    $[y, dy, \Sigma_f, \Sigma_{df}, T, Y, dY, N] \leftarrow$evaluateObjective$(tt, x_0, \alpha_0, d, T, Y, dY, N, \beta, f)$

    ▶ update the GP which is now of size $2 \cdot N$.
    $[G, A] \leftarrow$updateGP$(T, Y, dY, N, \sigma_f, \sigma_{df})$

    ▶ initialize storage for candidates. Dynamic arrays of maximum size $[N \times 1]$.
    $T_{\text{cand}} \leftarrow [\,]$ ▷ scaled position of candidates
    $M_{\text{cand}} \leftarrow [\,]$ ▷ GP mean of candidates
    $S_{\text{cand}} \leftarrow [\,]$ ▷ GP standard deviation of candidates

    ▶ current point is above the Wolfe threshold? If yes, accept point and return.
    if probWolfe(tt, T, A, G) **then**
      $output \leftarrow$rescaleOutput$(x_0, f_0, \alpha_0, d, tt, y, dy, \Sigma_f, \Sigma_{df}, \beta)$
      return $output$
    end if

    ▶ Wolfe conditions not satisfied at this point.
    ▶ find suitable candidates for next evaluation.

    ▶ GP mean of function values and corresponding gradients at points in T.
    $M \leftarrow$ map function m$(_, T, A)$ over T
    $dM \leftarrow$ map function d1m$(_, T, A)$ over T

```
    ▶ candidates 1: local minima of GP mean.
    T_sorted ⇐ sort T in ascending order
    T_Wolfes ⇐[ ]                                   ▷ prepare list of acceptable points

    ▶ iterate through all N − 1 cells, compute locations of local minima.
    for n from 1 to N − 1 do
        T_n ⇐ value of T_sorted at n
        T_{n+1} ⇐ value of T_sorted at n + 1

        ▶ add a little offset for numerical stability
        t_rep ⇐ T_n + 10^{−6} · (T_{n+1} − T_n)

        ▶ compute location of cubic minimum in n^th cell
        t_cubMin ⇐cubicMinimum(t_rep, T, A, N)

        ▶ add point to candidate list if minimum lies in between T_n and T_{n+1}
        if t_cubMin > T_n and t_cubMin < T_{n+1} then
            if (not isnanOrIsinf(t_cubMin)) and (t_cubMin > 0) then
                T_cand ⇐ append t_cubMin
                M_cand ⇐ append M(t_cubMin, T, A)
                S_cand ⇐ append V(t_cubMin, T, G)
            end if
        else

            ▶ most likely uphill? If yes, break.
            if n = 1 and d1m(0, T, A) > 0 then
                r ⇐ 0.01
                tt ⇐ r · (T_n + T_{n+1})

                ▶ evaluate objective function at tt and return.
                [y, dy, Σ_f, Σ_df, T, Y, dY, N] ⇐evaluateObjective(tt, x_0, α_0, d, T, Y, dY, N, β, f)

                output ⇐rescaleOutput(x_0, f_0, α_0, d, tt, y, dy, Σ_f, Σ_df, β)
                return output
            end if
        end if

        ▶ check whether there is an acceptable point among the old evaluations
        if n > 1 and probWolfe(T_n, T, A, G) then
            T_Wolfes ⇐ append T_n
        end if
    end for

    ▶ check if acceptable points exists and return
    if T_Wolfes is not empty then
```



```
        ▶ if last evaluated point is among acceptable ones, return it.
        if tt in T_Wolfes then
            output ⇐rescaleOutput(x_0, f_0, α_0, d, tt, y, dy, Σ_f, Σ_df, β)
            return output
        end if

        ▶ else, choose the one with the lowest GP mean and re-evaluate its gradient.
        M_Wolfes ⇐ map M(_, T, A) over T_Wolfes
        tt ⇐ value of T_Wolfes at index of min(M_Wolfes)

        ▶ evaluate objective function at tt.
        [y, dy, Σ_f, Σ_df, T, Y, dY, N] ⇐evaluateObjective(tt, x_0, α_0, d, T, Y, dY, N, β, f)

        output ⇐rescaleOutput(x_0, f_0, α_0, d, tt, y, dy, Σ_f, Σ_df, β)
        return output
    end if

    ▶ candidates 2: one extrapolation step
    T_cand ⇐ append max(T) + t_ext
    M_cand ⇐ append M(max(T) + t_ext, T, A)
    S_cand ⇐ append V(max(T) + t_ext, T, G)^{1/2}

    ▶ find minimal mean among M.
    μ_EI ⇐ minimal value of M

    ▶ compute expected improvement and Wolfe probabilities at T_cand
    EI_cand ⇐expectedImprovement(M_cand, S_cand, μ_EI)
    PW_cand ⇐ map probWolfe(_, T, A, G) over T_cand

    ▶ choose point among candidates that maximizes EI_cand ∧ PW_cand
    i_bestCand ⇐ index of max(EI_cand ⊙ PW_cand)
    tt_bestCand ⇐ value of T_cand at i_bestCand

    ▶ extend extrapolation step if necessary
    if tt_bestCand is equal to tt + t_ext then
        t_ext ⇐ 2 · t_ext
    end if

    ▶ set location for next evaluation
    tt ⇐ tt_bestCand
end for

▶ limit reached: evaluate a final time and return the point with lowest GP mean
[y, dy, Σ_f, Σ_df, T, Y, dY, N] ⇐evaluateObjective(tt, x_0, α_0, d, T, Y, dY, N, β, f)
```

▶ update the GP which is now of size $2 \cdot N$.
$[G, A] \leftarrow \text{updateGP}(T, Y, dY, N, \sigma_f, \sigma_{df})$

▶ check last point for acceptance
if probWolfe(tt, T, A, G) **then**
  $output \leftarrow \text{rescaleOutput}(x_0, f_0, \alpha_0, d, tt, y, dy, \Sigma_f, \Sigma_{df}, \beta)$
  return *output*
end if

▶ at the end of budget return point with the lowest GP mean
▶ compute GP means at T
$M \leftarrow$ map $\text{M}(\_, T, A)$ over T
$i_{\text{lowest}} \leftarrow$ index of minimal value in M
$t_{\text{lowest}} \leftarrow$ value of T at i_{lowest}

▶ if t_{lowest} is the last evaluated point, return
if t_{lowest} **is equal to** tt **then**
  $output \leftarrow \text{rescaleOutput}(x_0, f_0, \alpha_0, d, tt, y, dy, \Sigma_f, \Sigma_{df}, \beta)$
  return *output*
end if

▶ else, re-evaluate its gradient and return
$tt \leftarrow$ value of t_{lowest}

▶ evaluate objective function at tt.
$[y, dy, \Sigma_f, \Sigma_{df}, T, Y, dY, N] \leftarrow \text{evaluateObjective}(tt, x_0, \alpha_0, d, T, Y, dY, N, \beta, f)$

$output \leftarrow \text{rescaleOutput}(x_0, f_0, \alpha_0, d, tt, y, dy, \Sigma_f, \Sigma_{df}, \beta)$
return *output*
end function

function rescaleOutput$(x_0, f_0, \alpha_0, d, tt, y, dy, \Sigma_f, \Sigma_{df}, \beta, \alpha_{\text{stats}})$
  ▶ design parameters
  $\alpha_{\text{ext}} \leftarrow 1.3$ ▷ extrapolation parameter
  $\theta_{\text{reset}} \leftarrow 100$ ▷ reset threshold for GP scale

  ▶ rescale accepted step size
  $\alpha_{\text{acc}} \leftarrow tt \cdot \alpha_0$

  ▶ update weights
  $x_{\text{acc}} \leftarrow x_0 + \alpha_{\text{acc}} \cdot d$

  ▶ rescale accepted function value
  $f_{\text{acc}} \leftarrow y \cdot (\alpha_0 \cdot \beta) + f_0$

  ▶ accepted gradient
  $df_{\text{acc}} \leftarrow dy$

  ▶ sample variance of f_{acc}
  $\Sigma_{f_{\text{acc}}} \leftarrow \Sigma_f$

  ▶ sample variances of df_{acc}
  $\Sigma_{df_{\text{acc}}} \leftarrow \Sigma_{df}$

  ▶ update exponential running average of scalings
  $\gamma \leftarrow 0.95$
  $\alpha_{\text{stats}} \leftarrow \gamma \cdot \alpha_{\text{stats}} + (1 - \gamma) \cdot \alpha_{\text{acc}}$

  ▶ next initial step size
  $\alpha_{\text{next}} \leftarrow \alpha_{\text{acc}} \cdot \alpha_{\text{ext}}$

  ▶ if new GP scaling is drastically different than previous ones reset it.
  if $(\alpha_{\text{next}} < \alpha_{\text{stats}}/\theta_{\text{reset}})$ **or** $(\alpha_{\text{next}} > \alpha_{\text{stats}} \cdot \theta_{\text{reset}})$ **then**
    $\alpha_{\text{next}} \leftarrow \alpha_{\text{stats}}$
  end if

  ▶ compressed output for readability of pseudocode
  $output \leftarrow [\alpha_{\text{next}}, \alpha_{\text{stats}}, x_{\text{acc}}, f_{\text{acc}}, df_{\text{acc}}, \Sigma_{f_{\text{acc}}}, \Sigma_{df_{\text{acc}}}]$

  return *output*
end function

function evaluateObjective$(tt, x_0, \alpha_0, d, T, Y, dY, N, \beta, f)$
  ▶ evaluate objective function at tt
  $[y, dy, \Sigma_f, \Sigma_{df}] \leftarrow f(x_0 + tt \cdot \alpha_0 \cdot d)$

  ▶ scale output
  $y \leftarrow (y - f_0)/(\alpha_0 \cdot \beta)$
  $dy \leftarrow (dy' \cdot d)/\beta$

  ▶ storage
  $T \leftarrow$ append tt
  $Y \leftarrow$ append y
  $dY \leftarrow$ append dy
  $N \leftarrow N + 1$

  return $[y, dy, \Sigma_f, \Sigma_{df}, T, Y, dY, N]$
end function

function cubicMinimum(t, T, A, N)
  ▶ compute necessary derivatives of GP mean at t
  $d1m_t \leftarrow$ D1M(t, T, A)
  $d2m_t \leftarrow$ D2M(t, T, A)
  $d3m_t \leftarrow$ D3M(t, T, A, N)
  $a \leftarrow 0.5 \cdot d3m_t$
  $b \leftarrow d2m_t - t \cdot d3m_t$
  $c \leftarrow d1m_t - d2m_t \cdot t + 0.5 \cdot d3m_t \cdot t^2$

  ▶ third derivative is almost zero → essentially a quadratic, single extremum
  if $|d3m_t| < 1^{-9}$ **then**
    $t_{\text{cubMin}} \leftarrow -(d1m_t - t \cdot d2m_t)/d2m_t$
    return t_{cubMin}
  end if

  ▶ roots are complex, no extremum
  $\lambda \leftarrow b^2 - 4 \cdot a \cdot c$
  if $\lambda < 0$ **then**
    $t_{\text{cubMin}} \leftarrow +\infty$
    return t_{cubMin}
  end if

  ▶ compute the two possible roots
  $LR \leftarrow (-b - \text{sign}(a) \cdot \sqrt{\lambda})/(2 \cdot a)$ ▷ left root
  $RR \leftarrow (-b + \text{sign}(a) \cdot \sqrt{\lambda})/(2 \cdot a)$ ▷ right root

  ▶ calculate the two values of the cubic at those points (up to a constant)
  $dt_L \leftarrow LR - t$ ▷ distance to left root
  $dt_R \leftarrow RR - t$ ▷ distance to right root
  $CV_L \leftarrow d1m_t \cdot dt_L + 0.5 \cdot d2m_t \cdot dt_L^2 + (d3m_t \cdot dt_L^3)/6$ ▷ left cubic value
  $CV_R \leftarrow d1m_t \cdot dt_R + 0.5 \cdot d2m_t \cdot dt_R^2 + (d3m_t \cdot dt_R^3)/6$ ▷ right cubic value

  ▶ find the minimum and return it.
  if $CV_L < CV_R$ **then**
    $t_{\text{cubMin}} \leftarrow LR$
  else
    $t_{\text{cubMin}} \leftarrow RR$
  end if

  return t_{cubMin}

end function

function updateGP($T, Y, dY, N, \sigma_f, \sigma_{df}$)
  ▶ initialize kernel matrices
  $k_{TT} \leftarrow [N \times N]$ matrix with zeros ▷ covariance of function values
  $kd_{TT} \leftarrow [N \times N]$ matrix with zeros ▷ covariance of function values and gradients
  $dkd_{TT} \leftarrow [N \times N]$ matrix with zeros ▷ covariance of gradients

  ▶ fill kernel matrices
  for $i = 1$ to N **do**
    for $j = 1$ to N **do**
      $k_{TT}(i,j) \leftarrow$ K$(T(i), T(j))$
      $kd_{TT}(i,j) \leftarrow$ KD$(T(i), T(j))$
      $dkd_{TT}(i,j) \leftarrow$ DKD$(T(i), T(j))$
    end for
  end for

  ▶ build diagonal covariance matrix of Gaussian likelihood $[2N \times 2N]$.
  $\Lambda \leftarrow \begin{bmatrix} \text{diag}({\sigma_f}^2)_{N\times N} & 0_{N\times N} \\ 0_{N\times N} & \text{diag}({\sigma_{df}}^2)_{N\times N} \end{bmatrix}$

  $G \leftarrow \begin{pmatrix} k_{TT} & kd_{TT} \\ kd'_{TT} & dkd_{TT} \end{pmatrix} + \Lambda$ ▷ $[2N \times 2N]$ matrix

  ▶ residual between observed and predicted data
  $\Delta \leftarrow \begin{pmatrix} Y \\ dY \end{pmatrix}$ ▷ $[2N \times 1]$ vector

  ▶ compute weighted observations A.
  $A \leftarrow G \backslash \Delta$ ▷ $[2N \times 1]$ vector

  return $[G, A]$

end function

function $\textsc{m}(t, T, A)$
  ▶ posterior mean at t
  return $[\textsc{k}(t, T'),\ \textsc{kd}(t, T')] \cdot A$
end function

function $\textsc{d1m}(t, T, A)$
  ▶ first derivative of mean at t
  return $[\textsc{dk}(t, T'),\ \textsc{dkd}(t, T')] \cdot A$
end function

function $\textsc{d2m}(t, T, A)$
  ▶ second derivative of mean at t
  return $[\textsc{ddk}(t, T'),\ \textsc{ddkd}(t, T')] \cdot A$
end function

function $\textsc{d3m}(t, T, A, N)$
  ▶ third derivative of mean at t
  return $[\textsc{dddk}(t, T'),\ \textsc{zeros}(1, \mathrm{N})] \cdot A$
end function

function $\textsc{v}(t, T, G)$
  ▶ posterior variance of function values at t
  return $\textsc{k}(t, t) - [\textsc{k}((t, T'),\ \textsc{kd}(t, T')] \cdot (G \backslash [\textsc{k}(t, T'),\ \textsc{kd}(t, T')]')$
end function

function $\textsc{vd}(t, T, G)$
  ▶ posterior variance of function values and derivatives at t
  return $\textsc{kd}(t, t) - [\textsc{k}(t, T'),\ \textsc{kd}(t, T')] \cdot (G \backslash [\textsc{dk}(t, T'),\ \textsc{dkd}(t, T')]')$
end function

function $\textsc{dvd}(t, T, G)$
  ▶ posterior variance of derivatives at t
  return $\textsc{dkd}(t, t) - [\textsc{dk}(t, T'),\ \textsc{dkd}(t, T')] \cdot (G \backslash [\textsc{dk}(t, T'),\ \textsc{dkd}(t, T')]')$
end function

function $\textsc{v0f}(t, T, G)$
  ▶ posterior covariances of function values at $t = 0$ and t
  return $\textsc{k}(0, t) - [\textsc{k}(0, T'),\ \textsc{kd}(0, T')] \cdot (G \backslash [\textsc{k}(t, T'),\ \textsc{kd}(t, T')]')$
end function

function $\textsc{vd0f}(t, T, G)$
  ▶ posterior covariance of gradient and function value at $t = 0$ and t respectively
  return $\textsc{dk}(0, t) - [\textsc{dk}(0, T'),\ \textsc{dkd}(0, T')] \cdot (G \backslash [\textsc{k}(t, T'),\ \textsc{kd}(t, T')]')$
end function

function $\textsc{v0df}(t, T, G)$
  ▶ posterior covariance of function value and gradient at $t = 0$ and t respectively
  return $\textsc{kd}(0, t) - [\textsc{k}(0, T'),\ \textsc{kd}(0, T')] \cdot (G \backslash [\textsc{dk}(t, T'),\ \textsc{dkd}(t, T')]')$
end function

function $\textsc{vd0df}(t, T, G)$
  ▶ same as $\textsc{v0f}(\_)$ but for gradients
  return $\textsc{dkd}(0, t) - [\textsc{dk}(0, T'),\ \textsc{dkd}(0, T')] \cdot (G \backslash [\textsc{dk}(t, T'),\ \textsc{dkd}(t, T')]')$
end function

▶ all following procedures use the same design parameter:
$\tau \leftarrow 10$
---

function $\textsc{k}(a, b)$
  ▶ Wiener kernel integrated once in each argument
  return $1/3 \odot \min(a + \tau, b + \tau)^{\odot 3} + 0.5 \odot |a - b| \odot \min(a + \tau, b + \tau)^{\odot 2}$
end function

function $\textsc{kd}(a, b)$
  ▶ Wiener kernel integrated in first argument
  return $0.5 \odot (a < b) \odot (a + \tau)^{\odot 2} + (a \geq b) \odot \big((a + \tau) \cdot (b + \tau) - 0.5 \odot (b + \tau)^{\odot 2}\big)$
end function

function $\textsc{dk}(a, b)$
  ▶ Wiener kernel integrated in second argument
  return $0.5 \odot (a > b) \odot (b + \tau)^{\odot 2} + (a \leq b) \odot ((a + \tau) \cdot (b + \tau) - 0.5 \odot (a + \tau)^{\odot 2})$
end function

function $\textsc{dkd}(a, b)$
  ▶ Wiener kernel
  return $\min(a + \tau, b + \tau)$
end function

function $\textsc{ddk}(a, b)$
  ▶ Wiener kernel integrated in second argument and 1x derived in first argument
  return $(a \leq b) \odot (b - a)$
end function

function $\textsc{ddkd}(a, b)$
  ▶ Wiener kernel 1x derived in first argument
  return $(a \leq b)$
end function

function DDDK(a, b)
    ▶ Wiener kernel 2x derived in first argument and integrated in second argument
    return $-(a \leq b)$
end function

function probWolfe(t, T, A, G)
    ▶ design parameters
    $c_1 \leftarrow 0.05$ ▷ constant for Armijo condition
    $c_2 \leftarrow 0.5$ ▷ constant for curvature condition
    $c_W \leftarrow 0.3$ ▷ threshold for Wolfe probability

    ▶ mean and covariance values at start position ($t = 0$)
    $m_0 \leftarrow \mathrm{M}(0, T, A)$
    $dm_0 \leftarrow \mathrm{D1M}(0, T, A)$
    $V_0 \leftarrow \mathrm{V}(0, T, G)$
    $Vd_0 \leftarrow \mathrm{VD}(0, T, G)$
    $dVd_0 \leftarrow \mathrm{DVD}(0, T, G)$

    ▶ marginal mean and variance for Armijo condition
    $m_a \leftarrow m_0 - \mathrm{M}(t, T, A) + c_1 \cdot t \cdot dm_0$
    $V_{aa} \leftarrow V_0 + (c_1 \cdot t)^2 \cdot dVd_0 + \mathrm{V}(t) + 2 \cdot (c_1 \cdot t \cdot (Vd_0 - \mathrm{VD0F}(t)) - \mathrm{V0F}(t))$

    ▶ marginal mean and variance for curvature condition
    $m_b \leftarrow \mathrm{D1M}(t) - c_2 \cdot dm_0$
    $V_{bb} \leftarrow {c_2}^2 \cdot dVd_0 - 2 \cdot c_2 \cdot \mathrm{VD0DF}(t) + \mathrm{DVD}(t)$

    ▶ covariance between conditions
    $V_{ab} \leftarrow -c_2 \cdot (Vd_0 + c_1 \cdot t \cdot dVd_0) + c_2 \cdot \mathrm{VD0F}(t) + \mathrm{V0DF}(t) + c_1 \cdot t \cdot \mathrm{VD0DF}(t) - \mathrm{VD}(t)$

    ▶ extremely small variances → very certain (deterministic evaluation)
    if $V_{aa} \leq 10^{-9}$ **and** $V_{bb} \leq 10^{-9}$ **then**
        $p_{\text{Wolfe}} \leftarrow (m_a \geq 0) \cdot (m_b \geq 0)$

        ▶ accept?
        $p_{\text{acc}} \leftarrow p_{\text{Wolfe}} > c_W$
        return p_{acc}
    end if

    ▶ zero or negative variances (maybe something went wrong?)
    if $V_{aa} \leq 0$ **or** $V_{bb} \leq 0$ **then**
        return 0
    end if

    ▶ noisy case (everything is alright)
    ▶ correlation
    $\rho \leftarrow V_{ab} / \sqrt{V_{aa} \cdot V_{bb}}$

    ▶ lower and upper integral limits for Armijo condition
    $low_a \leftarrow -m_a / \sqrt{V_{aa}}$
    $up_a \leftarrow +\infty$

    ▶ lower and upper integral limits for curvature condition
    $low_b \leftarrow -m_b / \sqrt{V_{bb}}$
    $up_b \leftarrow \left(2 \cdot c_2 \cdot \left(|dm_0| + 2 \cdot \sqrt{dVd_0}\right) - m_b\right) / \sqrt{V_{bb}}$

    ▶ compute Wolfe probability
    $p_{\text{Wolfe}} \leftarrow \mathrm{BVN}(low_a, up_a, low_b, up_b, \rho)$

    ▶ accept?
    $p_{\text{acc}} \leftarrow p_{\text{Wolfe}} > c_W$
    return p_{acc}

The function $\mathrm{BVN}(low_a, up_a, low_b, up_b, \rho)$ evaluates the 2D-integral

$$\int_{low_a}^{up_a} \int_{low_b}^{up_b} \mathcal{N}\left(\begin{bmatrix} a \\ b \end{bmatrix}; \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right) da db.$$

end function

function gaussCDF(z)
    ▶ Gauss cumulative density function
    return $0.5 \odot \left(1 + \mathrm{erf}(z/\sqrt{2})\right)$
end function

function gaussPDF(z)
    ▶ Gauss probability density function
    return $\exp\left(-0.5 \odot z^{\odot 2}\right) \oslash \sqrt{2\pi}$
end function

function expectedImprovement(m, s, η)
    ▶ Jones et al. (1998)
    return $(\eta - m) \odot \mathrm{gaussCDF}((\eta - m) \oslash s) + s \odot \mathrm{gaussPDF}((\eta - m) \oslash s)$
end function

References

R.J. Adler. *The Geometry of Random Fields*. Wiley, 1981.

S.-I. Amari, H. Park, and K. Fukumizu. Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, 12(6):1399–1409, 2000.

M. Andrychowicz, M. Denil, S. Gomez-Colmenarejo, M.W. Hoffman, D. Pfau, T. Schaul, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. *CoRR*, abs/1606.04474, 2016. URL `http://arxiv.org/abs/1606.04474`.

L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1):1–3, 1966.

L. Balles, J. Romero, and P. Hennig. Coupling Adaptive Batch Sizes with Learning Rates. *ArXiv e-prints*, December 2016.

L. Balles, M. Mahsereci, and P. Hennig. Automating stochastic optimization with gradient variance estimates. *ICML AutoML Workshop*, 2017.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th Int. Conf. on Computational Statistic (COMPSTAT)*, pages 177–186. Springer, 2010.

T. Broderick, N. Boyd, A. Wibisono, A.C. Wilson, and M.I. Jordan. Streaming variational Bayes. In *Advances in Neural Information Processing Systems (NIPS 26)*, pages 1727–1735, 2013.

C.G. Broyden. A new double-rank minimization algorithm. *Notices of the AMS*, 16(4):670, 1969.

Z. Drezner and G.O. Wesolowsky. On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, 35(1-2):101–107, 1990.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

R. Fletcher. A new approach to variable metric algorithms. *The Computer Journal*, 13(3):317, 1970.

R. Fletcher and C.M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 1964.

A.P. George and W.B. Powell. Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming. *Machine Learning*, 65(1):167–198, 2006.

D. Goldfarb. A family of variable metric updates derived by variational means. *Math. Comp.*, 24 (109):23–26, 1970.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

I. Guyon, S. Gunn, A. Ben-Hur, and G.n Dror. Result analysis of the nips 2003 feature selection challenge. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 545–552. MIT Press, 2005.

E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I – Nonstiff Problems*. Springer, 1987.

S. Hansen. Using deep q-learning to control optimization hyperparameters. *CoRR*, abs/1602.04062, 2016. URL `http://arxiv.org/abs/1602.04062`.

P. Hennig. Fast Probabilistic Optimization from Noisy Gradients. In *30th International Conference on Machine Learning (ICML)*, 2013.

J. Hensman, M. Rattray, and N.D. Lawrence. Fast variational inference in the conjugate exponential family. In *Advances in Neural Information Processing Systems (NIPS 25)*, pages 2888–2896, 2012.

G.E. Hinton. *A Practical Guide to Training Restricted Boltzmann Machines*, pages 599–619. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35289-8. doi: 10.1007/978-3-642-35289-8_32. URL `http://dx.doi.org/10.1007/978-3-642-35289-8_32`.

M.D. Hoffman, D.M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL `http://arxiv.org/abs/1412.6980`.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. URL `https://www.cs.toronto.edu/~kriz/cifar.html`.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

K. Li and J. Malik. Learning to optimize. *CoRR*, abs/1606.01885, 2016. URL `http://arxiv.org/abs/1606.01885`.

M. Mahsereci and P. Hennig. Probabilistic line searches for stochastic optimization. In *Advances in Neural Information Processing Systems 28*, pages 181–189, 2015.

J. Martens. Deep learning via Hessian-free optimization. In *International Conference on Machine Learning (ICML)*, 2010.

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Verlag, 1999.

A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 3rd ed. edition, 1991.

R. Rajesh, W. Chong, D. Blei, and E. Xing. An adaptive learning rate for stochastic variational inference. In *30th International Conference on Machine Learning (ICML)*, pages 298–306, 2013.

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT, 2006.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, Sep. 1951.

N.L. Roux and A.W. Fitzgibbon. A fast natural Newton method. In *27th International Conference on Machine Learning (ICML)*, pages 623–630, 2010.

S. Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.

T. Schaul, S. Zhang, and Y. LeCun. No more pesky learning rates. In *30th International Conference on Machine Learning (ICML-13)*, pages 343–351, 2013.

N.N. Schraudolph. Local gain adaptation in stochastic gradient descent. In *Ninth International Conference on Artificial Neural Networks (ICANN) 99*, volume 2, pages 569–574, 1999.

D.F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Math. Comp.*, 24 (111):647–656, 1970.

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *International Conference on Machine Learning (ICML)*, 2010.

I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28. JMLR Workshop and Conference Proceedings, 2013.

G. Wahba. *Spline models for observational data*. Number 59 in CBMS-NSF Regional Conferences series in applied mathematics. SIAM, 1990.

W.H. Wolberg, W.N Street, and O.L. Mangasarian. *UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set*, January 2011. URL `http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`.

P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, pages 226–235, 1969.

T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Twenty-first International Conference on Machine Learning (ICML 2004)*, 2004.

Journal of Machine Learning Research 18 (2017) 1-49 Submitted 1/17; Revised 8/17; Published 11/17

Learning Instrumental Variables with Structural and Non-Gaussianity Assumptions

Ricardo Silva RICARDO@STATS.UCL.AC.UK
Department of Statistical Science and Centre for Computational Statistics and Machine Learning
University College London, WC1E 6BT, UK
The Alan Turing Institute, 96 Euston Road, London NW1 2DB, UK

Shohei Shimizu SHOHEI-SHIMIZU@BIWAKO.SHIGA-U.AC.JP
The Center for Data Science Education and Research
Shiga University, 1-1-1 Banba Hikone, Shiga 522-8522, Japan
The Institute of Scientific and Industrial Research, Osaka University, Japan
RIKEN Center for Advanced Intelligence Project, Japan

Editor: Joris Mooij

Abstract

Learning a causal effect from observational data requires strong assumptions. One possible method is to use instrumental variables, which are typically justified by background knowledge. It is possible, under further assumptions, to discover whether a variable is structurally instrumental to a target causal effect $X \to Y$. However, the few existing approaches are lacking on how general these assumptions can be, and how to express possible equivalence classes of solutions. We present instrumental variable discovery methods that systematically characterize which set of causal effects can and cannot be discovered under local graphical criteria that define instrumental variables, without reconstructing full causal graphs. We also introduce the first methods to exploit non-Gaussianity assumptions, highlighting identifiability problems and solutions. Due to the difficulty of estimating such models from finite data, we investigate how to strengthen assumptions in order to make the statistical problem more manageable.

Keywords: causality, causal discovery, instrumental variables

1. Contribution

Given observational data for a treatment variable X, an outcome Y, and a set of observed covariates $\mathbf{O}$ that precede X an Y causally, we present methods to estimate the causal effect of X on Y when hidden common causes between X and Y cannot be blocked by conditioning on observed variables. The main methodology consists of discovering *instrumental variables* among $\mathbf{O}$, which informally can be described as surrogate experiments on X that do not affect Y directly. This complements approaches where hidden common causes can be blocked. We will assume that the model is linear, although this assumption can be relaxed to some extent, as discussed in the conclusion. Much of the contribution is theoretical, and intended to describe what to the best of our knowledge is the first graphical account of the limits of what can be discovered about instrumental variables from constraints in the observational distribution. We also discuss a pragmatic implementation of such ideas and clarify their practical limitations.

Consider a linear graphical causal model (Spirtes et al., 2000; Pearl, 2000) where, given a directed acyclic graph (DAG) $\mathcal{G}$ with vertex set $\mathbf{V}$, we define a joint distribution in terms of conditional relationships between each variable V_i and its given *parents* in $\mathcal{G}$:

$$V_i = \sum_{V_j \in par_{\mathcal{G}}(i)} \lambda_{ij} V_j + e_i. \quad (1)$$

That is, each random variable V_i corresponds to a vertex in $\mathcal{G}$, where $par_{\mathcal{G}}(i)$ are the parents of V_i in $\mathcal{G}$ and e_i is an independent error term. Equation (1) is called a *structural equation* in the sense that it encodes a relationship that remains stable under a *perfect intervention* on other variables. Following the notation of Pearl (2000), we use the index "$do(V_k = v_k)$" to denote the regime under which variable V_k is fixed to some level v_k by an external agent. If V_k is a parent of V_i, the *(differential) causal effect* of V_k on V_i is defined as:

$$\frac{\partial \mathbb{E}[V_i \mid do(V_k = v_k)]}{\partial v_k} = \lambda_{ik}. \quad (2)$$

Each λ_{ik} will be referred to as a *structural coefficient*. Our goal is to estimate the differential causal effect of some treatment X on some outcome Y from observational data. If the common hidden causes of these two variables can be blocked by other observable variables, a formula such as the back-door adjustment of Pearl (2000) or the Prediction Algorithm of Spirtes et al. (2000) can be used to infer it. In general, unmeasured confounders of X and Y might remain unblocked.

When unmeasured confounding remains, and where it is reasonable to assume the linear structure (1), a possibility is to exploit an *instrumental variable* (or *instrument*, or *IV*; Morgan and Winship, 2015): some observable variable W that is not an effect of either X and Y, it is unconfounded with Y, and has no direct effect on Y.

Figure 1 illustrates one possible DAG containing an instrument. Let σ_{ab} represent the covariance of two variables A and B. The parameterization in (1) implies $\sigma_{wx} = \lambda_{xw}\sigma_{ww}$ and $\sigma_{wy} = \lambda_{yx}\lambda_{xw}\sigma_{ww}$. It follows that $\lambda_{yx} = \sigma_{wy}/\sigma_{wx}$. We can estimate σ_{wy} and σ_{wx} from observations, allowing for a consistent estimate of λ_{yx}. Notice that $\sigma_{wx} \neq 0$ is required. W in this case is called an instrumental variable for the causal relationship $X \to Y$.

It is not possible to test whether some observable variable is an IV from its joint distribution with X and Y alone. IV assumptions can nevertheless be falsified by exploiting constraints in the joint distribution of multiple observable variables (Chu et al., 2001; Brito and Pearl, 2002; Kuroki and Cai, 2005), where such constraints are necessary but not sufficient to identify IVs and the corresponding causal effects. Our contribution are IV discovery algorithms for causal effect estimation. We characterize in which ways such algorithms can find the correct causal effect, and in which sense they will fail. We also introduce variations of the assumptions that are needed for practical reasons, complementing existing methods that rely on other sets of assumptions. Some of the ideas used in our methods are based on principles from causal discovery in linear non-Gaussian models (Shimizu et al., 2006).

The structure of the paper is as follows. In Section 2, we discuss the challenges of inferring causal effects when treatment and outcome are confounded by unobserved variables, and

©2017 Ricardo Silva and Shohei Shimizu.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-014.html.

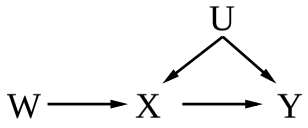


Figure 1: A graph illustrating a possible IV structure. X and Y have an unmeasured confounder U. W is an instrument as it is unconfounded with Y, has no direct effect on it, and causes X. In this paper, variables named "U" will denote hidden variables.

provide an overview of our approach. In Section 3, we discuss the theory behind two classes of testable constraints that can be detected from data. The resulting algorithm has several practical issues, and a more realistic alternative is provided in Section 4, which is then validated experimentally in Section 5. Other related approaches are discussed in Section 6.

2. Outline of Methodology: Learning under Unmeasured Confounding

We assume that the system of interest is a linear causal model which is Markov with respect to an unknown DAG $\mathcal{G}$ with vertices $\mathbf{V} \equiv \mathbf{O} \cup \mathbf{U} \cup \{X, Y\}$. Set $\mathbf{O}$ is the set of observable variables besides *treatment* variable X and *outcome* variable Y. Set $\mathbf{U}$ is the set of unobservable (also called *latent*, or *hidden*) variables. X and Y do not precede any element of $\mathbf{V} \backslash \{X, Y\}$. Y does not precede X. The task is to estimate the differential causal effect of X on Y. Our problems of interest are formulated so that all have a *target causal effect* λ_{yx} corresponding to edge $X \to Y$. To avoid constant repetition, in several results that appear in this paper, we will often refer to vertices and edges of a (possibly unnamed) causal graph $\mathcal{G}$ while implicitly assuming the structural assumptions and notation introduced in this paragraph.

Focusing on a single λ_{yx} is common in applied sciences: in particular, scenarios where we have a target causal effect $X \to Y$ to be estimated, and a set $\mathbf{O}$ of covariates preceding X and Y is available. Morgan and Winship (2015) provide several examples. This is in contrast to the more familiar causal structure discovery tasks in the machine learning literature, where an equivalence class of a whole causal system is learned from data, and where some causal queries may or may not be identifiable (Spirtes et al., 2000). The focus here in on quantifying the strength of a particular causal effect λ_{yx}, as opposed to unveiling the directionalities and connections of a causal graph. This allows more focused algorithms that bypass a full graph estimation. This philosophy was exploited by Entner et al. (2012) in the task of finding possible sets of observable variables that can block the effect of any hidden common cause of X and Y.

However, the approach by Entner et al. will not provide a causal effect estimate if such a set does not exist. For instance, in Figure 1, if U is a latent variable, their algorithm will provide no answer concerning the causal effect of X on Y. *Our goal is to cover this scenario, which complements approaches that require unmeasured confounding to be blocked.* The methodological framework to accomplish this task is by discovering candidate instru-

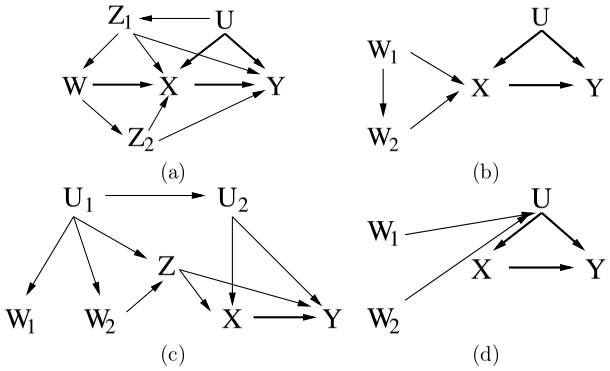


Figure 2: (a) Variable W is an instrument for the relation $X \to Y$ conditioning on $\{Z_1, Z_2\}$. This can be verified by noticing that W is d-separated from Y given $\{Z_1, Z_2\}$ in the graph obtained by removing edge $X \to Y$ from the given figure. (b) Both W_1 and W_2 are instruments conditioning on the empty set $\emptyset$. (c) The typical covariance constraints ("tetrads") that are implied by instrumental variables also happen in the case where no instruments exist, implying that rank constraints in the covariance matrix are not enough information to discover IVs. (d) A case that is difficult even when considering information from non-Gaussian distributions.

mental variables without prior knowledge of the causal structure, besides the stated ordering assumptions about $\{X, Y\}$. The challenge is that we cannot guarantee which candidate instrumental variables are actual instruments without further assumptions.

We will make use of structural characterizations of causality using graphical models (Pearl, 2000; Spirtes et al., 2000). Prior exposure to causal graphical models is assumed, with key definitions summarized in Section 2.1. In Section 2.2, we outline the challenges and explain the general concept of *equivalence class of causal effects*, a concept adapted from the work of Maathuis et al. (2009) to the instrumental variable case. Finally, in Section 2.3 we provide a road map of the steps used in our approach.

2.1 Graphical Terminology

Directed acyclic graphs (DAGs) encode independencies among vertices, which correspond to independencies among random variables. This follows from the usual one-to-one relationship between vertices in a graph and random variables in a distribution. In a DAG, any two vertices $\{V_i, V_j\}$ can be connected by up to one *directed edge*. Given an edge $V_i \to V_j$, we say that V_i is the *tail* and V_j is the *head* of the edge. We also say that V_i and V_j are *endpoints* of edge $V_i \to V_j$ where V_i is a *parent* of V_j and V_j is a *child* or V_i.

A *path* in a graph is a sequence of edges, where any two consecutive edges in the sequence share a common endpoint. Paths are *simple* if no vertex appears twice, and *non-simple*

otherwise. For instance, $W \to X \leftarrow U \to Y$ is a path in Figure 1. An example of a non-simple path is $W \to X \leftarrow U \to X \leftarrow U \to Y$. For most of our presentation, it is sufficient to just assume that the paths referred to are simple unless stated otherwise. A *collider* in a path is a vertex that is common to two consecutive edges such that this vertex is the head of both edges. For instance, X is a collider in the path given in the previous example, while U is not. A vertex V is an *endpoint of a path* P if it is one of the endpoints of the first or last edge E in the path, and V is not shared with the (possible) edge next to E in P. For instance, W and Y are the endpoints of $W \to X \leftarrow U \to Y$.

A path P is *between* vertices V_i and V_j if V_i and V_j are the endpoints of P. A *trek* is a path with no colliders. For instance, $W \to X \leftarrow U \to Y$ is not a trek, but $X \leftarrow U \to Y$ and $W \to X \to Y$ are. A trek P must have an unique *source*, the vertex in P that is not the head of any edge in P. A special case of a trek is a *directed path*, which is a trek where the source is one of the endpoints. For instance, $W \to X \to Y$ is a directed path between W and Y where W is the source. We also say that this path is *from* source (W) *into* the other endpoint (Y). The source V_i in a directed path P is an *ancestor* of all elements V_j in P, while V_j is a *descendant* of V_i. It is possible that $V_i = V_j$, so V_i is an ancestor and descendant of itself. A *non-directed path* is a path that is not directed.

A *back-door (path)* between V_i and V_j is a trek that is into V_i and V_j. For instance, $X \leftarrow U \to Y$ is a back-door between X and Y.

A vertex V is *active on a path* with respect to some vertex set $\mathbf{S}$ if it is either (i) a collider in this path and itself or one of its descendants is in $\mathbf{S}$; or (ii) not a collider and not in $\mathbf{S}$. A path is *active* if all of its vertices are active, *blocked* otherwise. The notion of active and blocked paths will be important in the sequel, as activation has implications on which variables can be considered to be an instrument with respect to which conditioning sets.

These definitions lead to the concept of *d-separation* (Pearl, 2000). A vertex V_i is d-separated from a vertex V_j given a set $\mathbf{S}$ if and only if every path between V_i and V_j is blocked by $\mathbf{S}$. The interpretation of this definition is explained at length in the graphical modeling literature and we will assume familiarity with it. We say a probabilistic model $\mathcal{M}$ is *Markov* with respect to a DAG $\mathcal{G}$ if every d-separation in $\mathcal{G}$ corresponds to a conditional independence constraint in $\mathcal{M}$. A model $\mathcal{M}$ is *faithful* to $\mathcal{G}$ if every d-separation in $\mathcal{G}$ corresponds to a conditional independence constraint in $\mathcal{M}$ and vice-versa. We sometimes use the notation $V_i \perp\!\!\!\perp V_j \mid \mathbf{S}$ (Dawid, 1979) to denote conditional independence of V_i and V_j given $\mathbf{S}$ in a distribution, and also to denote d-separation when applied to vertex sets in a graph.

2.2 Scope and Fundamental Challenges

The identification of structural coefficients from given causal graphs is a classical problem in the structural equation modeling literature (Bollen, 1989). Much progress has been achieved on describing increasingly intricate combinations of structural features that lead to the identification of such coefficients (Brito and Pearl, 2002; Foygel et al., 2011; Chen et al., 2014). As these sophisticated criteria also lead to constraints which are hard to detect from data, we focus instead on the class of structures that corresponds to classical accounts of instrumental variables (Angrist and Pischke, 2009) as described by Brito and Pearl (2002).

Brito and Pearl's criteria are as follows. Given the causal graph $\mathcal{G}$ of a system that includes an edge $X \to Y$, a vertex W is a (conditional) instrument variable for $X \to Y$ given $\mathbf{Z}$ if and only if:

1. $\mathbf{Z}$ does not d-separate W from X in $\mathcal{G}$, and
2. $\mathbf{Z}$ d-separates W from Y in the graph obtained by removing the edge $X \to Y$ from $\mathcal{G}$, and
3. $\mathbf{Z}$ are non-descendants of X and Y in $\mathcal{G}$.

For the rest of the paper, we will call the above conditions the *graphical criteria for instrumental variable validity*, or simply "Graphical Criteria." Notice that the validity of a vertex as an IV is dependent on which set $\mathbf{Z}$ we condition on. That is, if in the corresponding causal graph we find some set $\mathbf{Z}$ that blocks all and only the paths relevant to the Graphical Criteria, then we can identify λ_{yx} as $\sigma_{wy.\mathbf{z}}/\sigma_{wx.\mathbf{z}}$, where $\sigma_{ab.\mathbf{s}}$ represents the partial covariance[1] of two variables A and B given some set $\mathbf{S}$. Figure 2(a) illustrates a case.

Unless strong background knowledge is available, the relevant structure needs to be learned from the data. As an example, consider Figure 1. When U is unobserved and the distribution is faithful to the graph, the absence of the edge $W \to Y$ is not testable (Chu et al., 2001). However, in a situation such as Figure 2(b), the *simultaneous* lack of edges $W_1 \to Y$ and $W_2 \to Y$ has a testable implication, as the absence of each edge implies $\lambda_{yx} = \sigma_{w_1 y}/\sigma_{w_1 x}$ and $\lambda_{yx} = \sigma_{w_2 y}/\sigma_{w_2 x}$, respectively. This leads to a *tetrad constraint*,

$$\sigma_{w_1 y}\sigma_{w_2 x} - \sigma_{w_1 x}\sigma_{w_2 y} = 0, \tag{3}$$

which can be tested using observable data. Unfortunately, the tetrad constraint is necessary, but not sufficient, to establish that both elements in this pair of variables are instrumental. As an example, consider Figure 2(c). It is not hard to show that $\sigma_{w_1 y.z}\sigma_{w_2 x.z} - \sigma_{w_1 x.z}\sigma_{w_2 y.z} = 0$. However, the Graphical Criteria for IVs are not satisfied as W_1 is not d-separated from Y given Z if we remove edge $X \to Y$. This is because path $W_1 \leftarrow U_1 \to U_2 \to Y$ is active. Indeed, in this case λ_{yx} can be much different from $\sigma_{w_1 y.z}/\sigma_{w_1 x.z}$. A major component of our contribution is to characterize graphically in which ways the solution is not unique.

We can summarize the main identification challenge as follows. In the Graphical Criteria, the challenging condition is the second, as the first is easily testable by faithfulness and the third is given by assumption. Another way of phrasing condition 2, with respect to conditioning set $\mathbf{Z}$, is as follows:

2a. there is no active non-directed path between W and Y that does not include X (that is, no active back-door path nor any active path that includes a collider), and

2b. there is no active directed path from W to Y that does not include X.

While we will be able to check 2a, we will not be able to verify 2b, learning to multiple candidate causal effects.

1. Analogous to the (more widely used) notion of partial correlation, we define partial covariance as the covariance between the residuals of the least-squares regression of A and B on $\mathbf{S}$, given by $\sigma_{ab.\mathbf{s}} \equiv \sigma_{ab} - \Sigma_{ab,\mathbf{s}}\Sigma_{\mathbf{ss}}^{-1}\Sigma_{\mathbf{s},ba}$, where Σ is the joint covariance matrix and $\sigma_{ij} \equiv \Sigma_{ij}$ for scalars i, j.

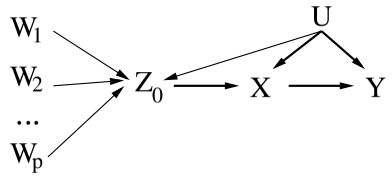


Figure 3: In this model, variables $W_1, \dots W_p$ are instrumental variables conditioning on the empty set. However, conditioning on Z_0 will introduce an active path from each W_i to Y via U, destroying their validity. This is particularly an issue for algorithms such as sisVIVE (Kang et al., 2016) and IV-BY-MAJORITY$_\infty$ (Section 3), where each variable is either deemed an IV or a conditioning variable. An algorithm such as IV-TETRAD$_\infty$ (Section 3) is able to avoid this mistake, although the computational cost can be high.

2.3 Roadmap

In general, the best we can do is to provide a *set* of candidate causal effects, one of which will be correct if at least two instrumental variables (under the same conditioning set) are present in the true graph. In this case, the set can be used, for instance, to provide lower and upper bounds on the causal effect. We will discuss our methods in the context of different assumptions about the existence of instruments, particularly about the *number* of instruments that exist in the true unknown graph.

That is, we propose algorithms that in theory return *equivalence classes of causal effects*. Such a class of algorithms will be sound in this equivalence class sense of returning a set of candidate effects that includes the true effect *if* there are instrumental variables, but will be incorrect otherwise. Following Cooper (1997), Mani et al. (2006) and Entner et al. (2012), we will not need to discover full graphs in order to identify the causal effect, but we will not also find all causal effects that are identifiable from faithfulness assumptions. In particular, we will consider in which sense our algorithms are complete: that is, if there are instrumental variables satisfying the Graphical Criteria, we will characterize under which conditions we will find them. Computational considerations are relevant here.

In the next section, we introduce algorithms that return an equivalence class of causal effects using tetrad constraints, which are complemented by non-Gaussianity assumptions. Motivated by simplicity of presentation, all approaches in Section 3 assume that the distribution of the population is known and that computational resources are unbounded. We do not claim these algorithms are practical —the goal is to use them as a theoretical basis to choose and justify stronger assumptions that achieve practical learning. In Section 4, we discuss practical methods for learning from data and the computational and identification compromises we adopt.

A roadmap for all algorithms introduced in this paper is shown in Table 1. In the table, we provide the main assumptions used by each method and whether or not the algorithm

Algorithm	Main Assumptions	Tractable
1: IV-BY-MAJORITY$_\infty$	there exists a set $\mathbf{W} \subseteq \mathbf{O}$, of size larger than $\lvert\mathbf{O}\rvert/2$, such that each $W \in \mathbf{W}$ is a valid IV given $\mathbf{O}\backslash\{W\}$.	Yes
2: IV-TETRAD$_\infty$	there exists a set $\mathbf{W} \subseteq \mathbf{O}$, of size at least 2, and a set $\mathbf{Z} \subseteq \mathbf{O}\backslash\mathbf{W}$, such that each $W \in \mathbf{W}$ is a valid IV given $\mathbf{Z}$.	No
3: IV-TETRAD$^+_\infty$	the assumptions of IV-TETRAD$_\infty$, plus the assumption that no non-directed paths between each W and Y are active given $\mathbf{Z} \cup \mathbf{W}\backslash\{W\}$, and that the distribution is non-Gaussian.	No
4: IV-TETRAD$^+$	the assumptions of IV-TETRAD$^+_\infty$, plus the assumptions that $\mathbf{Z} = \mathbf{O}\backslash\mathbf{W}$ and $\lvert\mathbf{W}\rvert \geq K$ for some given $K \geq 2$.	Yes
5: IV-TETRAD^{++}	the assumptions of IV-TETRAD$^+$.	Yes (by MCMC)

Table 1: All algorithms introduced in this paper, along with their main assumptions and whether they are computationally tractable. Algorithms 2-5 assume also a form of faithfulness discussed in Section 3. Further assumptions can be added on top of the assumptions listed below in order to reduce the size of the equivalence class returned, as discussed in Sections 3 and 4.

is tractable as a function of $|\mathbf{O}|$, the number of observed variables other than treatment X and outcome Y. The relationship among these algorithms is a follows.

Algorithm 1, introduced at the beginning of Section 3, is inspired by current approaches for learning instrumental variables. One of its main requirements is the existence of a large number of valid instruments, which grows with $|\mathbf{O}|$. It does not, however, require faithfulness.

Algorithms 2 and 3, introduced in Sections 3.1 and 3.3 respectively, are motivated by the requirement of only two valid instruments. Also, there is no requirement concerning a special choice for the respective conditioning set: it can be a strict subset of the remaining variables. The two algorithms, however, assume a variant of faithfulness explained in the next section. Algorithm 3 requires stronger assumptions than Algorithm 2, but as it will be shown, it will return an equivalence class no larger than Algorithm 2 while sometimes guaranteeing a strictly smaller equivalence class.

Unlike the previous algorithms, Algorithm 4, introduced in Section 4, is aimed at practical learning with finite sample sizes and limited computational resources. That is, here computational tractability and the handling of statistical variability are important issues to be considered. We achieve a reduction on statistical variability by enforcing a stronger constraint: the unknown true model must contain a set of valid IVs no smaller than K,

Algorithm 1 IV-BY-MAJORITY$_\infty$

1: **Input:** set of random variables $\mathbf{O} \cup \{X, Y\}$ with known joint distribution
2: **Output:** the causal effect of X on Y, or a value (NA) indicating lack of knowledge
3: **for** each $W_i \in \mathbf{O}$ **do**
4: $\quad \mathbf{Z}_i \leftarrow \mathbf{O} \backslash \{W_i\}$
5: $\quad \beta_i \leftarrow \sigma_{w_i y.\mathbf{z}_i} / \sigma_{w_i x.\mathbf{z}_i}$
6: **end for**
7: **if** more than half of set $\{\beta_i\}$ is equal to the same finite value β **then**
8: $\quad$ **return** β
9: **end if**
10: **return** NA

a user-defined parameter. We achieve tractability, and also a further reduction on statistical variability, by requiring the conditioning set to be automatically determined given a candidate choice of IVs, as in Algorithm 1.

Finally, Algorithm 5 is introduced late in Section 5.3.2 as an example of a Bayesian variation of Algorithm 4. The goal is to illustrate how variations of Algorithm 4 can be created to respond to particular needs of an individual application, such as the provision of Bayesian measures of uncertainty of our estimates. The algorithm is still tractable assuming Markov chain Monte Carlo (MCMC) can be reliably run on an acceptable amount of time.

Although it seems that we take a step backwards in Algorithm 4, by enforcing back assumptions about $\mathbf{Z}$ that are similar to Algorithm 1, we will explain in Section 4 that these are not fundamental. However, practical learning with finite sample sizes is hard to achieve otherwise.

3. From Structural Constraints to Instruments and Causal Effects

Consider Algorithm 1 as a method for learning causal effects given the distribution of the population (hence, the "∞" symbol in the name of the algorithm, indicating that this is equivalent to having infinite sample sizes). If W_i is an IV conditioned on $\mathbf{Z}_i \equiv \mathbf{O} \backslash W_i$, then $\beta_i \equiv \sigma_{w_i y.\mathbf{z}_i} / \sigma_{w_i x.\mathbf{z}_i} = \lambda_{yx}$, the true causal effect. Without knowing whether W_i is a conditional IV with respect to $\mathbf{Z}_i$, we cannot make claims about the causal effect.

However, if more than half of the elements $W_i \in \mathbf{O}$ are conditional IVs given the respective $\mathbf{Z}_i$, then it follows that more than half of the elements in set $\{\beta_i\}$ will be equal, and equal to λ_{yx}. This is the same core assumption introduced by Kang et al. (2016). It sidesteps the problems introduced by models such as the one in Figure 2(c) by assuming that more than half of $\mathbf{O}$ are "valid" IVs. That is, we can partition $\mathbf{O}$ into two sets, $\mathbf{O} = \mathbf{W} \cup \mathbf{Z}$, such that each $W \in \mathbf{W}$ is a conditional IV given $\mathbf{Z} \cup \mathbf{W} \backslash \{W\}$. This is done without knowledge of which variables are valid and which are not. As discussed by Kang et al. (2016), there are situations where this assumption is plausible, or at least weaker than in standard approaches, as in some genetic studies where $\mathbf{O}$ are genetic features of a cell and X, Y are phenotypes. Kang et al. (2016) introduced the sisVIVE algorithm ("some invalid, some

Algorithm 2 IV-TETRAD$_\infty$

1: **Input:** set of random variables $\mathbf{O} \cup \{X, Y\}$ with known joint distribution
2: **Output:** $\mathcal{C}$, a set of candidate differential causal effects of X on Y
3: Initialize $\mathcal{C} \leftarrow \emptyset$
4: **for** each pair $\{W_i, W_j\} \subseteq \mathbf{O}$ **do**
5: $\quad$ **for** every set $\mathbf{Z} \subseteq \mathbf{O} \backslash \{W_i, W_j\}$ **do**
6: $\quad\quad$ **if** $\sigma_{w_i x.\mathbf{z}} = 0$ **or** $\sigma_{w_j x.\mathbf{z}} = 0$ **then**
7: $\quad\quad\quad$ **next**
8: $\quad\quad$ **end if**
9: $\quad\quad$ **if** $\sigma_{w_i x.\mathbf{z}} \sigma_{w_j y.\mathbf{z}} \neq \sigma_{w_i y.\mathbf{z}} \sigma_{w_j x.\mathbf{z}}$ **then**
10: $\quad\quad\quad$ **next**
11: $\quad\quad$ **end if**
12: $\quad\quad$ $\mathcal{C} \leftarrow \mathcal{C} \cup \{\sigma_{w_i y.\mathbf{z}} / \sigma_{w_i x.\mathbf{z}}\}$
13: $\quad$ **end for**
14: **end for**
15: **return** $\mathcal{C}$

valid IV estimator") which, although very different[2] from Algorithm 1, relies on the same fundamental assumptions and as such will be used in our benchmark methods in Section 5.

However, this assumption can be false even when nearly all of $\mathbf{O}$ are possibly valid instruments. Consider Figure 3 where we have an arbitrary number of IVs $W_1, \ldots, W_p$ that are valid by conditioning on the empty set. *None* of them are valid by conditioning on Z_0. In this situation, IV-BY-MAJORITY$_\infty$ will return NA and sisVIVE loses its guarantees, potentially performing badly. We will characterize how this happens by using faithfulness.

3.1 Structural Signatures of Tetrad Constraints and the Graphical Criteria

Consider Algorithm 2 as a method for learning causal effects, disregarding for now its computational complexity. The idea is to find triplets $(W_i, W_j, \mathbf{Z})$ such that $(W_i, \mathbf{Z})$ and $(W_j, \mathbf{Z})$ both satisfy the Graphical Criteria. Under the assumption of *linear faithfulness* (Spirtes et al., 2000), Line 6 of the algorithm is equivalent to item 1 of the Graphical Criteria. To characterize what Algorithm 2 can say about item 2 of the Graphical Criteria, we will need other, less usual, graphical definitions introduced in the next section.

Algorithm 2 is simple, but to the best of our knowledge is has not been described in the literature. A possible reason for the absence of this algorithm in the literature is that characterizing its limitations is not straightforward. One of the main contributions of this paper is the analysis of IV-TETRAD$_\infty$ in Section 3.2, for which we will first need the following theory.

2. sisVIVE uses a modified sparse regression method that is arguably more statistically elegant than using, for instance, hypothesis testing to decide which estimated λ_i are statistically the same. The method remains computationally efficient. We introduced Algorithm 1 because it is closely related to the other algorithms discussed in this paper and hence provides more direct insights on the interplay between different assumptions and identification issues. Notice, however, that sisVIVE never returns NA as it assumes there is always a majority.

3.1.1 t-Separation and Rank Constraints

In this section, we describe the notion of *t-separation* (Sullivant et al., 2010), the most important graphical concept in our approach. Readers familiar with this concept can skip this section.

Recall the basic definition of a trek from Section 2.1. Another way of describing a trek T is by ordering its endpoints (such that a trek is "from" V_i "to" V_j), implying an ordered pair of (possibly empty) directed paths $(P_1; P_2)$ where: P_1 has *sink* (vertex without children in T) V_i; P_2 has sink V_j; and P_1, P_2 have the same *source* (vertex in T without parents in T).

Definition 1 (t-separation) *The ordered pair of vertex sets* $(\mathbf{C_A}; \mathbf{C_B})$ t-separates *vertex set* $\mathbf{A}$ *from vertex set* $\mathbf{B}$ *if, for every trek* $(P_1; P_2)$ *from a vertex in* $\mathbf{A}$ *to a vertex in* $\mathbf{B}$*, either* P_1 *contains a vertex in* $\mathbf{C_A}$ *or* P_2 *contains a vertex in* $\mathbf{C_B}$*.*

Spirtes (2013) and Sullivant et al. (2010) discuss a generalization of this notion and further examples. As the definition is somewhat complex, later in this section we provide some basic examples of this concept in the context of instrumental variables.

Like d-separation, t-separation is relevant if it implies testable implications that decrease the set of graphical structures compatible with the data (Spirtes et al., 2000). While we will see that t-separation can also imply independence constraints among observable variables, this would be of limited interest as d-separation can also imply such constraints. The importance of t-separation is the possibility of implying testable consequence of d-separations *given unobserved variables*. This is done under the assumption of linearity, which will lead to rank constraints (Spirtes, 2013).

Definition 2 (rank constraint) *A* rank constraint *in a matrix* M *is any constraint of the type* $rank(M) \leq r$*, where* r *is some constant and* $rank(M)$ *is the rank of matrix* M*.*

If M is the cross-covariance submatrix given by variables $\{V_i, V_j\}$ indexing the rows, and by $\{V_k, V_l\}$ indexing the columns, then the rank constraint $rank(M) \leq 1$ implies $\sigma_{ik}\sigma_{jl} - \sigma_{il}\sigma_{jk} = 0$. That is, the determinant of M is zero, and a corresponding tetrad constraint holds.

Let $\Sigma_{\mathbf{AB}}$ be the cross-covariance matrix of set $\mathbf{A}$ (rows) and set $\mathbf{B}$ (columns). The DAG Trek Separation Theorem of Sullivant et al. (2010) says:

Theorem 3 (Trek Separation for DAGs) *Let* $\mathcal{G}$ *be a DAG with vertex set* $\mathbf{V}$*. Let* $\mathbf{A}$ *and* $\mathbf{B}$ *be subsets of* $\mathbf{V}$*. We have* $rank(\Sigma_{\mathbf{AB}}) \leq r$ *in all linear structural equation models with graph* $\mathcal{G}$ *if and only if there exist subsets* $\mathbf{C_A}$ *and* $\mathbf{C_B}$ *of* $\mathbf{V}$ *with* $|\mathbf{C_A}| + |\mathbf{C_B}| \leq r$ *such that* $(\mathbf{C_A}; \mathbf{C_B})$ *t-separates* $\mathbf{A}$ *from* $\mathbf{B}$*.*

To jump from (testable) rank constraints to (unobservable) structural constraints in $\mathcal{G}$, we assume our model distribution $\mathcal{P}$ is *rank-faithful* to a DAG $\mathcal{G}$ (Spirtes, 2013):

Definition 4 ((linear) rank-faithfulness) *Distribution* $\mathcal{P}$ *is* rank-faithful *to a DAG* $\mathcal{G}$ *if every rank constraint holding on a covariance (sub)matrix derived from* $\mathcal{P}$ *is entailed by every linear structural equation model Markov with respect to* $\mathcal{G}$*.*

One implication of rank-faithfulness is that $|\Sigma_{\mathbf{ZZ}}| > 0$ for any set $\mathbf{Z}$ of vertices in $\mathcal{G}$, a fact which we will make use of in some proofs.

Unlike d-separation, t-separation is defined by a pair of conditioning sets. Moreover, when we say that V_i is t-separated from V_j given some ordered pair $(\mathbf{C}_I; \mathbf{C}_J)$, the order of the sets in the conditioning pair matters. These two sets do not need to be disjoint and can also contain V_i and V_j.

Example 1 *Consider Figure 1 again. In this example,* X *alone cannot d-separate* W *from* Y*, due to the active path* $W \to X \leftarrow U \to Y$*. We need also to condition on* U*, but we cannot directly test the respective conditional independence if* U *is a latent variable. It is however the case that if* $\mathbf{A} \equiv \{W\}$ *and* $\mathbf{B} \equiv \{Y\}$*, then* $(\emptyset; \{X\})$ *t-separates* $\mathbf{A}$ *from* $\mathbf{B}$*: using the notation of Definition 1, the only trek between* W *and* Y *is* $(\emptyset; W \to X \to Y)$*.* $\mathbf{C_A} = \emptyset$ *and* $\mathbf{C_B} = \{X\}$ *will suffice to imply the t-separation.*

Moreover, by Theorem 3, cross-covariance matrix $\Sigma_{\mathbf{AB}}$ has rank at most $|\emptyset| + |\{X\}| = 1$. Notice, however, that this is true regardless or not this t-separation holds. Therefore, without knowing the graph, we still cannot exclude edge $W \to Y$.

If we try to redefine the problem such that $\mathbf{A} \equiv \{W, X\}$ and $\mathbf{B} \equiv \{X, Y\}$, hoping to show that $\Sigma_{\mathbf{AB}}$ will not be full rank, then this will not work either. This is because $(\emptyset, X)$ does not imply the t-separation anymore, as there is now the trek $(X \leftarrow U; U \to Y)$ between $\mathbf{A}$ and $\mathbf{B}$. Pair $(\emptyset; \{X\})$ does not block this trek. We need the conditioning set pair to be $(\{X\}, \{X\})$. By the notation of Theorem 3, this means $\mathbf{C_A} = \mathbf{C_B} = \{X\}$ and all the theorem implies is that $\Sigma_{\mathbf{AB}}$ has rank at most $|\{X\}| + |\{X\}| = 2$. Once again, this is true regardless of the absence of the edge $W \to Y$. □

Example 2 *Consider Figure 2(b) again. Pair* $\mathbf{A} \equiv \{W_1, W_2\}$ *is t-separated from* $\mathbf{B} \equiv \{X, Y\}$ *by* $(\emptyset, \{X\})$*.*

To check this, enumerate all simple treks $(P_1; P_2)$ such that path P_1 has its sink in $\{W_1, W_2\}$ and path P_2 has its sink in $\{X, Y\}$. We obtain:

- $(P_1^{[1]}; P_2^{[1]}) \equiv (\emptyset; W_1 \to X)$,
- $(P_1^{[2]}; P_2^{[2]}) \equiv (\emptyset; W_1 \to X \to Y)$,
- $(P_1^{[3]}; P_2^{[3]}) \equiv (\emptyset; W_1 \to W_2 \to X)$,
- $(P_1^{[4]}; P_2^{[4]}) \equiv (\emptyset; W_1 \to W_2 \to X \to Y)$,
- $(P_1^{[5]}; P_2^{[5]}) \equiv (\emptyset; W_2 \to X)$,
- $(P_1^{[6]}; P_2^{[6]}) \equiv (\emptyset; W_2 \to X \to Y)$,
- $(P_1^{[7]}; P_2^{[7]}) \equiv (W_2 \leftarrow W_1; W_1 \to X)$,
- $(P_1^{[8]}; P_2^{[8]}) \equiv (W_2 \leftarrow W_1; W_1 \to X \to Y)$.

We can then verify that, for each case, it is indeed true that $\{X\}$ blocks all paths $P_2^{[1]}, \ldots, P_2^{[8]}$,. Since $|\emptyset| + |\{X\}| = 1$, then $|\Sigma_{\mathbf{AB}}|$ has rank at most 1. This constraint is *not* satisfied under rank-faithfulness if, for instance, edge $W_1 \to Y$ were to be present. That is, even though X does not d-separate $\{W_1, W_2\}$ from Y, it alone is responsible for a t-separation and no hidden variable is evoked in its testable claim of $rank(\Sigma_{\mathbf{AB}}) \leq 1$. □

Example 3 *Consider Figure 2(c) again. Using standard d-separation, we can verify that Z alone does not d-separate $\{W_1, W_2\}$ from $\{X, Y\}$. Instead, $\{U_1, Z\}$, or $\{U_2, Z\}$, should be used instead. In the absence of the causal graph, we cannot discover this d-separation by directly testing the corresponding conditional independence given a latent variable. We need indirect evidence for this d-separation that uses observable variables only.*

That is where *conditional* tetrad constraints play a role. Assuming Z is observable, we could de-activate all directed paths from $\{W_1, W_2\}$ to $\{X, Y\}$ that go through Z and consider the *partial* cross-covariance matrix of $\{W_1, W_2\}$ against $\{X, Y\}$ given Z. The Trek Separation Theorem for DAGs, however, says nothing explicit about conditional cross-covariances. The independence model given by conditioning on a variable is sometimes a DAG itself, but this is not true in general (Richardson and Spirtes, 2002). Sullivant et al. (2010) present versions of the theorem for some classes of DAGs under conditioning, but not in a completely general way, as further explained in Appendix A.

Fortunately, for our purposes, conditioning can be easily dealt with: let us simply introduce the conditioning variables in both sets $\mathbf{A}$ and $\mathbf{B}$. In this example, define $\mathbf{A} \equiv \{W_1, W_2, Z\}$ and $\mathbf{B} \equiv \{X, Y, Z\}$. Then it is not hard to verify that $(\emptyset; \{U_1, Z\})$ t-separates $\mathbf{A}$ from $\mathbf{B}$: this follows by first verifying that all treks between $\{W_1, W_2, Z\}$ and $\{X, Y, Z\}$ include U_1 or Z.[3] Cross-covariance $\Sigma_{\mathbf{AB}}$ is now rank-deficient since $|\emptyset| + |\{U_1, Z\}| = 2 < 3$, which means its determinant is zero. Because its determinant can be written as $\sigma_{zz}(\sigma_{w_1x.z}\sigma_{w_2y.z} - \sigma_{w_1y.z}\sigma_{w_2x.z})$ and we assume $\sigma_{zz} \neq 0$, the (conditional) tetrad constraint will hold. This means that without observing U_1 we can make claims about the conditional structure between $\{W_1, W_2\}$ and $\{X, Y\}$. □

Like d-separation, it is possible that more than one conditioning set pair implies t-separation between two sets $\mathbf{A}$ and $\mathbf{B}$. In the last example, it is also the case that $(\emptyset; \{U_2, Z\})$ implies t-separation.

In Appendix A, we provide further examples of t-separation. *Linear faithfulness*, the assumption that vanishing partial correlations hold in the distribution if and only if a corresponding d-separation also holds in $\mathcal{G}$ (Spirtes et al., 2000), is a special case of rank faithfulness since t-separation subsumes d-separation (Sullivant et al., 2010), as also discussed in Appendix A.

3.2 Analysis of Algorithm 2

The algorithm searches for rank constraints among quartets $\{W_i, W_j\} \times \{X, Y\}$ where X is the treatment and Y is the outcome.[4] We need to characterize which structures are

3. The "trivial trek Z," composed of the single vertex Z and no edges, does not violate this statement, as Z is t-separated from itself by $(\emptyset; \{U_1, Z\})$ if we define this trivial trek to be $(\emptyset; Z)$.
4. For simplicity, we assume that there is an edge $X \to Y$ corresponding to a non-zero coefficient λ_{yx}. Otherwise, if W_i satisfies the Graphical Criteria for $X \to Y$ given $\mathbf{Z}$, then under unmeasured confounding between treatment and outcome we have that the lack of an edge $X \to Y$ implies $W_i \perp\!\!\!\perp Y \mid \mathbf{Z}$ and $W_i \not\perp\!\!\!\perp Y \mid \mathbf{Z} \cup \{X\}$. This corresponds to the collider orientation rule of the FCI algorithm (Spirtes et al., 2000) that implies $\lambda_{yx} = 0$. If on top of the lack of edge $X \to Y$ we have that there is no unblocked unmeasured confounding between X and Y, then $X \perp\!\!\!\perp Y \mid \mathbf{Z}$, which again by faithfulness will allow us to infer $\lambda_{yx} = 0$ (Spirtes et al., 2000). Algorithm 2 is assumed to be invoked only if the FCI algorithm and the method of Entner et al. (2012) do not provide any results.

compatible with the Trek Separation Theorem and, among those, which satisfy the Graphical Criteria. We start with the following observation.

Proposition 5 *In Line 9 of Algorithm 2, under rank-faithfulness the conditional tetrad constraint is equivalent to*

$$rank(\Sigma_{\mathbf{AB}}) \leq |\mathbf{Z}| + 1, \tag{4}$$

for $\mathbf{A} = \{W_i, W_j\} \cup \mathbf{Z}$ and $\mathbf{B} = \{X, Y\} \cup \mathbf{Z}$.

Proof of Proposition 5. As in Example 2, the determinant $|\Sigma_{\mathbf{AB}}| = |\Sigma_{\mathbf{ZZ}}|(\sigma_{w_ix.\mathbf{Z}}\sigma_{w_jy.\mathbf{Z}} - \sigma_{w_iy.\mathbf{Z}}\sigma_{w_jx.\mathbf{Z}})$ is zero if and only if the conditional tetrad constraint $\sigma_{w_ix.\mathbf{Z}}\sigma_{w_jy.\mathbf{Z}} = \sigma_{w_iy.\mathbf{Z}}\sigma_{w_jx.\mathbf{Z}}$ holds, as $|\Sigma_{\mathbf{ZZ}}| > 0$ by rank-faithfulness. Hence, $\Sigma_{\mathbf{AB}}$ is not full rank, that is, its rank is strictly less than $\min(|\mathbf{A}|, |\mathbf{B}|) = |\mathbf{Z}| + 2$. □

If the constraint holds, by Theorem 3 it follows that there is some pair $(\mathbf{C_A}; \mathbf{C_B})$ that t-separates $\{W_i, W_j\} \cup \mathbf{Z}$ from $\{X, Y\} \cup \mathbf{Z}$. It is required that $\mathbf{Z} \subseteq \mathbf{C_A} \cup \mathbf{C_B}$, since $\mathbf{Z}$ is contained in both $\mathbf{A}$ and $\mathbf{B}$. This implies the lower bound $|\mathbf{C_A}| + |\mathbf{C_B}| \geq |\mathbf{Z}|$ so that at most one element of $\mathbf{Z}$ belongs to $\mathbf{C_A} \cap \mathbf{C_B}$, and at most one element not in $\mathbf{Z}$ can belong to $\mathbf{C_A} \cup \mathbf{C_B}$. The main result of this section is a characterization of the ways this may happen such that we fail to discover the correct causal effect.

Consider the case where there exists a trek T, containing no element of $\mathbf{Z}$, which is between some $W \in \{W_i, W_j\}$ and $\{X, Y\}$. $\mathbf{C_A} \cup \mathbf{C_B}$ must contain an element of T for the t-separation to hold. Since by (4) and Theorem 3 we have that $|\mathbf{C_A}| + |\mathbf{C_B}| \leq |\mathbf{Z}| + 1$, then we can use exactly one (possibly latent) element out of $\mathbf{Z}$ but in $\mathbf{C_A} \cup \mathbf{C_B}$ to explain this t-separation. Moreover, since $\mathbf{Z} \subseteq \mathbf{C_A} \cup \mathbf{C_B}$, no vertex can appear in both $\mathbf{C_A}$ and $\mathbf{C_B}$, or otherwise $|\mathbf{C_A}| + |\mathbf{C_B}| > |\mathbf{Z}| + 1$. That is, the existence of T implies $\mathbf{C_A} \cap \mathbf{C_B} = \emptyset$.

We call any vertex in T satisfying the definition below a *conditional choke point*. A conditional choke point can be a "left" or "right" choke point:

Definition 6 (left conditional choke point) *In a causal DAG $\mathcal{G}$ with vertex set $\mathbf{V}$, V_0 is a* left conditional choke point *for pairs $\{W_i, W_j\} \times \{X, Y\}$ given $\mathbf{Z}$, where $\{V_0, W_i, W_j, X, Y\} \cup \mathbf{Z} \subseteq \mathbf{V}$, if*

1. *some $(\mathbf{C_A}; \mathbf{C_B})$ t-separates $\{W_i, W_j\}$ from $\{X, Y\}$ in $\mathcal{G}$, such that $V_0 \in \mathbf{C_A} \backslash \{\mathbf{Z} \cup \mathbf{C_B}\}$, $\mathbf{C_A} \cup \mathbf{C_B} = \mathbf{Z} \cup \{V_0\}$, $\mathbf{C_A} \cap \mathbf{C_B} = \emptyset$, and*
2. *for any trek $(P_1; P_2)$ from a vertex in $\{W_i, W_j\}$ to a vertex in $\{X, Y\}$ in $\mathcal{G}$ that contains no member of $\mathbf{Z}$, P_1 contains V_0, and*
3. *there exists at least one such a trek.*

A "right conditional choke point" follows an equivalent definition with respect to $\mathbf{C_B}$ and P_2. The literature has characterizations of *unconditional* choke points (Shafer et al., 1993; Sullivant et al., 2010), relating them to unconditional tetrad constraints. To the best of our knowledge, this is the first time that conditional choke points are explicitly defined and used. The reason for the name "choke point" is explained later in this section.

A second category of choke point, now for vertices in $\mathbf{Z}$, will be defined as follows:

Definition 7 (collider choke point) *In a causal DAG $\mathcal{G}$ with vertex set $\mathbf{V}$, Z_0 is a* collider choke point *for pairs $\{W_i, W_j\} \times \{X, Y\}$ given $\mathbf{Z}$, where $\{W_i, W_j, X, Y\} \cup \mathbf{Z} \subseteq \mathbf{V}$, if*

1. *some $(\mathbf{C_A}; \mathbf{C_B})$ t-separates $\{W_i, W_j\}$ from $\{X, Y\}$ in $\mathcal{G}$, such that $Z_0 \in \mathbf{Z}$, $\mathbf{C_A} \cup \mathbf{C_B} = \mathbf{Z}$, $\mathbf{C_A} \cap \mathbf{C_B} = Z_0$, and*
2. *for any (possibly non-simple) path P from a vertex in $\{W_i, W_j\}$ to a vertex in $\{X, Y\}$ in $\mathcal{G}$ that contains a collider in $\mathbf{Z}$, P contains Z_0, and*
3. *there exists at least one such a path.*

The same vertex can be a conditional and a collider choke point depending on the conditioning set. For instance, Z_0 is a (right) conditional choke point in Figure 3 for each $\{W_i, W_j\} \times \{X, Y\}$ given the empty set, since $(\emptyset; Z_0)$ t-separates each pair in $\{W_1, \dots W_p\} \times \{X, Y\}$ and it lies on all of such treks. The tetrad constraint $\sigma_{w_i x}\sigma_{w_j y} = \sigma_{w_i y}\sigma_{w_j x}$ holds, and we can recover the correct causal effect from $\sigma_{w_i y}/\sigma_{w_i x}$ as the Graphical Criteria is satisfied. On the other hand, if we take the conditioning set to be Z_0 itself, then the conditional tetrad constraint $\sigma_{w_i x.z_0}\sigma_{w_j y.z_0} = \sigma_{w_i y.z_0}\sigma_{w_j x.z_0}$ *also* holds, as well as the corresponding t-separation. However, there is no conditional choke point as all treks in $\{W_i, W_j\} \times \{X, Y\}$ contain an element of the conditioning set, Z_0 itself. Z_0 is now a collider choke point, as it lies as a collider on all active paths between $\{W_1, \dots, W_p\}$ and $\{X, Y\}$. Notice that the Graphical Criteria does not hold as W_1 is not d-separated from Y given Z_0 in the graph that modifies Figure 3 by removing edge $X \to Y$.

Choke points do not need to be unique, in the same way that t-separation between sets may be implied by more than one conditioning pair. For instance, both U_1 and U_2 are right conditional choke points in Figure 2(c), corresponding to the different choices of $\mathbf{C_B}$ discussed in Example 3.

We are now ready to characterize when Algorithm 2 may provide an incorrect value for the target causal effect. The proof of this result is given in Appendix B.

Theorem 8 *Let $\mathcal{C}$ be the outcome of Algorithm 2 as applied to a distribution which is rank-faithful to some causal DAG $\mathcal{G}$. Let $\beta_i \in \mathcal{C}$, and let $(W_i, W_j, \mathbf{Z})$ be the respective triplet that generated β_i by passing the check in Line 9. Then $(W_i, W_j, \mathbf{Z})$ fails to satisfy the Graphical Criteria if and only if exactly one of these is true:*

(i) there is at least one trek between $\{W_i, W_j\}$ and Y that do not contain any member of $\mathbf{Z} \cup \{X\}$, and all of those treks intersect at the same conditional choke point V_0;

(ii) there is at least one (possibly non-simple) active path between $\{W_i, W_j\}$ and Y that contains a collider in $\mathbf{Z}$ and does not contain X, and all of those paths intersect at the same collider choke point Z_0.

Moreover, it is not possible that for $(W_i, W_j, \mathbf{Z})$ both a conditional choke point and a collider choke point exist in $\mathcal{G}$ regardless of whether the Graphical Criteria hold.

That is, if a candidate triplet $(W_i, W_j, \mathbf{Z})$ does not correspond to valid (conditional) instrumental variables, then *all offending paths that avoid X and connect the candidate instruments to Y must pass through a single variable $V \in \mathbf{V} \backslash \{X\}$*. If V is in the conditioning set $\mathbf{Z}$, then this variable must be a collider in all of the offending paths. For instance, U is a conditional choke point with respect to $\mathbf{Z} = \emptyset$ in Figure 2(d) and Z_0 is a collider choke point for $\mathbf{Z} = \{Z_0\}$ in Figure 3. The difference between valid and invalid instruments that imply tetrad constraints is that, in the former, X is a conditional choke point. Although this is not used by Algorithm 2, if conditional tetrad constraints hold for all pairs in some set $\mathbf{W} \equiv \{W_1, \dots, W_p\}$ given $\mathbf{Z}$, it follows that all offending paths from $\mathbf{W}$ to Y go through a common variable V. As a matter of fact, the name "choke point," originally coined by Spirtes et al. (2000) for the special case $\mathbf{Z} = \emptyset$, comes from this idea of having all connecting paths (treks only, in the original case) between two sets passing through a single variable.

We can formalize a sufficient assumption that guarantees that Algorithm 2 does not return a incorrect causal effect as follows:

Corollary 9 *Let $\mathcal{G}$ be a causal DAG with observable vertices $\mathbf{O}$, treatment vertex X and outcome vertex Y. For all $\{W_i, W_j\} \cup \mathbf{Z} \subset \mathbf{O}$ where a tetrad constraint $\sigma_{w_i x.\mathbf{Z}}\sigma_{w_j y.\mathbf{Z}} = \sigma_{w_i y.\mathbf{Z}}\sigma_{w_j x.\mathbf{Z}}$ holds, if X is a corresponding conditional choke point in $\mathcal{G}$, then the output set of Algorithm* TETRAD-IV$_\infty$ *will not contain an incorrect causal effect.*

The result above allows for an empty output set $\mathcal{C}$, which will happen if no valid pair of instrumental variables exists. However, even if all elements of $\mathcal{C}$ are equal to some $\lambda_{\mathcal{C}}$, it does not mean that $\lambda_{\mathcal{C}} = \lambda_{yx}$ in case the assumptions of Corollary 9 fail. It is impossible to distinguish, based on tetrad relations, the graph in Figure 2(b) from the one in 2(d). In the former, X can take the role of a conditional choke point explaining the tetrad constraint. In the latter, X cannot take this role because of the active trek $W_1 \to U \to Y$, but vertex U can, resulting on a tetrad constraint but on invalid IVs. In general, we will have problems in a graph which includes an intermediate latent variable that is a common ancestor of X and Y and blocks all treks from W_1 and W_2 to X. As another example, vertex U_1 takes the role of a conditional choke point given Z in Figure 2(c), while X cannot take this role because of the active trek $W_1 \leftarrow U_1 \to U_2 \to Y$.

One might argue that the assumptions of Corollary 9 are not too restrictive, as failing them require that all offending paths intersect at an unique vertex. If two disjoint paths from W_1 to Y via latent variables exist, for instance, then no tetrad constraint will hold and at worst we would be conservative and not return any causal effect. However, we argue that these assumptions should not be taken lightly. The whole motivation for including a conditioning set $\mathbf{Z}$ is to block paths such that a "desirable" choke point emerges: treatment X. The assumptions that make the presence of an instrumental variable possible are the same that make the presence of undesirable choke points plausible.

Instead, the output of IV-TETRAD$_\infty$ is better interpreted as an *equivalence class of causal effects*, differing from the usual output of causal discovery algorithms such as the PC algorithm (Spirtes et al., 2000): the PC algorithm returns a set of graphs of which some graphical features might be the same (for instance, they all share the same directed edge),

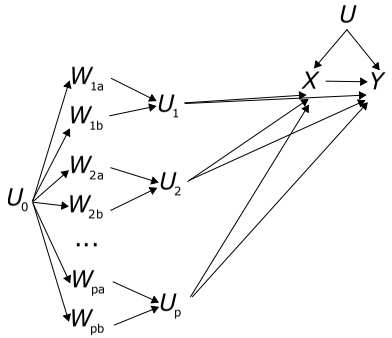


Figure 4: Vertices named $U_\star$ are latent variables, with X being the treatment and Y being the outcome. When given as input to algorithms for finding locally covariance equivalent solutions, the result will include a (potentially different) causal effect candidate for each of the pairwise groupings $\{W_{ia}, W_{ib}\}$, $i = 1, 2, \ldots, p$, where $p = |\mathbf{O}|/2$.

and where some causal effects might be identified based on the common features. That is, the PC algorithm either gives (in the limit of infinite data) the correct answer, or it answers "I don't know." The same is true of the method introduced by Entner et al. (2012).

In contrast, IV-TETRAD$_\infty$ may return an empty set of solutions, but at the same time *all* solutions in a non-empty output set $\mathcal{C}$ may be wrong in case the assumptions of Corollary 9 do not hold. The need to report an equivalence class is illustrated by the following result.

Proposition 10 *There exist problems where the output of* IV-TETRAD$_\infty$ *will contain* $\mathcal{O}(|\mathbf{O}|)$ *different elements, none of which is the correct solution* λ_{yx}.

Proof of Proposition 10. This is illustrated by Figure 4: given $\mathbf{O} = \{W_{1a}, W_{1b}, W_{2a}, \ldots, W_{pb}\}$, $p = |\mathbf{O}|/2$, the structure shown in the Figure will generate solutions for all pairs $\{W_{ia}, W_{ib}\} \times \{X, Y\}$ with corresponding conditioning set $\mathbf{Z}_i \equiv \mathbf{O} \backslash \{W_{ia}, W_{ib}\}$. For parameters generated randomly and independently, it is clear each resulting λ_i will be unique. The corresponding choke point in each case is U_i, and none of the returned causal effects will in general be correct. □

For practical purposes, one should be willing to to assume that *at least one* solution exists, that is, there exists at least one pair of IVs conditioned on the same set $\mathbf{Z}$. Under this condition, we obtain the following guarantee:

Corollary 11 *Let* $\mathcal{C}$ *be the output of* IV-TETRAD$_\infty$ *as applied to a distribution which is rank-faithful to a causal DAG* $\mathcal{G}$ *with observable vertices* $\mathbf{O} \cup \{X, Y\}$ *and target edge* $X \to Y$. *Let* λ_{yx} *be the corresponding true causal effect. Assuming that there exists at least one set* $\{W_i, W_j\} \cup \mathbf{Z} \subseteq \mathbf{O}$ *where* $\{W_i, W_j\}$ *satisfies the Graphical Criteria conditioned on* $\mathbf{Z}$, *then the minimum and maximum elements of* $\mathcal{C}$ *will provide lower and upper bounds on* λ_{yx}.

The proof of this Corollary is immediate from the previous discussions. Summaries other than lower bounds and upper bounds are possible: for instance, if the maximum of $\mathcal{C}$ is much larger than the second largest element, and background knowledge says that such a large value is unlikely, we may opt to exclude it or to reweight elements of $\mathcal{C}$ according to some prior. More of that is discussed in Section 5.

Assuming that one solution exists is a much weaker assumption than the one in IV-BY-MAJORITY / sisVIVE. However, it does require rank-faithfulness and it can only provide bounds and other summaries of an equivalence class. If in the application of the algorithm we have that all elements of $\mathcal{C}$ are the same, then from the assumption that there is at least one valid tuple it follows that we found the true causal effect. A discussion of what further assumptions are necessary in practical learning is postponed to Section 4.

The *soundness* of Algorithm IV-TETRAD$_\infty$ can be defined in terms of the bounds given by Corollary 11, or other summaries of interest, provably capturing the true λ_{yx}. Soundness is however of limited value if the equivalence class is too large or trivial (that is, if there is a lack of *specificity* in the constraints chosen to distinguish possible models), or if the algorithm cannot find all elements of the equivalence class (a lack of *completeness*). While soundness and completeness are a matter of mathematical proof, specificity must be argued in the context of the scientific plausibility of assumptions and practical matters of computational and statistical tractability.

The Markov equivalence class used by various causal discovery algorithms has a natural specificity, as independence constraints are nonparametric restrictions that require no further assumptions about the probability distribution underlying the causal model. The motivation for the specificity of our method is less clear and discussed by first formally defining the following concept:

Definition 12 (locally covariance equivalent causal effect) *We say that two ratios* $\beta_{i(\mathbf{z})} \equiv \sigma_{w_i y \cdot \mathbf{Z}} / \sigma_{w_i x \cdot \mathbf{Z}}$ *and* $\beta_{j(\mathbf{z}')} \equiv \sigma_{w_j y \cdot \mathbf{Z}'} / \sigma_{w_j x \cdot \mathbf{Z}'}$ *are locally covariance equivalent causal effects if there is some pair* $\{W_k, W_l\}$, $W_i \neq W_k$, $W_j \neq W_l$, *where* $\beta_{i(\mathbf{z})} = \beta_{k(\mathbf{z})}$ *and* $\beta_{j(\mathbf{z}')} = \beta_{l(\mathbf{z}')}$.

IV-TETRAD$_\infty$ is sound and complete in the sense that by construction it returns all and only locally covariance equivalent causal effects. The effects are "local" in the sense that it does not use covariance information from the joint distribution of $\{W_i, W_j, W_k, W_l\} \cup \mathbf{Z} \cup \mathbf{Z}'$ beyond what is found in the marginals $\{W_i, W_k\} \cup \mathbf{Z}$ and $\{W_j, W_l\} \cup \mathbf{Z}'$. This is motivated by the following desiderata:

Desiderata 1 (Local Covariance Discovery) *We say that an algorithm obeys the* local covariance discovery *desiderata if it follows these conditions:*

1. *only substructures where two or more variables follow the Graphical Criteria conditioned on a common set* $\mathbf{Z} \subset \mathbf{O}$ *can be used to select conditional instrumental variables;*

2. *only covariance information is used.*

Algorithm 3 IV-TETRAD$^{+}_{\infty}$

1: **Input:** set of zero-mean random variables $\mathbf{O} \cup \{X, Y\}$ with known joint distribution
2: **Output:** $\mathcal{C}$, a set of candidate differential causal effects of X on Y
3: Initialize $\mathcal{C} \leftarrow \emptyset$
4: **for** each pair $\{W_i, W_j\} \subseteq \mathbf{O}$ **do**
5: **for** every set $\mathbf{Z} \subseteq \mathbf{O} \backslash \{W_i, W_j\}$ **do**
6: **if** $\sigma_{w_i x.\mathbf{z}} = 0$ **or** $\sigma_{w_j x.\mathbf{z}} = 0$ **then**
7: **next**
8: **end if**
9: **if** $\sigma_{w_i x.\mathbf{z}} \sigma_{w_j y.\mathbf{z}} \neq \sigma_{w_i y.\mathbf{z}} \sigma_{w_j x.\mathbf{z}}$ **then**
10: **next**
11: **end if**
12: $r_{W_i} \leftarrow \text{resProj}(W_i, \mathbf{Z} \cup \{W_j\})$
13: $r_{W_j} \leftarrow \text{resProj}(W_j, \mathbf{Z} \cup \{W_i\})$
14: $r_Y \leftarrow \text{resProj}(Y, \mathbf{Z} \cup \{W_i, W_j\})$
15: **if** $r_{W_i} \perp\!\!\!\perp r_Y$ **and** $r_{W_j} \perp\!\!\!\perp r_Y$ **then**
16: $\mathcal{C} \leftarrow \mathcal{C} \cup \{\sigma_{w_i y.\mathbf{z}} / \sigma_{w_i x.\mathbf{z}}\}$
17: **end if**
18: **end for**
19: **end for**
20: **return** $\mathcal{C}$

The requirement for covariance information only allows us to use only second moments, which is relatively statistically efficient. This will be relaxed in the next section. The need for "two or more" variables in condition 1 is necessary, since the Graphical Criteria provides no testable implications for a single instrument, as discussed in the introduction. It is also desirable, as it considers variables that are jointly instrumental by conditioning on the same set $\mathbf{Z}$. To drop this criterion is to require a search space where different conditioning sets are required for each IV, resulting in an algorithm that effectively reconstructs a "non-local" graphical substructure that will require the identification of structural coefficients other than λ_{yx}. This is what is done, for instance, in one of the first published extensions of the Graphical Criteria (Brito and Pearl, 2002). However, identification of causal effects using given IVs is an easier problem than discovering IVs from limited structure information. Inferring structure from data is statistically challenging, and we claim that restricting a method to the desiderata above provides a baseline where stronger assumptions, and computational and statistical costs, can be exchanged for greater specificity of the solutions returned.

3.3 Exploiting Non-Gaussianity Assumptions

This Section introduces a variant of IV-TETRAD$_{\infty}$ that can verify the validity of item 2a of the Graphical Criteria, as discussed in Section 2.3. We assume our causal model is a *LiNGAM model*, a linear structural equation model with independent, non-Gaussian error terms, which may include latent variables (Shimizu et al., 2006). We call this Algorithm IV-TETRAD$^{+}_{\infty}$, which is shown in Algorithm 3. We will also discuss the difficulties posed by 2b even under the assumption of non-Gaussianity. The motivation is to reduce the size of the

equivalence class by trading it off with the addition of more assumptions (non-Gaussianity) and a weakening of completeness (some solutions might be missed, as we will characterize).

In Algorithm 3, function $\text{resProj}(V, \mathbf{S})$ is a function that returns the residual of the least-squares projection of V into the set of random variables $\mathbf{S}$ rearranged as a column vector,

$$\text{resProj}(V, \mathbf{S}) \equiv V - \mathbb{E}[\mathbf{S}^{\intercal} V]\mathbb{E}[\mathbf{S}\mathbf{S}^{\intercal}]^{-1}\mathbf{S},$$

which is itself a random variable. We assume that V and $\mathbf{S}$ have zero mean.

3.3.1 Main Result

The validity of Algorithm 3 holds "almost everywhere," in the sense it holds for all but a (Lebesgue) measure zero subset of the set of possible structural coefficients λ_{ij} and variances[5] ω_i of each respective error term e_i, $\theta_{\mathcal{G}} \equiv \{\lambda_{ij} \mid V_j \in par_{\mathcal{G}}(i)\} \cup \{\omega_i\}$. The motivation for this concept is analogous to the different variations of faithfulness, see the discussion on generic identifiability by Foygel et al. (2011) and Sullivant et al. (2010).

The main result of this section is the following:

Theorem 13 *Let $\mathbf{O} \cup \{Y\}$ be a subset of the variables in a zero-mean LiNGAM model with graph $\mathcal{G}$, where Y has no descendants in $\mathcal{G}$. Let $O_i \in \mathbf{O}$ and $\mathbf{Z} \subseteq \mathbf{O} \backslash \{O_i\}$, and let $r_i \equiv O_i - \mathbf{a}^{\intercal}\mathbf{Z}$ be the residual of the least-squares regression of O_i on $\mathbf{Z}$, with $\mathbf{a}$ being the corresponding least-squares coefficients. Analogously, let $r_y \equiv Y - b_i O_i - \mathbf{b}^{\intercal}\mathbf{Z}$ be the residual of the corresponding least-squares regression. Then, almost everywhere, $r_i \perp\!\!\!\perp r_y$ if and only if there are no active non-directed paths between O_i and Y conditioned on $\mathbf{Z}$*

The proof is given in Appendix B. This generalizes the main result of Entner et al. (2012), which considers the case where $\mathbf{Z}$ contains no descendant of O_i.

Even under non-Gaussianity, we will see that it is still possible for conditional choke points in the true graph to add incorrect causal effect estimates to the output of Algorithm 3. However, only the following special case can occur:

Definition 14 (downstream conditional choke point) *In a causal DAG $\mathcal{G}$ with vertex set $\mathbf{V}$, V_0 is a* downstream conditional choke point *for pairs $\{W_i, W_j\} \times \{X, Y\}$ given $\mathbf{Z}$, where $\{V_0, W_i, W_j, X, Y\} \cup \mathbf{Z} \subseteq \mathbf{V}$, if V_0 is a conditional choke point which is a descendant of $\{W_i, W_j\}$ in $\mathcal{G}$, an ancestor of both X and Y, such that no active back-door paths between V_0 and $\{W_i, W_j\}$ exist in $\mathcal{G}$.*

This implies that collider choke points are now ruled out, as shown in the following result proved in Appendix B:

Theorem 15 *Let $\mathcal{C}$ be the outcome of Algorithm 3 as applied to a distribution which is rank-faithful to some causal DAG $\mathcal{G}$. Let $\beta_i \in \mathcal{C}$, and let $(W_i, W_j, \mathbf{Z})$ be the respective triplet that generated β_i by passing the check in Line 15. Then $(W_i, W_j, \mathbf{Z})$ fails to satisfy the Graphical Criteria if and only if: there is at least one directed path between $\{W_i, W_j\}$ and Y that does not contain any member of $\mathbf{Z} \cup \{X\}$, and all of those directed paths intersect at the same downstream conditional choke point V_0.*

5. As error terms are non-Gaussian, other parameters are expected to exist, but our results are agnostic with respect to them.

By further restricting the set of possible choke points, we refine the explanation of possible disparities obtained by Algorithm 3 compared to Algorithm 2. By knowing that there is a *single* choke point per pair which is *unconfounded* with the candidate IVs, *and* which lies on all unblocked directed paths from $\{W_i, W_j\}$ to Y, the output of the algorithm, combined with background knowledge, can be analyzed to exploit this piece of information. For example, any downstream conditional choke point must be a latent confounder of X and Y, which is itself caused by $\{W_i, W_j\}$: efforts might be directed to try to interpret and measure such confounders as possible explanations for the disparities in the output of Algorithm 3.

Examples of advantages and remaining limitations of IV-TETRAD$^+_\infty$ are as follows.

Example 4 *Consider Figure 2(c) again. As discussed before, t-separation between* $\{W_1, W_2, Z\}$ *and* $\{X, Y, Z\}$ *holds given* $(\emptyset; \{U_1, Z\})$*. However, there is an active path between* W_1 *and* Y *that does not include* X*. Algorithm 2 will include the candidate causal effect corresponding the invalid IV triplet* (W_1, W_2, Z) *in its output. In contrast, Algorithm 3 will detect the active back-door path* $W_1 \leftarrow U_1 \rightarrow U_2 \rightarrow Y$ *by the residual test of Line 15 and reject* (W_1, W_2, Z)*.* □

Example 5 *Consider Figure 2(d) again. T-separation between* $\{W_1, W_2\}$ *and* $\{X, Y\}$ *holds given* $(\emptyset; U)$*. However, there are active paths between* $\{W_1, W_2\}$ *and* Y *that do not include* X *and intersect at the downstream conditional choke point* U*. Algorithm 3 will not detect these offending paths and will incorrectly include a value different from* λ_{yx} *in its output set.* □

Example 6 *Consider Figure 3 again. T-separation between* $\{W_1, W_2, Z_0\}$ *and* $\{X, Y, Z_0\}$ *holds given* $(\emptyset; \{Z_0\})$*. Algorithm 2 will include the candidate causal effect corresponding the invalid IV triplet* (W_i, W_j, Z_0) *in its output. In contrast, Algorithm 3 will detect the active non-directed paths* $W_i \rightarrow Z_0 \leftarrow U \rightarrow Y$ *by the residual test of Line 15.* □

Example 7 *Figure 5 shows a more complex example where a latent variable* U *invalidates one particular set of candidate IVs (*$\{V_1, V_2\}$ *given* V_6*). Meanwhile, in the same graph, conditioning on* V_2 *is essential to validate another set of candidate IVs. These two cases cannot be distinguished by Algorithm 2 but can be by Algorithm 3.* □

Notice that Lines 12-14 in IV-TETRAD$^+_\infty$ include W_j in the conditioning set for W_i and vice-versa. Including W_j in the conditioning set for W_i is acceptable, given that if W_i and W_j each satisfy the Graphical Criteria with respect to $X \rightarrow Y$ given $\mathbf{Z}$, then W_i satisfies the Graphical Criteria with respect to $X \rightarrow Y$ given $\mathbf{Z} \cup \{W_j\}$:

Proposition 16 *If* W_i *and* W_j *each satisfy the Graphical Criteria with respect to* $X \rightarrow Y$ *given* $\mathbf{Z}$*, then* W_i *satisfies the Graphical Criteria with respect to* $X \rightarrow Y$ *given* $\mathbf{Z} \cup \{W_j\}$*.*

Proof of Proposition 16. Given $\mathbf{Z}$, both W_i and W_j are d-separated from Y in the graph where edge $X \rightarrow Y$ has been removed. If $\mathbf{Z} \cup \{W_j\}$ d-connected W_i to Y in this mutilated graph, then there should be an active path from W_i to Y where W_j is a collider. But this would imply an active path from W_j to Y given $\mathbf{Z}$, contrary to our assumption. □

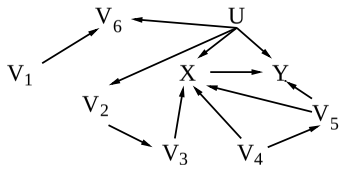


Figure 5: In this system, $\mathbf{W} \equiv \{V_1, V_2\}$ satisfies the conditional tetrad constraints $\{V_1, V_2\} \times \{X, Y\}$ given $\mathbf{Z} \equiv \{V_3, V_6\}$. The same is true for $\mathbf{W}' \equiv \{V_3, V_4\}$ for $\mathbf{Z}' \equiv \{V_2, V_5\}$. Only $\mathbf{W}'$ represents a set of (conditionally) valid IVs. The non-Gaussianity criteria of IV-TETRAD$^+_\infty$ helps to rule out the former.

The motivation for including W_j in the conditioning set of W_i is that, if W_i and W_j share a hidden common cause that is independent of X and Y and W_j is an unblocked ancestor of X, then not conditioning on W_j will induce an active back-door between W_i and X that will make the algorithm unnecessarily remove $\sigma_{w_i y.\mathbf{z}.}/\sigma_{w_i x.\mathbf{z}.}$ from the output.

3.3.2 Implications to Completeness

The following result is a direct consequence of the assumptions and the exhaustive search done by IV-TETRAD$^+_\infty$. The proof of this result follows immediately from Theorem 13 and the existence of edge $X \rightarrow Y$:

Proposition 17 *If there is a pair of observable variables* $\{W_i, W_j\}$ *which are IVs conditioned on some* $\mathbf{Z}$ *according to the Graphical Criteria, and some* $W \in \{W_i, W_j\}$ *has an active non-directed path with* X *given* $\mathbf{Z}$*, and* X *is a parent of* Y*, then Algorithm 3 will not include* $\{W_i, W_j\}$ *in its output.*

The implication of this is as follows. A positive property of IV-TETRAD$^+_\infty$ is that it will exclude from the equivalence class some candidate IVs that do not obey the Graphical Criteria but which are returned by IV-TETRAD$_\infty$. However, IV-TETRAD$_\infty$ was complete in the sense of always including in its output the correct causal effect, under the assumption that at least one pair of valid IVs existed given a common conditioning set. This assumption is not strong enough in the case of IV-TETRAD$^+_\infty$. As an example, suppose that there exists only one pair $\{W_i, W_j\}$ of valid IVs, and its conditioning set $\mathbf{Z}$ does not block all back-doors into X. IV-TETRAD$^+_\infty$ may return a non-empty set but it will not include the correct causal effect.

Example 8 W *is a valid IV in the model with paths* $W \rightarrow X \rightarrow Y$*,* $W \leftarrow U_1 \rightarrow X \leftarrow U_2 \rightarrow Y$*, but* W *will be discarded due to the back-door path between* W *and* Y *that is unblocked by not conditioning on* X*.* □

As a result, the following is a modification of Corollary 11 to the alternative algorithm IV-TETRAD$^+_\infty$. The proof follows immediately from the previous results.

Algorithm 4 IV-TETRAD$^+$

1: **Input:** $\mathcal{D}$, a sample from the joint distribution of random variables $\mathbf{O} \cup \{X, Y\}$; K, number of instruments to consider
2: **Output:** two sets of estimates of the differential causal effect of X on Y
3: Remove from $\mathbf{O}$ any element that is not in the Markov blanket of either X or Y, and also any element which is marginally uncorrelated with X or Y
4: Let $\hat{\lambda}_i \leftarrow \hat{\sigma}_{w_i y.\mathbf{O}\setminus i}/\hat{\sigma}_{w_i x.\mathbf{O}\setminus i}$ for each $W_i \in \mathbf{O}$
5: Initialize $\mathcal{C}$ and $\mathcal{C}_R$ as empty sets, and $\mathbf{W}$ as $\mathbf{O}$
6: **while** $|\mathbf{W}| \geq K$ **do**
7: Sort $\mathbf{W}$ as $W^{(1)}, \ldots, W^{(|\mathbf{W}|)}$ according to the corresponding $\{\hat{\lambda}\}$
8: Find $j \in \{1, 2, \ldots, |\mathbf{W}|\}$ that minimizes ScoreTetrads$(W^{(j:j+K-1)}, \mathbf{O}, X, Y, \mathcal{D})$
9: $\mathbf{W}_j \leftarrow$ ExpandTetrads$(j, K, \mathbf{W}, \mathbf{O}, X, Y, \mathcal{D})$
10: **if** TestTetrads$(\mathbf{W}_j, \mathbf{O}, X, Y, \mathcal{D})$ **then**
11: $\hat{\lambda}_{TSLS} \leftarrow$ TSLS$(\mathbf{W}_j, \mathbf{O}\setminus\mathbf{W}_j, X, Y, \mathcal{D})$
12: **if** TestResiduals$(\mathbf{W}_j, \mathbf{O}, X, Y, \mathcal{D})$ **then**
13: $\mathcal{C} \leftarrow \mathcal{C} \cup \hat{\lambda}_{TSLS}$
14: **else**
15: $\mathcal{C}_R \leftarrow \mathcal{C}_R \cup \hat{\lambda}_{TSLS}$
16: **end if**
17: **end if**
18: $\mathbf{W} \leftarrow \mathbf{W}\setminus\mathbf{W}_j$
19: **end while**
20: **return** $(\mathcal{C}, \mathcal{C}_R)$

Corollary 18 *Let $\mathcal{C}$ be the output of* IV-TETRAD$^+_\infty$ *as applied to a distribution which is rank-faithful to a causal DAG $\mathcal{G}$ with observable vertices $\mathbf{O}\cup\{X, Y\}$ and target edge $X \to Y$. Let λ_{yx} be the corresponding true causal effect. Assuming that there exists at least one set $\{W_i, W_j\} \cup \mathbf{Z} \subseteq \mathbf{O}$ where $\{W_i, W_j\}$ satisfies the Graphical Criteria conditioned on $\mathbf{Z}$, and that no non-directed path between $\{W_i, W_j\}$ and X is active given $\mathbf{Z}$, then the minimum and maximum elements of $\mathcal{C}$ will provide lower and upper bounds on λ_{yx}.*

The trade-off between IV-TETRAD$_\infty$ and IV-TETRAD$^+_\infty$ can be implemented in different ways within a practical algorithm, as we shall see next.

4. Practical Learning from Data

Testing tetrad constraints from data is difficult in practice, particularly without assuming Gaussianity and under conditioning. In order to search for candidate IVs, we will first adopt the stronger (but falsifiable) assumption that, if any set of valid conditional IVs exist, then there is at least one such set $\mathbf{W}$ of size K, which remains valid by conditioning on $\mathbf{O}\setminus\mathbf{W}$. One motivation is to avoid a combinatorial search for conditioning sets, while still having the option of rejecting a solution if confounding or collider biases remain by doing a test at the end. The other motivation is the statistical unreliability of candidate sets of small size: in a large system where the treatment may have many observed causes, instruments will in

general be weakly associated with the treatment, leading to high variance estimates. This issue of "weak instruments" is pervasive in real problems and one mitigation is to consider instrument sets of a minimum size.

The algorithm, IV-TETRAD$^+$, is shown in Algorithm 4. Its implementation diverges from a direct adaptation of IV-TETRAD$^+_\infty$ where hypothesis testing would be used to to check the required constraints. This is because we found that, in simulations, directly testing conditional tetrad constraints is not a very robust strategy. Instead, IV-TETRAD$^+$ includes intermediate steps along the lines of a "score-based approach" (Cooper, 1999) for causal discovery. In the algorithm, we search for a window of K estimated causal effects $\{\hat{\lambda}\}$ within a sorted list, scoring this set according to a particular measure of variability of its elements. Ideally, a perfect score of 0 is obtained when all $\hat{\lambda}$ in this window are equal. *This corresponds to an alternative view of Line 9 of* IV-TETRAD$^+_\infty$*, where we find a set of K variables W_i with corresponding ratios $\hat{\sigma}_{w_i y.\mathbf{O}\setminus i}/\hat{\sigma}_{w_i x.\mathbf{O}\setminus i}$ as close as possible to each other, in a formal sense defined below.*

The algorithm makes use of TSLS$(\mathbf{W}, \mathbf{Z}, X, Y, \mathcal{D})$, the two-stage least squares (TSLS) estimator for some set of conditional instruments $\mathbf{W}$ and conditioning set $\mathbf{Z}$, given a data set $\mathcal{D}$. The TSLS estimator[6] $\hat{\lambda}_{TSLS}$ is motivated by the redundant ways of estimating λ_{yx} if we have more than one instrument. For instance, we could have used either $\sigma_{w_i y.\mathbf{Z}}/\sigma_{w_i x.\mathbf{Z}}$ or $\sigma_{w_j y.\mathbf{Z}}/\sigma_{w_j x.\mathbf{Z}}$ in the previous algorithms without any difference. When finite sample covariance matrices are used instead of population covariance matrices, differences will appear. The TSLS is a classic way of combining all postulated instruments into a single estimator (Angrist and Pischke, 2009). This is particularly relevant in IV-TETRAD$^+$, where we use more than two instruments in each causal effect estimate inside function ScoreTetrads, as explained in the next paragraphs.

The output of the algorithm is a pair of two sets, $\mathcal{C}$ and $\mathcal{C}_R$, where the latter excludes from $\mathcal{C}$ all elements that correspond to each W_i detected to be connected to Y via an active non-directed path, as in IV-TETRAD$^+_\infty$.

4.1 Interpretation

The details of the algorithm are as follows:

- In Line 3, we remove variables from $\mathbf{O}$ that are considered to be conditionally independent of X or Y. *The deletion of variables outside the Markov blanket of X correspond to Line 6 in* IV-TETRAD$^+_\infty$. As discussed in the beginning of this section, we assume that if valid IVs exist, then $\mathbf{O}$ can be partitioned into instrument set $\mathbf{W}$ and conditioning set $\mathbf{Z} \equiv \mathbf{O}\setminus\mathbf{W}$. By Proposition 16, $W_i \in \mathbf{W}$ remains a valid IV if the conditioning set is $\mathbf{O}_i \equiv \mathbf{O}\setminus\{W_i\}$. We remove also variables which are statistically independent of Y given the remaining variables in $\mathbf{O}$. This is motivated by the instability of the ratio of estimates $\hat{\sigma}_{w_i y.\mathbf{Z}}/\hat{\sigma}_{w_i x.\mathbf{Z}}$ when both are close to zero. If W is d-connected to X and edge $X \to Y$ exists, then in theory we will not remove any valid IV by doing this test.

6. Assuming $\mathcal{D}$ is centered to have zero empirical mean, this is defined as follows: let $\mathbf{R}_W$, $\mathbf{R}_X$, $\mathbf{R}_Y$ denote the corresponding empirical residuals of the least-squares regression of $\mathbf{W}$, X and Y on $\mathbf{Z}$. The projection matrix $P_{\mathbf{W}}$ is defined as $\mathbf{R}_W(\mathbf{R}_W^\top\mathbf{R}_W)^{-1}\mathbf{R}_W^\top$. Let $\hat{\mathbf{R}}_X$ denote the predicted value $\hat{\mathbf{R}}_X \equiv P_{\mathbf{W}}\mathbf{R}_X$. The two-stage least squares coefficient is then given by the least-squares regression of $\mathbf{R}_Y$ on $\hat{\mathbf{R}}_X$, resulting in $\hat{\lambda}_{TSLS} \equiv (\mathbf{R}_X^\top P_{\mathbf{W}}\mathbf{R}_X)^{-1}\mathbf{R}_X^\top P_{\mathbf{W}}\mathbf{R}_Y$.

In our experiments in the next Section, Line 3 is implemented by running LARS (Efron et al., 2004), regressing separately X on $\mathbf{O}$ and Y on $\mathbf{O}$. We then remove from $\mathbf{O}$ any element that is given a coefficient of value zero in either regression.

Moreover, we also remove from $\mathbf{O}$ any element marginally uncorrelated with X or Y by a test of vanishing Spearman correlations at a 0.05 level;[7]

- In Line 4, each $\hat{\sigma}_{ab.\mathbf{z}}$ is an estimated partial covariance as given by the empirical covariance matrix derived from $\mathcal{D}$, and $\mathbf{O}_{\setminus i} \equiv \mathbf{O} \setminus \{W_i\}$.
- The loop that starts in Line 6 will search for a subset of $\mathbf{W}$ of size at least K that respects desirable tetrad and residual independence constraints. Every time a set of minimum size is found, it is removed from $\mathbf{W}$ and the process repeats;
- In Line 8, function ScoreTetrads($W^{(j:j+K-1)}, \mathbf{V}, X, Y, \mathcal{D}$) returns the median of the absolute differences between $\hat{\lambda}_{TSLS}$ and each $\hat{\lambda}^{(j)}, \dots \hat{\lambda}^{(j+K-1)}$, where $\hat{\lambda}_{TSLS} \equiv$ TSLS $(\{W^{(j)}, \dots, W^{(j+K-1)}\}, \mathbf{V} \setminus \{W^{(j)}, \dots, W^{(j+K-1)}\}, X, Y, \mathcal{D})$;
- In Line 9, function ExpandTetrads performs a greedy search to optimize ScoreTetrads by expanding the current set $\{W^{(j)}, \dots, W^{(j+K-1)}\}$ as either $\{W^{(j-1)}, \dots, W^{(j+K-1)}\}$ or $\{W^{(j)}, \dots, W^{(j+K)}\}$. This is done until a local maximum for ScoreTetrads is found;
- At the end of Line 9, we have a set of proposed instrumental variables which are found to approximately imply the same causal effect, as defined by Line 4. This also implies approximately satisfying the corresponding tetrad constraints. In Line 10, we formally check the validity of this candidate set by performing Wishart tests of conditional tetrad constraints (Wishart, 1928; Spirtes et al., 2000) for every possible pair in $\mathbf{W}_j$ against pair $\{X, Y\}$.[8] Function TestTetrads returns true if and only if more than half of the tested pairs return a test p-value greater than a particular pre-defined level. In our experiments in the next section, this level was set at 0.05. *This step has therefore a direct correspondence to the tetrad checks in* IV-TETRAD$^+_\infty$*, augmented with a pre-selection of pairs*;
- Finally, in Line 12, for each $W \in \mathbf{W}_j$, function TestResiduals is defined such that we first obtain the residual r_W of the least-squares regression of W on $\mathbf{O} \setminus W$, as well as the residual r_X of the regression of X on $\mathbf{O}$ (alternatively we could calculate r_Y, but we expect X to be more strongly associated with W as in theory there is no edge between W and Y). We perform a marginal independence test of these two residuals using HSIC (Gretton et al., 2007). As with the tetrad tests, TestResiduals will return true if and only if more than half of the p-values are above a particular threshold (again 0.05 in our experiments).

7. Although this rules out possible IVs which are dependent on X only if we condition in some other observed variable (for instance, if the only path between W_i and X is $W_i \to Z \leftarrow U_z \to X$ for some $Z \in \mathbf{Z}$ and $U_z \in \mathbf{U}$), in preliminary runs we found this to provide more stable outputs.
8. We ignore that the test distribution in Wishart's test assumes Gaussianity and marginal covariances. As discussed by Spirtes (2013), there is evidence that tests without Gaussianity assumptions (Bollen, 1990) have no clear advantage compared to Wishart's.

The rationale for Algorithm 4 is as follows. In the limit of infinite data, any set of K variables which satisfies the corresponding tetrad constraints will return a value of zero for ScoreTetrads, which will then be expanded by ExpandTetrads to include any remaining candidate IVs that imply the same ratio λ. Sets of size K that satisfy the Graphical Criteria will contribute with the true causal effect to $\mathcal{C}$, as will also sets of size K or larger corresponding to locally covariance equivalent causal effects. The algorithm can fail to produce information that a more exhaustive algorithm would produce if subsets of conditioning sets $\mathbf{O} \setminus W_i$ were considered instead (for instance, in the limit of infinite data, IV-TETRAD$^+$ will return an empty $\mathcal{C}_R$ for data generated by the model in Figure 3, which is in theory avoidable).

The separation between $\mathcal{C}$ and $\mathcal{C}_R$ is due to the fact that full automation of effect discovery is not recommended, as there are unidentifiability trade-offs. Recall that in the graph where W_1 and W_2 are valid IVs, but where W_1 and X share an unmeasured confounder, that the residual test will asymptotically reject this candidate pair. It is to remind the user of this possibility that the algorithm separates, in a second set, the outcomes filtered by a more stringent criteria. The price to be paid is the stronger assumption concerning the existence of a partition $(\mathbf{W}, \mathbf{Z})$ of $\mathbf{O}$ such that $\mathbf{W}$ are valid IVs given $\mathbf{Z}$ and that $|\mathbf{W}| \geq K$. If this existence criteria also includes that no elements in $\mathbf{W}$ has an active non-directed path with Y given $\mathbf{Z}$, then $\mathcal{C}_R$ will be no larger and sometimes smaller than $\mathcal{C}$. Both sets can be summarized in many ways, including using them to provide lower and upper bounds to λ_{yx} in a way analogous to Corollaries 11 and 18.

4.2 Further Comments

IV-TETRAD$^+$ can in principle be expanded to include a greedy search over sets $\mathbf{Z}$ which are strict subsets of $\mathbf{O} \setminus \{W_i\}$. Some non-trivial theory on how to do it with formal guarantees and non-trivial assumptions is left as future work. The most immediate modification of IV-TETRAD$^+$ would be to include a further outer loop, where other conditioning sets are used instead of $\mathbf{O}_{\setminus i}$, starting from the empty set. The outer loop would increase this set by one element at a time, optimizing ScoreTetrads. As this is a relatively straightforward heuristic without any insightful theory behind it, and the statistical variance of the final result will just increase considerably, we do not discuss it further. Our preferred solution is to let the algorithm return an empty set if no tetrad constraints and residual tests are satisfied, ignoring the search for a more specific conditioning set. The theory behind IV-TETRAD$^+_\infty$ is used in IV-TETRAD$^+$ to provide guarantees about $\mathcal{C}_R$. It also provides the explanation of what can go wrong by improperly conditioning on a covariate. This theory can be combined with background knowledge to manually select a pre-defined set of conditioning sets that are believed to be adequate for finding instruments, in case the default $\mathbf{O}_{\setminus i}$ is deemed insufficient.

The choice of K is not a straightforward question. This decision is a bias-variance trade-off: smaller K means weaker assumptions but higher variance. We do not provide a domain-free rule of thumb: our recommendation is to set K as high as possible according to knowledge from the domain of interest. Notice that, unlike sisVIVE, this number does not need to grow with $|\mathbf{O}|$. Sensitivity analysis, where different values of K are attempted, can be a valuable tool to diagnose the impact of this choice.

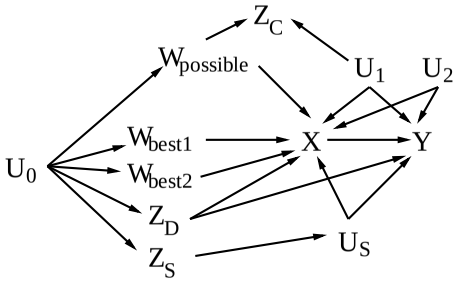


Figure 6: Example of a synthetic graph generated by the template used in Section 5. W_{best_1}, W_{best_2} and $W_{possible}$ are valid IVs conditioned on Z_D and Z_S only, as conditioning on Z_C activates the path $W_{possible} \rightarrow Z_C \leftarrow U_1 \rightarrow Y$, which invalidates that instrument. However, W_{best_1} and W_{best_2} remain valid even after conditioning on all other observed variables.

5. Experiments

We assess how IV-TETRAD$^+$ compares to other methods in a series of simulations. We then provide an example on how the method can be used in an empirical study and assess its robustness.

5.1 Synthetic Studies for Finding a Single Causal Effect Assuming IVs Exist

We generate synthetic models of particular structures, intended to capture features such as confounding and collider bias that should be avoided by proper conditioning. In order to allow for a direct a comparison of IV-TETRAD$^+$ against alternative methods that return a single causal effect estimate, we pragmatically force our algorithm to choose a single element in each output $\mathcal{C} \cup \mathcal{C}_R$ for the sake of simplifying comparisons.[9] The choice is the effect estimate corresponding to the set $\mathbf{W}_j$ that minimizes ScoreTetrads at the end of Line 9, which is then contrasted against the true effect.[10]

Simulations are performed as follows. Treatment X and outcome Y have two latent parents, U_1 and U_2. U_1 is connected to observed covariates as described below, so its variance decreases by conditioning. U_2 is not directly connected to the observed covariates. We then generate synthetic graphs by splitting the observed covariates in different groups.

9. Also, in this setup, the non-Gaussianity test of Line 12 does not play a role in the final output.
10. It is theoretically possible that this criterion will make IV-TETRAD$^+$ return the wrong causal effect in the limit of infinite data, as by design we introduce locally covariance equivalent causal effects. However, with finite data this criterion typically chooses solutions corresponding to correct IVs: in our setup, the non-IV vertices that imply locally covariance equivalent effects tend to be more weakly associated with outcome Y than the true IVs, as paths from such vertices to outcome Y pass through latent choke points. This inflates the variance of the corresponding $\hat{\lambda}_i$, leading to higher disparity compared to the effect estimated by two-stage least squares.

Confounding	$\lambda_{xu} = \lambda_{yu} = 0.125$			$\lambda_{xu} = \lambda_{yu} = 0.25$		
N	1000	5000	10000	1000	5000	10000
NAIVE1	0.25, 0.79	0.25, 0.81	0.25, 0.81	0.50, 0.58	0.50, 0.58	0.50, 0.59
NAIVE2	0.16, 0.84	0.16, 0.84	0.15, 0.85	0.37, 0.68	0.36, 0.66	0.35, 0.69
NAIVE3	0.54, 0.56	0.54, 0.56	0.54, 0.56	0.78, 0.43	0.78, 0.40	0.78, 0.42
ORACLE	0.04, 0.94	0.02, 0.95	0.01, 0.98	0.13, 0.88	0.04, 0.95	0.02, 0.98
S-ORACLE	0.06, 0.92	0.03, 0.95	0.02, 0.97	0.17, 0.78	0.06, 0.93	0.03, 0.93
W-ORACLE	0.25, 0.80	0.23, 0.79	0.21, 0.77	0.52, 0.58	0.47, 0.60	0.44, 0.63
SISVIVE	0.34, 0.73	0.35, 0.72	0.31, 0.75	0.64, 0.56	0.67, 0.56	0.68, 0.57
IV-TETRAD$^+$	0.23, 0.78	0.12, 0.86	**0.07, 0.86**	**0.56, 0.61**	0.26, 0.77	**0.10, 0.79**

Table 2: Experimental results for 8 methods as described in the text, including three oracle competitors that illustrate idealized scenarios where parts of the structure are known. Two experimental parameters are varied: sample size N and the amount $\lambda_{xu} = \lambda_{yu}$ of unmeasured confounding between X and Y. For each pair of method and experimental condition, we report the summarized performance over 100 trials in two ways: the median absolute difference between the estimated causal effect and true λ_{yx} (the smaller, the better); and the proportion in which the sign of the estimated effect and the sign of the true effect agree (the higher, the better). IV-TETRAD$^+$ beats all non-oracle competitors at sample size 10000, with a p-value $p < 0.001$ according to a binomial test (in bold). Notice that NAIVE2 is actually the closer competitor to IV-TETRAD$^+$, while in this particular setup sisVIVE does not improve even with large sample sizes.

Group $\mathbf{W}$ are variables which can be used as conditional IVs. This group has two subgroups, $\mathbf{W}_{best}$ and $\mathbf{W}_{possible}$. The former remains a set of valid IVs conditioned on all other observed covariates. The latter will have active collider paths with treatment X that can be in principle deactivated by a combinatorial search, which is not performed by any algorithm in our benchmark. Group $\mathbf{Z}_S$ is a set of covariates with a common latent child U_S which is also a parent of X and Y, making U_S a spurious choke point between $\mathbf{Z}_S$ and $\{X, Y\}$. Group $\mathbf{Z}_D$ is a set of covariates which are parents of both X and Y and as such are not blocked by choke points, latent or not. Group $\mathbf{Z}_C$ are variables which are children of $\mathbf{W}_{possible}$ and U_1. Finally, all variables in $\mathbf{WZ} \equiv \mathbf{W} \cup \mathbf{Z}_S \cup \mathbf{Z}_D$ have a common latent parent U_0, making conditioning on $\mathbf{WZ} \backslash W$ required for $W \in \mathbf{W}_{best}$ to be a valid IV. Figure 6 shows an example where $|\mathbf{W}_{best}| = 2, |\mathbf{W}_{possible}| = 1, |\mathbf{Z}_S| = 2, |\mathbf{Z}_D| = 1$ and $|\mathbf{Z}_C| = 1$.

The methods we compare against are:

1. NAIVE1, the least-squares regression coefficient of Y on X;
2. NAIVE2, two-stage least squares (TSLS) of Y on X using all variables $\mathbf{O} = \mathbf{W} \cup \mathbf{Z}_S \cup \mathbf{Z}_C \cup \mathbf{Z}_D$ as instruments;
3. NAIVE3, least-squares regression of Y on X and $\mathbf{O}$;

4. ORACLE, TSLS estimation using the correct set of IVs $\mathbf{W}$ and correct adjustment set $\mathbf{Z}_S \cup \mathbf{Z}_D$;

5. W-ORACLE, TSLS using $\mathbf{W}$ as IVs, but conditioning on all of the other variables $\mathbf{Z}_S \cup \mathbf{Z}_D \cup \mathbf{Z}_C$;

6. S-ORACLE, the algorithm of Kang et al. (2016) executed by first correctly removing the set $\mathbf{Z}_C$ from the input;

7. SISVIVE, the algorithm of Kang et al. (2016) executed in the standard way, by taking all variables $\mathbf{O}$ as input;

8. IV-TETRAD$^+$, our method, using all variables $\mathbf{O}$ as input and $K = 10$.

All error variables and latent variables are zero-mean Laplacian distributed, and coefficients are sampled from standard Gaussians and re-scaled such that the observed variables have a variance of 1. A simulation is rejected until $|\lambda_{yx}| > 0.05$. Coefficients between $\{X, Y\}$ and latent variables $\{U_1, U_2\}$ are set such that $\lambda_{xu_1} = \lambda_{yu_1} = \lambda_{yu_2} = \lambda_{yu_2}$, at two levels, $(0.125, 0.25)$. The difficulty of the problem increases with λ_{xu_1}, as this makes unmeasured confounding stronger. Comparisons are shown in Table 2, with the setup $|\mathbf{W}_{best}| = 15$, $|\mathbf{W}_{possible}| = 10$, $|\mathbf{Z}_S| = 10$, $|\mathbf{Z}_D| = 5$, $|\mathbf{Z}_C| = 10$. This satisfies the criterion of $|\mathbf{W}|$ being half the number of remaining variables, although only the 15 variables $\mathbf{Z}_S \cup \mathbf{Z}_D$ should be used as a conditioning set.

We generate 100 synthetic problems with a data set of 10,000 points each and assess methods using also subsets of size 1000 and 5000. For $\lambda_{xu} = 0.125$, the empirical distribution of effects λ_{yx} had a median of 0.34, an upper quartile of 0.55 and a maximum of 1.05. For $\lambda_{yu} = 0.25$, a median of 0.35, an upper quartile of 0.75 and a maximum of 1.28. For assessment, we use two measures that summarize absolute deviance from λ_{yx} and agreement with its sign.

The message in Table 2 is as follows. The parameter estimation problem is easy if one knows the correct IVs and background variables (ORACLE), but it shows already its challenges if one has potentially correct IVs but conditions on the wrong set (W-ORACLE). sisVIVE can work very well if one knows in advance which conditioning variables to discard (S-ORACLE). In general we will not have this knowledge, and sisVIVE can potentially behave badly, even worse than a naïve method (compare NAIVE3 to SISVIVE). The large contrast between S-ORACLE and SISVIVE is a warning against ignoring the effects of incorrect conditioning. Of all non-oracle methods, IV-TETRAD$^+$ is the clear winner, although as expected it is a high variance estimator and should be advantageous only for relatively large sample sizes. In Section 5.3 we illustrate ways in which the uncertainty of this estimator can be assessed.

5.2 Residual Test Assessment

We assess the robustness of the residual test validations of IV-TETRAD$^+$ in a series of simulations. We designed three scenarios as illustrated in Figure 7, where $\{U_0, U\}$ are latent variables:

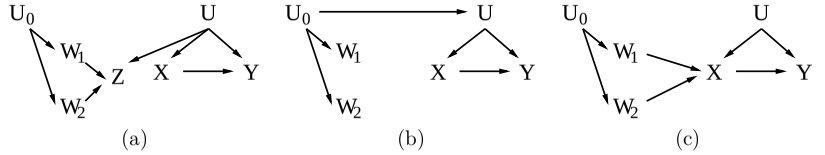


Figure 7: The three different configurations used in the assessment performed in Section 5.2.

S_1: The graph is given by Figure 7(a). The ideal output is to falsify the null hypothesis of residual independence between W_1 and X given $\{W_2, Z\}$, and W_2 and X given $\{W_1, Z\}$.

S_2: The graph is given by Figure 7(b). The ideal output is to falsify the null hypothesis of residual independence between W_1 and X given W_2, and W_2 and X given W_1.

S_3: The graph is given by Figure 7(c). The ideal output is *not* to falsify the null hypothesis of residual independence between W_1 and X given W_2, and W_2 and X given W_1.

In each scenario, we simulate 100 synthetic problems and generate a data set of size 10,000 from each. The simulation of parameters is similar to Section 5.1, also including two levels of confounding by setting $\lambda_{xu} = \lambda_{yu}$ to either 0.125 or 0.25. We also increase the non-Gaussianity of the error terms: each error term is generated by sampling a Laplace distributed random variable as before, and then raising it to the power of 1.5, preserving the sign, and rescaling it back to have the same variance required so that each latent and observed variable has a marginal variance of 1. We do not claim that the tests proposed are particularly strong when the sample size is not large or the random variables are not clearly non-Gaussian. We do illustrate that, according to the theory, we should get better decisions with increasing sample sizes.

A joint decision is given for the pair of variables $\{W_1, W_2\}$, as in the definition of TestResiduals used in Algorithm 4. This is done by voting: for each candidate IV $W \in \{W_1, W_2\}$, we test the independence of r_W against r_X using HSIC (Gretton et al., 2007) conditioning on the remaining observed variables. If more than half of the candidate IVs result in a corresponding p-value greater than the chosen level of 0.05 (in our setup, this means that both W_1 and W_2 pass the test), then the model passes the test and the IVs are not falsified. Otherwise, the model is rejected. In our simulation, we are correct in each simulated instance if we reject the candidate instrument pair in scenarios S_1 and S_2, or if we do not reject it in scenario S_3.

We use a Shapiro-Wilk test of Gaussianity followed by the HSIC test as implemented in the R package dHSIC,[11] both at a level 0.05. That is, if for candidate instrument W the Gaussianity assumption cannot be rejected for the residuals r_W or r_X, then W by default is

11. Available at the Comprehensive R Archive Network, CRAN, `https://cran.r-project.org/`.

Confounding	$N = 1000$			$N = 5000$			$N = 10000$		
	S_1	S_2	S_3	S_1	S_2	S_3	S_1	S_2	S_3
$\lambda_{yu} = 0.125$	0.54	0.28	0.87	0.77	0.85	0.89	0.82	0.93	0.83
$\lambda_{yu} = 0.25$	0.63	0.59	0.79	0.85	0.98	0.83	0.86	1.00	0.83

Table 3: Experimental results for assessing the detection of invalid instrument candidates based on the dependencies between least-squares residuals of instruments and treatments on a common pool of covariates. Within each sample size and amount of confounding, we show the proportion of times we correctly decided whether the corresponding pair of candidate IVs are indeed valid or not (out of 100 simulations). The three problems S_1, S_2 and S_3 are discussed in the main text. The correct decisions are to reject the candidates in problems of type S_1 and S_2, and not reject them in S_3.

not rejected as a plausible instrument.[12] Only default dHSIC hyperparameters were used to make the results more conservative: we believe better results can be obtained by more sophisticated approaches for hyperparameter selection. Similarly, the p-value threshold of 0.05 should not be seen the level of the test but as a regularization parameter, and should be adapted as the sample size increases (Kalisch and Bühlmann, 2007). Results are summarized in Table 3.

As expected, correctly rejecting the model in scenarios S_1 and S_2 is easier when confounding is stronger, which is precisely when we would like to be more conservative regarding our choice of instruments. Scenario S_2 is particularly hard under smaller sample sizes, as correlations quickly go to zero given a single path of length 3 between candidate instruments and treatment/outcome. The performance in Scenario S_3 plateaus, meaning that in this particular simulated distribution of parameters we still erroneously reject the valid IVs approximately 17% of the time. The Type I error probability is not that straightforward to calculate due to the voting mechanism and the dependency of the least-squares rejection estimates and the residual estimates. However, as it is known, in any constraint-based search algorithm we should decrease the test level hyperparameter from 0.05 towards zero as sample size increases if we want consistency. This is analogous to the behavior of other algorithms such as the PC algorithm (Spirtes et al., 2000).

To conclude, despite the noted shortcomings of requiring detectable non-Gaussianity and large sample sizes, the tests do provide a validation of proposed instruments as advertised. Moreover, they are expected to be more effective in discarding possibly invalid IVs as unmeasured confounding gets stronger, which is a desired behavior. However, we still recommend that we output candidate effects both with and without the residual independence tests as done by IV-TETRAD$^+$, for the reasons explained.

12. In theory the goal is to save computational time. In this cases the independence assumption is likely not to be rejected anyway, but the HSIC test can be costly if done many times so we want to avoid performing it whenever possible.

5.3 Empirical Illustration

We consider the application of our method to the study by Sachs et al. (2005). That study collected cell activity measurements (concentration of proteins and lipids) for single cell data under a variety of conditions. While searching for instrumental variables under a combination of experimental conditions is an interesting topic, we will focus on a single condition (described in the paper as stimulation with anti-CD3 and anti-CD28). There are 11 variables and 853 data points, of which we selected the manipulation of concentration levels of molecule *Erk* as the treatment, and concentration of *Akt* as the outcome. We use as background knowledge the model inference result shown as Figure 3A presented by Sachs et al. (2005), encoding that the other 9 cell products are not causally affected by either *Erk* or *Akt*. Although previously unknown, evidence for this causal link was also given support by an experiment performed by Sachs et al. (2005).

The data shows some weak correlations, but we will assume for simplicity that no conditional independencies will hold between the treatment variable and the remaining 10 variables. The motivation for Step 3 in Algorithm 4 was primary as a variance reduction technique. Instead, here we will be primarily concerned about illustrating how we could assert uncertainty in our estimates using the modified Bayesian approach introduced by Silva and Kalaitzis (2015).

Running the standard regression adjustment NAIVE3 on this data, we get a differential causal effect of 1.36. sisVIVE reports an effect of 1.58, where all variables were chosen as instruments and as such this estimate here is the same as NAIVE2. This suggests that unmeasured confounding may be weakening the association between treatment and outcome, but sampling variability may be high as discussed in the next section. We run IV-TETRAD$^+$ as in Section 5.1 (and skipping Step 3), where a single effect of 1.43 is returned by minimizing ScoreTetrads. For this run we chose a window size $K = 3$, as $K = 2$ is in general too noisy of a choice since it cannot exploit any redundancies of the tetrad constraints implied by the candidate instrument sets. TestResiduals does not reject any candidate set proposed by the tetrad search. For $K = 4$ and $K > 4$ the results were 1.38 and 1.58 respectively, illustrating some of the difficulties of working with a relatively small sample size (853 points) using an algorithm that has been shown to require large samples. To illustrate what a practitioner should do in this case, we describe ways of assessing the uncertainty of the output of IV-TETRAD$^+$.

5.3.1 Exploration with the Bootstrap

First, we show a simple comparison based on the bootstrap: we sample the data with replacement and show the corresponding (bootstrapped) sampling distribution of the estimates obtained by NAIVE3, sisVIVE and the single-output IV-TETRAD$^+$ with $K = 3$. A smoothed estimate of the respective bootstrapped outcomes is shown in Figure 8. It is evident that as the method gets more flexible, the entropy of the respective sampling distribution of the estimates also increases. There is some evidence of bimodality of the sampling distribution, with IV-TETRAD$^+$ being particularly more extreme.

Bootstrap can in principle be used to find confidence intervals for the corresponding causal effects. However, it is relevant also to discuss Bayesian approaches for quantifying uncertainty. This is particularly interesting for a method like IV-TETRAD$^+$, which po-

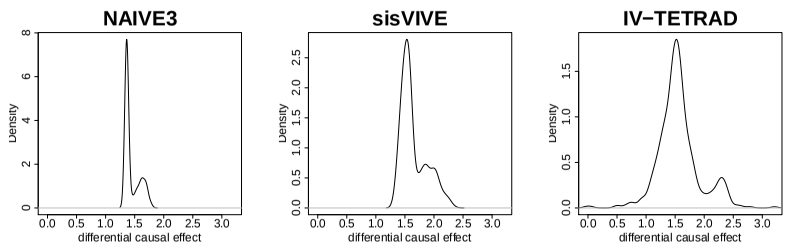


Figure 8: Smoothed sampling distribution of the estimated causal effects for the three discussed methods using 500 bootstrapped samples and kernel density estimation.

tentially returns sets of causal effects of varying sizes. It is not clear what a confidence interval (or confidence set) would be here. Returning a summary such as the minimum and maximum elements of the set, and building confidence intervals on bounds of the causal effect, may be too conservative and not very robust to outliers. In the next section, we will show a Bayesian solution to that problem. First, let us start by still considering the case where IV-TETRAD$^+$ returns a single element as chosen by ScoreTetrads.

Obtaining a posterior distribution over the causal effect of interest requires a likelihood function that includes all observed variables. The usual Bayesian approach for graphical modeling would require exploring a space of conditional tetrad and residual independence constraints, where any chosen set of constraints defines a model. We would then evaluate the marginal likelihood of each model and search for the maximum a posteriori or list several models of high probability. Alternatively, we in principle could perform Markov chain Monte Carlo (MCMC) sampling in the space of parameters and model constraints. A very different algorithm from IV-TETRAD$^+$ would be necessary, as it does not search among complete models. Such an algorithm would be computationally very intense, and it is not clear how to parameterize a joint likelihood for any given set of constraints, whether latent variables are explicitly introduced or not.

5.3.2 Learning by Bayesian Projections

Instead, we adopt the much simpler "Bayesian projections" approach suggested by Silva and Kalaitzis (2015), which has a relation to several other approaches for model selection that avoid introducing constraints directly in the likelihood function, such as the one by Goutis and Robert (1998). We define a black-box non-Gaussian likelihood function for the observed variables, generate posterior samples from it by MCMC, and perform a causal effect search algorithm similar to IV-TETRAD$^+$. The main difference here is that instead of performing statistical tests, we instead reject constraints if they violate a threshold of misfit.

To make it more explicit, we define an algorithm which we will call IV-TETRAD^{++}. In this algorithm, we are given M posterior samples of the model parameters from a black-box

mixture of Gaussians likelihood. Within each sample $1 \leq m \leq M$, we run IV-TETRAD$^+$ with the following modifications:

1. Empirical covariances $\hat{\sigma}_{ab.\mathbf{s}}$ are replaced with the corresponding model covariances $\sigma^{(m)}_{ab.\mathbf{s}}$, which can be computed analytically for a mixture of Gaussians model;
2. We ignore Line 3 of IV-TETRAD$^+$, the one where we originally estimated Markov blankets. Instead, we discard every λ_i such that $|\lambda_i| > \alpha_{cut}$. Hyperparameter α_{cut} is given as an input, implying the removal from the output of unusually high causal effects, including some of those resulting from variables V_i such that $\sigma_{v_i x.\mathbf{v}_{\setminus i}} \approx 0$.
3. TestTetrads is modified to reject a tetrad constraint if
$$|\rho^{(m)}_{w_i x.\mathbf{o}_{\setminus i}} \rho^{(m)}_{w_j y.\mathbf{o}_{\setminus j}} - \rho^{(m)}_{w_j i y.\mathbf{o}_{\setminus i}} \rho^{(m)}_{w_j x.\mathbf{o}_{\setminus j}}| > \tau,$$
where $\rho_{ab.\mathbf{s}}$ is the partial correlation of A and B given $\mathbf{S}$. Hyperparameter τ is given as an input, and provides the rule that decides on whether a tetrad constraint fits the model given by the m-th MCMC sample with model partial correlations $\rho^{(m)}_{ab.\mathbf{s}}$.
4. For simplicity, we do not perform TestResiduals. It could in principle be implemented by sampling a large synthetic data set from model m, estimating a measure of dependence between the residuals, and rejecting independence with this measure is larger than a given hyperparameter. Due to its computational cost and the fact it does not change the message of this section, we will ignore this step.

A more formal description of the algorithm is given in Appendix C.

Silva and Kalaitzis (2015) discuss the shortcomings of algorithms based on Bayesian projections compared to a traditional Bayesian approach where the likelihood enforces constraints explicitly. However, the computational advantages are major, as sampling from the posterior of a more standard black-box model is much simpler than searching within a complex likelihood space. As a matter of fact, at the present time it is not known how to define such a likelihood function. A Bayesian projection approach has some similarities to frequentist bootstrap, where a black-box object (the empirical distribution, in the nonparametric bootstrap case) provides the source of variability (the sampling distribution, in the bootstrap case). This allows us to leverage constraint-based algorithms with few modifications.

The most delicate design choices are the choice of α_{cut} and τ. One possibility to assess its impact is by sampling from the prior and plotting the implied distribution on differential causal effects. This is done in Figure 9, where the black-box distribution is a mixture of 5 Gaussians with the following priors: the mixing distribution is given a Dirichlet prior with hyperparameter $(2, 2, 2, 2, 2)$; the covariance matrix Σ_c of each mixture component c is given an inverse Wishart prior with $\nu = 11 + 3$ degrees of freedom and a $\nu\mathbf{I}$ scale matrix, with $\mathbf{I}$ being the 11×11 identity matrix; the mean vector μ_c of each component is given a multivariate Gaussian prior with zero mean and covariance matrix Σ_c.

Given the prior for the base distribution, Figure 9 shows a smoothed depiction of 1000 samples from the priors defined by sampling the mixture model parameters from the base distribution and passing them through IV-TETRAD^{++}, where we choose the single element that minimizes ScoreTetrads. To generate the figure, we used the the following configurations: i. $(\alpha_{cut} = 10, \tau = 0.01)$; ii. $(\alpha_{cut} = 3, \tau = 0.01)$ and iii. $(\alpha_{cut} = 10, \tau = 0.001)$.

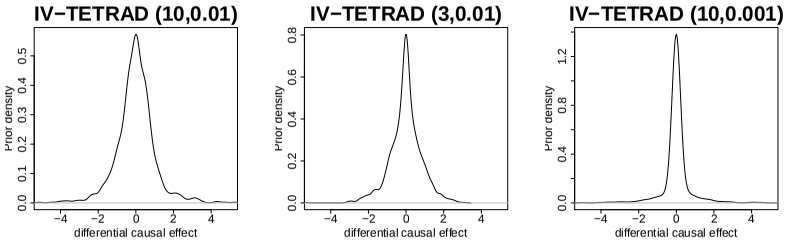
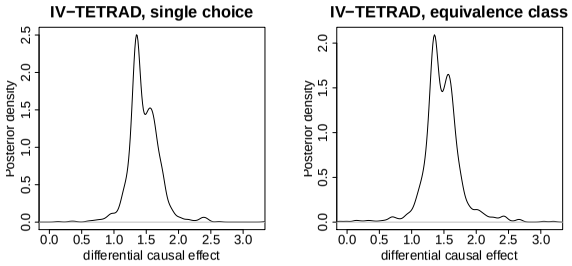


Figure 9: Examples of the implied prior on the differential causal effect using the Bayesian projections algorithm IV-TETRAD^{++} with three choices of hyperparameters ($\alpha_{cut} = 10, \tau = 0.01$), ($\alpha_{cut} = 3, \tau = 0.01$) and ($\alpha_{cut} = 10, \tau = 0.001$).



(a) (b)

Figure 10: (a) The posterior distribution when a single value is chosen as the output of IV-TETRAD^{++}. (b) The resulting posterior where we uniformly sample one of the elements of the equivalence class.

As expected, decreasing these two hyperparameters makes the prior narrower. At the same time, since the effect is given by the ratio of covariance entries, we expect the implied prior to be heavy-tailed. For illustration purposes, we proceed to analyze the Sachs et al. data by choosing ($\alpha_{cut} = 10, \tau = 0.01$) as the prior configuration.

5.3.3 Results

Figure 10(a) shows the resulting posterior after running MCMC for the mixture of Gaussians for 10,000 iterations and thinning it down to 1000 samples. The evidence that 5 mixture components is enough is given by the fact that the component of smallest posterior probability was assigned almost zero mass. The figure illustrates the difficulty of the problem and

why a single point estimate might not be a good summary of the outcome even in the case where a single causal effect is returned.

More interestingly, the Bayesian approach allows us to generate posteriors over equivalence classes in a relatively simple way, without resorting to potentially uninformative summaries based on bounds. This however requires subject matter information, as the data cannot distinguish the elements in an equivalence class. Following the philosophy of Richardson et al. (2011), we separate priors for elements which are informed by the data (equivalence class) from priors for elements which cannot be distinguished (the choice *within* the equivalence class, given the equivalence class). Figure 10(b) shows the posterior distribution that, for each sample m of the Markov chain, generates a sample causal effect by sampling uniformly at random from the elements in the corresponding equivalence class of sample m. That is, the extra information here is the uniform prior, which may be taken as a default choice. In principle, another prior can be used. For instance, if it is agreed that the choice should be made based, for instance, on a distribution that assigns mass in a way that is inversely proportional to the magnitude of each element in the class (this assumes that smaller effects are more likely than large effects). In any case, the important lesson is the understanding that this final component of the posterior is independent of the data given the equivalence class.[13] The conclusion from Figure 10(b) is that the bimodality is explained not only by the data, but also by the indeterminacy of the solution, in contrast to the bimodality found in Figure 8, which is an artefact of the sampling distribution of the corresponding estimators.

6. Related Work

The notion of "equivalence class of causal effects" is not new, even if not previously named as such. Hoyer et al. (2008) discuss how causal effects in LiNGAM models can be partially identified if the number of latent confounders is known. Their method also returns sets of causal effects, but without any need for further covariates and instruments. However, the size of the equivalence class of causal effects grows rapidly with the number of assumed latent variables and it is unclear how to determine the number of unmeasured confounders. Maathuis et al. (2009) present extensions of the PC algorithm which return sets of causal effects that follow from different members of an equivalence class of DAGs compatible with the observable independence constraints.

Didelez and Sheehan (2007) present a survey on instrumental variables from the point of view of applications in Mendelian randomization. A complementary method to sisVIVE based on similar assumptions, but performing estimation by the reweighted median of covariance ratios, is discussed by Bowden et al. (2016). The difficulty of estimating effects with weak instruments is well-known in the statistics literature. A review that covers some background on the estimation with weak instruments is provided by Burgess et al. (2017).

A different way of inducing "approximate" instrumental variables was introduced by Silva and Evans (2016). In that case, only (approximate) independence constraints are exploited. The same machinery in principle could include approximate tetrad constraints, although

13. This idea could also be adapted to the case where the empty set is returned in some of the MCMC samples: the posterior would be a mixture model with positive mass on the empty set and the remaining mass spread over the values found.

further research on that would be necessary. The problem of equivalence class of causal effects is also present in that approach. In such a scenario, however, there are infinitely many alternatives, which can be represented as an interval. The width of the interval is typically large and uninformative in classical bounding methods for instrumental variables (Didelez and Sheehan, 2007), but the approach introduced by Silva and Evans (2016) allows for the introduction of stronger assumptions as a way of reducing the size of the equivalence class. This borrows some concepts from earlier work such as the one Ramsahai (2012).

The use of tetrad constraints for testing the validity of particular edge exclusions in linear causal models has a long history, dating back at least to Spearman (1904). Although the generalizations discussed by Sullivant et al. (2010) were presented in the context of Gaussian models, this distributional assumption is not necessary, with their rank constraint results exploited even in the context of partially non-linear models by Spirtes (2013). More recently, tetrad constraints have been used in the discovery of latent variable model structure (Silva et al., 2006; Spirtes, 2013), where structures such as Figure 2(c) emerge but no direct relationships among observables (such as $X \to Y$) are discoverable. The combination of tetrad constraints and non-Gaussianity assumptions has been exploited by Shimizu et al. (2009), again with the target being relationships among latent variables. Tetrad tests for the validity of postulated IVs were discussed by Kuroki and Cai (2005).

The literature on learning algorithms allowing for latent variables has been growing steadily, including the Fast Causal Inference algorithm of Spirtes et al. (2000) and more recent methods that exploit constraints other than independence constraints (Tashiro et al., 2014; Nowzohour et al., 2015), but none of these methods allow for the estimation of the causal effect of X and Y when there is an unblocked unmeasured confounder between them. Phiromswad and Hoover (2013) introduced an algorithm for IV discovery, but it does not take into account unidentifiability issues that can be solved by exploring constraints other than covariance matrix constraints. It also returns equivalence classes of graphs and requires searching for multiple causal effects at the same time, contrary to our goal of focusing on a given causal effect λ_{yx}.

7. Conclusion

Finding instrumental variables is one of the most fundamental problems in causal inference. To the best of our knowledge, this paper provides the first treatment on how this can be systematically achieved by exploiting rank constraints and clarifying to which extent an equivalence class of solutions remains. We then proceeded to show how non-Gaussianity can be exploited in a pragmatic way, by adapting a state-of-the-art algorithm. Finally, we illustrated how empirical improvement can be obtained.

We expect that theoretical challenges in instrumental variable discovery can be further tackled by building on the findings shown here. In particular, as also hinted by Kang et al. (2016), some of the ideas here raised extend to non-linear (additive), heterogeneous effects and binary models. Methods developed by Peters et al. (2014) can potentially provide a starting point on how to allow for non-linearities in the context of instrumental variables. As discussed by Spirtes (2013), linearity is only really needed "downstream" of the choke point: that is, it would be enough that only the structural equation for outcome Y is

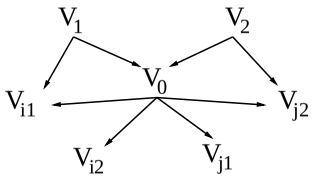


Figure 11: The graph used in Appendix A for more complex examples of t-separation. We are interested on making claims about this structure by observing some, but not all, of its vertices.

linear. Theoretical details need to be sorted out, but IV-TETRAD$^+$ should in principle be practically applicable to non-linear models with linearity for Y only.

More sophisticated graphical criteria for the identification of causal effects in linear systems were introduced by Brito and Pearl (2002). Further work has led to rich graphical criteria to identify causal effects in confounded pairs (Foygel et al., 2011), going beyond the IV criteria discussed here. This opens up the possibility elaborated discovery algorithms where back-door blocking (Entner et al., 2012) and the methods in this paper cannot provide a solution. How to perform this task in a computationally and statistically tractable way remains an open question.

Code for all experiments is available at `http://www.homepages.ucl.ac.uk/~ucgtrbd/code/iv_discovery`.

Acknowledgments

We thank Robin Evans, Peter Spirtes and Kun Zhang for several useful discussions. Part of the discussions were possible by travel funding supporting the attendance of RS and SS in the Short Thematic Program "Statistical Causal Inference and Applications to Genetics" organized and sponsored by the Centre the Recherches Mathématiques, University of Montreal, Canada. RS's visit was also partially supported by a grant from Adobe Research. RS also acknowledges travel support from the Center for Causal Discovery in Pittsburgh, PA, where part of the discussions that originated this paper took place. This work was partially supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1.

This manuscript has strongly benefited from suggestions and clarification questions raised by three anonymous reviewers.

Appendix A. More about T-separation

We start with a few more examples of t-separation. Consider Figure 11. Vertex V_0 does not d-separate V_{i1} from V_{j2}. However, *pair* $(V_0; V_0)$ t-separates V_{i1} from V_{j2}. To see that, let us list all three (simple) treks between V_{i1} and V_{j2}:

- $(P_1^{[1]}; P_2^{[1]}) \equiv (V_{i1} \leftarrow V_1; V_1 \rightarrow V_0 \rightarrow V_{j2})$
- $(P_1^{[2]}; P_2^{[2]}) \equiv (V_{i1} \leftarrow V_0; V_0 \rightarrow V_{j2})$
- $(P_1^{[3]}; P_2^{[3]}) \equiv (V_{i1} \leftarrow V_0 \leftarrow V_2; V_2 \rightarrow V_{j2})$

In the first trek, V_0 is contained in $P_2^{[1]}$; in the second case, both paths in the trek satisfy the criterion of containing V_0; in the final trek, $P_1^{[3]}$ plays this role. Notice that $(V_0; \emptyset)$ does not t-separate V_{i1} from V_{j2} (first trek remains "unblocked") and neither does $(\emptyset; V_0)$ (third trek remains "unblocked").

That V_{i1} is t-separated from V_{j2} given $(V_0; V_0)$, however, brings out no useful implication about the cross-covariance matrix $\Sigma_{\mathbf{AB}}$ of ($\mathbf{A} \equiv \{V_{i1}\}$, $\mathbf{B} \equiv \{V_{j2}\}$): all it says is that $rank(\Sigma_{\{V_{i1}\}\{V_{j2}\}}) \leq 2$, which is a vacuous claim. We may attempt to introduce V_0 in the two sets $\mathbf{A}$ and $\mathbf{B}$, assuming V_0 is observable, as this is compatible with the notion that V_0 can t-separate itself from itself. Unfortunately, there is nothing to be gained, as $rank(\Sigma_{\{V_{i1}V_0\}\{V_{j2}V_0\}}) \leq 2$, which is again a vacuous claim.

Pair $(\emptyset; V_0)$ does t-separate V_{i1} and V_{i2} with a testable implication $rank(\Sigma_{\{V_{i1}V_0\}\{V_{i2}V_0\}}) \leq 1$, but this does not provide anything useful to a structure learning algorithm, as d-separation of V_{i1} and V_{i2} given V_0 can be tested directly instead.

Consider now $\mathbf{A} \equiv \{V_{i1}, V_{i2}\}$, $\mathbf{B} \equiv \{V_{j1}, V_{j2}\}$. To be useful, we need a conditioning pair $(\mathbf{C_A}; \mathbf{C_B})$ where $|\mathbf{C_A}|+|\mathbf{C_B}| \leq 1$. Following the previous examples, it should be clear that no such a pair will exist out of the given variables. However, we could consider de-activating the path $V_{i1} \leftarrow V_1 \rightarrow V_0$ by considering the *partial* cross-covariance matrix of $\{V_{i1}, V_{i2}\}$ against $\{V_{j1}, V_{j2}\}$ given V_1. The Trek Separation Theorem for DAGs, however, says nothing explicit about partial cross-covariances. Sullivant et al. (2010) present versions of the theorem for some classes of DAGs under conditioning, but not in a completely general way. Theorem 2.17 of Sullivant et al. (2010) assumes that conditioning can be encoded by undirected edges so that no vertices adjacent to an undirected edge can also be at the arrowhead endpoint of a directed edge. Although one can show this does not affect the Markov properties of general DAGs under conditioning, as ancestral graphs (Richardson and Spirtes, 2002) are a special case of this, it is not immediately obvious how this affects parametric constraints.

Fortunately, even in the conditioning case we do not need to consider anything but DAGs: simply introduce the conditioning variables in both sets $\mathbf{A}$ and $\mathbf{B}$. In this example, define $\mathbf{A} \equiv \{V_{i1}, V_{i2}, V_1\}$ and $\mathbf{B} \equiv \{V_{j1}, V_{j2}, V_1\}$. Then it is not hard to verify that $(V_0; V_1)$ t-separates $\mathbf{A}$ from $\mathbf{B}$ (recall that V_1 "t-separates itself from itself" using the conditioning set pair $(\emptyset; V_1)$). Cross-covariance $\Sigma_{\mathbf{AB}}$ is rank-deficient, which means its determinant is zero. Because its determinant can be written as $\sigma_{11}(\sigma_{i1j1.1}\sigma_{i2j2.1} - \sigma_{i1j2.1}\sigma_{i2j1.1})$ and we assume $\sigma_{11} \neq 0$, the (conditional) tetrad constraint will hold.

The relationship between t-separation and d-separation is given by Theorem 2.11 of Sullivant et al. (2010). Namely, d-separation between set $\mathbf{A}$ and set $\mathbf{B}$ given set $\mathbf{C}$ holds if and only if we can partition $\mathbf{C}$ as two sets $\mathbf{C_A}$ and $\mathbf{C_B}$ such that $(\mathbf{C_A}; \mathbf{C_B})$ t-separates $\mathbf{A} \cup \mathbf{C_A}$ from $\mathbf{B} \cup \mathbf{C_B}$. That is, any d-separation can be cast as some type of t-separation. However, as t-separation of $\mathbf{A} \cup \mathbf{C_A}$ from $\mathbf{B} \cup \mathbf{C_B}$ implies t-separation of $\mathbf{A}$ from $\mathbf{B}$, this allows us to derive rank constraints using the marginal distribution of $\mathbf{A} \cup \mathbf{B}$ only, unlike the conditional independence constraints directly implied by conditioning on $\mathbf{C}$. If $\mathbf{C}$ contains

hidden variables, this is another way of understanding how t-separation implies testable constraints that cannot be achieved directly by testing for conditional independencies.

Appendix B. Proofs

We being by proving the main result of Section 3.2.

Proof of Theorem 8. If $(W_i, W_j, \mathbf{Z})$ satisfies the Graphical Criteria, then there should be no active path between any element of $\{W_i, W_j\}$ and Y that does not include X. Hence, neither point (i) nor point (ii) in the theorem statement are possible.

Assume now that $(W_i, W_j, \mathbf{Z})$ does not satisfy the Graphical Criteria. Then there must be at least one active path, conditioned on $\mathbf{Z}$, connecting some $W \in \{W_i, W_j\}$ to Y which does not contain X. Otherwise, if all active paths contained X, then they would all be interrupted by the removal of edge $X \rightarrow Y$, as all other edges adjacent to X are into X, and X is not in the conditioning set $\mathbf{Z}$.

Suppose one of these active paths is a trek T, which should not contain any member of $\mathbf{Z}$ or otherwise it would not be active. $\{W_i, W_j, \mathbf{Z}\}$ must be t-separated from $\{X, Y, \mathbf{Z}\}$ by some $(\mathbf{C_A}; \mathbf{C_B})$, as implied by Theorem 3 and the conditional tetrad constraint detected in Line 9. As argued in the discussion leading to the definition of conditional choke point, $\mathbf{C_A} \cup \mathbf{C_B} = \mathbf{Z} \cup \{V_0\}$ for some conditional choke point V_0 not in $\mathbf{Z}$. If there is some other trek that is active and contains no member of $\mathbf{Z} \cup \{V_0\}$, then this would imply the existence of another element of $\mathbf{C_A} \cup \mathbf{C_B}$ not in $\mathbf{Z} \cup \{V_0\}$, contrary to the fact that $|\mathbf{C_A}|+|\mathbf{C_B}| \leq |\mathbf{Z}|+1$. Hence, scenario (i) in the theorem statement will happen if we do not satisfy the Graphical Criteria and there exists an active trek between $\{W_i, W_j\}$ and Y.

Assume that there exists a active path C' with endpoints $W \in \{W_i, W_j\}$ and Y, and which contains at least one collider in $\mathbf{Z}$. Allowing this path to be non-simple, we have that all colliders in this path must be in $\mathbf{Z}$, and no non-colliders are in $\mathbf{Z}$. Let $Z_1, Z_2, \ldots, Z_K$ be the colliders in C' as we move from W towards Y. Then the trek $(P_1^K; P_2^K)$ linking Z_K to Y is such that Z_K is the sink of P_1^K and hence $Z_K \in \mathbf{C_A}$. Likewise, Z_1 will be the sink of P_2^1 in the trek $(P_1^1; P_2^1)$ between W and Z_1, so that $Z_1 \in \mathbf{C_B}$. As we move from Z_1 towards Z_K in C', we will be forced to make one element of $\mathbf{Z}$ to belong to both $\mathbf{C_A}$ and $\mathbf{C_B}$, as and we cannot have all consecutive treks $Z_i \leftarrow \cdots \rightarrow Z_{i+1}$ such that Z_i and Z_{i+1} are both only in $\mathbf{C_A}$ or only in $\mathbf{C_B}$. This is illustrated in Figure 12. Therefore, there exists some $Z' \in \mathbf{C_A} \cap \mathbf{C_B}$. Since $|\mathbf{C_A}| + |\mathbf{C_B}| \leq |\mathbf{Z}| + 1$, we cannot have both (i) and (ii) being true, as Z' must be counted twice and $\mathbf{Z} \subseteq \mathbf{C_A} \cup \mathbf{C_B}$ (as we have seen, if there is a conditional choke point then $\mathbf{C_A} \cap \mathbf{C_B} = \emptyset$). Moreover, we cannot have another path C'' satisfying the properties of C' but which does not contain Z', as the existence of another $Z'' \neq Z'$ would again violate $|\mathbf{C_A}| + |\mathbf{C_B}| \leq |\mathbf{Z}| + 1$.

As all active paths connecting $\{W_i, W_j\}$ to Y must be either treks or paths where all colliders are ancestors of $\mathbf{Z}$, then these two cases cover all possibilities. □

We now proof the other core result from Section 3.3:

Proof of Theorem 15. By the same reasoning of Theorem 8, passing Line 9 leaves us with the two possible scenarios of that theorem. However, scenario (ii) cannot take place since, by Theorem 13 and Line 15, there must be no active non-directed paths between $\{W_i, W_j\}$

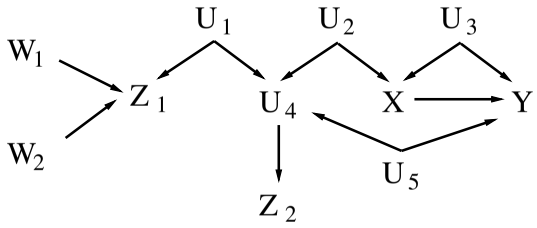


Figure 12: Path $W_1 \to Z_1 \leftarrow U_1 \to U_4 \to Z_2 \leftarrow U_4 \leftarrow U_5 \to Y$ is an example of a (non-simple) active path from some vertex in $\{W_1, W_2\}$ to Y given $\mathbf{Z} \equiv \{Z_1, Z_2\}$ where both Z_1 and Z_2 are colliders in this path. Here, Z_2 is on the "left side" of the trek $Z_2 \leftarrow U_4 \leftarrow U_5 \to Y$ than links an element of $\mathbf{A} \equiv \{W_1, W_2, Z_1, Z_2\}$ to $\mathbf{B} \equiv \{X, Y, Z_1, Z_2\}$. Z_1 is on the "right side" of the trek $W_1 \to Z_1$ that links an element of $\mathbf{A}$ to an element of $\mathbf{B}$. Trek $Z_1 \leftarrow U_1 \to U_4 \to Z_2$ also links en element of $\mathbf{A}$ to an element of $\mathbf{B}$. If we use Z_1 to block this trek, then Z_1 is now on the "left side" of a trek. If we use Z_2, then Z_2 is now on the "right side" of a trek. Hence, in order to t-separate $\mathbf{A}$ from $\mathbf{B}$, we need either $(\{Z_2\}; \{Z_1, Z_2\})$ or $(\{Z_1, Z_2\}; \{Z_1\})$, both of size 3 even though only two vertices are used.

and Y. For the same reasons, choke point V_0 cannot lie on an active non-directed path between $W \in \{W_i, W_j\}$ and Y, nor it can be in an active back-door path with endpoints in W and Y. As Y is not an ancestor of any vertex, V_0 must be a descendant of $\{W_i, W_j\}$ and an ancestor or Y. What is left is showing that V_0 is also an ancestor of X.

There must be an active path between W and X given $\mathbf{Z}$ by linear faithfulness and Line 6 of the algorithm. Assume V_0 is not an ancestor of X. There must be an active path between W and X which is not a trek or otherwise it would be blocked by $\mathbf{Z} \cup \{V_0\}$. However, using the same argument as in Theorem 8, this path with colliders in $\mathbf{Z}$ would also require a member of $\mathbf{Z}$ to be in both sets of the conditioning set pair $(\mathbf{C_A}; \mathbf{C_B})$, contradicting that $|\mathbf{C_A}| + |\mathbf{C_B}| \leq |\mathbf{Z}| + 1$. Therefore, this makes V_0 also an ancestor or X, fulfilling all requirements of a downstream choke point. □

The result for Theorem 13 depends on this standard theorem (Darmois, 1953; Skitovitch, 1953):

Theorem 19 (Darmois-Skitovitch Theorem) *Let $e_1, \ldots, e_n$ be independent random variables, $n \geq 2$. Let $v_1 = \sum_{i=1}^{n} \alpha_i e_i$, $v_2 = \sum_{i=1}^{n} \beta_i e_i$ for some coefficients $\{\alpha_i\}$, $\{\beta_i\}$. If v_1 and v_2 are independent, then those e_j for which $\alpha_j \neq 0$, $\beta_j \neq 0$ are Gaussian.*

The idea is that if we assume $\{e_i\}$ are not Gaussian, $\{V_1, V_2\}$ share a common source if and only if they are dependent. Shimizu et al. (2006) and Entner et al. (2012) present a deeper discussion on how this theorem is used in causal discovery.

For the main results, we will assume that particular algebraic (polynomial) identities implied by the model graph do not vanish at the particular parameter values of the given model (which we called "almost everywhere" results in the theorem). We will in particular consider ways of "expanding" the structural equations of each vertex according to *exogenous* variables, that is, any variable which is either an error term or latent variable (assuming without loss of generality that latent variables have no parents).

For each vertex V_k in the model, and each exogenous ancestor E_m of V_k, let $\mathcal{P}_{km}$ be the set of all directed paths from E_m to V_k. For each path $p \in \mathcal{P}_{km}$, define $\phi_{kmp} = \prod_j \lambda_{jj'}$, the product of all coefficients along this path for $V_j \in \mathbf{V} \cap p$ where $V_{j'} = p \cap par_{\mathcal{G}}(j)$ (that is, $V_{j'}$ is the parent of V_j in this path. We multiply coefficients following a sequence $E_m \to \cdots \to V_{j'} \to V_j \to \cdots \to V_k$). From this,

$$V_k = \sum_{E_m \in \mathcal{A}_{\mathcal{G}}(k)} \sum_{p \in \mathcal{P}_{km}} \phi_{kmp} E_m, \tag{5}$$

where $A_{\mathcal{G}}(k)$ is the set of exogenous ancestors of V_k, where for $E_m = e_k$ we have $\phi_{kmp} \equiv 1$ and path p is given by the single edge $e_k \to V_k$. We refer to the idea of expansion a few times in the proofs as a way of describing how models can be written as polynomial functions of the parameters $\theta_{\mathcal{G}} = \{\lambda_{ij} \mid V_j \in par_{\mathcal{G}}(i)\} \cup \{\omega_i\}$.

Overall, for a LiNGAM model $\mathcal{M}$ with DAG $\mathcal{G}$, we denote by $\mathcal{X}_{\mathcal{G}}$ the set of exogenous variables of $\mathcal{M}$, and by the *expanded graph* of $\mathcal{M}$ the graph $\mathcal{G}$ augmented with the error terms and the corresponding edges $e_i \to V_i$ for all observable vertices V_i in $\mathcal{G}$.

The main result used in the proof of Theorem 13 comes from the following Lemma. Notice that the non-Gaussianity assumption and the Darmois-Skitovitch Theorem are not necessary for its proof.

Lemma 20 *Let $\mathbf{O} \cup \mathbf{U}$ be the set of variables in a zero-mean LiNGAM model $\mathcal{M}$ with graph $\mathcal{G}$, where $\mathbf{U}$ are the latent variables of the model. Let $\mathbf{O}_{\setminus i} \equiv \mathbf{O} \setminus \{O_i\}$ for any $O_i \in \mathbf{O}$. Let $r_i \equiv O_i - \mathbf{a}^{\intercal}\mathbf{O}_{\setminus i}$ be the residual of the least-squares regression of O_i on $\mathbf{O}_{\setminus i}$, with $\mathbf{a}$ being the corresponding least-squares coefficients. Then, almost everywhere, r_i can be written as a linear function of the exogenous variables of $\mathcal{M}$, $r_i = \sum_{E_m \in \mathcal{X}_{\mathcal{G}}} c_m E_m$, where $c_m \neq 0$ if and only if O_i is d-connected to E_m given $\mathbf{O}_{\setminus i}$ in the expanded graph of $\mathcal{M}$.*

Proof of Lemma 20. Without loss of generality, assume that each latent variable in $\mathbf{U}$ has no parents. We will sometimes use X_k as another representation of any particular model variable (observable, latent or error term), with the index k indicating particular variables in $\mathbf{O} \cup \mathbf{U}$ and error terms, depending on the context.

Choose some $O_i \in \mathbf{O}$. One way of obtaining r_i is by first performing least-squares regression of each model variable X_k on O_j, for some $O_j \neq O_i$ in $\mathbf{O}$, and calculating residuals $X_k^{(1)}$. Among model variables in $\mathbf{O}$, define $\mathbf{O}^{(1)}$ as the set of all residuals $\{O_k^{(1)}\}$, $k \neq j$. That is, the superscript (1) here denotes that we have regressed all observable variables on one arbitrary observable $O_j \neq O_i$, collecting the residuals $O_k^{(1)}$ of the regression of O_k on O_j for all variables other than O_j. We will repeat the process while incrementing the superscript and removing one element out of $\mathbf{O}^{(m)}$ at a time for $m = 2, 3, \ldots, n-1$, where n is the size of $\mathbf{O}$. This is done by the process of picking some $O_{j'}^{(1)} \in \mathbf{O}^{(1)} \setminus \{O_i^{(1)}\}$ and regressing

$\{O^{(1)}_{k\neq j'}\}$ on $O^{(1)}_{j'}$ to obtain $\mathbf{O}^{(2)}$, iterating until we are left with $\mathbf{O}^{(n-1)}$ containing the single element $O^{(n-1)}_i$, resulting in $O^{(n-1)}_i = r_i$. The elimination sequence can be arbitrary.

Let O_j be a vertex in $\mathbf{O}_{\setminus i}$. Let λ_{km} be the structural coefficient between O_k and any $X_m \in \mathbf{O} \cup \mathbf{U}$. We define $\lambda_{km} \equiv 0$ if X_m is not a parent of O_k in $\mathcal{G}$. Since

$$O_k = \lambda_{kj} O_j + \sum_{X_m \in par_{\mathcal{G}}(k) \setminus O_j} \lambda_{km} X_m + e_k,$$

we have

$$\sigma_{kj} = \lambda_{kj}\sigma_{jj} + \sum_{X_m \in par_{\mathcal{G}}(k) \setminus O_j} \lambda_{km}\sigma_{mj} + \sigma_{e_k j},$$

where $\sigma_{e_k j}$ is the covariance of e_k and O_j and σ_{mj} here represents the covariance of X_m and O_j. Dividing both sides by σ_{jj} implies

$$a^{(1)}_{kj} = \lambda_{kj} + \sum_{X_m \in par_{\mathcal{G}}(k) \setminus O_j} \lambda_{km} a^{(1)}_{mj} + a^{(1)}_{e_k j}, \tag{6}$$

where $a^{(1)}_{e_k j}$ is the least-squares regression coefficient of e_k on O_j. This means $O^{(1)}_k = O_k - a^{(1)}_{kj} O_j$ can be written as

$$O^{(1)}_k = \sum_{X_m \in par_{\mathcal{G}}(k) \setminus O_j} \lambda_{km} X^{(1)}_m + e^{(1)}_k \tag{7}$$

with $X^{(1)}_m$ and $e^{(1)}_k$ defined analogously.

We can iterate this process until we are left with r_i:

$$r_i = \sum_{U_k \in par_{\mathcal{G}}(i) \cap \mathbf{U}} \lambda_{ik} U^{(n-1)}_k + e^{(n-1)}_i. \tag{8}$$

Variable $U^{(n-1)}_k$ is the residual of the regression of U_k on $\mathbf{O}_{\setminus i}$, similarly for $e^{(n-1)}_i$.

What we will show next is that within (8) each $U^{(n-1)}_k$ and $e^{(n-1)}_i$ can be expanded as polynomial functions of $\theta_{\mathcal{G}}$ and $\mathcal{X}_{\mathcal{G}}$, and the end result will contain non-vanishing monomials that are a (linear) function of only the exogenous variables E_m which are d-connected to O_i given $\mathbf{O}\setminus O_i$ in the expanded graph of $\mathcal{M}$. Since the monomials cannot vanish except for a strict subset of lower dimensionality than that of the set of possible $\theta_{\mathcal{G}}$, the result will hold almost everywhere.

Since we are free to choose the elimination ordering leading to r_i, as they all lead to the same equivalent relation (8), let us define it in a way that a vertex can be eliminated at stage t only when its has no ancestors in $\mathbf{O}^{(t-1)}$ (where $\mathbf{O}^{(0)} \equiv \mathbf{O}_{\setminus i}$). We will define the meaning of the ancestral relationship for elements of $\mathbf{O}^{(t)}$ shortly in terms of DAGs $\mathcal{G}^{(t)}$, but for $t = 0$ it is the one implied by the expanded version of $\mathcal{G}$.

For $t = 1$, the only exogenous variables which will have a non-zero coefficient multiplying O_j in the least-squares regression are the parents of O_j in the expanded graph, since O_j has

no other ancestors.[14] Let $U^{(1)}_k$ be the residual of some latent parent of O_j,

$$U^{(1)}_k = U_k - a_{kj} \left(\sum_{E_m \in \mathcal{A}_{\mathcal{G}}(j)} \sum_{p \in \mathcal{P}_{jm}} \phi_{jmp} E_m \right), \tag{9}$$

where $\phi_{jmp} \equiv \lambda_{jm}$ if E_m is a latent variable, or 1 if $E_m = e_j$. Moreover, $a_{kj} = \lambda_{jk}\omega_{kk}/\sigma_{jj}$, where ω_{kk} is the variance parameter of U_k and σ_{jj} is a polynomial function of $\theta_{\mathcal{G}}$. We can multiply both sides of the equation above by σ_{jj} (as well all equations referring to any $O^{(1)}_k$ or $X^{(1)}_m$ such as (7)) to get a new system of variables that is polynomial in $\theta_{\mathcal{G}}$. We will adopt this step implicitly and claim that from (9) we have that $U^{(1)}_k$ can be expanded as parameters that are polynomial functions of $\theta_{\mathcal{G}}$. Moreover, it is clear from (9) that there will be at least one non-vanishing monomial containing each E_m. In what follows, we refer to any expression analogous to (9) as the *expansion* of $U^{(t)}_k$ for $t = 1, 2, \ldots, n-1$.

We define a DAG $\mathcal{G}^{(1)}$ with vertices $\{X^{(1)}_m\}$, where the children of $X^{(1)}_m$ are defined as all and only the $O^{(1)}_k$ such that X_m is a parent of O_k in the original extended graph of the model. That is, $\mathcal{G}^{(1)}$ is the extended graph over residuals after the first regression. The respective model $\mathcal{M}^{(1)}$ is given by equations of type (7) with parameters coming from $\theta_{\mathcal{G}}$.[15]

For any $t > 1$, let $O^{(t)}_j$ be the vertex being eliminated. Each $U^{(t)}_k$ in which $U^{(t-1)}_k$ is a parent of $O^{(t-1)}_j$ in $\mathcal{G}^{(t-1)}$ will be a polynomial function of $\theta_{\mathcal{G}}$ and a linear function of the union of the exogenous variables present in the expansion of each parent of $O^{(t-1)}_j$: the expansion analogous to (9) in the new model will always introduce new symbols $\lambda_{j\star}$ into existing monomials, or create new monomials with e_j, as vertex $O^{(t-1)}_j$ had no eliminated descendants up to iteration t. As such, no exogenous variable will be eliminated from the algebraic expansion of the respective $U^{(t)}_k$.

Finally, the expansion of $\lambda_{ik}U^{(n-1)}_k$ in (8) will not cancel any monomial in the expansion of some other $\lambda_{ik'}U^{(n-1)}_{k'}$: since U_k and $U_{k'}$ are both parents of O_i, no monomial in the expansion of U_k can differ from a monomial in the expansion of $U_{k'}$ by a factor of $\lambda_{ik}\lambda_{ik'}$. So (8) will depend algebraically on the union of the exogenous terms leading to each $U^{(n-1)}_k$.

To prove the Lemma, we start by pointing out that $U^{(n-1)}_k$ will have a latent/error parent of some O_j in its expansion if and only if there is at least one sequence of vertices $(O_c, \ldots, O_j)$ where O_c is an observable child of U_k and any two consecutive elements in this sequence have at least one common latent parent in $\mathcal{G}$ (the sequence can be a singleton, $O_c = O_j$). To see this, notice that the different $U^{(t)}_k$ form an equivalence relation: each $U^{(t)}_k$ with a $O^{(t-1)}_j$ child which is being eliminated at iteration t will include into its expansion the exogenous variables found in the expansion of the other parents of $O^{(t-1)}_j$. This partitions $\mathbf{O}_{\setminus i}$ into sets in which each vertex O_j can "reach" some other vertex O_k by first moving to some $O_{j'}$ which shares a latent parent with O_j and which can "reach" O_k. The latent parents of $\mathbf{O}$ are then partitioned according to their observed children.

14. Assuming O_j is not a child of O_i. In this case, without loss of generality we assume that the parents of O_i are added to the parents of O_j, and remove O_i from the model at any iteration t.
15. To be more precise, polynomial functions of such parameters, as we are implicitly multiplying each equation by σ_{jj}.

To finalize the proof, suppose O_i is d-separated from a latent/error parent E_m of O_j given $\mathbf{O}\backslash O_i$. This happens if and only if all latent parents of O_i (and e_i) are d-separated from E_m given $\mathbf{O}\backslash O_i$. Let U_k be a latent parent of O_i (or its error term). Then $U_k^{(n-1)}$ cannot have E_m in its expansion. If this was the case, by the previous paragraph U_k would be d-connected to all latent parents of O_j, meaning O_i would be d-connected to them. This implies $c_m = 0$. Conversely, suppose O_i is d-connected to the error term or a latent parents of O_j given $\mathbf{O}\backslash O_i$. Then again by the previous paragraph, for any latent parent U_k of O_i, $U_k^{(n-1)}$ will have the latent parents of O_j as terms in its expansion, implying $c_m \neq 0$ almost everywhere. □

We can now prove Theorem 13.

Proof of Theorem 13. Considering the system for $\{O_i, Y\} \cup \mathbf{Z}$, we can represent the model in an equivalent way where all other variables are latent variables, and all latent variables are exogenous. Applying Lemma 20 to both r_i and r_y, and by Theorem 19, these variables will be dependent if and only if they are a non-trivial linear function of at least one common exogenous variable E_m in the model. Again by Lemma 20, this happens if and only if O_i is d-connected to E_m given $\mathbf{Z}$, and Y is d-connected to E_m given O_i and $\mathbf{Z}$. If Y is d-connected to E_m given $\mathbf{Z}$ only, we can create an active non-directed path between O_i and Y given $\mathbf{Z}$ by the concatenation of the (O_i, E_m) path with (Y, E_m) path: since E_m is exogenous, the path is non-directed as it should include the subpath $V \leftarrow E_m \rightarrow V'$ for some V, V'. If Y is not d-connected to E_m given $\mathbf{Z}$ only —that is, we must condition on O_i also to d-connect them— then Y must be d-connected to O_i given $\mathbf{Z}$ by a path that is into O_i. This path is non-directed as Y is not an ancestor of O_i by assumption, and the claim again follows. □

Remarks: The assumptions are stronger than, for instance, the ones used in the proofs of Tashiro et al. (2014). A closely related result in that paper is its Lemma 2, a result identifying the dependence between the residual of the regression of a variable on its children. It does not use any variation of the faithfulness assumption. This is because, in their context, it is enough to detect the dependence between the residual and *some* children. So if some path cancellations take place, some other path cancellations cannot occur. But we need the dependence of our r_i and every relevant error term, because we cannot claim that r_i depends on *some* error terms or latent variables, while r_y depends on *some* error terms or latent variables, if these two sets do not overlap. Although some of the ideas by Tashiro et al. (2014) could be used in our context to build partial models and from them deduce instrumental variables, it goes against our framework of solving a particular prediction problem (causal effect of a target treatment-outcome) directly, instead of doing it by recovering parts of a broader causal graph.

Finally, we have not provided an explicit discussion on how to validate the non-Gaussianity assumption by testing the non-Gaussianity of the residuals, as done by Entner et al. (2012). Or, more precisely, showing which assumptions are necessary so that testing non-Gaussianity of the residuals is equivalent to testing non-Gaussianity of the error terms. This is left as future work.

Algorithm 5 IV-TETRAD^{++}

Input: $\mathcal{D}$, a sample from the joint distribution of random variables $\mathbf{O} \cup \{X, Y\}$; K, number of instruments to consider; S, the number of mixture components; M, the number of MCMC iterations; π, a prior for a mixture of Gaussian model for $\mathbf{O} \cup \{X, Y\}$ with M components; α_{cut} and τ, two threshold parameters
Output: a distribution over equivalence classes of differential causal effect of X on Y
Initialize $\mathcal{C}_{Bayes}$ as the empty set
Run a MCMC algorithm to obtain M samples from the posterior distribution of a Bayesian mixture of S Gaussians model with data $\mathcal{D}$ and prior π
for $m = 1, 2, \ldots, M$ **do**
  Let $\Sigma^{(m)}$ be the m-th MCMC sample of the covariance matrix of $\mathbf{O} \cup \{X, Y\}$
  Let $\hat{\lambda}_i \leftarrow \sigma^{(m)}_{w_i y.\mathbf{O}_{\backslash i}} / \sigma^{(m)}_{w_i x.\mathbf{O}_{\backslash i}}$ for each $W_i \in \mathbf{O}$
  Discard all $\hat{\lambda}_i$ such that $|\hat{\lambda}_i| > \alpha_{cut}$
  Initialize $\mathcal{C}^{(m)}$ as the empty set, and $\mathbf{W}$ as $\mathbf{O}$
  while $|\mathbf{W}| \geq K$ **do**
    Sort $\mathbf{W}$ as $W^{(1)}, \ldots, W^{(|\mathbf{W}|)}$ according to the corresponding $\{\hat{\lambda}\}$
    Find $j \in \{1, 2, \ldots, |\mathbf{W}|\}$ that minimizes $\textsc{ScoreTetrads}^{\star}(W^{(j:j+K-1)}, \mathbf{O}, X, Y, \Sigma^{(m)})$
    $\mathbf{W}_j \leftarrow \textsc{ExpandTetrads}^{\star}(j, K, \mathbf{W}, \mathbf{O}, X, Y, \Sigma^{(m)})$
    if $\textsc{TestTetrads}^{\star}(\mathbf{W}_j, \mathbf{O}, X, Y, \Sigma^{(m)})$ **then**
      $\hat{\lambda}_{TSLS} \leftarrow \text{TSLS}^{\star}(\mathbf{W}_j, \mathbf{O}\backslash\mathbf{W}_j, X, Y, \Sigma^{(m)})$
      $\mathcal{C} \leftarrow \mathcal{C} \cup \hat{\lambda}_{TSLS}$
    end if
    $\mathbf{W} \leftarrow \mathbf{W}\backslash\mathbf{W}_j$
  end while
  Add set $\mathcal{C}^{(m)}$ as an element of $\mathcal{C}_{Bayes}$
end for
return $\mathcal{C}_{Bayes}$

Appendix C. A Bayesian Projections Variation of IV-TETRAD^{+}

Algorithm 5 is a method that relies on substituting hypothesis testing by a thresholding mechanism over posterior samples of an unconstrained distribution. This algorithm is referred to in Section 5.3.2.

TestTetrads* takes the correlation matrix implied by $\Sigma^{(m)}$ and returns false if

$$|\rho^{(m)}_{w_i x.\mathbf{O}_{\backslash i}} \rho^{(m)}_{w_j y.\mathbf{O}_{\backslash j}} - \rho^{(m)}_{w_i y.\mathbf{O}_{\backslash i}} \rho^{(m)}_{w_j x.\mathbf{O}_{\backslash j}}| > \tau.$$

ExpandTetrads* and TSLS* are just the straightforward variations of the original functions that take a covariance matrix instead of a data set as input.

References

J. Angrist and J-S Pischke. *Mostly Harmless Econometrics: An Empiricist's Companion.* Princeton University Press, 2009.

K. Bollen. *Structural Equations with Latent Variables.* John Wiley & Sons, 1989.

K. Bollen. Outlier screening and a distribution-free test for vanishing tetrads. *Sociological Methods and Research*, 19(1):80–92, 1990.

J. Bowden, G. Smith, P. Haycock, and S. Burgess. Consistent estimation in Mendelian randomization with some invalid instruments using a weighted median estimator. *Genetic Epidemiology*, 40(4):304–314, 2016.

C. Brito and J. Pearl. Generalized instrumental variables. *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI 2002)*, 2002.

S. Burgess, D. Small, and S. Thompson. A review of instrumental variable estimators for mendelian randomization. *Statistical Methods in Medical Research*, 26(5):2333–2355, 2017.

B. Chen, J. Tian, and J. Pearl. Testable implications of linear structural equations models. *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014)*, pages 2424–2430, 2014.

T. Chu, R. Scheines, and P. Spirtes. Semi-instrumental variables: a test for instrument admissibility. *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI 2001)*, pages 83–90, 2001.

G. Cooper. A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, 1(2):203–224, 1997.

G. Cooper. An overview of the representation and discovery of causal relationships using Bayesian networks. In C. Glymour and G. Cooper, editors, *Computation, Causation and Discovery*, pages 3–62. MIT Press, 1999.

G. Darmois. Analyse générale des liaisons stochastiques. *Review of the International Statistical Institute*, 21:2–8, 1953.

A. P. Dawid. Conditional independence in statistical theory. *Journal of the Royal Statistical Society, Series B*, 41(1):1–31, 1979.

V. Didelez and N. Sheehan. Mendelian randomization as an instrumental variable approach to causal inference. *Statistical Methods in Medical Research*, 16(4):309–330, 2007.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

D. Entner, P.O. Hoyer, and P. Spirtes. Statistical test for consistent estimation of causal effects in linear non-Gaussian models. *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, pages 364–372, 2012.

R. Foygel, J. Draisma, and M. Drton. Half-trek criterion for generic identifiability of linear structural equation models. *Annals of Statistics*, 40(3):1682–1713, 2011.

C. Goutis and C. Robert. Model choice in generalised linear models: a Bayesian approach via Kullback-Leibler projections. *Biometrika*, 85(1):29–37, 1998.

A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. *Advances in Neural Information Processing Systems*, 20:585–592, 2007.

P. Hoyer, S. Shimizu, A. Kerminen, and M. Palviainen. Estimation of causal effects using linear non-Gaussian causal models with hidden variables. *International Journal of Approximate Reasoning*, 49(2):362–378, 2008.

M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8(Mar):613–636, 2007.

H. Kang, A. Zhang, T. Cai, and D. Small. Instrumental variables estimation with some invalid instruments and its application to Mendelian randomization. *Journal of the American Statistical Association*, 111(513):132–144, 2016.

M. Kuroki and Z. Cai. Instrumental variable tests for directed acyclic graph models. *Proceedings of the 10th Workshop on Artificial Intelligence and Statistics (AISTATS 2005)*, pages 190–197, 2005.

M. H. Maathuis, M. Kalisch, and P. Bühlmann. Estimating high-dimensional intervention effects from observational data. *Annals of Statistics*, 37(6A):3133–3164, 2009.

S. Mani, G. Cooper, and P. Spirtes. A theoretical study of Y structures for causal discovery. *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI2006)*, pages 314–323, 2006.

S. Morgan and C. Winship. *Counterfactuals and Causal Inference: Methods and Principles for Social Research*. Cambridge University Press, 2nd edition, 2015.

C. Nowzohour, M. Maathuis, and P. Bühlmann. Structure learning with bow-free acyclic path diagrams. *arXiv:1508.01717*, 2015.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf. Causal discovery with continuous additive noise models. *Journal of Machine Learning Research*, 15(Jun):2009–2053, 2014.

P. Phiromswad and K. Hoover. Selecting instrumental variables: A graph-theoretic approach. *Working paper. Available at SSRN: http://ssrn.com/abstract=2318552 or http://dx.doi.org/10.2139/ssrn.2318552*, 2013.

R. Ramsahai. Causal bounds and observable constraints for non-deterministic models. *Journal of Machine Learning Research*, 13(Mar):829–848, 2012.

T. Richardson and P. Spirtes. Ancestral graph Markov models. *Annals of Statistics*, 30(4): 962–1030, 2002.

T. Richardson, R. Evans, and J. Robins. Transparent parameterizatios of models for potential outcomes. In J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, editors, *Bayesian Statistics 9*, pages 569–610. Oxford University Press, 2011.

K. Sachs, O. Perez, D. Pe'er, D. Lauffenburger, and G. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

G. Shafer, A. Kogan, and P.Spirtes. Generalization of the tetrad representation theorem. *DIMACS Technical Report 93-68*, 1993.

S. Shimizu, P. Hoyer, A. Hyvärinen, and Antti Kerminen. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030, 2006.

S. Shimizu, P. Hoyer, and A. Hyvärinen. Estimation of linear non-Gaussian acyclic models for latent factors. *Neurocomputing*, 72(7):2024–2027, 2009.

R. Silva and R. Evans. Causal inference through a witness protection program. *Journal of Machine Learning Research*, 17(56):1–53, 2016.

R. Silva and A. Kalaitzis. Bayesian inference via projections. *Statistics and Computing*, 25 (4):739–753, 2015.

R. Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7(Feb):191–246, 2006.

W. Skitovitch. On a property of the normal distribution. *Doklady Akademii Nauk SSSR*, 89:217–219, 1953.

C. Spearman. "General intelligence," objectively determined and measured. *American Journal of Psychology*, 15(2):210–293, 1904.

P. Spirtes. Calculation of entailed rank constraints in partially non-linear and cyclic models. *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI 2013)*, pages 606–615, 2013.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. Cambridge University Press, 2000.

S. Sullivant, K. Talaska, and J. Draisma. Trek separation for Gaussian graphical models. *Annals of Statistics*, 38(3):1665–1685, 2010.

T. Tashiro, S. Shimizu, A. Hyvärinen, and T. Washio. ParceLiNGAM: A causal ordering method robust against latent confounders. *Neural Computation*, 26(1):57–83, 2014.

J. Wishart. Sampling errors in the theory of two factors. *British Journal of Psychology*, 19 (2):180–187, 1928.

Journal of Machine Learning Research 18 (2017) 1-34 Submitted 8/15; Revised 5/17; Published 11/17

Classification of Time Sequences using Graphs of Temporal Constraints

Mathieu Guillame-Bert mathieug@andrew.cmu.edu
Artur Dubrawski awd@cs.cmu.edu
Auton Lab, The Robotics Institute, School of Computer Science
Carnegie Mellon University, Pittsburgh, United States

Editor: Luc De Raedt

Abstract

We introduce two algorithms that learn to classify Symbolic and Scalar Time Sequences (SSTS); an extension of multivariate time series. An SSTS is a set of *events* and a set of *scalars*. An *event* is defined by a symbol and a time-stamp. A *scalar* is defined by a symbol and a function mapping a number for each possible time stamp of the data. The proposed algorithms rely on temporal patterns called Graph of Temporal Constraints (GTC). A GTC is a directed graph in which vertices express occurrences of specific events, and edges express temporal constraints between occurrences of pairs of events. Additionally, each vertex of a GTC can be augmented with numeric constraints on scalar values. We allow GTCs to be cyclic and/or disconnected. The first of the introduced algorithms extracts sets of co-dependent GTCs to be used in a voting mechanism. The second algorithm builds decision forest like representations where each node is a GTC. In both algorithms, extraction of GTCs and model building are interleaved. Both algorithms are closely related to each other and they exhibit complementary properties including complexity, performance, and interpretability. The main novelties of this work reside in direct building of the model and efficient learning of GTC structures. We explain the proposed algorithms and evaluate their performance against a diverse collection of 59 benchmark data sets. In these experiments, our algorithms come across as highly competitive and in most cases closely match or outperform state-of-the-art alternatives in terms of the computational speed while dominating in terms of the accuracy of classification of time sequences.

Keywords: classification, temporal data, sequential data, graphical constraint models, decision forests, symbolic and scalar time sequences, supervised learning

1. Introduction

Learning to classify from labeled training data is one of the flagship capabilities of machine learning. Classification is commonly used to model flat-table data, and in applications involving more complex representations such as (time) sequences or graphs, the information is typically "featurized" or "flattened" to fit the flat-table representation paradigm, so that standard classification algorithms can be readily applicable. In the case of symbolic time-stamped sequences, popular featurization techniques include representing time-since-previous-occurrence of an event, or computing frequency of occurrence (or other statistics or measures) over moving windows of time. Perhaps, a preferable alternative is to instead use classification algorithms capable of natively handling the temporal aspects of time-stamped sequences.

In this paper we introduce two parametric algorithms that belong to that specialized class. Both algorithms are designed to classify Symbolic and Scalar Time Sequences (SSTS); an expressive extension of the Time Series (TS) representation that remedies some of its limitations. In particular, SSTS extends the standard TS model by allowing non-uniform and asynchronous sampling, representing instantaneous sparse events as well as dense numeric sequences, and scaling well to highly-dimensional (multi-channel) application scenarios.

Both algorithms rely on a simple but expressive graphical pattern model for SSTS data called Graph of Temporal Constraints (GTC). A GTC is a conjunction of three types of conditions: (1) Existence of an event, (2) Constraint on the timestamp difference between two given events, and (3) Inequality constraints on the numerical values of a numerical time sequence at a time defined by a given event. Each GTC defines a Boolean function on an SSTS (or a TS).

The first algorithm presented in this paper, called the "GTC-Set", bootstraps training SSTSs and extracts fitting GTCs to form an ensemble model. Unlike methods relying on extracting all possible patterns and then applying a feature selection, our first algorithm directly build a final set co-dependent discriminant GTCs (all extracted GTC are used and no feature selection is applied). For this reason, the set of build GTCs is much smaller than the set of all possible GTCs. At the end of the learning process, each thusly extracted GTC is used as an individual "voter" in the classification procedure.

The second algorithm, called "GTC-DF", builds a bootstrapped decision forest where the condition in each decision node is defined as a GTC. Path of inference through each decision tree in this forest depends on whether a query STSS fulfills the constraints of the particular node's GTC or not. Here again, the GTC attached to each node is directly constructed on-the-fly instead of being selected from a pool of existing GTCs.

To assess performance of the GTC-Set and the GTC-DF algorithms, we evaluated them on a diverse collection of 59 temporal sequence classification data sets (all of them are available publicly), and compared against 13 alternative algorithms found in the literature. Results show that our algorithms are highly competitive and in most cases outperform state-of-the-art considered alternatives, both in terms of computing speed and accuracy of classification. Our algorithms are especially competitive for unaligned temporal sequence data, i.e. the data for which the value of an observation at a particular time t has no distinct predictive value, or in other words: classification performance does not depend on when we start observing the data. We also show how the users can interpret learned models based on the proposed GTC representation. Interpretability is often an important factor that determines practical usability of machine learning solutions in the real world, and our presented framework satisfies this requirement well.

This paper is structured as follows: Section 2 describes the SSTS representation and formalizes the classification problem for SSTS. Section 3 discusses related work and proposes a three-category taxonomy of time series classification algorithms. Sections 4 and 5 introduce the GTC representation and the GTC-Set/DF models. Section 6 presents and details our two classification algorithms. Section 7 summarizes results of empirical evaluations and comparisons, and Subsection 7.7 shows how a user can interpret a GTC-Set model. Concluding remarks wrap up the paper in Section 8.

©2017 Mathieu Guillame-Bert and Artur Dubrawski.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/15-403.html.

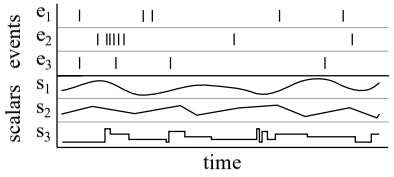


Figure 1: Example of an SSTS with three event symbols and three scalar variables.

2. Problem Definition

A Symbolic and Scalar Time Sequence (SSTS) is a temporal data set representation with a high power of expression. It is defined as follows:

Let $E = \{e_1, \ldots, e_n\}$ be a set of n distinct symbols called *event symbols*. Let $S = \{s_1, \ldots, s_m\}$ be a set of m distinct symbols called *scalar symbols*. An *event* is a tuple $\langle e_i, x\rangle$ where $x \in \mathbb{R}$ is a timestamp. A *scalar mapping* is a tuple $\langle s_i, f_i\rangle$ where f_i is a function $\mathbb{R} \mapsto \mathbb{R}; t \to v$ that maps each time-stamp t to a value v. Finally, a *symbolic and scalar time sequence* $d = \langle P, Q\rangle$ is a tuple where P is a set of events and Q is a set of scalar mapping. By convention, an event of type e_i is said to "happen" at time t if $\langle e_i, t\rangle \in P$, and the value of the scalar s_j at time t is $f_i(t)$ with $\langle s_i, f_i\rangle \in Q$. Figure 1 shows an example of SSTS. Note that an SSTS can contain several events with the same (event) symbol.

The SSTS classification problem is defined as follows: Given a set of labeled SSTSs and a set of unlabeled SSTSs drawn from the same underlying distribution, infer the labels of the unlabeled SSTSs.

3. Related Work

Published research into temporal data classification can be grouped into three categories: *distance based methods*, *flattening methods* and *model based methods*. This categorization does not take into account possible data pre-processing steps.

Distance based methods rely on a definition of a distance between pairs of time series, and the use of a distance-based classification algorithm (such as e.g. k-nearest neighbors). If the distance metric requires tuning of its parameters to optimize the data fit, the observations available for training are split into separate training and validation samples. Methods falling into this category include nearest neighbor and Euclidean distance (NN+ED) (Ding et al., 2008), NN+L1 (Ding et al., 2008), NN+Linf (Ding et al., 2008), DTW (Berndt and Clifford, 1994), LCSS (Vlachos et al., 2002), ERP (Chen et al., 2005), EDR (Chen and Ng, 2004), DISSIM (Frentzos et al., 2007), SWALE (Morse and Patel, 2007), TQUEST (Aßfalg et al., 2006).

Flattening methods work in two steps. Firstly, temporal data is transformed into a flat table in which each time series is decomposed into a set of features and then it is represented by a single row of the table. Secondly, a standard classification algorithm is applied to thusly prepared flat-table data (Random Forest (Breiman, 2001), SVM (Meyer et al., 2003), Artificial Neural Networks, or AdaBoost (Freund and Schapire, 2009) have all been used in this context). Types of features that are commonly used for flattening include basic

statistics of the temporal data computed over moving time windows (mean, standard deviation, skewness, kurtosis, etc.), parameters prevalent in forecasting applications (trends, periodicity), and popular temporal signal projections (Fourier Transform, Wavelet Transforms, Singular Value Decomposition, etc.). Methods that rely on extracting a large amount of patterns from raw data followed by either applying feature selection and/or applying one of the classification algorithms to identify combinations of patterns that are effective on the task, fall into this category (Guillame-Bert and Dubrawski, 2014; Batal et al., 2012; Liu et al., 1998; Li et al., 2001). In this case, large amount of patterns are extracted according to user defined constraints (e.g. minimum support) and scores (e.g. F1), before being fed into a classifier. However, these methods are often slow because of the large amount of patterns to extract. And because discriminant power for classification does not always imply high score for a pattern, the methods may miss important patterns.

Finally, *direct-model-based methods* directly extract from data a model that contains temporal information. Examples of such methods include Rodríguez and Alonso (2004) and Deng et al. (2013) who have modified the decision forest framework (Breiman, 2001) to handle time series. In these approaches, conditions in the non-leaf nodes of the decision trees are of the form $f_i(t_1, t_2) < \alpha$ with $f_i(t_1, t_2)$ function computing a predefined statistic on all time series points between t_1 and t_2. For Rodríguez and Alonso (2004), $f_i(t_1, t_2)$ is the mean, standard deviation or the dynamic time warping distance to a template time series. For Deng et al. (2013), f is either the mean, standard deviation, or an estimation of the slope. Methods presented in Rodriguez et al., Deng et al. and Ye and Keogh (2009) do not fit the flattening paradigm because the features or conditions are determined at the time when the model is built instead of at a separate pre-processing stage. Inductive Logic Programming (Muggleton, 1991) (ILP) is a general framework that is capable to express many temporal patterns. Rodriguez et al. (2000) has shown an example of how to use ILP to classify time series. However, from our experience and because of their general nature, it is unclear how ILP solvers could learn efficiently temporal patterns containing time constraints which are not predefined by the users. The two algorithms proposed in this paper fall in this last category of methods.

One of the two proposed algorithms (GTC-Set) extracts large amount of GTC patterns (see definition in section 4), and use them with a simple voting mechanism (each active pattern casts a vote—the most represented class is predicted). This algorithm can be described trough the pattern-based classification framework described by Bringmann et al. Bringmann et al. (2011). In this framework, a pattern-based classification algorithm is divided into three parts: (1) The extraction of a pool of patterns, (2) the selection of a subset of patterns, and (3) the training of a classifier based on the patterns. The class of an algorithm is define by whether or not the labels are used in the stage (1) and (2) i.e. supervised or non-supervised pattern extraction and pattern selection: The main advantages of using labels in early stages is to "direct" the pattern extraction and selection, and to generate a smaller and more specialized (and hopefully more performant) set of patterns. On the other hand, not using labels in the first stage allows the use well studied of frequent pattern mining algorithms (e.g. frequent itemset mining (Agrawal and Srikant, 1994; Han et al., 2004), frequent sequence mining (Srikant and Agrawal, 1996; Zaki, 2001; Pei et al., 2001), frequent graph mining (Yan et al., 2008; Nowozin et al., 2007)). In this framework, the GTC-Set algorithm we propose is a "Model-dependent iterative mining"

algorithm (i.e. the labels are used in the pattern extraction stage), and it does not contains a pattern selection nor training of a classifier stages: All extracted patterns are directly and independently used. Such simple application of extracted patterns is only possible because of the specific extraction of the GTC-Set algorithm. For example, directly and independently applying all patterns extracted from a frequent itemset mining algorithm would lead to poor results.

Supervised Descriptive Rule Discovery (Novak et al., 2009) (proposed as a unification of Contrast Set Mining (Bay and Pazzani, 2001), Emerging Pattern Mining (Dong and Li, 1999), and Subgroup Discovery (Klösgen, 1996)) is the supervised extraction of association rules from labeled transactional data sets for the purpose of user interpretation and data discovery. Rules are extracted to maximize a user defined interest score often measuring a lift-life measure or a statistical distance from a null-space model. Unlike pattern-based classification, these rules are not designed to be used for classification. In section 7.7, we discuss how to extract a small but representative subset of GTC patterns from set of patterns extracted by the GTC-Set algorithm. While being less performant for the classification task than the original set of patterns, the selected set of patterns aims to be easier to interpret by a user. The GTC-Set algorithm conjugated with pattern selection method can be seen as a solution to Supervised Descriptive Rule Discovery on SSTS. Interestingly, and contrary to most of the Supervised Descriptive Rule Discovery literature, the proposed method build an interpretable model from a classifier.

Unlike itemset mining, sequential mining allows for the definition and use of a relation of order in between items. This order is often used to represent the temporal sequence of the items: One item can be anterior, posterior or concurrent to another item. However, time distance is not taken into account by sequential mining, and in many dynamical systems, relations such as the time distance or the overlapping between items is discriminant.

In the literature, learning algorithms dealing with temporally discrete data (e.g. time point events) face the challenge of representing and extracting temporal constraints between pairs of time points. In most of the literature such constraints are represented with a subset C of $\mathbb{R}$: Given two data points with timestamps respectively t_1 and t_2, the constraint is satisfied if and only if $t_2 - t_1 \in C$.

This formulation has a high power of expression, but it can be prone to over-fitting given the potentially infinite number of candidate patterns and many available and applicable algorithms for pattern extraction are intractable. Solutions found in the literature rely on two principles: Constraining possible values of C, and/or defining an inexpensive to optimize score function on C. The simplest and the most popular solution is to require for the user to specify a small set of candidates for C (Mannila et al., 1997). Rodríguez and Alonso (2004) and Rodriguez et al. (2000) only consider segments of the form $[2^i, 2^j]$ where i and j are integers. Deng et al. (2013) consider a small number of randomly generated segments. Ye and Keogh (2009) show the use of a decision tree where node conditions are expressed as thresholds on the distance between the signal and a subset of a time series called "shapelet". During training, candidate shapelets are extracted from all possible subsets of training data, and pruning-based optimization is used to ensure tractability. Dousson and Vu Duong (1999) consider all segments. Recursively, the algorithm selects the segment $[a, b]$ that minimizes the surface $b - a$ while holding a user-defined minimum support. Guillame-Bert (2012) considers all possible unions from a large set of segments

defined by the user $\{[ai + b, a(i + 1) + b]\}_{i\in[m,n]}$. Recursively, the algorithm selects the set of segments that maximizes information gain. The maximization problem is solved approximately using a forward or backward gradient descent optimization. The resulting constraint is not necessarily a contiguous segment. Guillame-Bert and Dubrawski (2014) consider all combinations $[b_i, b_j]$ where $\{b_i\}$ are provided by the user, and i and j are selected to maximize information gain. All possible combinations of i and j can be evaluated thanks to an efficient specialized data encoding.

The algorithms proposed in this work rely on learning constraints of the type $[a, b]$ (a and b are not constrained). a and b are selected to maximize information gain. This is a strict improvement over Guillame-Bert and Dubrawski (2014). Efficient and exact, according to the information gain criterion, extraction of a and b is one of the main results of this work.

4. Graphs of Temporal Constraints

A Graph of Temporal Constraints (GTC) is a set of temporal constraints represented as a graph.

Definition 1 *A GTC is defined by a set of* tests $Q = \{q_1, \ldots, q_n\}$ *and a set of* constraints $C = \{c_1, \ldots, c_m\}$.

A test q_i is defined by an event symbol*, a positive or negative* flag*, and a (possibly empty) set of* scalar tests*. A q_i test with a positive flag is true at time t if an event with the given event symbol exists at time t, and if all its scalar tests are true at time t. A test with a negative flag is true if and only if the same test with a positive flag is impossible to satisfy.*

A scalar test *is an inequality constraint based on a scalar mapping value at a time t defined by the node. In this work, scalar tests are restricted to be inequalities between a single scalar mapping term and a threshold value.*

A constraint c_k *is defined by two tests q_i and q_j, a positive or negative flag, and $R \subset \mathbb{R}$ (a subset of the real numbers). A constraint with a positive flag is true if its both tests are true respectively at time t_i and t_j and if $t_j - t_i \in R$. A constraint with a negative flag is true if and only if the same constraint with a positive flag is false. In other words, a negative constraint $[a, b]$ is equivalent to the disjunctive constraint $]-\inf, a[\cup]b, +\inf[$. Additionally, we restrict GTC not to have conditions/edges starting from a negative test/node.*

This restriction is similar to the apriori constraint for association rule mining, and it guaranties that if built iteratively, the GTCs at successive steps have decreasing support.

A GTC can be seen as a directed graph were each vertex expresses the existential condition for an event (conditioned by a symbol), i.e. a test, and where each edge expresses a temporal constraint between the vertices it connects, i.e. a constraint. A GTC can contain undirected cycles and disconnected components. Fig. 2 shows two examples of GTCs.

Let us now introduce some aspects of notation used for describing GTCs. A GTC instance $\{k_i\}_{i\in1,\ldots,n}$ assigns times k_i to each of the tests q_i of GTC A so that all tests and all constraints of A are true. By convention, $A(d)$ denotes the set of instances of A in an SSTS d. Note that different instances can partially overlap with each other. By convention, A is true in an SSTS d if d contains at least one instance of A, i.e. $|A(d)| > 0$. Trace $T(q_i, d)$ of the test q_i of A on the SSTS d is defined as the set of timestamps associated

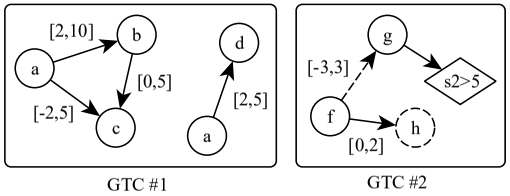


Figure 2: Two examples of GTCs. Dashed vertex (or edge) represents a vertex (or edge) with a negative flag. Diamond shapes represent scalar tests.

with the test q_i for any instances of A on d: $T(q_i, d) = \{k_i \mid \forall\{k_1, \ldots, k_i, \ldots, k_n\} \in A(d)\}$. Given a set of SSTS $D = \{d_1, \ldots, d_n\}$, $A(D)$ is the subset of SSTS for which A is true: $A(D) = \{d_i \mid d_i \in D \text{ with } |A(d_i)| > 0\}$. By convention, a GTC without tests is always true. Finally, given a set of SSTS D, the support of A is the fraction of SSTS where A is true: $supp(A) = |A(D)|/|D|$.

Testing the existence of a GTC in an SSTS (i.e. computing the existence of instances) can be done by (1) Extracting a set of spanning trees of the GTC (one tree for each of the GTCs disconnected components), (2) Independently matching each tree to the SSTS records while continuously, and (3) Checking any additional constraints that were not the part of the spanning trees. The cost of testing a GTC is polynomial with the number of events and exponential with the number of nodes. The smaller the surfaces of the temporal constraints between nodes, the less extensive the exploration, and the faster the GTC matching. However, in practice (cf. Experimental evaluation section below), the cost of testing GTCs is insignificant in comparison to the cost of learning these GTCs. It appears that because the constraints on the edges are selected to be discriminant, they tend to be restrictive and help in pruning large parts of the exploration space.

To the best of our knowledge, GTC is a novel representation, but it can be related to some existing temporal pattern representation frameworks. The closest are the Temporal Constraint Network (TCN) of Dechter (1991), the Chronicles of Dousson and Vu Duong (1999) and the Tita rules of Guillame-Bert (2012). TCN and Chronicles can be seen as reduced cases of GTCs in which tests and constraints do not have negative flags and in which there are no scalar tests. Tita rules allow node negation but do not allow undirected cycles or disjoint components.

5. Classification with GTCs

In this section, we introduce two SSTS classification models that use GTCs as their building blocks. The first model we call *GTC-Set*. A GTC-Set is an unstructured set of labeled (assigned to represent a particular unique class of STSS) GTCs. Given an unlabeled SSTS d, the predicted label for d is the most common label among all of the GTCs that are true on d. Note that this model would perform poorly if it were to contain all possible GTCs (even with a support constraint). Figure 3 shows an example of a GTC-Set.

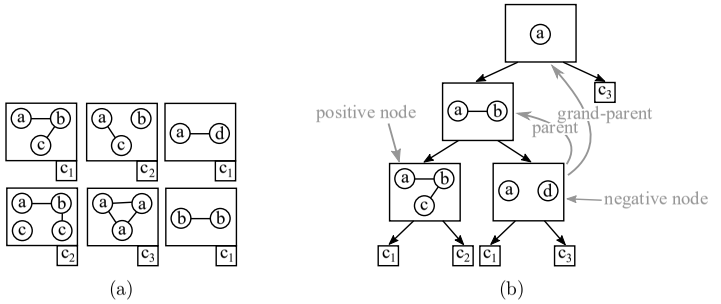


Figure 3: Examples of a GTC-Set (a) and a tree in a GTC-DF (b). Class labels in the leaf nodes are denoted with c_i.

The second model we call *GTC Decision Forest* (GTC-DF). A GTC-DF is a set of decision trees where each non-leaf node is associated with a particular GTC and it has two children. Additionally, the GTC attached to a decision tree node is constrained to be directly derived from the GTC of the parent node if connected by a positive decision tree branch, and from the grandparent node otherwise. This constraint is important and it will be discussed and justified in the next section. When applying a GTC-DF to classify a query in the form of a STSS, the decision trees are independently traversed in a familiar way. Upon arrival in a node, its associated GTC is tested for fitness to the data and the result of this test determines the direction of further traversal. As in the original decision forest algorithm (Breiman, 2001), class label distributions are agglomerated across the final active leaf nodes of each component tree, in order to obtain the joint prediction. Figure 3 shows an example of a tree in a GTC-DF.

Independence of components in GTC-Sets makes these models somewhat easier to interpret by the users than GTC Decision Forests because each of them can be considered individually. Conversely, interpretation of a GTC Decision Forest model, as well as its inference process in response to a particular query, requires consideration of GTCs corresponding to the subsequent nodes in each tree.

6. Learning Temporal Patterns

We introduce two algorithms to respectively extract GTC-Sets and GTC-DFs from a set of labeled SSTS instances. Both algorithms share the same sub-routines but differ in the structure of their main loops and therefore in the structure of the extracted GTCs. These algorithms can be seen as complementary. Experimentation shows that extracting GTC-DF is on average 40% faster, and it yields less complex models which typically perform slightly better when compared to GTC-Set models obtained from the same data (cf. Section 7.6). On the other hand, GTC-Sets allow easier interpretation of the results and the models themselves by the users (Section 7.7).

Both algorithms build GTCs by recursively applying constructive operations. In the process, selection of the next operation to apply is guided by maximizing the local information gain. This section presents GTC building operations, defines the concept of information gain of an operation, presents how to efficiently identify the operation with the highest information gain, and details the main loops of the two algorithms. Our proposed solution to how to efficiently extract optimal temporal constraints for the edges in GTCs is one of the main novelties of this work.

6.1 Constructive Operations on GTCs

Constructive operations on a GTC include:
O1: Addition of a new test (i.e. addition of a vertex),
O2: Addition of a constraint between two existing tests (i.e. addition of an edge),
O3: Addition of a scalar test to an existing test,
O4: Addition of a new test and addition of a constraint between this test and an already existing test.

Each operation requires specific parameters. O1 requires an event symbol and a positive/negative flag. O2 requires two existing tests, a flag and a subset of $\mathbb{R}$. O3 requires an existing test and a scalar test defined by a scalar symbol, a threshold value and an inequality direction ($<$ or $\geq$). O4 requires a combination of arguments of O1 and O2. Note that, because of the threshold value of the scalar test in O3, and because the need for a subset of $\mathbb{R}$ in O2 and O4, these operations have an infinite number of possible parameter values. Operation O4 is equivalent to operations 1 and 2 combined. We justify its existence later.

Both proposed algorithms require for the set of positive SSTS according to a GTC to be monotone decreasing when applying operations: Given a set D of SSTSs, a GTC A, an operation O and B the result of O on A. Algorithms require for the set of positive SSTSs according to B to be contained in the set of positive SSTSs according to A i.e. $B(D) \subset A(D)$. This implies that each operation reduces (albeit not strictly) the support of the GTC it is applied to. For this reason, operations O2-O4 cannot be applied on negative nodes.

The result of the negation of an operation O on a GTC A is a GTC B' such that $\forall d \in D, A(d) \Rightarrow (B(d) \Leftrightarrow \neg B'(d))$ and $\neg A(d) \Rightarrow (\neg B(d) \wedge \neg B'(d))$, where B is the result of O on A. Note: $\Leftrightarrow$ and $\neg$ are respectively the logical equivalence and logical negation. Except for trivial cases, the GTC resulting from the negation of an operation is more complex than the result of the operation. In the case of fully connected GTCs, the size of the GTC resulting from the negative operation is twice the size of the result from the direct operation. Instead, we define the approximate result of the negation of an operation. This approximation is not exact but it provides reasonable results as presented in Section 7.

Whether to use or not the approximate negation is the main difference between the two proposed algorithms. Figure 4 illustrates these operations and their associated approximate negations. Adding a constraint to a test with a negative flag would violate the support reduction property and it is therefore forbidden. The goal of O4 is to allow creating a connected test with a negative flag without violating the support reduction requirement.

As a simple example, consider the four GTCs shown in figure 5a. B is the result of an operation of type $O4$ on A. B' is the result of the negation of this operation. B'' is the

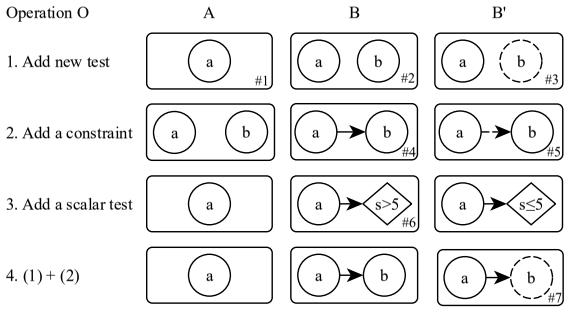
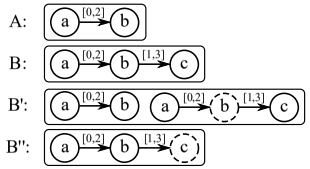


Figure 4: Illustration of the constructive operations on GTCs. Each row of graphs represents a specific operation. Column A represents the initial GTC. Column B shows the result of executing the particular operation on the initial GTC. Column B' depicts the result of the approximate negation of the operation. Dashed nodes and edges represent respectively tests and constraints with negative flags. Diamonds represent scalar tests. Description of the GTCs: #1: An event of type a is present. #2: An event of type a and an event of type b are present. #3: An event of type a is present, but no event of type b is present. #4: An event of type a and an event of type b are present, and they satisfy the constraint of the edge. #5: An event of type a and an event of type b are present, and they satisfy the complementary constraint of the edge. #6: An event of type a is present at time t such that the scalar s value is greater than 5 at time t. #7: An event of type a is present at time t_a such that there is no event b at time t_b such that t_a and t_b satisfy the constraint.

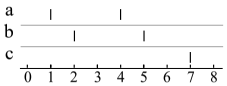
approximate result of the negation of this operation. A expresses that "There is an event a followed by an event b between 0 and 2 time units". B expresses that "There is an event a followed by an event b between 0 and 2 time units, itself followed by an event c between 1 and 3 time units". B' expresses that "There is an event a followed by an event b between 0 and 2 time units, but there is not an event a followed by an event b between 0 and 2 time units, itself followed by an event c between 1 and 3 time units". B'' expresses that "There is an event a followed by an event b between 0 and 2 time units, itself not followed by an event c between 1 and 3 time units". Figure 5b shows an SSTS B'' is true but where B' is not.

6.2 Information Gain Maximization

Both of the proposed GTC extraction algorithms are driven by the information gain maximization criterion. We define entropy H of a set of labeled SSTS D as the entropy of the



(a) Suppose a GTC A. B is the result of an operation of type $O4$ on A. B' is the result of the negation of this operation. B'' is the approximate result of the negation of this operation.



(b) Example of SSTS with B' false but B'' true.

Figure 5: Illustration of the negation and the approximate negation of an operation.

distribution of their labels

$$H(D) = \sum_i -p_i(D) \log p_i(D)$$

, with $p_i(D)$ representing the relative frequency of label i in D. Additionally, we assert $0 \log 0 = 0$.

Given a set of SSTSs $D = d_1, \ldots, d_n$, a GTC A such that A is true for all d_i, and an operation O, if B is the result of O on A, then information gain IG of the operation O on A can be defined as

$$IG = H(D) - \frac{|B(D)|}{|D|} H(B(D)) - \frac{|D \setminus B(D)|}{|D|} H(D \setminus B(D))$$

, where $H(D)$ is commonly called the initial (prior) entropy, and the rest of the right hand side of the formula is the weighed final (posterior) entropy.

During the training stage, given a GTC A, both algorithms rely on finding such constructive operation and its parameters that maximize the information gain. Except for operation O1, testing and evaluating the information gain over the range of possible parameter values is in most cases infeasible because of the infinite number of tests it might involve. We propose a solution based on the following four considerations: (1) Instead of considering all subsets of $\mathbb{R}$ (operations O2 and O3), we only consider contiguous segments of $\mathbb{R}$; (2) Since each finite data set consists of a finite number of SSTSs, and each SSTS is composed of a finite number of events, there is only a finite number of parameter values that need to be considered; (3) The procedure of extracting a plausible threshold for an operation O3, and the corresponding segments of $\mathbb{R}$, can be optimized for computational speed with an appropriate algorithm and an appropriate internal data structure; and (4) Random sampling can be used as an option to further speed up the extraction of the optimum threshold and the optimum segments of $\mathbb{R}$.

The next two subsections show how to build decision forests and sets of GTCs models using information gain maximization. Subsection 6.5 shows how to compute efficiently the threshold of operation O3 that maximize their information gain. Subsection 6.6 shows how to compute efficiently the segment of R that maximize their information gain for operations O2 and O4.

6.3 Learning Decision Trees and Forests of GTCs

We build decision trees and forests of GTCs similarly to how standard decision trees and forests are built (Quinlan, 1993; Breiman, 2001). The two differences are that: (1) Each decision node of a GTC tree is assigned a GTC instead of a single threshold condition on a single data feature as it is the case of the standard model; and (2) The structure of a GTC representing a node of a GTC tree is heavily constrained by the structure of the GTCs of its parent and grand-parent nodes.

The algorithm for training a GTC tree starts with a single empty GTC and it looks for the operation on it that yields the highest information gain. This operation is then applied, and the resulting GTC is used to define the root node constraint. The root node constraint is used to split the training SSTSs into two complementary subsets. The same procedure is further recursively applied to these subsets and additional nodes are appended to the tree until a stopping criterion is met. The GTC of a positive node of the decision tree will be derived by applying a single operation on the GTC attached to its parent. The GTC of a negative node of the decision tree will be derived by applying a single operation on the GTC attached to its grand-parent. This restriction has three benefits: (1) Search space for the new GTC is limited and therefore fast to explore. (2) The new GTC is constrained and therefore the risk for it to over-fit is mitigated. (3) GTCs of an arbitrary complexity can still be derived. A GTC Decision Forest (GTC-DF) can be obtained by bootstrapping data for GTC decision tree learning from the training set of SSTSs. Note that we typically also bootstrap the GTC-Set models and in fact use ensembles of GTC-Sets, cf. Algorithm 2. Pseudocode in Algorithm 1 provides details.

Algorithm 1: Extraction of a GTC decision forest from data.

input : D: A set of SSTSs
b: The number of bootstrap iterations (e.g. 20)
l: The minimum number of times a GTC is used (e.g. 5)
output: M: The set of extracted labeled GTCs
Algorithm
  for $i \leftarrow 1$ **to** b **do**
    $A \leftarrow$ an empty GTC
    `recursive_build`(D, A)
Subroutine `recursive_build`(D,*reference to* A)
  $M \leftarrow M \cup A$
  if $|A(D)| < l$ **then**
    $label(A) \leftarrow$ most frequent class in D with A true
    return
  $o \leftarrow$ find the operation on A with the highest information gain
  $B_2 \leftarrow A$
  $A \leftarrow$ result of O on A
  $B_1 \leftarrow A$
  set B_1 and B_2 to be respectively the positive and negative children of A
  `recursive_build`$(B_1(D), B_1)$
  `recursive_build`$(D \setminus B_2(D), B_2)$

6.4 Learning Sets of GTCs

This sub-section introduces the algorithm to extract a GTC set from data. This algorithm starts with a single empty GTC A and looks for the operation with the maximum information gain to extend its structure. This operation and the approximate negation of it are applied to the GTC A to create two new GTCs, respectively B_1 and B_2. A is then removed. The same process is next recursively applied to both B_1 and B_2 until a stopping criterion is met. Finally, each remaining (i.e. non removed) GTC is assigned to the most common class of the SSTS where it is true. Note that because of the approximated negation $B_1(D)$ and $B_2(D)$ are not necessary disjoint (the last line of the Algorithm 2).

Algorithm 2: Extraction of a GTC set.

input : D: A set of SSTSs
b: The number of bootstrap iterations to execute (e.g. 20)
l: The minimum number of times a GTC is used (e.g. 5)
output: M: The set of extracted labeled GTCs
Algorithm
  $A \leftarrow$ an empty GTC
  for $i \leftarrow 1$ **to** b **do**
    $M \leftarrow M \cup$ `recursive_build`(D, A)
Subroutine `recursive_build(D,A)`
  if $|D(A)| < l$ **then**
    $label(A) \leftarrow$ most frequent class in D with A true
    return A
  $o_1 \leftarrow$ find the operation on A with the highest information gain
  $o_2 \leftarrow$ approximate negation of o_1
  $B_1 \leftarrow$ result of o_1 on A
  $B_2 \leftarrow$ result of o_2 on A
  return `recursive_build`$(B_1(D), B_1) \cup$ `recursive_build`$(B_2(D), B_2)$ `// Note that it is not required nor expected that` $B_1(D) \cap B2(D) = \emptyset$

6.5 Finding the Optimal Scalar Tests

A scalar test is defined by an existing test q_i, a scalar symbol s_j, a threshold value α and the direction of inequality ($<$ or $\geq$). The problem of finding the optimal threshold value is different but related to the problem of finding the optimal threshold in the decision tree learning algorithm C4.5 (Quinlan, 1993). Unlike C4.5, our algorithms need to consider several scalar values for each observation/SSTS: one scalar value for each GTC instance on a given SSTS.

To identify the optimal scalar test, the algorithm considers all possible existing tests q_i with a positive flag, every scalar symbol s_j, and both inequality directions $<$ and $\geq$. For each of these values, the algorithm computes the optimal threshold value α as follows. Consider a GTC A, the set of training SSTS $D = \{d_i\}_{1..n}$, the scalar symbol s_j, the existing test q_i of A, and the inequality direction $<$ or $\geq$. We first compute a $n \times 2$ matrix L defined as follows:

$$L[j,1] = \left\{ \begin{array}{ll} \min(\{s_j(x) | \forall i, x \in T(q_j, d_i)\}) & \text{if the inequality is } < \\ \max(\{s_j(x) | \forall i, x \in T(q_j, d_i)\}) & \text{if the inequality is } \geq \end{array} \right\}$$
$$L[j,2] = \text{label}(d_j)$$

where $T(q_j, d_i)$ is the trace of the test q_j on the SSTS d_i (see the definition in Section 4). Next, this matrix is sorted according to its first column and all possible threshold values are evaluated in a sequence. α is defined as the threshold with the highest information gain. Algorithm 3 shows the details of these operations:

Algorithm 3: Finding the optimal threshold value α for a new scalar test.

input : L: A $n \times 2$ matrix as described in section 6.5.
output: α: the optimal threshold
Sort L according to the first column values
$h' \leftarrow +\infty$; $i' \leftarrow 0$; $w \leftarrow false$
$X \leftarrow \{0\} \times C$ with C the number of classes
$Y \leftarrow$ such that Y_i is the number of SSTS of class c_i
for $i \leftarrow 1$ **to** $n-1$ **do**
  $X_{L[i,2]} \leftarrow X_{L[i,2]} + 1$
  $Y_{L[i,2]} \leftarrow Y_{L[i,2]} - 1$
  if $L[i+1,1] \neq L[i,1] \wedge (L[i+1,2] \neq L[i,2] \vee w)$ **then**
    $w \leftarrow false$
    $h \leftarrow entropy(X)i + entropy(Y)(n-i)$
    if $h < h'$ **then**
      $h' \leftarrow h$; $i' \leftarrow i$
  else
    $w \leftarrow true$
$\alpha \leftarrow (L[i',1] + L[i'+1,1])/2$

6.6 Finding the Optimal Segment of R

Operations O2 and O4 require specification of a time constraint. In this work, we restrict time constraints to be contiguous segments $[a, b] \in \mathbb{R}$, albeit in general this restriction can be lifted without harm. The data set being finite, the number of meaningful candidate boundaries for a and b is also finite.

Given a GTC A and an operation O2 or O4 to apply, we define $M(d_k) \subset \mathbb{R}$ as follows. For operation O2, given two tests q_i and q_j of A, M is defined as $M(d_k) = \{x_1 - x_2 | \forall x_1 \in T(q_i, d_k), x_2 \in T(q_j, d_k)\}$. For operation O4, given an existing test q_i of A and an event symbol e_j, M is defined as $M(d_k) = \{x_1 - x_2 | \forall x \in T(q_i, d_k), \exists \langle e_j, x_2 \rangle \in d_k\}$. Note that M is computed from the trace T that is itself computed during the GTC evaluation.

By design, (1) If A is not valid on an SSTS d_k, then $M(d_k)$ is empty; and (2) If A is valid on an SSTS d_k, then the GTC resulting from applying the current operation with constraint $[a, b]$ on A is valid on d_k if and only if $\exists x \in M(d_k)$ such that $x \in [a, b]$.

We define $Y = \bigcup_k M(d_k)$, and Y_i to be the ith smallest element of Y. Finally, we define Z as the set of all candidate boundaries (i.e. $a, b \in Z$) as $Z_i = \{(Y_i + Y_{i+1})/2\}_{i \in [1, |Y|]} \cup$

$\{Y_1\} \cup \{Y_{|Y|}\}$. In case of large data sets, or in case of small data sets with high risk of over-fitting, the set of candidate Z can be randomly sampled down. Such down sampling is in essence equivalent to attribute sampling used e.g. in Random Forest. In practice, this significantly speeds up the search for the optimal time constraint without reducing the model accuracy because of the redundant nature of the decision tree or decision set models. We observed experimentally that limiting Z to 40 elements (i.e. 40^2 interval candidates) does not impact the models' empirical accuracy.

Finding the segment $[a, b]$ that maximizes the information gain is an optimization problem in a finite and countable two dimensional space. The naive and direct solution to this problem requires evaluating a GTC on each SSTS for each unique point in the search space (i.e. for each possible $(a, b) \in Z^2$ with $a < b$). This solution leads to the worst case computational cost of $O(|Z|^2 nE)$ with n being the number of SSTSs and E the computational cost to compute the instances of a GTC on an SSTS. This cost may easily become prohibitively expensive for any realistically complex application. An alternative, more efficient solution is to iterate over M to evaluate application of the GTC. This solution only requires evaluating the initial GTC once, it does not require evaluating the resulting GTC, and it has a computational cost of $O(|Z|^2 n + En)$. It allows us to handle much larger problems than when using the naive approach.

Instead of these two implementations, we propose a third solution with a time complexity of $O(|Z|^2 + nE)$ (instead of $O(|Z|^2 nE)$ of $O(|Z|^2 n + En)$, as above). This new approach relies on the two following properties: (1) If $h \in M(d_k)$, then the GTC resulting from the current operation with the time constraint $[h, h]$ is true on the SSTS d_k; (2) If the GTC with the new constraint $[a, b]$ is true on the SSTS d_k, then all GTCs with a time constraint $[a', b'] \supset [a, b]$ will also be true on the SSTS d_k.

The last solution relies on building a $|Z| \times |Z| \times (C-1)$ matrix P where C is the number of classes in the classification problem. After being built, each element of $P[x, y, z]$ represents the number of SSTSs in training data which belong to class z and that match the GTC A after it was subjected to the new operation with the time constraint $[Z_x, Z_y]$. Once the matrix P is computed, search for the optimal time constraint can be done in one pass over P by iterating over x and y, then over z in the inner loop. Algorithm 4 shows how to efficiently construct matrix P. Note that in practice, search for the optimal time constraint can be merged with the last "for" loop of the algorithm. Figure 6 illustrates the Algorithm 4 step by step using a small example. Also note that while we use P to optimize information gain, P can be used to optimize any metric that relies on GTC matching counts.

7. Experimental Evaluation

In this section we report and discuss performance of the two proposed algorithms (GTC-DF and GTC-Set) as compared to a wide variety of alternative approaches found in literature. The algorithms are evaluated on a collection of 59 unique data sets described below. All these data sets are publicly available online.

7.1 Synthetic Data (1 set)

The synthetic data set is composed of a relatively simple collection of 200 computer-generated, purely symbolic SSTSs. Each of them contains one instance of each type of

Algorithm 4: Building matrix P.

input : Z: The set of candidate boundaries.
M: The "meet" sets.
output: P: Matrix P.

```
// a. Initialization
P ← a |Z| × |Z| × (C − 1) matrix filled with zeros
for k = 1 to |D| do
    k ← −1; c ← class of d_k
    if c = C then continue
    m ← M(d_k) // m_i is the i^th smallest element of m.
    for i = 1 to |m| do
        j ← such that Z_j ≤ m_i < Z_{j+1} // Note: i ↦ j is monotonically increasing
        P[j, j, c] ← P[j, j, c] + 1
        if k ≠ −1 then P[j, k, c] ← P[i, k, c] − 1
        k ← i
// b. Vertical upward spread
for i = 1 to |Z| do
    X ← {0} × (C − 1)
    for j = i to |Z| do
        X ← X + P[j, i, _]
        P[j, i, _] ← X
// c. Horizontal left spread
for i = 1 to |Z| do
    X ← {0} × (C − 1)
    for j = i to 1 do
        X ← X + P[i, j, _]
        // Note: Here X contains the class distribution for the candidate GTC
           with time constraints [Z_j, Z_i].
        P[i, j, _] ← X
```

events from the set A, B and C, respectively at times t_a, t_b and t_c. For each SSTS, t_a is sampled uniformly from $[0, 200]$, and t_b and t_c are sampled uniformly from $[t_a - 15, t_a + 15]$. If $|t_a - t_b| < 10$ and $|t_a - t_c| < 10$ and $|t_b - t_c| < 10$, then the SSTS is labeled as class 1. Else, if $|t_a - t_b| < 10$, then the SSTS is labeled as class 2. If none of these conditions are valid, the SSTS is labeled as class 3. This data set aims to evaluate our algorithm capability of correctly learning temporal constraints from data. The synthetic data set as well as a Python script used to generate it are available at *mathieu.guillame-bert.com/dataset*.

7.2 UCR Time Series Classification Repository (41 UCR Data Sets)

The University of California at Riverside (UCR) benchmark data is a collection of 41 diverse data sets representing time series classification problems of varying complexity. These data sets have the particularity to be *aligned* i.e. given a data set, the i-th observations of all time series reflect the same phenomenon observed at the same moment of time. Because of this alignment, "visual" measures of distance between instances of such data (e.g. Euclidean or

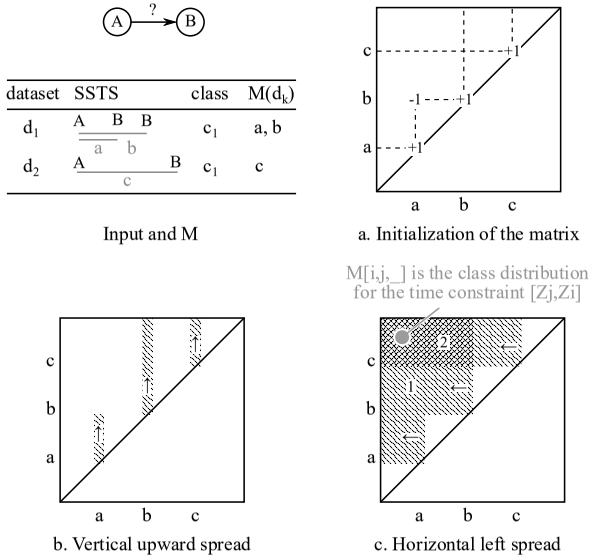


Figure 6: Step by step illustration of Algorithm 4 to construct M, matrix P, and the optimal time interval constraint.

Dynamic Time Warping (DTW) restricted to a very small shift coupled with one nearest neighbor) and algorithms designed for "flat" data sets (such as e.g. SVM) can be very competitive. The UCR data sets are organized into two groups: Group 1 (18 sets) and Group 2 (23 sets). The number of classes, number of time series, and their lengths, vary in these data sets respectively from-to $[2, 50]$, $[56, 7164]$ and $[60, 637]$ for Group 1, and $[2, 25]$, $[322, 9236]$ and $[28, 1882]$ for Group 2. It appears that the sets in Group 2 are on average larger than sets of Group 1. These data are available at *http://www.cs.ucr.edu/∼eamonn/time_series_data*.

7.3 16 Rotated UCR Data Sets

The UCR data sets are *aligned*, and it is likely that some classification models applied to them may rely on and benefit from this alignment. These data sets also contain rich local patterns. We propose to modify the UCR data by breaking their alignment in order evaluate the ability of classification algorithms to learn discriminative local patterns. In this way, the modified UCR collection can be used to evaluate algorithms on the significant problem of classifying non-aligned time series and SSTSs. One such scenario involves problems where the starting point of time series observation is arbitrary, which is the case of many practical applications in which we may not have the luxury of observing a modeled process from a

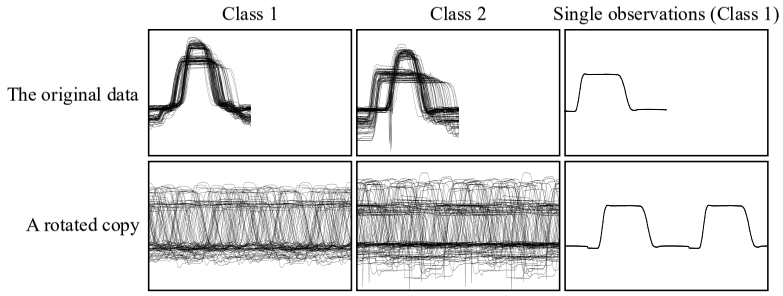


Figure 7: The original and one rotated copy of the UCR Gun Point data set.

well-defined initial setting. We call this modified collection of benchmark data the "Rotated UCR data sets".

The rotation is performed as follows: We break any potentially pre-existing alignment of a data set by randomly and independently shifting each time series in it versus others. Given a time series $X = [x_1, \ldots, x_n]$, the rotated copy is defined as $X' = [x_i, \ldots, x_n, x_1, \ldots, x_n, x_1, \ldots, x_{i-1}]$ with i being a uniformly random integer drawn from the range $[1, n]$ and unique to each time series. In order to avoid disturbing local patterns (if they exist), the rotated series are doubled in length with regard to the originals. To enable comprehensive evaluation, we independently generate 10 rotated copies for each of the originals and use each of these copies independently in experiments. The reported classification performance will be computed by averaging the results obtained from the 10 rotated copies. To avoid excessive computation costs, we chose to rotate all but two UCR Group 1 sets. As an example, Figure 7 shows the original UCR Gun Point data and one of its rotated copies.

7.4 Internal Bleeding Detection Data Set (1 Set)

Undetected internal bleeding during and after surgical procedures is a major medical concern. Early and reliable detection of internal bleeding is therefore considered a significant medical research problem. We assembled the *Bleeding Detection Data Set* from vital signs measured at high frequency (250Hz) using a bed-side hemodynamic monitoring system. The collected measurements include arterial blood pressure, central venous pressure, and airway pressure. The data has been collected from a cohort of 52 healthy pigs subjected to induced slow bleeding. Each animal has been sedated, instrumented, left to rest for half an hour, and then bled at the fixed rate of 20mL/min. From the vital sign records of each pig, we randomly sampled two 30 second long segments of data: One from the resting period before and one approximately 2 minutes into the bleeding. Pre- and post-bleeding segments are respectively labeled as negative and positive. Unlike in the UCR data sets, the SSTSs (vital signs) are not temporally aligned: The starting point of observation is effectively arbitrary. Moreover, the data is multivariate.

For reference, current clinical practice is usually unable to detect internal bleeding earlier than 10 to 15 minutes after the onset of a slow internal bleeding episode. We try to push

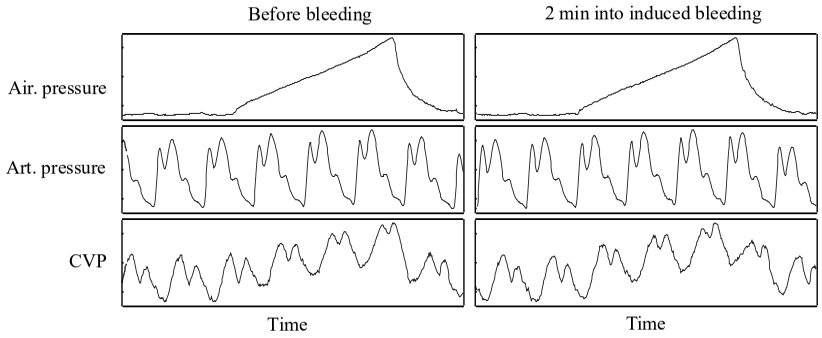


Figure 8: Four second samples of vital sign time series collected from one pig in pre-bleed phase (left) and 2 minutes into the induced bleeding (right). Note that in practice, vital signs of different animals tend to look less similar than vital signs of a same animal before and after bleeding. Air. pressure stands for Airway pressure. Art. pressure stands for arterial blood pressure, and CVP stands for central venous pressure.

that limit by considering SSTSs observed at a 2 minute mark into bleeding, when the data is not yet visually differentiable to expert clinicians. Figure 8 shows small parts of the pre-bleed and 2-minutes-in snapshots of vital signs data observed for one animal. The internal bleeding detection data set is available at *mathieu.guillame-bert.com* [Guillame-Bert].

7.5 Experimental Setup

Purposive featurization of time sequence data can sometimes significantly impact performance of the resulting models. In order to avoid obfuscating the relative performance of various algorithms being compared in this paper, we decided to not invest time and effort to produce advanced and potentially informative algorithm-specific featurizations of raw data. Instead, we relied on two basic statistics (moving averages and moving standard deviations), and we marked easily detectable peaks manifesting in the scalar data using a simple threshold criterion on the signal's second order derivative.

All considered algorithms have been implemented (or re-implemented) in C++ to ensure fairness of comparison. The computation times reported include training and evaluation over the complete 10-fold cross-validation cycles. This allows fair comparisons of lazy (e.g. 10NN) and non-lazy (e.g. our methods) algorithms. Experiments have been performed on a 3.4GHz i7 8-core processor with 16GB of main memory. Multiple algorithms were slow when applied on the UCR data set *StarLightCurves* (StarLC). Algorithms taking more than one day of computation for the 10-fold cross-validation were stopped and the expected total computation cost was estimated from the completed iterations. For this reason, results on *StarLightCurves* data are not used in computation of the global rankings.

7.6 Results

We compare our two algorithms against 13 related and relevant alternative classification algorithms for time series. The reference list includes: *ED+NN*: Euclidean Distance (i.e. L_2 distance) with Single Nearest Neighbor (Ding et al., 2008), *ED+kNN*: Euclidean Distance with k Nearest Neighbors (k is specified by the user), *ED+CVkNN*: Euclidean Distance with k Nearest Neighbors (k is determined through an internal loop 10-fold cross-validation executed on each training fold of the main 10-fold cross-validation loop), *DTW+NN*: Dynamic Time Warping (Berndt and Clifford, 1994) with Single Nearest Neighbor, *DTWC+NN*: Dynamic Time Warping with Constrained Warping Window (Berndt and Clifford, 1994) with Single Nearest Neighbor, *EDR+NN*: Edit Distance on Real Sequence (Chen and Ng, 2004) with Single Nearest Neighbor, *ERP+NN*: Edit Distance with Real Penalty (Chen et al., 2005) with Single Nearest Neighbor, *LCSS+NN*: Longest Common Subsequence (Vlachos et al., 2002) with Single Nearest Neighbor, *LCSSC+NN*: Longest Common Subsequence with Constrained Warping Window (Vlachos et al., 2002) with Single Nearest Neighbor, *L-1+NN*: L_1 distance with Single Nearest Neighbor, *L-inf+NN*: L_∞ distance with Single Nearest Neighbor, *RF*: Random Decision Forest (Breiman, 2001), *SVM*: Support Vector Machine Meyer et al., 2003, *Default*: Always return the most frequent class.

Algorithm parameters have not been tuned and were assigned to their meaningful default settings (except for k in *ED+kNNCV*), as follows. *RF*: maxNumTrees(30), minNumObservations(5), maxDepth(10). *LCSS+NN*: e(0.05 standard deviation). *EDR+NN*: e(0.25 standard deviation). *LCSSC+NN*: The maximum window is set to 25% of the data set length. It is worth noting that one could argue that the maximum window setting (c) of 25% is too high to operate DTWC and LCSSC on the UCR data sets, but we have chosen that for the three following reasons. Firstly, we have not tuned any algorithm parameters for specific data sets except for k in *ED+kNNCV*. Secondly, we have evaluated Euclidean Distance method being equivalent to DTWC with c=0%, and DTW being equivalent to DTWC with c=100%. Thirdly, c is a constraint parameter that only and strictly expands search space of the closest element when increased. We have experimented with alternative settings of this parameter and we have indeed obtained slight variations of empirical accuracy of DTWC, but these variations have not materially changed performance rankings reported below. We therefore chose to evaluate DTWC+NN (and other constrained dynamic programming distances) with c=25% for consistency. It is significant to remark that we evaluated the original implementation of DTW (Berndt and Clifford, 1994). Several pruning criteria have been studied to reduce the computation time without impacting the classification performances, including the work of Keogh and Ratanamahatana (2005). *DTWC+NN*: The maximum window is set to 25% of the data set length. *ED+CVkNN*: k is searched between 1 and 10. It is likely that optimizing parameters of each algorithm using internal cross-validation might improve their results, but it would significantly increase the computation time.

All reported results have been computed using the same 10-fold partitioning of data. In the case of the rotated UCR data sets, the 10-fold cross-validation is repeated for each of the 10 rotations of each rotated data set (e.g. each reported error rate is computed from 100 train and test experiments). In Bleeding Detection Data Set, we ensure that the

complete record for each animal is either entirely used for training or for testing in each cross-validation iteration.

Reported evaluation figures for the UCR data sets are not directly comparable with figures reported by Ding et al. (2008) or the figures available on the UCR Time Series web page: In Ding et al. (2008), evaluation results are computed with a "reversed cross-validation" where each of the n folds is successively used for training while the remaining $n-1$ folds are used for testing. On the UCR Time Series web page, evaluation is performed with a train and test protocol.

Tables 1 and 3 respectively compile the observed classification errors of all algorithms on the UCR data sets and rotated UCR data sets. The average rank and median rank are also reported for each algorithm. Table 2 shows the cumulative computation times of training and testing cycles for each algorithm. Table 4 shows the accuracy, classification error, AUC scores and computation times of each algorithm when applied to the Bleeding Detection Data Set. Figure 9 shows the ROC of the GTC-DF and Random Forest models obtained for this data set.

First, both our algorithms show a 100% accuracy on the synthetic data. This simple experiment validates their ability to identify explicit temporal sequence patterns when they exist.

The experiments on the UCR data, rotated UCR data, and Bleeding Detection Data Set (Tables 1, 3 and 4), show consistently high performance of our algorithms in comparison to state-of-the-art alternatives. On the original UCR data, and when considering 13 other methods, GTC-DF and GTC-Set have the highest average and median ranks of accuracy (respectively a median rank of 2 and 2.5 for Group 1, and 2.5 and 2.5 for Group 2). The next best methods are ERP+NN (med. rank of 4.8) and DTW+NN (med. rank of 6) for Group 1, and SVM (med.rank of 4.5) and ERP+NN (med. rank of 5) for Group 2. It is worth noting that since the original UCR data sets are internally aligned, methods that rely on basic distance metrics (e.g. Euclidean Distance or DTWC with very strong constraints e.g. c=0.5%) tend to perform relatively well and are also cheap to compute.

The Wilcoxon signed-rank test (Demšar, 2006) was applied to evaluate the significance of differences of performance between each algorithms. The resulting p-values are reported in Table 5. GTC-DF performs significantly better than all the other evaluated methods (including GTC-Set). Additionally, GTC-Set, while performing significantly worse than GTC-DF, performs significantly better than all the other evaluated methods.

Table 1: Empirical classification error rates observed for the algorithms under consideration on the UCR data sets from Groups 1 and 2. The first column contains the acronym names of the algorithms (our methods are labeled GTC-DF and GTC-Set), and the next two columns show average rank and median rank of each algorithm based on their accuracy performance measured on the individual data sets. The algorithms are sorted according to their average rank, and the two proposed methods top the sorted list.

(a) Error rates for the UCR Group 1 data sets.

Algo.	avg.rank	med.rank	50words	Adiac	Beef	CBF	Coffee	ECG200	FaceAll	Face4	GunPoint	Lighting2	Lighting7	OSULeaf	Trace	2Patterns	Fish	Synthetic	Wafer	Yoga
GTC DF	**3.5**	**2**	.225	**.237**	.300	.005	**.036**	.115	.045	.018	.015	.223	**.196**	.242	**.000**	.001	.001	**.066**	**.007**	**.000**	.045
GTC Set	3.7	2.5	.235	.256	**.250**	.004	**.036**	.120	.045	.018	**.010**	.223	.203	.265	**.000**	.001	.074	.012	.001	.045
ERP+NN	4.8	5.5	.254	.344	.550	**.000**	.107	.125	.010	.018	.015	**.107**	.315	.240	.090	**.000**	.143	.023	.002	.037
DTW+NN	5.9	6	.261	.359	.550	**.000**	.107	.175	.012	.027	.060	.132	.294	.269	.005	**.000**	.177	.018	.003	.055
DTWC+NN	5.9	6	.250	.359	.550	**.000**	.089	.175	.012	.027	.060	.132	.301	.269	.005	**.000**	.177	.018	.003	.062
EDR+NN	6.7	5	**.171**	.830	.567	.009	**.036**	.105	**.008**	**.009**	.055	.182	.308	**.143**	.030	**.000**	.226	.112	.002	.073
SVM	7.1	6.5	.277	.344	.667	.002	.071	.115	.023	.062	.015	.248	.350	.301	.130	.013	.094	.018	.001	.042
RF	7.8	8.5	.349	.297	.517	.002	.268	.120	.051	**.009**	.025	.190	.238	.367	.080	.012	.189	.023	.004	.066
ED+NN	8.1	9	.322	.329	.517	.009	.125	.105	.038	.071	.050	.231	.343	.342	.130	.013	.174	.078	.001	.056
L1+NN	8.3	8	.282	.350	.550	.015	.143	**.065**	.031	.027	.050	.190	.322	.328	.135	.003	.191	.100	.002	.056
LCSSC+NN	9.5	11	.360	.362	.550	.012	.161	.210	.279	.152	.035	.223	.441	.176	.060	.002	.091	.270	.004	**.027**
LCSS+NN	9.6	11	.360	.362	.550	.012	.161	.215	.283	.152	.035	.223	.441	.179	.060	.001	.091	.262	.004	**.027**
ED+5NN	11.3	11	.345	.387	.550	.009	.304	.110	.080	.125	.095	.281	.378	.391	.315	.020	.180	.092	.003	.078
Linf+NN	12.4	14	.429	.306	.483	.181	.071	.135	.148	.268	.095	.372	.580	.389	.150	.695	.171	.137	.007	.083
ED+CVkNN	12.6	14	.402	.435	.567	.008	.286	.095	.111	.152	.120	.355	.427	.457	.445	.032	.211	.113	.004	.097
ED+10NN	12.6	14	.402	.435	.567	.008	.286	.095	.111	.152	.120	.355	.427	.457	.445	.032	.211	.113	.004	.097
default	17.0	17	0.88	0.986	0.983	0.712	0.625	0.335	0.855	0.696	0.62	0.397	0.734	0.781	0.85	0.739	0.937	0.885	0.106	0.464

(b) Error rates for the UCR Group 2 data sets.

Algo.	avg.rank	med.rank	ChlorineC	ECGtorso	CricketX	CricketY	CricketZ	DiatSRed	ECGD	FaceallUCR	Haptics	InlineS	KPwrDnd	MALLAT	MedImage	MoteStrain	SonyRDeS	SonyRDeS2	Symbols	2LeadECG	WordsSyn	uWvGLibX	uWvGLibY	uWvGLibZ	StarLC*
GTC DF	**4.3**	**2.5**	.194	.004	.171	**.167**	**.162**	.006	**.000**	.040	.475	.348	.031	**.000**	**.155**	.039	.011	.035	.021	.003	.210	.158	**.215**	**.206**	**.022**
GTC Set	4.5	**2.5**	.236	.001	**.169**	.171	.173	.006	.001	.044	.473	**.335**	.033	**.000**	.162	.039	.016	.036	.020	.003	.224	**.156**	.221	.209	.025
SVM	4.6	4.5	.045	**.000**	.291	.279	.272	.001	**.000**	.021	**.467**	.423	.034	.017	.198	.057	**.006**	**.006**	.031	**.002**	.267	.184	.255	.243	.029
ERP+NN	5.7	5	.008	.002	.212	.177	.208	.003	.005	**.011**	.540	.395	.045	.006	.199	.031	.018	.017	.023	**.002**	.228	.221	.208	.274	2.000
DTW+NN	7.0	8	.008	.001	.321	.309	.328	**.000**	.001	.032	.605	.469	.037	.006	.215	.031	.021	.012	.034	.006	.256	.228	.283	.296	2.000
DTWC+NN	7.0	8	.008	.001	.321	.309	.328	**.000**	.001	.032	.605	.469	.037	.006	.215	.031	.021	.012	.034	.006	.256	.228	.283	.296	2.000
L1+NN	7.0	8	.008	.001	.321	.309	.328	**.000**	.001	.032	.605	.469	.037	.006	.215	.031	.021	.012	.034	.006	.256	.228	.283	.296	.111
LCSS+NN	7.3	5.5	.077	.003	.276	.219	.264	.019	.005	.028	.533	.415	.059	.026	.225	**.023**	.095	.105	**.014**	**.002**	**.177**	.208	.286	.263	2.000
LCSSC+NN	7.3	6	.077	.003	.277	.219	.263	.019	.005	.029	.533	.415	.059	.026	.225	**.023**	.095	.105	**.014**	**.002**	**.177**	.208	.286	.262	2.000
RF	7.5	7	.214	.008	.362	.251	.282	.016	.015	.053	.527	.458	**.026**	.003	.193	.036	.013	.034	.032	.019	.319	.197	.247	.233	.035
ED+NN	8.3	9	**.003**	.001	.341	.346	.332	**.000**	.005	.040	.583	.477	.034	.015	.230	.079	.014	.015	.035	.004	.284	.229	.274	.291	.111
ED+5NN	10.5	12	.078	.004	.421	.371	.415	.016	.014	.078	.577	.546	.027	.020	.240	.079	.018	.031	.041	.004	.330	.236	.274	.287	.117
ED+CVkNN	11.6	13	.171	.006	.503	.406	.486	**.000**	.029	.109	.583	.618	.034	.021	.264	.071	.023	.041	.049	.005	.381	.247	.275	.297	.137
ED+10NN	12.1	13	.171	.006	.503	.406	.486	.019	.029	.109	.583	.618	.034	.021	.264	.071	.023	.041	.049	.005	.381	.247	.275	.297	.137
EDR+NN	12.1	16	.374	.001	.317	.285	.286	.677	.002	.027	.680	.782	.192	.356	.300	.065	.027	.023	.086	.007	.265	.263	.304	.378	2.000
Linf+NN	12.3	15	.019	.014	.563	.529	.578	**.000**	.002	.148	.613	.514	.036	.035	.274	.153	.024	.034	.047	**.002**	.391	.259	.317	.318	.120
default	17.0	17	.864	.790	.967	.940	.964	.764	.549	.858	.819	.851	.526	.940	.479	.462	.448	.384	.847	.539	.779	.897	.907	.907	.423

JMLR 18(121):1-34, 2017

Table 2: Computing times of training and testing on the UCR data sets from Groups 1 and 2. Sorting of algorithms is the same as in Table 1b. Note that our algorithms very often remain competitive in terms of computational costs when compared to the most accurate alternatives. The ordering of the algorithms in this table is the same as in Table 1b. The bottom part of each table summarizes basic statistics of the data sets. Color is used to indicate value of each cell and it is normalized separately for each column.

(a) Computing times for the UCR Group 1 data sets

Algo.	avg.rank	med.rank	50words	Adiac	Beef	CBF	Coffee	ECG200	FaceAll	Face4	GunPoint	Lighting2	Lighting7	OSULeaf	Trace	2Patterns	Fish	Synthetic	Wafer	Yoga
GTC DF	3.5	2	51mn 39s	12mn 1.6s	12s	1mn 35s	3.6s	19s	**1h 9mn 52s**	24s	14s	30s	**2mn 34s**	6mn 20s	18s	2h 21mn	2mn 46s	1mn 46s	10mn 57s	27mn 48s
GTC Set	3.7	2.5	**1h 24mn**	**16mn 2.3s**	12s	2mn 2.7s	4.3s	**26s**	38mn 15s	36s	18s	41s	2mn 3s	12mn 41s	23s	30mn 28s	4mn 5.1s	**3mn 11s**	12mn 0.53s	37mn 3s
ERP+NN	4.8	5.5	24mn 27s	7mn 41s	28s	6mn 56s	**9.5s**	8.7s	35mn 5.9s	37s	21s	2mn 35s	51s	15mn 12s	1mn 13s	**2h 45mn**	11mn 19s	30s	8h 35mn	14h 55mn
DTW+NN	5.9	6	21mn 0.86s	6mn 34s	**30s**	7mn 1.9s	8.2s	11s	29mn 55s	**51s**	**28s**	**4mn 18s**	44s	**20mn 0.7s**	1mn 9.8s	2h 21mn	**16mn 19s**	38s	**9h 14mn**	**15h 8mn**
DTWC+NN	5.9	6	12mn 1.7s	3mn 47s	15s	3mn 8.4s	3.7s	3.4s	15mn 53s	15s	8.4s	1mn 7.4s	25s	6mn 35s	29s	1h 20mn	4mn 51s	12s	3h 46mn	1h 3mn
EDR+NN	6.7	5	21mn 47s	10mn 3.4s	21s	**7mn 7.8s**	6s	8s	30mn 37s	34s	20s	2mn 22s	45s	13mn 58s	1mn 11s	2h 25mn	10mn 52s	27s	7h 8mn	14h 49mn
SVM	7.1	6.5	26s	11s	2.2s	13s	3s	3.6s	27s	5.4s	3.4s	11s	3.4s	23s	7.3s	1mn 13s	18s	5.4s	1mn 17s	2mn 3.5s
RF	7.8	8.5	1mn 20s	44s	4s	12s	2s	3.6s	1mn 43s	1.2s	0.9s	13s	8.1s	53s	9.1s	5mn 2.1s	38s	11s	7mn 28s	9mn 55s
ED+NN	8.1	9	13s	5.3s	0.17s	6.6s	0.11s	0.52s	30s	0.62s	0.64s	1.3s	0.5s	6.5s	1.2s	2mn 25s	4.6s	1.2s	7mn 36s	5mn 35s
L1+NN	8.3	8	14s	5.6s	0.17s	6.6s	0.21s	0.53s	31s	0.39s	0.79s	1.1s	0.51s	7.8s	0.82s	2mn 31s	5.8s	1.5s	7mn 53s	5mn 49s
LCSSC+NN	9.5	11	14mn 14s	3mn 20s	12s	2mn 15s	3s	2.6s	15mn 19s	11s	6.1s	52s	23s	5mn 9.1s	21s	1h 4mn	3mn 36s	9.1s	2h 19mn	5h 30s
LCSS+NN	9.6	11	15mn 37s	5mn 17s	24s	5mn 8s	6.4s	8.3s	31mn 38s	42s	19s	3mn 23s	47s	19mn 2.1s	**1mn 16s**	1h 44mn	11mn 2.1s	21s	6h 4mn	12h 34mn
ED+5NN	11.3	11	3.1s	1.3s	0.11s	1.3s	0.075s	0.13s	6.3s	0.18s	0.15s	0.35s	0.23s	1.4s	0.29s	28s	1.1s	0.32s	1mn 6.5s	55s
Linf+NN	12.4	14	13s	5.5s	0.18s	5.9s	0.19s	0.49s	30s	0.71s	0.43s	0.93s	0.51s	7s	1.4s	2mn 27s	4.9s	1.4s	7mn 45s	5mn 55s
ED+CVkNN	12.6	14	4mn 6.1s	1mn 42s	3s	1mn 34s	2.1s	3.4s	9mn 59s	6.4s	6.2s	14s	9.2s	1mn 39s	14s	47mn 11s	1mn 9.4s	17s	1h 48mn	1h 29mn
ED+10NN	12.6	14	2.6s	1.1s	0.06s	1.1s	0.041s	0.067s	5.6s	0.12s	0.1s	0.22s	0.15s	1.2s	0.21s	28s	0.82s	0.24s	1mn 4.1s	53s
default	**17.0**	**17**	1s	0.38s	0.071s	0.44s	0.043s	0.063s	1.1s	0.11s	0.076s	0.18s	0.14s	0.45s	0.14s	2.1s	0.35s	0.14s	3.1s	3.1s
#classes			50	37	5	3	2	2	14	4	2	2	7	6	4	4	7	6	2	2
length			270	176	470	128	286	96	131	350	150	637	319	427	275	128	463	60	152	426
#ssts			905	781	60	930	56	200	2250	112	200	121	143	442	200	5000	350	600	7164	3300
avg.err.			.341	.410	.545	.058	.171	.142	.126	.117	.087	.239	.370	.329	.172	.092	.200	.134	.009	.083
std.err.			.156	.196	.150	.174	.148	.064	.206	.167	.142	.087	.135	.149	.225	.235	.197	.209	.025	.100

(b) Computing times for the UCR Group 2 data sets

Algo.	avg.rank	med.rank	ChlorineC	ECGtorso	CricketX	CricketY	CricketZ	DiatSRed	ECG5D	FacesUCR	Haptics	InlineS	ItPwrDmd	MALLAT
GTC DF	4.3	2.5	47mn 58s	7mn 58s	22mn 30s	19mn 41s	22mn 29s	21s	1mn 23s	30mn 15s	10mn 50s	20mn 3.1s	49s	8mn 16s
GTC Set	4.5	2.5	56mn 6.7s	8mn 20s	**34mn 46s**	**28mn 58s**	**34mn 49s**	23s	**1mn 35s**	**37mn 21s**	14mn 36s	24mn 22s	**56s**	9mn 35s
SVM	4.6	4.5	57s	1mn 32s	13s	13s	13s	3.7s	4.2s	19s	24s	1mn 0.85s	1.3s	1mn 25s
ERP+NN	5.7	5	48mn 39s	11h 30mn	5mn 18s	5mn 17s	5mn 17s	1mn 17s	1mn 22s	8mn 11s	31mn 41s	3h 12mn	3.9s	12h 37mn
DTW+NN	7.0	8	1mn 29s	2h 36mn	9.1s	8.6s	8.7s	3.5s	3.1s	17s	6mn 52s	42mn 8s	0.53s	2h 44mn
DTWC+NN	7.0	8	1mn 33s	2h 34mn	9s	8.9s	8.8s	2.7s	3.1s	18s	6mn 58s	43mn 24s	0.6s	2h 45mn
L1+NN	7.0	8	26s	48s	2.5s	2.5s	2.5s	0.64s	1.1s	5.8s	4.2s	1mn 20s	0.28s	1mn 20s
LCSS+NN	7.3	5.5	31mn 4.9s	7h 48mn	3mn 21s	3mn 18s	3mn 22s	24s	36s	4mn 52s	25mn 11s	2h 15mn	2.2s	8h 7mn
LCSSC+NN	7.3	6	15mn 33s	4h 47mn	1mn 42s	1mn 41s	1mn 41s	14s	21s	2mn 31s	13mn 32s	1h 20mn	1.4s	4h 57mn
RF	7.5	7	3mn 25s	6mn 46s	51s	50s	51s	0.61s	16s	1mn 12s	1mn 38s	4mn 56s	4.3s	4mn 6.3s
ED+NN	8.3	9	27s	48s	2.7s	2.6s	2.6s	0.77s	1.3s	6.3s	4.4s	13s	0.3s	1mn 20s
ED+5NN	10.5	12	29s	49s	2.7s	2.6s	2.7s	0.73s	1.3s	6.3s	4.3s	13s	0.31s	1mn 21s
ED+CVkNN	11.6	13	43mn 39s	1h 13mn	3mn 20s	3mn 20s	3mn 21s	43s	1mn 30s	8mn 58s	5mn 27s	18mn 13s	18s	2h 6mn
ED+10NN	12.1	13	28s	48s	2.2s	2.2s	2.2s	0.51s	1s	5.7s	3.6s	12s	0.24s	1mn 17s
EDR+NN	12.1	16	**57mn 7.8s**	**11h 59mn**	5mn 17s	5mn 23s	5mn 27s	**2mn 32s**	1mn 32s	7mn 41s	**32mn 41s**	**3h 27mn**	5.1s	**14h 26mn**
Linf+NN	12.3	15	31s	48s	2.5s	2.6s	2.5s	0.67s	1.4s	6.7s	4.1s	13s	0.25s	1mn 16s
default	**17.0**	**17**	0.83s	1.6s	0.19s	0.2s	0.19s	0.075s	0.11s	0.36s	0.32s	0.76s	0.048s	1.7s
#classes			3	4	12	12	12	4	2	14	5	7	2	8
length			166	1639	300	300	300	345	136	131	1092	1882	24	1024
#ssts			4307	1420	780	780	780	322	884	2250	463	650	1096	2400
avg.err.			.138	.050	.367	.336	.362	.091	.038	.099	.577	.505	.075	.087
std.err.			.165	.190	.191	.187	.191	.228	.127	.198	.084	.139	.122	.228

Algo.	MedImgs	MoteStrain	SonyRbtS	SonyRbtS2	Symbols	2LeadECG	WordsSyn	uWvGLibX	uWvGLibY	uWvGLibZ	StarLC*
GTC DF	11mn 38s	3mn 59s	1mn 4.3s	2mn 42s	4mn 23s	1mn 0.84s	38mn 59s	1h 25mn	1h 34mn	1h 29mn	2h 28mn
GTC Set	**13mn 15s**	**4mn 36s**	**1mn 15s**	**3mn 48s**	7mn 35s	1mn 6.5s	**47mn 57s**	1h 58mn	1h 56mn	1h 55mn	3h 13mn
SVM	6.1s	4.5s	1.7s	2.6s	15s	3.5s	14s	1mn 21s	1mn 28s	1mn 27s	7mn 8s
ERP+NN	1mn 12s	1mn 4.5s	11s	23s	17mn 47s	51s	5mn 48s	**3h 12mn**	3h 12mn	**3h 12mn**	7day 18h
DTW+NN	3.2s	3.3s	0.73s	1.5s	3mn 58s	2.7s	11s	4mn 47s	4mn 49s	5mn 11s	1day 17h
DTWC+NN	3.2s	3.3s	0.72s	1.6s	3mn 59s	2.7s	9.8s	4mn 55s	4mn 49s	4mn 52s	1day 17h
L1+NN	1.1s	1.2s	0.36s	0.72s	5.4s	0.97s	2.9s	1mn 12s	1mn 11s	1mn 11s	18mn 39s
LCSS+NN	44s	40s	7.1s	14s	11mn 52s	25s	3mn 6.2s	1h 57mn	1h 58mn	1h 59mn	5day 16h
LCSSC+NN	21s	20s	3.6s	7.5s	7mn 0.73s	14s	1mn 42s	1h 7.8s	1h 43s	1h 37s	3day 4h
RF	25s	18s	2.4s	11s	60s	16s	54s	6mn 59s	7mn 17s	7mn 4.7s	1h 1mn
ED+NN	1.2s	1.2s	0.39s	0.81s	5.6s	1s	2.6s	1mn 10s	1mn 11s	1mn 8.9s	18mn 43s
ED+5NN	1.2s	1.2s	0.43s	0.74s	5.6s	1s	2.9s	1mn 11s	1mn 11s	1mn 12s	18mn 50s
ED+CVkNN	1mn 27s	1mn 29s	21s	47s	7mn 56s	**1mn 13s**	4mn 1.1s	1h 55mn	1h 56mn	1h 56mn	1day 7h
ED+10NN	1s	1s	0.29s	0.56s	5.5s	1s	2.8s	1mn 11s	1mn 11s	1mn 11s	18mn 36s
EDR+NN	1mn 31s	1mn 0.88s	11s	21s	**19mn 41s**	1mn 5.7s	5mn 39s	3h 4mn	**3h 14mn**	3h 11mn	**8day 20h**
Linf+NN	1.1s	1.1s	0.28s	0.52s	5.5s	1s	3s	1mn 12s	1mn 13s	1mn 12s	18mn 27s
default	0.12s	0.12s	0.058s	0.088s	0.28s	0.098s	0.19s	1.1s	1.1s	1.1s	6.5s
#classes	10	2	2	2	6	2	25	8	8	8	3
length	99	84	70	65	398	82	270	315	315	315	1024
#ssts	1141	1272	621	980	1020	1162	905	4478	4478	4478	9236
avg.err.	.241	.078	.052	.055	.089	.066	.305	.259	.316	.314	.780
std.err.	.077	.104	.103	.089	.199	.172	.139	.168	.155	.156	.933

Table 3: Classification error rates observed on the rotated UCR data sets. "O." stands for the original and "R." for the rotated data. The average and median ranks only reflect the rotated results. Algorithms are sorted according to their average rank on the rotated data sets. Note that GTC-DF and GTC-Set are consistently the highest on average ranking algorithms, while the runner-ups change when compared to the non-rotated data results.

Algo.	avg.rank	med.rank	50words		Adiac		Beef		CBF		Coffee		ECG200		FaceAll		Face4	
			O.	R.	O.	R.	O.	O.	R.	O.	R.	O.	O.	R.	O.	R.	O.	R.
GTC DF	**2.9**	**2**	.220	**.359**	**.243**	**.308**	.300	.667	.004	.008	**.036**	.320	.105	**.166**	.044	.060	.018	.188
GTC Set	3.6	2.5	.239	.379	.259	.317	**.267**	.658	.004	.011	**.036**	.338	.115	.168	.048	.066	.018	.188
ERP+NN	4.4	5	.254	.459	.344	.651	.550	.623	**.000**	**.000**	.107	.371	.125	.177	.010	**.027**	.018	.060
DTWC+NN	6.4	6.5	.250	.630	.359	.859	.550	.617	**.000**	.042	.089	.445	.175	.256	.012	.329	.027	.525
DTW+NN	6.5	7	.261	.630	.359	.859	.550	.617	**.000**	.042	.107	.445	.175	.256	.012	.329	.027	.525
LCSS+NN	6.8	5.5	.360	.360	.362	.865	.550	.757	.012	.001	.161	.293	.215	.182	.283	.060	.152	**.017**
LCSSC+NN	6.8	5.5	.360	.360	.362	.865	.550	.757	.012	.001	.161	.293	.210	.182	.279	.060	.152	**.017**
EDR+NN	7.3	6	**.171**	.546	.830	.977	.567	.800	.009	.012	**.036**	**.287**	.105	.227	**.008**	.105	**.009**	.156
L1+NN	8.0	8	.282	.630	.350	.859	.550	.617	.015	.040	.143	.466	**.065**	.257	.031	.329	.027	.519
SVM	8.9	10	.277	.658	.344	.882	.667	.648	.002	.031	.071	.559	.115	.263	.023	.435	.062	.627
RF	9.0	10	.349	.751	.297	.941	.517	**.587**	.002	.048	.268	.500	.120	.311	.051	.578	**.009**	.609
ED+NN	9.2	10	.322	.665	.329	.848	.517	.615	.009	.059	.125	.484	.105	.262	.038	.413	.071	.588
ED+5NN	11.5	12	.345	.721	.387	.924	.550	.607	.009	.045	.304	.507	.110	.274	.080	.600	.125	.686
Linf+NN	12.4	13.5	.429	.767	.306	.812	.483	.630	.181	.412	.071	.500	.135	.279	.148	.625	.268	.724
ED+10NN	13.4	14	.402	.750	.435	.947	.567	.612	.008	.076	.286	.521	.095	.282	.111	.678	.152	.738
default	16.5	17	.880	.880	.986	.986	.983	.983	.712	.712	.625	.625	.335	.335	.855	.855	.696	.696

Algo.	GunPoint		Lighting2		Lighting7		OSULeaf		Trace		2Patterns		Fish		Synthetic	
	O.	R.	O.	R.	O.	R.	O.	R.	O.	R.	O.	R.	O.	R.	O.	R.
GTC DF	**.015**	**.043**	.190	.217	**.203**	.290	.242	.232	**.000**	.006	.001	.081	.077	**.170**	**.008**	.171
GTC Set	**.015**	.050	.231	.240	**.203**	**.288**	.247	.245	**.000**	**.004**	.001	**.080**	**.063**	.184	**.008**	**.169**
ERP+NN	**.015**	.112	**.107**	**.172**	.315	.349	.240	.298	.090	.222	**.000**	.112	.143	.320	.023	.205
DTWC+NN	.060	.254	.132	.312	.301	.495	.269	.465	.005	.349	**.000**	.109	.177	.633	.018	.337
DTW+NN	.060	.254	.132	.312	.294	.495	.269	.465	.005	.349	**.000**	.109	.177	.633	.018	.337
LCSS+NN	.035	.062	.223	.210	.441	.374	.179	**.146**	.060	.095	.001	.107	.091	.251	.262	.253
LCSSC+NN	.035	.062	.223	.210	.441	.374	.176	**.146**	.060	.095	.002	.107	.091	.251	.270	.253
EDR+NN	.055	.397	.182	.317	.308	.522	**.143**	.348	.030	.549	**.000**	.104	.226	.864	.112	.235
L1+NN	.050	.267	.190	.320	.322	.495	.328	.462	.135	.340	.003	.109	.191	.631	.100	.339
SVM	**.015**	.218	.248	.413	.350	.682	.301	.470	.130	.430	.013	.137	.094	.591	.018	.253
RF	.025	.214	.190	.347	.238	.531	.367	.532	.080	.334	.012	.159	.189	.689	.023	.290
ED+NN	.050	.333	.231	.445	.343	.661	.342	.474	.130	.495	.013	.120	.174	.611	.078	.360
ED+5NN	.095	.384	.281	.440	.378	.720	.391	.518	.315	.659	.020	.116	.180	.754	.092	.359
Linf+NN	.095	.404	.372	.463	.580	.780	.389	.517	.150	.636	.693	.759	.171	.585	.137	.479
ED+10NN	.120	.449	.355	.513	.427	.760	.457	.563	.445	.723	.032	.135	.211	.805	.113	.405
default	.620	.620	.397	.397	.734	.734	.781	.781	.850	.850	.739	.739	.937	.937	.885	.885

Algorithm	Error rate	Accuracy	Duration	AUC
GTC DF	0.087	**0.913**	18mn	**0.975**
GTC Set	0.125	0.875	2h	0.958
ERP+NN	0.404	0.596	**36h**	0.524
LCSS+NN	0.423	0.577	32h	0.519
LCSSC+NN	0.423	0.577	9h	0.519
EDR+NN	0.462	0.538	**36h**	0.51
DTW+NN	0.471	0.529	1h	0.507
DTWC+NN	0.471	0.529	1h	0.507
L1+NN	0.471	0.529	24s	0.507
RF	0.471	0.529	7mn	0.507
ED+10NN	0.5	0.5	2.7s	0.500
SVM	0.5	0.5	3mn	0.488
default	0.5	0.5	0.67s	0.500
Avg. Vit	0.51	0.49	3.9s	0.492
ED+NN	0.529	0.471	26s	0.493
Linf+NN	**0.577**	0.423	25s	0.481

Table 4: Classification error, accuracy, computation time (of the complete 10-fold cross-validation cycle), and AUC obtained on the bleeding detection data set. GTC methods are by far the top performers on this data.

The rotated UCR data sets are harder to classify than the original, aligned UCR data sets. In most cases, the error rates on the rotated data sets are higher for the same algorithm than the rates observed on the non-rotated counterparts (see Table 3). The experiment shows that GTC-DF and GTC-Set are substantially less impacted by this randomization than any other of the evaluated methods, and that GTC-DF and GTC-Set remain the top performers in terms of the average and median ranks (respectively an average and median rank of 2.9 and 2 for GTC-DF, and 3.6 and 2.5 for GTC-Set (Table 3). As expected, GTC-DF and GTC-Set yield similar error rates, and therefore their ranking impact each other. Removing one of these two algorithms from the ranking would further increase the ranking difference between the remaining proposed algorithm and the other state of the art methods under consideration.

Finally, on the Bleeding Detection Data Set, GTC-DF shows an Area Under the Receiver Operating Characteristic Curve (AUC) score of 96.2% and the accuracy at the default sensitivity set point (50%) of 91.3%, while all other evaluated methods show performance very close to random based on their empirical AUC scores (Table 4). This last result is encouraging as it may allow developing effective cardio-respiratory monitors tailored to detect specific hard-to-discern conditions.

In addition to overall highly competitive performance, learning of GTC-DF and GTC-Set models is reasonably fast, even though these algorithms are not uniformly faster than all of the evaluated alternatives. The 10-fold cross-validation on the Yoga data set (the largest of the considered sets from Group 1) takes our approaches approximately 27 and 37 minutes respectively, while the next best on accuracy models require approximately 15 hours (ERP+NN and DTW+NN, Table 2).

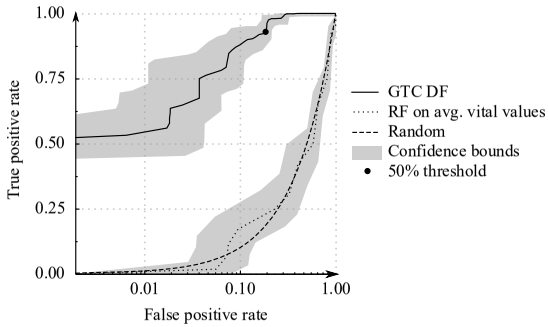


Figure 9: Receiver Operating Characteristic obtained for GTC-DF algorithm on the bleeding detection data set compared to the performance of the Random Forest algorithm on the flattened variant of the same data. False positive rate is scaled logarithmically.

Table 5: P-values of the Wilcoxon signed-rank test on the 41 UCR data sets and between each pair of the evaluated algorithms. Gray background is indicative of a positive significance of performances with a .05% error. A "+1NN" suffix has been removed from column labels for brevity.

Algorithm	GTC DF	GTC Set	ERP	DTW	DTWC	EDR	SVM	RF	ED	L1	LCSSC	LCSS	ED+5NN	Linf	ED+CVkNN	ED+10NN
GTC DF		6.6e-3	2.4e-3	1.5e-4	1.5e-4	1.7e-5	2.3e-4	5.7e-7	6.9e-6	9.7e-5	2.7e-5	2.8e-5	4.2e-7	2.0e-7	8.8e-8	5.4e-8
GTC Set	0.994		3.9e-3	3.0e-4	2.9e-4	2.4e-5	3.8e-4	4.4e-6	1.3e-5	1.3e-4	4.1e-5	4.5e-5	5.2e-7	2.8e-7	1.4e-7	8.8e-8
ERP+1NN	0.998	0.996		8.6e-4	2.2e-3	7.9e-5	8.0e-2	3.2e-2	2.7e-4	1.5e-4	8.5e-4	8.1e-4	6.5e-6	1.3e-5	2.4e-6	1.9e-6
DTW+1NN	1.000	1.000	0.999		0.819	1.8e-3	0.898	0.569	2.6e-2	5.4e-2	0.180	0.180	5.4e-5	1.1e-4	9.0e-6	6.5e-6
DTWC+1NN	1.000	1.000	0.998	0.292		1.7e-3	0.892	0.592	2.7e-2	7.1e-2	0.175	0.173	5.4e-5	7.5e-5	9.0e-6	6.5e-6
EDR+1NN	1.000	1.000	1.000	0.998	0.998		0.997	0.996	0.958	0.992	0.945	0.944	0.738	0.341	0.447	0.447
SVM	1.000	1.000	0.922	0.104	0.111	3.1e-3		0.104	2.2e-4	1.2e-2	2.5e-2	2.4e-2	4.0e-7	2.1e-6	3.0e-7	2.6e-7
RF	1.000	1.000	0.969	0.436	0.413	3.8e-3	0.898		2.9e-2	0.124	0.140	0.147	1.0e-5	4.8e-5	5.5e-7	3.4e-7
ED+1NN	1.000	1.000	1.000	0.975	0.973	4.4e-2	1.000	0.971		0.930	0.373	0.384	4.7e-7	2.4e-6	1.8e-7	1.2e-7
L1+1NN	1.000	1.000	1.000	0.951	0.935	8.3e-3	0.988	0.879	7.2e-2		0.288	0.291	7.7e-6	9.3e-6	2.6e-7	1.8e-7
LCSSC+1NN	1.000	1.000	0.999	0.823	0.829	5.6e-2	0.976	0.863	0.632	0.716		0.275	5.8e-2	9.2e-3	4.4e-3	4.0e-3
LCSS+1NN	1.000	1.000	0.999	0.823	0.830	5.8e-2	0.976	0.856	0.621	0.714	0.763		5.8e-2	9.3e-3	4.5e-3	4.1e-3
ED+5NN	1.000	1.000	1.000	1.000	1.000	0.266	1.000	1.000	1.000	1.000	0.944	0.944		3.4e-3	1.2e-6	4.2e-7
Linf+1NN	1.000	1.000	1.000	1.000	1.000	0.664	1.000	1.000	1.000	1.000	0.991	0.991	0.997		0.860	0.838
ED+CVkNN	1.000	1.000	1.000	1.000	1.000	0.559	1.000	1.000	1.000	1.000	0.996	0.996	1.000	0.143		0.500
ED+10NN	1.000	1.000	1.000	1.000	1.000	0.559	1.000	1.000	1.000	1.000	0.996	0.996	1.000	0.166	0.977	

7.7 Interpretability of Models

The ability to interpret a trained machine learning model typically improves understanding of the studied process, it may lead to useful discoveries, and it generally improves the acceptance of the model by its end-users. Many of the most powerful machine learning algorithms belong to the informal class of black (or opaque) box algorithms and specifically lack this useful ability. In our work, a single GTC is a graph of a usually small size (typically less than five tests, this number can however be controlled and further reduced

if needed). The resulting complexity of a single GTC is therefore manageable and the model can practically be read by a typical trained user. However, the large amount and the redundancy of GTCs in large GTC-DF or a GTC-Set models may still make it difficult for the user to comprehend and interpret their structures. In this section, we present a two straightforward methods to extract a small but representative subset of interpretable GTCs from a GTC-Set model.

The first method is a non-weighed coverage algorithm guided by immediate information gain. At each step of the process, all GTCs are evaluated and the one with the highest information gain is selected. Next, all SSTSs correctly classified by this selected GTC are removed form data. This sequence is repeated until all SSTSs have been removed, or until no remaining GTC has a strictly positive information gain, or until a user-specified number of GTCs have been extracted. We refer to this method as the "Coverage selection method".

This method is similar in nature to the feature selection steps in CBA (Liu et al., 1998) and CMAR (Li et al., 2001) algorithms. However, while the CBA and the CMAR use feature selection to build a classifier from a pool of already extracted patterns, our algorithms use the method presented below to extract an interpretable and informative set of patterns from an already existing and functional classifier.

The application of p boolean patterns as a list of exception rules defines $p + 1$ combinations (e.g. the second rule is only tested if the first one does not apply). However, the independent application of p boolean patterns defines 2^p combinations. The second method is a greedy forward selection guided by the immediate information gain from all the considered patterns i.e. by the 2^p combinations. This second method is more likely to over fit than the first method when a large number of patterns is selected. But it gives an equal importance to all the patterns and observations. We refer to this method as the "Combination selection method".

The primary goal of the extracted patterns is not to perform predictions but to help the user interpret the models. As shown below, the selected GTCs typically show reduced classification performance but lower complexity than the corresponding full GTC-Sets.

Figure 10 shows the first two selected GTCs by the Combination selection method on the first cross-validation folds of a few UCR data sets (Coffee, CBF, ECG200, Wager, Yoga) and the Bleeding Detection Data Set. Figure 11 shows the empirical class distribution of the held-out test data classified by these top GTCs (the held-out test sets have not been used either for training or selection of the GTCs). We can see a very good effectiveness of just the first two (or just one for the Coffee data set) greedily selected GTCs at expressing output class distributions. For instance, in case of the CBF data set using the first selected GTC enables perfect separation of class 2 from the rest of the data, and adding the second greedily selected GTC discriminates class 1 from class 3 data in the remainder. Other results shown in Figure 11 indicate a very strong, even if not perfect, utility of just the top 2 GTCs to express class distributions in data. This suggests a potential for obtaining highly interpretable and simple models with impressive classification performance if that is attainable, and gradually adding complexity as required to achieve desired accuracy of classification while limiting overall complexity of the resulting models and maintaining their interpretability.

Figure 12 illustrates an example of application of the single top GTC extracted for the first SSTS of the first test fold of the UCR Coffee data set. The time series graph depicts

the original scalar time series of the original data and the panel above it depicts occurrences of the discrete events extracted from it. This particular GTC requires that the standard deviation of the raw signal computed on a trailing time window 16 time units wide ought to peak-up (show a local maximum over time) during period between 226 and 252 time units from the time of occurrence of the starting event labeled as "begin". This is the positive node plotted in the graph diagram in the right part of Figure 12. Concurrently, this GTC also requires that no event of the standard deviation of the raw time series computed over the trailing window 16 time units wide increase, crossing the threshold of 0.75 units, during time interval between 103 and 109 time units from "begin" (the negative node in the GTC diagram) will occur. If these conditions are met, the GTC is matched, and using just this GTC to perform classification of the underlying time series would be predicted as an instance of class 1.

Interestingly, the class distribution of the first fold of the Coffee data set can be explained perfectly with just a single GTC of a relatively small size (Fig. 6). It only consists of three tests (Fig. 10). We conclude that this data set appears to be quite simple to handle. The first fold of the CBF data set can be explained perfectly and entirely with just two GTCs. In the ECG200 data set, the top two GTCs can classify correctly 90% of the held-out test data. Yoga appears to be the most complex of the data sets considered in Fig. 10). The top two GTCs extracted from its fist cross-validation fold are significantly more complex than in the case of other data sets, with 10 tests each. These two GTCs can correctly classify 76% of the held-out data. Interestingly, this finding is somewhat intuitive since for all evaluated algorithms this data takes the longest to process due to its inherent complexity. For Yoga data set, the GTC models achieve prevailing accuracy but they require more GTCs of greater complexity to accomplish that.

Table 6 summarizes the aggregated classification performance obtained with the two selection methods, when varying the number of GTCs used per data set, and compared to the results obtained when using the complete GTC-Set. We use UCR Group 1 and Bleeding Detection Data Set. The results reflect average performance from 10-fold cross-validation, so the extracted GTCs may vary between folds. In terms of error rates, UCR:Trace, UCR:Lighting2, UCR:ECG200, UCR:Coffee and UCR:Beef data are are well handled by the top two GTCs in comparison to the fully developed GTC-Set. On the other hand, performance revealed on UCR:2Patterns and UCR:Synthetic data substantially diminishes when just the simple two-GTC models are used. Finally, The Combination selection method performs better that the Coverage selection method on most data sets (16 over 18 for two selected GTCs, and 14 over 18 for four selected GTCs).

8. Conclusion

We introduced Graphs of Temporal Constraints (GTC), a new expressive model for temporal pattern representation. We also presented two methods (GTC-DF and GTC-Set) of inferring GTCs from data to enable classification of Symbolic and Scalar Time Sequences (SSTS). Our methods stand-out from previously known approaches because: (1) They offer a high power of expression, enabled by the parametric and structural nature of the proposed representation; (2) The computational cost of learning and applying these models is manageable and often competitive when compared to the most potent alternatives; (3) The

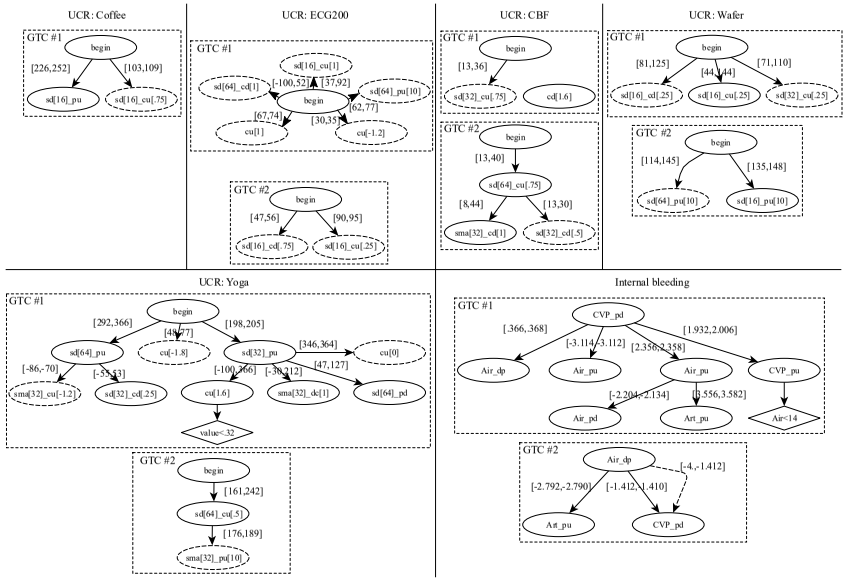


Figure 10: Graphical representations of the first two selected GTCs for the first cross-validation fold of a few evaluation data sets with the Combination selection method. Meaning of the event and scalar names: **begin**: begin of the SSTS, **sd[x]**: standard deviation over the trailing window x time units wide, **cd[x]**: signal crosses-down a threshold value x, **cu[x]**: signal crosses-up a threshold value x, **sma[x]**: simple moving average over the trailing window x time units wide, **pu**: peak up (local maximum over time), **pd**: peak down (local minimum over time), **CVP**: central venous pressure, **Air**: airway pressure, **Art**: arterial blood pressure.

users can read and interpret the models; and (4) In many cases, the users can easily project any new SSTS data and pursue its meaningful interpretation using only a handful of the most informative GTCs.

A wide range of experiments have been conducted to evaluate the proposed algorithms against a rich and diverse collection of benchmark data, and they have been compared to state-of-the-art alternative methods of temporal data classification. Experiments have shown that the proposed algorithms on average decisively outperform previously known best methods in terms accuracy of their predictions, while remaining highly competitive in terms of overall computational costs. Additionally, we have shown how easy it is to interpret the resulting models and we have presented a simple greedy procedure that in many cases yields a manageably small yet highly expressive and highly accurate models

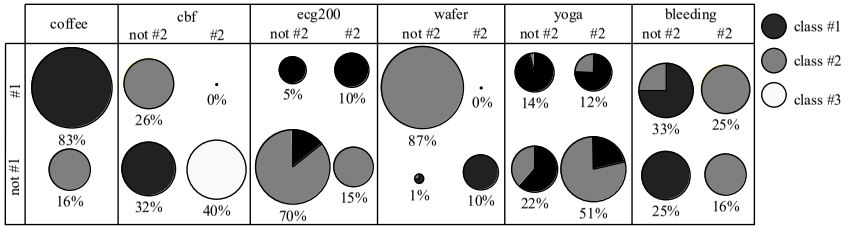


Figure 11: Held-out set empirical class separation achieved by simultaneously using the top two greedily selected GTCs for each of the considered data sets by the Combination selection method. For Coffee data set, only one GTC is needed to achieve the perfect class separation on the particular subset of testing data. The surface of each area represents the proportion of data it represents. This proportion is also shown with the percentage value beneath each graph. Pie charts are used to show distributions of target classes.

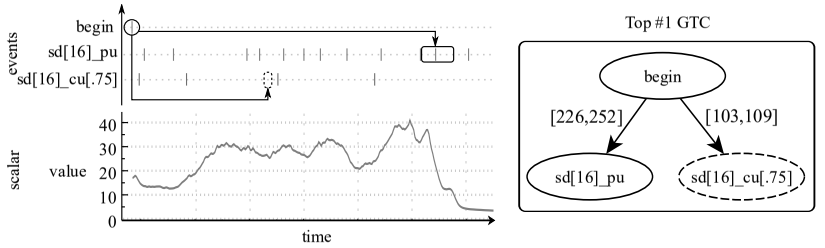


Figure 12: Instance of test SSTS of the UCR Coffee data set and illustration of the application of the first selected GTC extracted for it using the greedy procedure.

that can drive knowledge discovery and interpretability of the results in many real-world application scenarios.

Data Set	Classes	GTC Set	Top 2 Cov	Top 2 Cb	Cov/Cb	Gb/Orig	Top 4 Cov.	Top 4 Cb	Cov/Gb	Cb/Orig
Trace	5	**.000**	.325	**.000**		**.000**	.325	.005	65.000	
Lighting2	**2**	.223	.298	.289	1.029	1.296	.298	.248	1.200	**1.111**
ECG200	**2**	.120	.230	.190	1.211	1.583	.205	.175	1.171	1.458
Coffee	**2**	.036	.071	.071	1.000	2.000	.071	.071	1.000	2.000
Beef	6	.250	.683	.500	1.367	2.000	.567	.500	1.133	2.000
OSULeaf	7	.265	.520	.550	**.947**	2.077	.505	.419	1.205	1.581
Lighting7	7	.203	.566	.538	1.052	2.655	.469	.406	1.155	2.000
Wafer	**2**	.001	**.003**	.002	1.429	2.800	**.001**	**.002**	**.625**	3.200
50words	51	.235	.768	.754	1.019	3.202	.691	.625	1.104	2.657
CBF	4	.004	.014	.014	1.000	3.250	.013	.015	.857	3.500
Adiac	38	.256	.941	.930	1.012	3.630	.904	.816	1.108	3.185
Yoga	3	.045	.245	.222	1.102	4.946	.172	.198	.867	4.419
GunPoint	3	.010	.050	.050	1.000	5.000	.050	.055	.909	5.500
FaceFour	5	.018	.339	.098	3.455	5.500	.339	.116	2.923	6.500
Fish	8	.074	.700	.546	1.283	7.346	.500	.271	1.842	3.654
FaceAll	15	.045	.672	.684	.983	15.22	.557	.492	1.132	10.96
Synthetic	7	.012	.588	.478	1.230	41.00	.392	.143	2.733	12.28
2Patterns	5	.001	.301	.300	1.006	374.5	.224	.087	2.568	109.2

Table 6: Comparison of empirical classification error rates observed via 10-fold cross-validation when using only the top few (2 and 4) GTCs obtained using the Coverage (Cov) and Combination (Cb) selection methods, versus using the fully developed GTC-Set model. "Cb/Orig" denotes the ratio of error rates of the compact model extracted by the combination selection method vs. full GTC-Set. "Cov/Cb" denotes the ratio of error rates of the coverage selection method vs. the combination selection method. It is intended to reflect the drop of performance due to using a simplified representation. Entries are sorted by the ratio for the 2-GTC models ("Cb/Orig").

References

Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1-55860-153-8. URL `http://dl.acm.org/citation.cfm?id=645920.672836`.

Johannes Aßfalg, Hans-Peter Kriegel, Peer Kröger, Peter Kunath, Alexey Pryakhin, and Matthias Renz. Similarity Search on Time Series Based on Threshold Queries. *Advances in Database Technology*, pages 276–294, 2006.

Iyad Batal, Dmitriy Fradkin, James Harrison, Fabian Moerchen, and Milos Hauskrecht. Mining recent temporal patterns for event detection in multivariate time series data. *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*, page 280, 2012.

Stephen D. Bay and Michael J. Pazzani. Detecting group differences: Mining contrast sets. *Data Mining and Knowledge Discovery*, 5(3):213–246, 2001. ISSN 1573-756X. doi: 10.1023/A:1011429418057. URL `http://dx.doi.org/10.1023/A:1011429418057`.

Donald Berndt and James Clifford. Using dynamic time warping to find patterns in time series. *Workshop on Knowledge Knowledge Discovery in Databases*, 398:359–370, 1994.

Leo Breiman. Random forests. *Machine learning*, pages 5–32, 2001.

Björn Bringmann, Siegfried Nijssen, and Albrecht Zimmermann. Pattern-based classification: A unifying perspective. *CoRR*, abs/1111.6191, 2011. URL `http://arxiv.org/abs/1111.6191`.

Lei Chen and Raymond Ng. On the marriage of Lp-norms and edit distance. *Proceedings of the 30th International Conference on Very Large Data Bases*, 30:792–803, 2004.

Lei Chen, M. Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. *Proceedings of the 2005 ACM SIGMOD international conference on Management of data - SIGMOD '05*, page 491, 2005.

Rina Dechter. Temporal constraint networks, 1991.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006. ISSN 1532-4435.

Houtao Deng, George Runger, Eugene Tuv, and Martyanov Vladimir. A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153, 2013.

Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, and Eamonn Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 43–52, New York, NY, USA, 1999. ACM. ISBN 1-58113-143-7. doi: 10.1145/312129.312191. URL `http://doi.acm.org/10.1145/312129.312191`.

Christophe Dousson and Thang Vu Duong. Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. *IJCAI International Joint Conference on Artificial Intelligence*, 1:620–626, 1999.

Elias Frentzos, Kostas Gratsias, and Yannis Theodoridis. Index-Based Most Similar Search. *Proceedings of the 23th International Conference on Data Engineering (ICDE'07)*, pages 816–825, 2007.

Yoav Freund and Robert E. Schapire. A Short Introduction to Boosting. 14(5):771–780, 2009.

Mathieu Guillame-Bert. Datasets for gtc classification. URL `http://mathieu.guillame-bert.com/dataset`.

Mathieu Guillame-Bert. *Learning Temporal Association Rules on Symbolic Time Sequences*. PhD thesis, 2012.

Mathieu Guillame-Bert and Artur Dubrawski. Learning Temporal Rules to Forecast Events in Multivariate Time Sequences. In *NIPS Workshop : Machine Learning for Clinical Data Analysis, Healthcare and Genomics*, 2014.

Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.*, 8 (1):53–87, January 2004. ISSN 1384-5810. doi: 10.1023/B:DAMI.0000005258.31418.83. URL `http://dx.doi.org/10.1023/B:DAMI.0000005258.31418.83`.

Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, 7(3):358–386, March 2005.

Willi Klösgen. Advances in knowledge discovery and data mining. chapter Explora: A Multipattern and Multistrategy Discovery Assistant, pages 249–271. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996. ISBN 0-262-56097-6. URL `http://dl.acm.org/citation.cfm?id=257938.257965`.

Wenmin Li, Jiawei Han, and Jian Pei. Cmar: accurate and efficient classification based on multiple class-association rules. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 369–376, 2001.

Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. pages 80–86, 1998.

Heikki Mannila, Hannu Toivonen, and Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge*, 289:259–289, 1997.

David Meyer, Friedrich Leisch, and Kurt Hornik. The support vector machine under test. *Neurocomputing*, 55(1-2):169–186, 2003.

Michael D. Morse and Jignesh M. Patel. An efficient and accurate method for evaluating time series similarity. *ACM SIGMOD international conference on Management of data*, page 569, 2007.

Stephen Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.

Petra Kralj Novak, Nada Lavrač, and Geoffrey I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *J. Mach. Learn. Res.*, 10:377–403, June 2009. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1577069.1577083`.

S. Nowozin, K. Tsuda, T. Uno, T. Kudo, and G. Bakir. Weighted substructure mining for image analysis. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383171.

Jian Pei, Jiawei Han, B. Mortazavi-Asl, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. Prefixspan,: mining sequential patterns efficiently by prefix-projected pattern growth. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 215–224, 2001. doi: 10.1109/ICDE.2001.914830.

J Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

Juan J. Rodríguez and Carlos J. Alonso. Interval and dynamic time warping-based decision trees. *Proceedings of the 2004 ACM symposium on Applied computing - SAC '04*, page 548, 2004.

Juan J. Rodriguez, Carlos J. Alonso, and Henrik Bostrom. Learning first order logic time series classifiers: Rules and boosting. In DjamelA. Zighed, Jan Komorowski, and Jan Zytkow, editors, *Principles of Data Mining and Knowledge Discovery*, volume 1910 of *Lecture Notes in Computer Science*, pages 299–308. Springer Berlin Heidelberg, 2000.

Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Proceedings of the 5th International Conference on Extending Database Technology: Advances in Database Technology*, EDBT '96, pages 3–17, London, UK, UK, 1996. Springer-Verlag. ISBN 3-540-61057-X. URL `http://dl.acm.org/citation.cfm?id=645337.650382`.

M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. *Proceedings 18th International Conference on Data Engineering*, 2002.

Xifeng Yan, Hong Cheng, Jiawei Han, and Philip S. Yu. Mining significant graph patterns by leap search. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 433–444, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-102-6. doi: 10.1145/1376616.1376662. URL `http://doi.acm.org/10.1145/1376616.1376662`.

Lexiang Ye and Eamonn Keogh. Time series shapelets: a new primitive for data mining. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956, 2009.

Mohammed J. Zaki. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2):31–60, 2001.

Journal of Machine Learning Research 18 (2017) 1-43 Submitted 12/16; Revised 8/17; Published 11/17

Distributed Stochastic Variance Reduced Gradient Methods by Sampling Extra Data with Replacement

Jason D. Lee JASONLEE@MARSHALL.USC.EDU
Marshall School of Business
University of Southern California
Los Angeles, CA 90089, USA

Qihang Lin QIHANG-LIN@UIOWA.EDU
Tippie College of Business
University of Iowa
Iowa City, IA 52242, USA

Tengyu Ma TENGYU@CS.PRINCETON.EDU
Department of Computer Science
Princeton University
Princeton, NJ 08544, USA

Tianbao Yang TIANBAO-YANG@UIOWA.EDU
Department of Computer Science
University of Iowa
Iowa City, IA 52242, USA

Editor: Francis Bach

Abstract

We study the round complexity of minimizing the average of convex functions under a new setting of distributed optimization where each machine can receive two subsets of functions. The first subset is from a random partition and the second subset is randomly sampled with replacement. Under this setting, we define a broad class of distributed algorithms whose local computation can utilize both subsets and design a distributed stochastic variance reduced gradient method belonging to in this class. When the condition number of the problem is small, our method achieves the optimal parallel runtime, amount of communication and rounds of communication among all distributed first-order methods up to constant factors. When the condition number is relatively large, a lower bound is provided for the number of rounds of communication needed by any algorithm in this class. Then, we present an accelerated version of our method whose the rounds of communication matches the lower bound up to logarithmic terms, which establishes that this accelerated algorithm has the lowest round complexity among all algorithms in our class under this new setting.

Keywords: distributed optimization, communication complexity, first-order method, lower bound, stochastic variance reduced gradient

©2017 Jason D. Lee, Qihang Lin, Tengyu Ma and Tiangbao Yang.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-640.html.

1. Introduction

In this paper, we consider the *distributed optimization* problem of minimizing the average of N convex functions $f_i : \mathbb{R}^d \to \mathbb{R}$ for $i = 1, \ldots, N$, i.e.,

$$\min_{x\in\mathbb{R}^d} \left\{ f(x) := \frac{1}{N}\sum_{i=1}^{N} f_i(x) \right\} \tag{1}$$

using m machines. For simplicity of notation, we assume $N = mn$ for an integer n but all of our results can be easily generalized for an arbitrary N. In this paper, the norm $\|\cdot\|$ represents the Euclidean norm and $\langle\cdot,\cdot\rangle$ represents the inner product in $\mathbb{R}^d$. Throughout the whole paper, we make the following standard assumptions on problem (1).

Assumption 1 *The functions f_i for $i = 1, \ldots, N$ in (1) satisfy the following properties:*

- *Each function f_i is convex and L-smooth, which means f_i is differentiable and its gradient ∇f_i is L-Lipschitz continuous, i.e., $\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x-y\|$, $\forall x, y \in \mathbb{R}^d$.*
- *The average function f is μ-convex with $\mu \ge 0$, i.e., $f(x) \ge f(y) + \langle \nabla f(y), x - y\rangle + \frac{\mu}{2}\|x-y\|^2$, $\forall x, y \in \mathbb{R}^d$.*

When $\mu > 0$, f is *μ-strongly convex* and we call $\kappa = \frac{L}{\mu}$ the *condition number* of f. Note that the function f itself can be L_f-smooth, namely, $\|\nabla f(x) - \nabla f(y)\| \le L_f\|x-y\|$, $\forall x, y \in \mathbb{R}^d$, for a constant $L_f \le L$. Let x^* be an optimal solution of (1) and a deterministic solution $\hat{x}$ is called an *ϵ-optimal solution* to (1) if $f(\hat{x}) - f(x^*) \le \epsilon$. If $\hat{x}$ is a random variable generated by a stochastic algorithm, we call it an ϵ-optimal solution if $\mathbb{E}[f(\hat{x}) - f(x^*)] \le \epsilon$.

One of the most important applications of problem (1) is *empirical risk minimization* (ERM) in statistics and machine learning. Let $\{\xi_1, \xi_2, \ldots, \xi_N\}$ be a set of i.i.d. samples from an unknown distribution D. A regularized ERM problem with can be formulated as

$$\min_{x\in\mathbb{R}^d} \left\{ f(x) = \frac{1}{N}\sum_{i=1}^{N} \phi(x, \xi_i) + r(x) \right\}, \tag{2}$$

where x represents a vector of parameters of a predictive model, $\phi(x, \xi)$ is a loss function, and $r(x)$ is a regularization term. Note that (2) has the form of (1) with $f_i(x) = \phi(x, \xi_i) + r(x)$. A commonly used regularization term is $r(x) = \frac{\lambda}{2}\|x\|^2$ and the regularization parameter λ significantly influences μ and κ. The value of λ is typically in the order of $\Theta(1/\sqrt{N}) = \Theta(1/\sqrt{mn})$ as justified by Shamir and Srebro (2014), Shamir et al. (2014), Shalev-Shwartz et al. (2009) and Zhang and Lin (2015). For simplicity, in the rest of the paper, we will call f_i the ith data point or the ith function interchangeably.

We consider a situation where all N functions are initially stored in a large storage space that has limited computation power. Then, these N functions are partitioned, sampled and distributed to m machines where the main computation is performed for solving (1). One common practice under a classical setting is to randomly and evenly partition the N functions into subsets $S_1, S_2, \ldots, S_m$ with $|S_j| = \frac{N}{m} = n$ and store S_j in machine j. In this paper, we consider a *new setting* where an extra set of functions R_j with $|R_j| \le O(|S_j|)$ is sampled with replacement from these N functions so that the set $S_j \cup R_j$ is stored in

machine j for the local computation. Since no machine can access all N functions, we consider solving (1) by synchronized distributed algorithms that alternate between a local computation procedure at each machine, and a round of communication to share information among the machines.

One contribution of this paper is to develop a distributed algorithm for (1) under this new setting. Compared to the previous methods where only S_j is stored in machine j, our method requires more storage space or machines but significantly fewer rounds of communication to achieve an ϵ-optimal solution for (1) when the problem's parameters are in a practical regime, showing the advantage of the new setting. Moreover, we define a broad class of distributed algorithms for (1) under the new setting, which includes the proposed method, and establish a lower bound for the number of rounds of communication needed by any algorithm in this class to find an ϵ-optimal solution. This lower bound matches the number of rounds required by the proposed method up to logarithmic factors, indicating that our lower bound is tight and the proposed method is nearly optimal in this class.

1.1 Performance Metrics

In the design of distributed algorithms, one often needs to consider more performance metrics than in the single-machine scenario. A priori, the most important performance metric is the total runtime of algorithm—the time difference between the start of the first machine and the end of the last machine. However, there are other factors than the machines' CPU time that affect the total runtime such as the communication between machines. Moreover, the latency in initiating communication is also non-negligible, and often the bottleneck of the entire system.

We use the following synchronous message passing model from the distributed computation literature, e.g., Gropp et al. (1996) and Dean and Ghemawat (2008): We assume the communication occurs in rounds—in each round, machines exchanges messages with each other and, between two rounds, machines only perform computation based on their local information (local data points and messages received before). Under this model, three metrics are of main interests to us:

- **Parallel runtime**: The longest running time of the m machines spent in local parallel computation. In distributed first-order optimization methods, we measure it by the number of gradient computations, i.e., $\nabla f_i(x)$ computed for any i and x, in parallel.
- **The amount of communication**: The total amount of communication among m machines, measured by the number of bits transmitted through the network. It does not include the bits of the communication for distributing functions to machines before an algorithm starts. For the algorithms that only communicate iterates x and gradients $\nabla f_i(x)$, the amount of communication can be measured by the number of vectors of size $O(d)$ transmitted through the network.
- **The number of rounds of communication**: How many times m machines have to pause their local computation and initiate a round of communication to exchange messages. We also refer it as "rounds" or "round complexity" for simplicity.

Typically, trade-offs exist among these three metrics and algorithms need to be designed with a balance of these metrics. In this paper, we mostly focus on the third metric, the

number of rounds of communication an algorithm needs for solving (1), which has a significant impact on the total runtime of an algorithm when the speed of network transmission is slow or initiating the communication between machines in each round is time consuming.

1.2 Main Contributions

Distributed algorithms with low round complexity: First, we propose a *distributed stochastic variance reduced gradient* (DSVRG) method in the new distributed setting where machine j has access to two subsets of $\{f_i\}_{i\in[N]}$: a subset S_j of size n from a random partition of $\{f_i\}_{i\in[N]}$ and a subset R_j of size αn with $\alpha > 0$ sampled with replacement from $\{f_i\}_{i\in[N]}$. This method is a simple distributed implementation of the single-machine stochastic variance reduced gradient (SVRG) method (Johnson and Zhang, 2013; Mahdavi et al., 2013; Xiao and Zhang, 2014; Konečnỳ and Richtárik, 2017). We show in Theorem 14 that the DSVRG method with $\alpha = \Theta(1)$ requires $O\left(\left(\frac{\kappa}{n}\right)\log\left(\frac{1}{\epsilon}\right)\right)$ rounds of communication to find an ϵ-optimal solution for (1) in a realistic regime of ϵ and κ that contains most of ERM problems in practices.

Second, using the acceleration techniques developed by Frostig et al. (2015) and Lin et al. (2015), we propose a *distributed accelerated stochastic variance reduced gradient* (DASVRG) method that further improves the theoretical performance of DSVRG. In a practical regime of ϵ and κ similar to DSVRG, we show in Theorem 16 that DASVRG with $\alpha = \Theta(1)$ requires only $O\left(\left(\sqrt{\frac{\kappa}{n}}\right)\log\left(\frac{\kappa}{n}\right)\log\left(\frac{1}{\epsilon}\right)\right)$ rounds of communication to find an ϵ-optimal solution, which is lower than the round complexity of many existing techniques in the traditional setting with $R_j = \emptyset$, showing the advantage of the new distributed setting. The proposed algorithms may require more machines than previous approaches in order to have extra space for storing R_j in additional to S_j. However, when $\alpha = \Theta(1)$ (e.g., $\alpha = 1$), the required number of machines will increase only by a constant factor under the new setting.

Round complexity lower bounds: We also contribute to the answer of a fundamental question about the round complexity for solving (1) under the new setting:

When each machine receives functions from both random partition and sampling with replacement, what is the minimum number of rounds of communication a distributed optimization algorithm needs in order to obtain an ϵ-optimal solution for (1)?

Since different algorithms utilize different operations in local computation which may influence how many rounds are needed, a class of algorithms must be specified before we can answer the question above affirmatively. Hence, we define a broad class of algorithms called *distributed (extended) first-order algorithm*, denoted by $\mathcal{F}_\alpha$ (see Definition 1).

In $\mathcal{F}_\alpha$ with $\alpha \geq 0$, machine j has access to the subsets S_j and R_j of $\{f_i\}_{i\in[N]}$ as described in the DSVRG and DASVRG methods above with $|R_j| = \alpha n$. One realistic setting is when $\alpha \leq O(1)$ so that the memory space required in each machine remains $|R_j| + |S_j| = \Theta(|S_j|) = \Theta(n)$. It is only interesting to consider $\alpha < m - 1$, since, otherwise, $|R_j| + |S_j| \geq N$ which means the memory space of machine j is large enough to store all the N functions so that distributed computing is no longer needed. In $\mathcal{F}_\alpha$, each machine locally maintains a set of vectors W which can be shared with other machines and merged the W sets sent from other machines during a round of communication. Between the rounds of communication, machine j can add to W arbitrarily many linear combinations of the vectors in W and the (proximal) gradients of the functions in $S_j \cup R_j$ evaluated on the

vectors in W. With flexible schemes of communication and computation, the class $\mathcal{F}_\alpha$ with $\alpha = 0$ (i.e., $R_j = \emptyset$) covers many exist methods in the classical setting. When $\alpha > 0$, the class $\mathcal{F}_\alpha$ covers the proposed DSVRG and DASVRG methods.

We show that, when f is μ-strongly convex ($\mu > 0$), any algorithm in $\mathcal{F}_\alpha$ needs at least $\tilde{\Omega}\Big(\sqrt{\frac{\kappa/n+\alpha}{(\alpha+1)^3}}\log\big(\frac{1}{\epsilon}\big)\Big)$ rounds of communication in order to find an ϵ-optimal solution. Here and elsewhere in the paper, $\tilde{\Omega}(\cdot)$ and $\tilde{O}(\cdot)$ hide a logarithmic factor of m. In the scenario where $\alpha \leq O(1)$, this lower bound becomes $\tilde{\Omega}\left(\sqrt{\frac{\kappa}{n}}\log\big(\frac{1}{\epsilon}\big)\right)$, which suggests that the round complexity of DASVRG with $\alpha = \Theta(1)$ is nearly optimal and our lower bound is tight up to logarithmic factors for a practical regime of parameters. Briefly speaking, this lower bound is proved by carefully designing the "worst" $\{f_i\}_{i\in[N]}$ such that, with a high probability, the random subset $S_j \cup R_j$ in any single machine j does not contain enough (first-order) information of f in (1) so that many rounds of communication must be performed. Since this proof is much more challenging than the round complexity analysis of the DSVRG and DASVRG methods and is the main technical result of this paper, we provide it first in the main body of the paper before introducing the new algorithms.

Optimal parallel runtime, amount of communication, and rounds: In some practical regime of parameters, we show that our DSVRG algorithm can be tuned so that it achieves the optimal parallel runtime, the optimal amount of communication, and the optimal number of rounds of communication *simultaneously* for solving (1). Concretely, when $\kappa = O(n^{1-2\delta})$ with a constant $0 < \delta < \frac{1}{2}$, with appropriate choices of the control parameters (shown in Proposition 15), DSVRG finds an ϵ-optimal solution with a parallel runtime of $O(n)$, an $O(m)$ amount of communication and $O(1)$ rounds of communication for any $\epsilon = \Omega(\frac{1}{n^s})$ where s is any universal positive constant.

Note that $\kappa = O(n^{1-2\delta})$ is a common setting for machine learning applications. For example, as argued in Shamir and Srebro (2014), Shamir et al. (2014), Shalev-Shwartz et al. (2009) and Zhang and Lin (2015), ERM often has the condition number κ in the order of $\Theta(\sqrt{N}) = \Theta(\sqrt{mn})$. Therefore, when the number of machines m is not too large, e.g., when $m \leq n^{0.8}$, we have that $\kappa = \Theta(\sqrt{mn}) \leq n^{0.9}$ (so that $\delta = 0.05$).[1] Moreover, $\epsilon = n^{-10}$ (so that $s = 10$) is certainly a high enough accuracy for most machine learning applications since it exceeds the statistically optimal accuracy.

These performance guarantees of DSVRG, under the specific setting where $\kappa = O(n^{1-2\delta})$ and $\epsilon = \Omega(\frac{1}{n^s})$, are optimal up to constant factors among all the distributed first-order methods. First, to solve (1), all m machines together need to compute at least $\Omega(N)$ gradients (Agarwal and Bottou, 2015) in total so that each function in $\{f_i\}_{i\in[N]}$ can be accessed at least once. Therefore, at least one machine needs to compute at least $\Omega(n)$ gradients in serial given any possible allocation of functions. Second, the amount of communication is at least $\Omega(m)$ for even simple Gaussian mean estimation problems (Braverman et al., 2016), which can be solved as optimization (1). Third, at least $O(1)$ rounds of communication is needed to integrate the computation results from machines into a final output.

Extension to non-strongly convex problem: Although we mainly focus on strongly convex problems, some of our results can be extended to the non-strongly convex case. Using a similar approach to the strongly convex case, we show that when $\mu = 0$, any algorithm in $\mathcal{F}_\alpha$ needs at least $\tilde{\Omega}\Big(\sqrt{\frac{L}{(\alpha+1)^3 n\epsilon}}\Big)$ rounds of communication to find an

1. If $n = 10^5$, then $n^{0.8} = 10^4$ which is already much more than the number of machines in most clusters.

ϵ-optimal solution for (1). In the scenario where $\alpha \leq O(1)$, this lower bound becomes $\tilde{\Omega}\Big(\sqrt{\frac{L}{n\epsilon}}\Big)$. To achieve of this lower bound, we apply DASVRG to the strongly convex problem $\min_{x\in\mathbb{R}^d}\{f(x) + \frac{\sigma_\epsilon}{2}\|x\|^2\}$ with $\sigma_\epsilon = \Theta(\epsilon)$. According to the round complexity of DASVRG in the strongly convex case, we show in Theorem 17 that, in a certain regime of ϵ and κ, an ϵ-optimal solution can be found by DASVRG with $\alpha = \Theta(1)$ in $O\Big(\Big(\sqrt{\frac{L}{n\epsilon}}\Big)\log\Big(\frac{L}{n\epsilon}\Big)\log\big(\frac{1}{\epsilon}\big)\Big)$ rounds of communication, showing that our lower bound is tight up to logarithmic factors in that regime.

1.3 Notations and Outline

For a positive integer K, let $[K]$ denote the set $\{1, 2, \ldots, K\}$. We define a *multi-set* as a collection of items where some items can be repeated. We say two multi-sets are equal if they contain the same sets of unique elements and each unique element appears for the same number of times in both multi-sets. For a multi-set R, the notation $R\backslash\{i\}$ represents a multi-set identical to R except that the occurrence of i, if $i \in R$, is reduced by one. The union of a regular set S and a multi-set R, denoted by $S \cup R$, is defined as a regular set consisting of the items in either S or R without repetition. Let $A \otimes B$ denote the Kronecker product of matrices A and B and let $\mathbf{1}_n$ denotes the all-one vector in $\mathbb{R}^n$.

The rest of this paper is organized as follows. In Section 2, we review the related literature and compare the theoretical performance of our methods with existing work in distributed optimization. In Section 3, we define a broad family of distributed optimization algorithms and state our main results on the lower bounds for the rounds of communication for this family. In Section 4, we provide the proof of the main results. In Section 5 and Section 6, we present the DSVRG and DASVRG algorithms, respectively, and show their theoretical guarantees. We briefly describe how to apply DASVRG to non-strongly convex problems in Section 7. Finally, we present some numerical results in Section 8 and conclude the paper in Section 9.

2. Related Work

The work most closely related to our paper is Arjevani and Shamir (2015), where a lower bound for the rounds of communication was established for solving

$$\min_{x\in\mathbb{R}^d}\left\{\frac{1}{m}\sum_{j=1}^{m}\bar{f}_j(x)\right\} \tag{3}$$

using a broad class of distributed algorithms when machine j has only access to the local function $\bar{f}_j(x)$ for $j = 1, 2, \ldots, m$. To connect (3) to (1), we can define the local function as

$$\bar{f}_j(x) := \frac{1}{|S_j|}\sum_{i\in S_j} f_i(x) \tag{4}$$

for a given partition $\{S_j\}_{j\in[m]}$ of $\{f_i\}_{i\in[N]}$. Arjevani and Shamir (2015) proved that, if the local functions $\{\bar{f}_j\}_{j\in[m]}$ are δ-related (see Arjevani and Shamir (2015) for the definition) and f is strongly convex, the class of algorithms they considered needs at least $\Omega\big(\big(\sqrt{\delta\kappa}\big)\log\big(\frac{1}{\epsilon}\big)\big)$

rounds of communication to find an ϵ-optimal solution for (3). When $\delta = \Omega(1)$, their lower bound can be achieved by a straightforward centralized distributed implementation of accelerated gradient methods. In a specific context of linear regression with functions in S_j being a i.i.d. sample, it was shown by Shamir et al. (2014) that $\delta = O(\frac{1}{\sqrt{n}})$. In this case, the lower bound by Arjevani and Shamir (2015) becomes $O(\frac{\sqrt{\kappa}}{n^{.25}}\log(\frac{1}{\epsilon}))$, which has already been achieved by DISCO (Zhang and Lin, 2015) and AIDE (Reddi et al., 2016).

With our notation, the aforementioned algorithms all belong to the family $\mathcal{F}_0$ where each machine only receives the subset S_j from the random partition of $\{f_i\}_{i\in[N]}$. Unlike all previous work, the DSVRG and DASVRG algorithms we propose belong to $\mathcal{F}_\alpha$ with $\alpha > 0$ so that each machine has access to a subset R_j sampled with replacement from $\{f_i\}_{i\in[N]}$ in addition to S_j. Moreover, in the setting of Arjevani and Shamir (2015), machine j processes the entire local average function $\bar{f}_j$ in (4) in the local computation while machine j in our algorithms can process each individual function in $S_j \cup R_j$. Because of these key differences, DSVRG and DASVRG do not fall into the class of algorithms subject to the lower bound in Arjevani and Shamir (2015). That's why the communication lower bound we establish for $\mathcal{F}_\alpha$ and the communication upper bound ensured by DASVRG can be both lower than the lower bound by Arjevani and Shamir (2015). The "worst" set of functions $\{f_i\}_{i\in[N]}$ constructed in our main proof is a generalization of the set of functions in Arjevani and Shamir (2015). Chen et al. (2017) established lower bounds for the round complexity when the data is partitioned on features which is different from the setting in our paper and Arjevani and Shamir (2015) where the data is partitioned on samples.

Recently, there have been many distributed optimization algorithms proposed for problem (1) when f is strongly convex. We list several of them, including a distributed implementation of the accelerated gradient method (Accel Grad) by Nesterov (2013), in Table 1 and present their rounds and key assumptions for a clear comparison. Except DSVRG and DASVRG, all algorithms in Table 1 belong to $\mathcal{F}_0$ (i.e., $R_j = \emptyset$).

The $O(\sqrt{\kappa_f}\log(\frac{1}{\epsilon}))$ rounds of communication ($\kappa_f := \frac{L_f}{\mu}$) of Accel Grad in Table 1 is directly from the iteration complexity of the single-machine accelerated gradient method, and the $O(\sqrt{\kappa}\log(\frac{1}{\epsilon}))$ round complexity of centralized ADMM was given by Deng and Yin (2016). The distributed dual coordinate ascent method, including DisDCA (Yang, 2013), CoCoA (Jaggi et al., 2014) and CoCoA+ (Ma et al., 2015), is a class of distributed coordinate optimization algorithms which can be applied to the conjugate dual formulation of (2) when $f_i(x) = \phi(x, \xi_i) = \phi_i(\xi_i^T x)$ for ϕ_i on $\mathbb{R}^1$. By Ma et al. (2015) and Jaggi et al. (2014), CoCoA+ requires $O(\kappa\log(1/\epsilon))$ rounds of communication to find an ϵ-optimal solution.[2] Therefore, when applicable, both DSVRG and DASVRG with $\alpha = \Theta(1)$ require fewer rounds of communication than the methods mentioned above.

Assuming the problem (1) has the form of (2) with ξ_i's i.i.d. sampled from a distribution D (denoted by $\xi_i \overset{iid}{\sim} D$), the DANE (Shamir et al., 2014) and DISCO (Zhang and Lin, 2015) algorithms require $O((1+\frac{\sqrt{\kappa}}{n^{.25}})\log(1/\epsilon))$ and $O((1+\frac{\kappa^2}{n})\log(1/\epsilon))$ rounds of communication, respectively. Hence, DSVRG uses fewer rounds of communication than DANE and fewer than DISCO when $\kappa \le n^{1.5}$. If applicable, DASVRG always uses fewer rounds of communication than DISCO and DANE. Furthermore, DANE only has the theoretical guarantee

2. CoCoA+ has a better theoretical performance than CoCoA. According to Ma et al. (2015), CoCoA+ is equivalent to DisDCA with "practical updates" (Yang, 2013) under certain choices of parameters.

Algorithm	Rounds	Settings (when Assumption 1 holds)
DSVRG	$(1+\frac{\kappa}{n})\log\frac{1}{\epsilon}$	$\kappa\log\left(\frac{1}{\epsilon}\right) \le O(N)$, $\alpha = \Theta(1)$
DASVRG	$(1+\sqrt{\frac{\kappa}{n}})\log(1+\frac{\kappa}{n})\log\frac{1}{\epsilon}$	$(1+\sqrt{\frac{\kappa}{n}})\log(1+\frac{\kappa}{n})\log\frac{1}{\epsilon} \le O(m)$, $\alpha = \Theta(1)$
DISCO	$(1+\frac{\sqrt{\kappa}}{n^{.25}})\log\frac{1}{\epsilon}$ $d^{.25}\left((1+\frac{\sqrt{\kappa}}{n^{.25}})\log\frac{1}{\epsilon} + \frac{\kappa^{1.5}}{n^{.75}}\right)$	(2) with $\xi_i \overset{iid}{\sim} D$ and quadratic $\phi(x,\xi_i)$ (2) with $\xi_i \overset{iid}{\sim} D$
DANE	$\delta^2\kappa^2\log\frac{1}{\epsilon}$ $(1+\frac{\kappa^2}{n})\log\frac{1}{\epsilon}$ $\kappa_f\log\frac{1}{\epsilon}$	δ-related $\{\bar{f}_j\}_{j\in[m]}$ (2) with $\xi_i \overset{iid}{\sim} D$ and quadratic $\phi(x,\xi_i)$
AIDE	$\sqrt{\delta\kappa}\log\frac{1}{\epsilon}$ $(1+\frac{\sqrt{\kappa}}{n^{.25}})\log\frac{1}{\epsilon}$ $\sqrt{\kappa_f}\log\frac{1}{\epsilon}$	δ-related $\{\bar{f}_j\}_{j\in[m]}$ (2) with $\xi_i \overset{iid}{\sim} D$ and quadratic $\phi(x,\xi_i)$
CoCoA+	$\kappa\log\frac{1}{\epsilon}$	(2) with $\phi(x,\xi_i) = \phi_i(\xi_i^T x)$
Accel Grad	$\sqrt{\kappa_f}\log\frac{1}{\epsilon}$	
ADMM	$\sqrt{\kappa}\log\frac{1}{\epsilon}$	μ-strongly convex f_i

Table 1: Rounds and settings of different distributed optimization algorithms. Except DSVRG and DASVRG, all algorithms in this table only require $\alpha = 0$ (i.e., they do not require a subset R_j sampled with replacement).

mentioned above when it is applied to quadratic problems. Also, DISCO only applies to self-concordant functions with easily computed Hessian[3] and makes strong statistical assumptions on the data points. On the contrary, DSVRG and DASVRG works for a more general problem (1) and do not assume $\xi_i \overset{iid}{\sim} D$ for (2).

In each iteration, DANE requires solving a non-trivial sub-problem to optimality. Reddi et al. (2016) proposed an InexactDane algorithm that is similar to DANE but only needs to solve the sub-problem approximately and still achieves the same round complexity as DANE up to a logarithmic factor. Applying the acceleration techniques developed by Frostig et al. (2015) and Lin et al. (2015) to InexactDane, Reddi et al. (2016) further proposed an Accelerated Inexact DanE (AIDE) algorithm that matches the round complexity of DISCO up to a logarithmic factor.

Shamir (2016) consider a without-replacement SVRG which, if implemented in a distributed way, is similar to the DSVRG proposed here. The main difference between his method and DSVRG is that his method sequentially accesses the data points in S_j while while DSVRG accesses the data points in R_j for the local update. Since Shamir (2016) assumes that the concatenation of $\{S_j\}_{j\in[m]}$ forms a random permutation of $\{f_i\}_{[N]}$, his method essentially performs the iterative update of SVRG by sampling from $\{f_i\}_{[N]}$ without replacement. When κ is smaller than N and ϵ is not extremely small (the same as what DSVRG requires), his method obtains the same round complexity as DSVRG. However, his complexity analysis only applies for quadratic $\{f_i\}_{[N]}$. Konečný et al. (2016) proposed a

3. The examples in Zhang and Lin (2015) all take the form of $f_i(x) = \phi_i(\xi_i^T x)$ for some function ϕ_i on $\mathbb{R}^1$, which is more specific than $\phi(x,\xi_i)$, so that it is relatively easy to compute the Hessian of f_i.

Federated SVRG algorithm which is also a distributed implementation of SVRG and can be viewed as a special case of InexactDane with specific choices of parameters (Reddi et al., 2016). However, the performance analysis of InexactDane under those specific choices of parameters is only available when $\{f_i\}_{[N]}$ are quadratic and δ-related.

The methods in comparison in Table 1 are centralized methods which are typically implemented in a star network with a center server. There also exist many distributed (sub)gradient methods (Nedić and Ozdaglar, 2007, 2009; Duchi et al., 2012; Jakovetić et al., 2012; Chen and Ozdaglar, 2012; Jakovetić et al., 2014; Beck et al., 2014; Shi et al., 2015a,b; Aybat et al., 2015) and distributed ADMM methods (Boyd et al., 2011; Wei and Ozdaglar, 2012; Mota et al., 2013; Shi et al., 2014; Chang et al., 2015; Makhdoumi and Ozdaglar, 2017; Deng and Yin, 2016; Mokhtari et al., 2016) for decentralized optimization over general networks. In general, the decentralized distributed methods require more rounds of communication than the centralized ones. For example, the number of communication rounds needed by the decentralized ADMM in Makhdoumi and Ozdaglar (2017) is $O(\sqrt{\kappa}\log(\frac{1}{\epsilon}))$ multiplied by a network-dependent factor that is typically greater than one.

3. Lower Bounds on Rounds of Communication

In this section, we first define a broad family of distributed (extended) first-order methods which contains many existing distributed optimization algorithms in the literature. Then, we prove the lower bounds for the round complexity for this family of algorithms to find an ϵ-optimal solution for (1) under Assumption 1 with $\mu > 0$ and $\mu = 0$, respectively.

The algorithms in the family we consider consist of a data distribution stage where the functions $\{f_i\}_{i\in[N]}$ are distributed onto m machines, and a distributed computation stage where, in each round, machines can not only use first-order information of the functions stored locally but also apply preconditioning using local second-order information.

Definition 1 (Distributed (extended) first-order algorithms $\mathcal{F}_\alpha$) *Suppose there is a stage of data distribution where $\{f_i\}_{i\in[N]}$ are distributed to m machines such that:*

1. *The index set $[N]$ is randomly and evenly partitioned into $S_1, S_2, \ldots, S_m$ with $|S_j| = n$ for $j = 1, \ldots, m$.*

2. *A multi-set R_j of size αn is created by sampling with replacement from $[N]$ for $j = 1, \ldots, m$, where $\alpha \geq 0$ is a constant. When $\alpha = 0$, we have $R_j = \emptyset$ for all j.*

3. *Let $S'_j = S_j \cup R_j$. Machine j acquires functions in $\{f_i\}_{i\in S'_j}$ for $j = 1, \ldots, m$.*

We say an algorithm $\mathcal{A}$ for solving (1) belongs to the family $\mathcal{F}_\alpha$ ($\mathcal{A} \in \mathcal{F}_\alpha$) of distributed (extended) first-order algorithms if the machines do the following operations in rounds:

1. *Machine j maintains a local working set of vectors $W_j \subset \mathbb{R}^d$ initialized to be $W_j = \{\mathbf{0}\}$, where $\mathbf{0}$ is the all-zero vector in $\mathbb{R}^d$.*

2. *In each round, for arbitrarily many times, machine j can add any $w \in \mathbb{R}^d$ to W_j if w satisfies (c.f. Arjevani and Shamir (2015))*

$$\gamma w + \nu \nabla F_j(w) \in span\Big\{ w', \nabla F_j(w'), (\nabla^2 F_j(w') + D)w'', (\nabla^2 F_j(w') + D)^{-1} w'' \Big| \\ w', w'' \in W_j, D \text{ is diagonal}, \nabla^2 F_j(w') \text{ and } (\nabla^2 F_j(w') + D)^{-1} \text{ exist} \Big\} \quad (5)$$

   for some constants $\gamma \geq 0$ and $\nu \geq 0$ with $\gamma\nu \neq 0$, where $F_j = \sum_{i\in U_j \subset S'_j} f_i$ with an arbitrary subset U_j of S'_j.

3. *At the end of the round, all machines can simultaneously send any vectors in W_j to any other machines, and each machine can add the vectors received from other machines to its local working set.*

4. *The final output is a vector in the linear span of W_j for any j.*

We use $\mathcal{A}(\{f_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H)$ to represent the output vector of $\mathcal{A}$ when it is applied to (1) for H rounds with the inputs $\{f_i\}_{i\in[N]}$ and the sets $\{S'_j\}_{j\in[m]}$ generated in the data distribution stage described above.

Besides the randomness due to the data distribution stage, the algorithm $\mathcal{A}$ itself can be a randomized algorithm. Hence, the output $\mathcal{A}(\{f_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H)$ can be a random vector. Next, we provide some discussions on this class of algorithms.

In $\mathcal{F}_\alpha$, machines can only share iterates but not data points explicitly. Therefore, the sets $\{S'_j\}_{j\in[m]}$ will not be changed during the algorithm. The class $\mathcal{F}_\alpha$ allows gradient update $w = w' - \eta\nabla F_j(w')$ with $\eta > 0$ in local computation, which corresponds to (5) with $\gamma = 1, \nu = 0$. It also allows proximal mapping update $w = \arg\min_v \frac{1}{2\eta}\|v - w'\|^2 + F_j(v)$ with $\eta > 0$ in local computation, which has an optimality condition $w + \eta\nabla F_j(w) = w'$ and thus corresponds to (5) with $\gamma = 1, \nu = \eta$. As a result, the class $\mathcal{F}_0$ covers distributed Accel Grad, ADMM, DANE and AIDE. Although the algorithms in $\mathcal{F}_\alpha$ can use the local second-order information like $\nabla^2 f_i(x)$ in each machine, Newton's method is still not contained in $\mathcal{F}_\alpha$ since Newton's method requires the access to the global second-order information such as $\nabla^2 f(x)$ (machines are not allowed to share Hessian matrices with each other). That being said, one can still use a distributed iteration method which can multiply $\nabla^2 f_i(x)$ to a local vector in order to solve the inversion of $\nabla^2 f(x)$ approximately. This scheme will lead to a distributed inexact Newton method such as DISCO, which also belongs to $\mathcal{F}_0$. The DSVRG and DASVRG algorithms proposed in this paper belong to $\mathcal{F}_\alpha$ with $\alpha > 0$.

We are ready to present the lower bounds for the rounds of communications.

Theorem 2 *Suppose $m \geq (e + 4\max\{1, \alpha\})^2$ and there exists an algorithm $\mathcal{A} \in \mathcal{F}_\alpha$ with the following property:*

"For any $\epsilon > 0$ and any N convex functions $\{f_i\}_{i\in[N]}$ satisfying Assumption 1 with $\mu > 0$ and $\kappa \geq 1.5n$, there exists H_ϵ such that the output $\hat{x} = \mathcal{A}(\{f_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H_\epsilon)$ satisfies $\mathbb{E}[f(\hat{x}) - f(x^*)] \leq \epsilon$, where the sets $\{S'_j\}_{j\in[m]}$ are generated as in Definition 1."

Then, with $k \equiv \lceil (e + 4\max\{1, \alpha\})\log m \rceil$, we must have

$$H_\epsilon \geq \left(\frac{\sqrt{2\kappa/(3n) + k - 1} - \sqrt{k}}{4k\sqrt{k}} \right) \log\left(\left(1 - \frac{1}{e}\right) \frac{\mu\|x^*\|^2}{4\epsilon} \right) \geq \tilde{\Omega}\left(\sqrt{\frac{\kappa/n + \alpha}{(\alpha+1)^3}} \log\left(\frac{\mu\|x^*\|^2}{\epsilon} \right) \right).$$

In the definition of $\mathcal{F}_\alpha$, we allow the algorithm to access both randomly partitioned data and independently sampled data, and allow the algorithm to use local Hessian for preconditioning. This makes our lower bounds in Theorem 2 **stronger**: Even with more options in distributing data and with algorithms more powerful than first-order methods (in terms of the class of operations it can take), the number of rounds needed to find an ϵ-optimal solution still cannot be reduced.

The $\kappa \geq 1.5n$ condition is not critical because, when $\kappa \leq \frac{n^{1-2\delta}}{32} < 1.5n$ for a constant $0 < \delta < \frac{1}{2}$, we will show in Proposition 15 in Section 5.2 that DSVRG only needs $O(\frac{\log(1/\epsilon)}{\delta \log n})$ rounds, which almost meets the trivial lower bound $\Omega(1)$.

A similar result can be obtained for the non-strongly convex case.

Theorem 3 *Suppose $m \geq (e + 4\max\{1, \alpha\})^2$ and there exists an algorithm $\mathcal{A} \in \mathcal{F}_\alpha$ with the following property:*

"For any $\epsilon > 0$ and any N convex functions $\{f_i\}_{i\in[N]}$ satisfying Assumption 1 with $\mu = 0$, there exists H_ϵ such that the output $\hat{x} = \mathcal{A}(\{f_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H_\epsilon)$ satisfies $\mathbb{E}[f(\hat{x}) - f(x^*)] \leq \epsilon$, where the sets $\{S'_j\}_{j\in[m]}$ are generated as in Definition 1."

Then, with $k \equiv \lceil (e + 4\max\{1, \alpha\}) \log m \rceil$, we must have $H_\epsilon \geq \sqrt{(1 - \frac{1}{e}) \frac{L\|x^*\|^2}{64k^3 n\epsilon}}$.

4. Proofs of Main Results

In this section, we will provide the proofs for Theorem 2 and 3. Before presenting all technical details, we will first give an informal sketch of the proofs.

4.1 Sketch of Proof for Main Results

We will only provide an informal proof sketch for Theorem 2. The proof of Theorem 3 follows a similar strategy.

Let k be an integer with $k = \Theta(\log(m))$, $\mu' = n\mu$ and $\kappa' = \frac{L}{\mu'}$. For simplicity, we assume $\frac{m}{k}$ is an integer in this proof sketch although Theorem 2 and 3 hold without this assumption. Let $E_0 = \{\mathbf{0}\}$ where $\mathbf{0}$ is the all-zero vector in $\mathbb{R}^b$ and let E_t be the subspace of $\mathbb{R}^b$ consisting of the vectors whose non-zero values only appear in the first t coordinates.

Motivated by the proof for the lower bound of the iteration complexity of first-order methods (Nesterov, 2013), we construct a convex quadratic function $\bar{p} : \mathbb{R}^b \to \mathbb{R}$ and represent it as the average of k convex quadratic functions $p_s : \mathbb{R}^b \to \mathbb{R}$ for $s = 1, 2, \ldots, k$. In particular, we construct $\bar{p}(w) = \frac{1}{2}w^T\Sigma w + q^T w + \frac{\mu'}{2}\|w\|^2 = \frac{1}{k}\sum_{s=1}^k p_s$, where $\bar{p}_s(w) = \frac{1}{2}w^T\Sigma_s w + q_s^T w + \frac{\mu'}{2}\|w\|^2$ for $s = 1, 2, \ldots, k$. We are able to choose Σ, Σ_s, q and q_s here such that the following properties are satisfied

1. The matrix $\Sigma = \frac{1}{k}\sum_{s=1}^k \Sigma_s$. The matrix Σ is tridiagonal and the matrix Σ_s is block diagonal with a 2×2 block and $k - 2$ consecutive zero entries alternatively appearing along its diagonal. The sparsity patterns of Σ and Σ_s can be seen in Figure 1 in an example with $b = 15$ and $k = 3$.

2. The vector $q = \frac{1}{k}\sum_{s=1}^k q_s$ with $q, q_s \in E_1$ for $s = 1, \ldots, k$.

3. Each p_s is L-smooth and $\bar{p}$ is μ'-strongly convex.

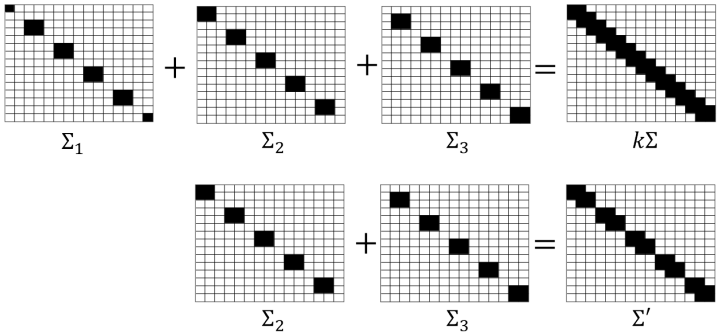


Figure 1: The sparsity pattern of Σ and Σ' when $U = \{2, 3\}$, $b = 15$ and $k = 3$.

4. The vector $w^* = \arg\min_w \bar{p}(w)$ is non-zero in all coordinates and, when b is sufficiently larger than t, $\|\hat{w} - w^*\|^2 \geq \Omega((1 - \frac{1}{\sqrt{\kappa'}})^{2t})$ for any $\hat{w} \in E_t$.

Because the gradient $\nabla\bar{p}(w) = (\Sigma + \mu' I)w + q$, where Σ is tridiagonal and $q \in E_1$, if a first-order method is applied to $\min_w \bar{p}(w)$ with an initial solution of $\mathbf{0}$, the solution will stay in E_t after t iterations. According to the 4th property above, a large number of iterations is needed for the first-order method to achieve an ϵ-optimal solution. We will use a similar argument to show that, when $\min_w \bar{p}(w)$ is solved by a distributed first-order method but each machine does not have a full collection of $\{p_s\}_{s\in[k]}$, a large number of rounds of communication is needed in order to find an ϵ-optimal solution.

Suppose a machine can only locally access the functions in $\{p_s\}_{s\in U}$ with $U \subsetneqq [k]$. By the sparsity patterns of Σ and Σ_s, $\Sigma' = \sum_{s\in U}\Sigma_s$ will be block diagonal with the size of each block being at most $(k + 1) \times (k + 1)$. The sparsity patterns of Σ' can be seen in Figure 1 for an example with $U = \{2, 3\}$, $b = 15$ and $k = 3$. Since $q, q_s \in E_1$, if the local computation of this machine starts with an initial solution in E_t and the solution is updated only using linear combinations of gradients $\nabla p_s(w) = (\Sigma_s + \mu' I)w + q_s$ for $s \in U$, this machine can only produce a solution in E_{t+k} no matter how many local iterations are performed. According to the 4th property above, this machine has to receive a solution outside E_{t+k} from other machines in order to make progress in approximting w^*. Hence, if no machine in a distributed first-order method has a full collection of $\{p_s\}_{s\in[k]}$, many rounds of communication is needed in order to generate an ϵ-optimal solution for $\min_w \bar{p}(w)$.

Since $\bar{p}$ is the average of only $k = \Theta(\log(m))$ functions while the communication lower bound is given for minimizing the average of N functions, we will make a few copies of $\{p_s\}_{s\in[k]}$ and add some zero functions to them. In particular, let $v = \frac{m}{k}$ and $\{g_i\}_{i\in[N]}$ be the multi-set of functions on $\mathbb{R}^b$ that consists of v copies of $\{p_s\}_{s\in[k]}$ and $N - vk$ zero functions. Therefore, $\bar{g}(w) \equiv \frac{1}{N}\sum_{i=1}^N g_i(w) = \frac{\bar{p}(w)}{n}$ so that $w^* = \arg\min_w \bar{g}(w)$. In addition, each g_i is L-smooth due to the 3rd property above and g is $\frac{\mu'}{n}$-strongly (i.e., μ-strongly) convex because m g_i's are μ'-strongly convex and $N - m$ g_i's are zero.

Suppose an algorithm in $\mathcal{F}_\alpha$ is applied to $\min_w \bar{g}(w)$ with the initial solution $\mathbf{0}$. Each machine will receives n functions from $\{g_i\}_{i\in[N]}$ by random partition and another αn func-

tions from $\{g_i\}_{i\in[N]}$ by random sampling with replacement. Since $k = \Theta(\log(m))$ and $\{g_i\}_{i\in[N]}$ contains only $vk = m$ functions from $\{p_s\}_{s\in[k]}$ and $N - m$ zero functions, we can show that, with a high probability, no machine will receive k or more functions from $\{p_s\}_{s\in[k]}$. According to the previous discussion, the number of non-zero coordinate in the solution generated by the algorithm will only increase by at most k between two consecutive rounds of communication. In other words, if $\hat{w}$ is the solution found by $\mathcal{F}_\alpha$ after H rounds of communication, we must have $\hat{w} \in E_t$ with $t = Hk$. By the 4th property above, the distance $\|\hat{w} - w^*\|^2 \geq \Omega((1 - \frac{1}{\sqrt{\kappa'}})^{2t})$. Since $\bar{g}$ is strongly convex, this distance implies an objective function gap of $\bar{g}(\hat{w}) - \bar{g}(w^*) \geq \Omega((1 - \frac{1}{\sqrt{\kappa'}})^{2t})$. Hence, in order to ensure $\bar{g}(\hat{w}) - \bar{g}(w^*) \leq \epsilon$, with a high probably, the rounds of communication must satisfy $H = \frac{t}{k} \geq \Omega(\sqrt{\kappa'}) \log\left(\frac{1}{\epsilon}\right) = \Omega(\sqrt{\frac{\kappa}{n}}) \log\left(\frac{1}{\epsilon}\right)$. This provides the lower bounds for the rounds of communication we found in this paper. Finally, we show that the same lower bound can be proved without having any zero function in the N functions. To do so, we raise the dimension of the problem from b to $d = nb$ by constructing a problem like (1), where $\{f_i\}_{i\in[N]}$ consists of vn copies of $\{p_s\}_{s\in[k]}$ with the first v copies defined on the first b coordinates of x, the second v copies defined on the second b coordinates of x and so on. Problem (1) with such $\{f_i\}_{i\in[N]}$ can be solved as n independent problems on $\mathbb{R}^b$ with each problem equivalent to $\min_w \bar{g}(w)$ so that the same communication lower bound can be obtained.

4.2 Some Definitions and Lemmas

Before we start the formal proof for Theorem 2, some definitions and technical lemmas are provided in this section. Given a vector $x \in \mathbb{R}^d$ and a set of indices $D \subset [d]$, we use x_D to represent the sub-vector of x that consists of the coordinates of x indexed by D.

Definition 4 *A function $f : \mathbb{R}^d \to \mathbb{R}$ is **decomposable** with respect to a partition $D_1, \ldots, D_r$ of coordinates $[d]$ if $f(x) = g^1(x_{D_1}) + \cdots + g^r(x_{D_r})$ with $g^l : \mathbb{R}^{|D_l|} \to \mathbb{R}$ for $l = 1, \ldots, r$. A set of functions $\{f_i\}_{i\in[N]}$ is **simultaneously decomposable** with respect to a partition $D_1, \ldots, D_r$ if each f_i is decomposable with respect to $D_1, \ldots, D_r$.*

It follows the Definition 1 and Definition 4 straightforwardly that:

Proposition 5 *Suppose the functions $\{f_i\}_{i\in[N]}$ in (1) are simultaneously decomposable with respect to a partition $D_1, \ldots, D_r$ so that $f_i(x) = \sum_{l=1}^r g_i^l(x_{D_l})$ with $g_i^l : \mathbb{R}^{|D_l|} \to \mathbb{R}$ for $i = 1, \ldots, N$ and $l = 1, \ldots, r$. Let x^* be the optimal solution of (1). We must have*

$$x^*_{D_l} \in \arg\min_{w\in\mathbb{R}^{|D_l|}} \left\{ \bar{g}^l(w) \equiv \frac{1}{N} \sum_{i=1}^N g_i^l(w) \right\}, \text{ for } l = 1, 2, \ldots, r. \tag{6}$$

Moreover, any algorithm $\mathcal{A} \in \mathcal{F}_\alpha$, when applied to $\{f_i\}_{i\in[N]}$, becomes decomposable with respect to the same partition $D_1, \ldots, D_r$ in the following sense: For $l = 1, \ldots, r$, there exists an algorithm $\mathcal{A}_l \in \mathcal{F}_\alpha$ such that, after any number of rounds H,

$$\mathbb{E}[f(\hat{x}) - f(x^*)] = \sum_{l=1}^r \mathbb{E}[\bar{g}^l(\hat{w}^l) - \bar{g}^l(x^*_{D_l})],$$

where $\hat{x} = \mathcal{A}(\{f_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H) \in \mathbb{R}^d$ and $\hat{w}^l = \mathcal{A}_l(\{g_i^l\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H) \in \mathbb{R}^{|D_l|}$ for $l = 1, 2, \ldots, r$ and $\{S'_j\}_{j\in[m]}$ is generated as in Definition 1.

Proof The proof of this proposition is straightforward. Since $\{f_i\}_{i\in[N]}$ in (1) are simultaneously decomposable with respect to a partition $D_1, \ldots, D_r$, we have

$$f(x) = \frac{1}{N} \sum_{i=1}^N \sum_{l=1}^r g_i^l(x_{D_l}) = \sum_{l=1}^r \bar{g}^l(x_{D_l})$$

so that the problem (1) can be solved by solving (6) for each l separately and $x^*_{D_l}$ must be the solution of the l-th problem in (6).

In addition, the function F_j in Definition 1 is also decomposable with respect to the same partition $D_1, \ldots, D_r$. As a result, its gradient $\nabla F_j(x)$ also has a decomposed structure in the sense that the block $[\nabla F_j(x)]_{D_l}$ only depends on x_{D_l}. Similarly, its Hessian matrix $\nabla^2 F_j(x)$ is a block diagonal matrix with r blocks and the l-th block only depends on x_{D_l}. These properties ensure that each operation $\mathcal{A}$ is able to apply (as in Definition 1) to some $x \in \mathbb{R}^d$ can be decomposed into r independent operations applied on $x_{D_1}, x_{D_2}, \ldots, x_{D_r}$ separately. Hence, given the sets $\{S'_j\}_{j\in[m]}$, the sequence of operations conducted by $\mathcal{A}$ on x_{D_l} can be viewed as an algorithm $\mathcal{A}_l \in \mathcal{F}_\alpha$ applied to $\{g_i^l\}_{i\in[N]}$ so that $\hat{w}^l = \mathcal{A}_l(\{g_i^l\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H)$ is indeed the sub-vector $\hat{x}_{D_l}$ of the vector $\hat{x} = \mathcal{A}(\{f_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H)$ for $l = 1, 2, \ldots, r$ and any H. ■

Because $m \geq (e + 4\max\{1, \alpha\})^2$ and $k \equiv \lceil (e + 4\max\{1, \alpha\}) \log m \rceil$, we can show that

$$\frac{m}{k} \geq \frac{m}{1.25(e + 4\max\{1, \alpha\}) \log m} \geq \frac{e + 4\max\{1, \alpha\}}{2.5 \log(e + 4\max\{1, \alpha\})} \geq \frac{e + 4}{2.5 \log(e + 4)} > 3$$

for any $\alpha \geq 0$, where we use the fact that $1.25x \geq \lceil x \rceil$ for $x \geq 4$ and the fact that $\frac{x}{\log x}$ is monotonically increasing on $[e, +\infty)$. In the rest of the paper, we define

$$v \equiv \left\lfloor \frac{m}{k} \right\rfloor \text{ and } \theta \equiv \frac{vk}{m}.$$

It is easy to show that $\frac{2}{3} < 1 - \frac{k}{m} < \theta \leq 1$, where the first inequality is because $\frac{m}{k} > 3$ and the second and the last inequalities are by the definitions of v and θ.

4.3 Proof for Theorem 2

Now we are ready to give the proof for Theorem 2 in this subsection. Let

$$\mu' \equiv \frac{\mu n}{\theta}, \quad \kappa' \equiv \frac{L}{\mu'} = \frac{\theta\kappa}{n} > \frac{2\kappa}{3n} \geq 1. \tag{7}$$

We first generalize the machinery by Arjevani and Shamir (2015) to construct k functions on $\mathbb{R}^b$ where $b = uk$ for some integers $k, u \geq 1$. The values of k and u will be specified later. In particular, for $i, j = 1, \ldots, b$, let $\delta_{i,j}$ be an $b \times b$ matrix with its (i, j) entry being one and others being zeros. Let $M_0, M_1, \ldots, M_{b-1}$ be $b \times b$ matrices defined as

$$M_i = \begin{cases} \delta_{1,1} & \text{for } i = 0 \\ \delta_{i,i} - \delta_{i,i+1} - \delta_{i+1,i} + \delta_{i+1,i+1} & \text{for } 1 \leq i \leq b-2 \\ \delta_{b-1,b-1} - \delta_{b-1,b} - \delta_{b,b-1} + \frac{\sqrt{\kappa'+k-1}+3\sqrt{k}}{\sqrt{\kappa'+k-1}+\sqrt{k}} \delta_{b,b} & \text{for } i = b-1. \end{cases} \tag{8}$$

For $s \in [k]$, let

$$\Sigma_s = \sum_{i=0}^{u-1} M_{ik+s-1}. \tag{9}$$

For example, when $u = 2$ and $k = 3$ (so $b = 6$), the matrices Σ_s's are given as follows

$$\Sigma_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & \mathcal{K} \end{bmatrix},$$

where $\mathcal{K} = \frac{\sqrt{\kappa'+k-1}+3\sqrt{k}}{\sqrt{\kappa'+k-1}+\sqrt{k}}$.

We define k functions $p_1, \ldots, p_k : \mathbb{R}^b \to \mathbb{R}$ as follows

$$p_s(w) = \begin{cases} \frac{L}{4}\left[\left(1-\frac{\mu'}{L}\right)\frac{w^T\Sigma_1 w}{2} - \left(1-\frac{\mu'}{L}\right)e_1^\top w\right] + \frac{\mu'}{2}\|w\|^2 & \text{for } s = 1 \\ \frac{L}{4}\left[\left(1-\frac{\mu'}{L}\right)\frac{w^T\Sigma_s w}{2}\right] + \frac{\mu'}{2}\|w\|^2 & \text{for } s = 2, \ldots, k, \end{cases} \tag{10}$$

where $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^b$, and denote their average by

$$\bar{p} \equiv \frac{1}{k}\sum_{s=1}^{k} p_s = \frac{L-\mu'}{4k}\left[\frac{1}{2}w^T\left(\sum_{s=1}^{k}\Sigma_s\right)w - e_1^T w\right] + \frac{\mu'}{2}\|w\|^2. \tag{11}$$

According to (7), we have $1 - \frac{\mu'}{L} > 0$ so that p_s for any $s \in [k]$ and $\bar{p}$ are all μ'-strongly convex functions. It is also easy to show that, for each s, $\lambda_{\max}(\Sigma_s) \leq 4$ so that ∇p_s has a Lipschitz continuity constant of $L\left(1-\frac{\mu'}{L}\right) + \mu' = L$ and p_s is L-smooth.

Next, we characterize the optimal solution of $\min_{w\in\mathbb{R}^b} \bar{p}(w)$.

Lemma 6 *Let $h \in \mathbb{R}$ be the smaller root of the equation $h^2 - 2\left(\frac{\kappa'-1+2k}{\kappa'-1}\right)h + 1 = 0$, namely, $h \equiv \frac{\sqrt{\kappa'+k-1}-\sqrt{k}}{\sqrt{\kappa'+k-1}+\sqrt{k}}$. Then, $w^* = (w_1^*, w_2^*, \ldots, w_b^*)^T \in \mathbb{R}^b$ with*

$$w_j^* = h^j, \text{ for } j = 1, 2, \ldots, b \tag{12}$$

is the optimal solutions of $\min_{w\in\mathbb{R}^b} \bar{p}(w)$.

Proof By (9), we have

$$\sum_{s=1}^{k}\Sigma_s = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & \frac{\sqrt{\kappa'+k-1}+3\sqrt{k}}{\sqrt{\kappa'+k-1}+\sqrt{k}} \end{bmatrix}.$$

With this tridiagonal structure in its Hessian matrix, $\bar{p}(w)$ is related to but different from the functions constructed by Nesterov (2013) and Lan and Zhou (2017) to establish the iteration lower bound for single-machine first-order algorithms. We can show that the optimal solution of $\min_{w\in\mathbb{R}^b} \bar{p}(w)$ must satisfy the following optimality conditions

$$\begin{aligned} w_2^* - 2\left(\frac{\kappa'-1+2k}{\kappa'-1}\right)w_1^* + 1 &= 0 \\ w_{j+1}^* - 2\left(\frac{\kappa'-1+2k}{\kappa'-1}\right)w_j^* + w_{j-1}^* &= 0, \text{ for } j = 2, 3, \ldots, b-1 \\ -\left(\frac{\sqrt{\kappa'+k-1}+3\sqrt{k}}{\sqrt{\kappa'+k-1}+\sqrt{k}} + \frac{4k}{\kappa'-1}\right)w_b^* + w_{b-1}^* &= 0. \end{aligned} \tag{13}$$

We can easily verify that $w_j^* = h^j$ for $j = 1, 2, \ldots, b$ satisfy all equations in (13) such that w^* is the optimal solution of $\min_{w\in\mathbb{R}^b} \bar{p}(w)$. ■

We claim that $\{p_s\}_{s\in[k]}$ in (10) has the following property according to our construction.

Lemma 7 *Suppose U is a* **strict** *subset of $\{p_s\}_{s\in[k]}$ defined in (10), and q is an arbitrary linear combination of some functions p_s in U. The Hessian of q is a block diagonal matrix where each block is a square matrix of a size at most k.*

Proof There must exist some $s' \in [k]$ with $p_{s'} \notin U$. Since q is a linear combination of p_s's in U, according to the construction in (10), the Hessian of q is a tridiagonal matrix because it is a linear combination of a diagonal matrix (the Hessian of $\frac{\mu'}{2}\|w\|^2$) and all the matrices Σ_s defined in (9) except $\Sigma_{s'}$. We note that $\Sigma_{s'}$ is the only matrix among all Σ_s's that has non-zero entries in the positions $(s'-1+ik, s'+ik)$ and $(s'+ik, s'-1+ik)$ for $i = 0, 1, \ldots, u-1$ and both the row and column indices of these positions are equally spaced with a gap of k. Therefore, without $\Sigma_{s'}$ involved in the linear combination, the tridiagonal Hessian of q becomes block diagonal with each block of a size at most k. ■

To complete the proof of Theorem 2, the following lemma is critical. This lemma tells us that the property given by Lemma 7 forces the machines to perform a large number of rounds of communication in order to minimize $\bar{p}$ whenever $\{p_s\}_{s\in[k]}$ do not appear together in any machine.

Lemma 8 *For any $b = uk$ with integers $k, u \geq 1$, let $\{g_i\}_{i\in[N]}$ be the multi-set of functions on $\mathbb{R}^b$ that consists of v copies of $\{p_s\}_{s\in[k]}$ defined as (10) and $N - vk$ zero functions. More specifically,*

$$g_i(w) = \begin{cases} p_s(w) & \text{if } i = s, s+k, s+2k, \ldots, s+(v-1)k, \\ 0 & \text{if } i \geq vk+1. \end{cases} \tag{14}$$

Suppose an algorithm $\mathcal{A} \in \mathcal{F}_\alpha$ is applied to $\{g_i\}_{i\in[N]}$ to solve

$$\min_{w\in\mathbb{R}^b} \bar{g}(w) \equiv \frac{1}{N}\sum_{i=1}^N g_i \tag{15}$$

and the sets $\{S'_j\}_{j\in[m]}$ are generated as in Definition 1. Let $\mathcal{E}$ be the random event that none of the m machines has all functions in $\{p_s\}_{s\in[k]}$. Let $\hat{w} = \mathcal{A}(\{g_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H)$. Then, for any $\epsilon > 0$ and $k \geq 1$, there exist u and $b = uk$ such that $\mathbb{E}[g(\hat{w}) - g(w^*)|\mathcal{E}] \leq \epsilon$ only if $H \geq \left(\frac{\sqrt{\kappa'+k-1}-\sqrt{k}}{4k\sqrt{k}}\right)\log\left(\frac{\mu\|w^*\|^2}{4\epsilon}\right)$.

Proof Let $w^* \in \mathbb{R}^b$ defined as (12). Since $\bar{g} = \frac{1}{N}\sum_{i=1}^N g_i = \frac{v}{N}\sum_{s=1}^k p_s = \frac{\theta\bar{p}}{n}$, we have $w^* = \arg\min_{w\in\mathbb{R}^b}\bar{p}(w) = \arg\min_{w\in\mathbb{R}^b}\bar{g}(w)$ by Lemma 6.

Let $E_0 = \{\mathbf{0}\}$, where $\mathbf{0}$ is the all-zero vector in $\mathbb{R}^b$ and let E_t be the linear space in $\mathbb{R}^b$ spanned by the unit vectors $e_1, \ldots, e_t$ for $t = 1, \ldots, b$, where $e_s \in \mathbb{R}^b$ has one in its s-th component and zeros in other components.

Because $\mathcal{A}$ is applied to (15), by Definition 1, the working set W_j in machine j in each round can be updated for arbitrarily many times by including a vector w such that

$$\gamma w + \nu\nabla F_j(w) \in \text{span}\Big\{ w', \nabla F_j(w'), (\nabla^2 F_j(w') + D)w'', (\nabla^2 F_j(w') + D)^{-1}w'' \Big| \\ w', w'' \in W_j, D \text{ is diagonal}, \nabla^2 F_j(w') \text{ and } (\nabla^2 F_j(w') + D)^{-1} \text{ exist} \Big\} \tag{16}$$

for some constants $\gamma \geq 0$ and $\nu \geq 0$ with $\gamma\nu \neq 0$, where $F_j = \sum_{i\in U_j\subset S'_j} g_i$ with an arbitrary subset U_j of S'_j. Suppose event $\mathcal{E}$ happens such that S'_j does not contain all functions in $\{p_s\}_{s\in[k]}$ for any j. The set U_j in (16) will be a strict subset of $\{p_s\}_{s\in[k]}$ so that, according to Lemma 7, the Hessian matrix $\nabla^2 F_j$ in (16) is block-diagonal with a block size at most k. Because F_j is always a quadratic function, such a property of its Hessian matrix means, as long as W_j is contained by E_t, any vector w satisfying (16) will be in E_{t+k}. Therefore, we can show that, at the beginning of round ℓ in algorithm $\mathcal{A}$, if $\cup_j W_j \subset E_t$, then at the end of round ℓ (and at the beginning of round $\ell+1$), we have $\cup_j W_j \subset E_{t+k}$. Using this finding and the fact that $\cup_j W_j = \{\mathbf{0}\} = E_0$ initially, we conclude that, after H rounds in $\mathcal{A}$, we must have $\cup_j W_j \subset E_{Hk}$. Let $t = Hk$. Since $\hat{w} = \mathcal{A}(\{g_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H)$, we must have $\hat{w} \in E_t$ if $\mathcal{E}$ happens.

By (12), we can show that

$$\|w^*\|^2 = \sum_{j=1}^b (w_j^*)^2 = \sum_{j=1}^b h^{2j} = \frac{h^2(1-h^{2b})}{1-h^2}. \tag{17}$$

Because $\bar{p}$ defined in (11) is μ'-strong convex, we have

$$\frac{2}{\mu'}\mathbb{E}[\bar{p}(\hat{w}) - \bar{p}(w^*)|\mathcal{E}] \geq \mathbb{E}[\|\hat{w} - w^*\|^2|\mathcal{E}] \geq \sum_{j=t+1}^b \mathbb{E}[(w_j^*)^2|\mathcal{E}] \geq \frac{h^{2t+2}(1-h^{2b-2t})}{1-h^2} \geq \frac{\|w^*\|^2(h^{2t}-h^{2b})}{1-h^{2b}}$$

where the second inequality is because $\hat{w} \in E_t$ and the third inequality is due to (17). For any $k \geq 1$, when u is large enough so $b = uk$ is large enough, the inequality above implies $\mathbb{E}[\bar{p}(\hat{w}) - \bar{p}(w^*)|\mathcal{E}] \geq \frac{\mu'\|w^*\|^2 h^{2t}}{4} = \frac{\mu'\|w^*\|^2}{4}\left(\frac{\sqrt{\kappa'+k-1}-\sqrt{k}}{\sqrt{\kappa'+k-1}+\sqrt{k}}\right)^{2t}$. Based on this inequality, when $\mathbb{E}[\bar{g}(\hat{w}) - \bar{g}(w^*)|\mathcal{E}] \leq \epsilon$, or equivalently, when $\mathbb{E}[\bar{p}(\hat{w}) - \bar{p}(w^*)|\mathcal{E}] \leq \frac{n\epsilon}{\theta}$, we must have $\log\left(\frac{\mu'\theta\|w^*\|^2}{4n\epsilon}\right) \leq 2t\log\left(1 + \frac{2\sqrt{k}}{\sqrt{\kappa'+k-1}-\sqrt{k}}\right) \leq \frac{4t\sqrt{k}}{\sqrt{\kappa'+k-1}-\sqrt{k}}$, which further implies $H = \frac{t}{k} \geq \left(\frac{\sqrt{\kappa'+k-1}-\sqrt{k}}{4k\sqrt{k}}\right)\log\left(\frac{\mu\|w^*\|^2}{4\epsilon}\right)$. ■

In Lemma 8, the requirement of a large enough $b = uk$ is necessary since, if b is fixed and the number of rounds H satisfies $Hk \geq b$, the solution $\hat{w} = \mathcal{A}(\{g_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H)$ can be non-zero in all dimensions so that it may equal w^*. If this happens, $g(\hat{w}) - g(w^*) \leq \epsilon$ for any ϵ so that the lower bound for H in Lemma 8 does not hold. Because the communication lower bound we study does not depend on the dimension of the problem, we construct the functions with a dimension higher than Hk so that there are dimensions that will not be reached by the generated solutions so that the optimality gap has to reduce to zero slowly.

We now complete the proof of Theorem 2 by constructing N special functions $\{f_i\}_{i\in[N]}$ on $\mathbb{R}^d$ with $d = nb$ and $b = uk$ for a sufficiently large u based on $\{p_s\}_{s\in[k]}$ defined as (10), so that any algorithm $\mathcal{A} \in \mathcal{F}_\alpha$, when applied to $\{f_i\}_{i\in[N]}$, will need at least the targeted amount of rounds of communication.

For a large enough u, we partition the set of indices $[d] = \{1, \ldots, d\}$ into n disjoint subsets $D_1, D_2, \ldots, D_n$ with $|D_l| = b$ and $D_l = \{b(l-1)+1, , b(l-1)+2, \ldots, bl\}$. For any $l \in [n]$ and $s \in [k]$, let $q_{l,s}(x)$ be a function on $\mathbb{R}^d$ defined as

$$q_{l,s}(x) \equiv p_s(x_{D_l}), \tag{18}$$

where $\{p_s\}_{s\in[k]}$ is defined as (10). By its definition, the function $q_{l,s}(x)$ only depends on the b coordinates of x indexed by D_l. Therefore, we obtain nk different functions $\{q_{l,s}\}_{l\in[n],s\in[k]}$. Finally, we define $\{f_i\}_{i\in[N]}$ to be a set that consists of v copies of $\{q_{l,s}\}_{l\in[n],s\in[k]}$ and $N - vkn$ zero functions. In other words, the non-zero functions in the set $\{f_i\}_{i\in[N]}$ we constructed in this way constitutes a multi-set as follows

$$\left\{\begin{matrix} p_1(x_{D_1}), & p_2(x_{D_1}), & \cdots, & p_k(x_{D_1}), & \cdots, & p_1(x_{D_1}), & p_2(x_{D_1}), & \cdots, & p_k(x_{D_1}) \\ p_1(x_{D_2}), & p_2(x_{D_2}), & \cdots, & p_k(x_{D_2}), & \cdots, & p_1(x_{D_2}), & p_2(x_{D_2}), & \cdots, & p_k(x_{D_2}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ p_1(x_{D_n}), & p_2(x_{D_n}), & \cdots, & p_k(x_{D_n}), & \ldots, & p_1(x_{D_n}), & p_2(x_{D_n}), & \cdots, & p_k(x_{D_n}) \end{matrix}\right\}, \tag{19}$$

where the l-th row consists of v copies of $\{p_s\}_{s\in[k]}$ applied on x_{D_l}.

Because of Lemma 6 and the fact that

$$f(x) = \frac{1}{N}\sum_{i=1}^N f_i(x) = \frac{v}{N}\sum_{l=1}^n\sum_{s=1}^k q_{l,s}(x) = \frac{v}{N}\sum_{l=1}^n\sum_{s=1}^k p_s(x_{D_l}) = \frac{\theta}{n}\sum_{l=1}^n \bar{p}(x_{D_l}), \tag{20}$$

the optimal solution x^* for (1) with $\{f_i\}_{i\in[N]}$ constructed as (19) is $x^* = \mathbf{1}_n \otimes w^* = (w^*, w^*, \ldots, w^*)^T$ where $w^* \in \mathbb{R}^d$ is defined as (12) and repeated for n times in x^*.

Now, we want to verify that the functions $\{f_i\}_{i\in[N]}$ satisfy our assumptions. In fact, we have shown that p_s is L-smooth for each $s \in [k]$. Since f_i is either a zero function or equals $p_s(x_{D_l})$ for some $l \in [n]$ and $s \in [k]$, the function f_i is L-smooth for each $i \in [N]$ as well. Since $\bar{p}$ is μ'-strongly convex (on $\mathbb{R}^b$) and $\mu' = \frac{n\mu}{\theta}$, the function f defined in (1) must be μ-strongly convex (on $\mathbb{R}^d$) according to the relationship (20).

According to its construction, $\{f_i\}_{i\in[N]}$ are simultaneously decomposable with respect to the partition $D_1, \ldots, D_n$ with $D_l = \{b(l-1)+1, b(l-1)+2, \ldots, bl\}$ (see Definition 4). In fact, $f_i(x) = p_s(x_{D_l})$ for some $s \in [k]$ and $l \in [n]$ so that $f_i(x)$ can be represented as

$$f_i(x) = \sum_{l=1}^{n} g_i^l(x_{D_l}), \tag{21}$$

where $g_i^l \in \{p_s\}_{s\in[k]}$ for exactly one $l \in [n]$ and $g_i^l \equiv 0$ for other l's. Moreover, for any $l \in [n]$, the following equality holds between the two multi-sets

$$\{g_i^l\}_{i\in[N]} = \{g_i\}_{i\in[N]}, \tag{22}$$

where $\{g_i\}_{i\in[N]}$ are defined as (14).[4] By Proposition 5, $\mathcal{A}$ can be decomposed with respective to the same partition $D_1, D_2, \ldots, D_n$ into $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathcal{F}_\alpha$ and $\mathcal{A}_l$ is applied to

$$\min_{w\in\mathbb{R}^b} \left\{ \bar{g}^l \equiv \frac{1}{N}\sum_{i=1}^{N} g_i^l = \frac{1}{N}\sum_{i=1}^{N} g_i = \bar{g} \right\}, \tag{23}$$

which is exactly the problem (15) and has the form of (1). Recall Definition 1 for the procedure of generating $S_j' = S_j \cup R_j$, When $\{f_i | i \in S_j'\}$ is allocated to machine j for algorithm $\mathcal{A}$, what happens at the same time is that $\{g_i^l | i \in S_j'\}$ is allocated to machine j for algorithm $\mathcal{A}_l$ for $l = 1, 2, \ldots, n$.

We now focus on the solution generated by $\mathcal{A}_l$ for some l. For machine j in $\mathcal{A}_l$, let $Y_{1,j}$ be the number of functions in $\{p_s\}_{s\in[k]}$ (repetitions counted) that are contained in S_j and $Y_{2,j}$ be the number of functions in $\{p_s\}_{s\in[k]}$ (repetitions counted) that are contained in R_j.

Due to (14) and (22), exactly vk functions in $\{g_i^l\}_{i\in[N]}$ are from $\{p_s\}_{s\in[k]}$ and the other $N - vk$ functions are zero. Hence, $Y_{1,j}$ has a hypergeometric distribution where $\text{Prob}(Y_{1,j} = r)$ equals the probability of r successes in n draws, without replacement, from a population of size N that contains vk successes. Let $r = e\log m$. According to Chvatal (1979) and

4. The equation (22) does not means $g_i^l = g_i$ for any l and i. It only means, for any l and i, there exists an $i' \in [N]$ so that $g_i^l = g_{i'}$.

$\frac{vk}{N} \le \frac{1}{n}$, we have $\text{Prob}(Y_{1,j} \ge r) \le \left(\frac{vkn}{Nr}\right)^r \left(\frac{n-vkn/N}{n-r}\right)^{n-r} \le \left(\frac{1}{r}\right)^r \left(\frac{n-1}{n-r}\right)^{n-r}$, which implies

$$\begin{aligned}
\text{Prob}(Y_{1,j} \ge e\log m) &\le \left(\frac{1}{e\log m}\right)^{e\log m} \left(\frac{n-1}{n-e\log m}\right)^{n-e\log m} \\
&= \left(\frac{1}{e\log m}\right)^{e\log m} \left(1 + \frac{e\log m - 1}{n - e\log m}\right)^{n-e\log m} \\
&< \left(\frac{1}{e\log m}\right)^{e\log m} e^{e\log m - 1} = \frac{1}{e}\left(\frac{1}{\log m}\right)^{e\log m} \\
&\le \frac{1}{e}\left(\frac{1}{2\log(e+1)}\right)^{e\log m} \le \frac{1}{em^2},
\end{aligned} \tag{24}$$

where the second inequality is because $(1 + \frac{1}{x})^x < e$, the third inequality is because $m \ge (e + 4\max\{1,\alpha\})^2 \ge (e+1)^2$, and the last inequality is because $(2\log(e+1))^e > e^2$.

On the other hand, we have $Y_{2,j} = 0$ when $\alpha = 0$ and $Y_{2,j} = \sum_{i\in R_j} \mathbf{1}_{g_i^l \in \{p_s\}_{s\in[k]}}$ when $\alpha > 0$, where $\mathbf{1}_{g_i^l\in\{p_s\}_{s\in[k]}}$ for $i \in R_j$ are αn i.i.d. binary random variables which equal one with a probability of $\frac{vk}{N}$ and zero with a probability of $1 - \frac{vk}{N}$. Suppose $\alpha > 0$. The mean and the variance of $\mathbf{1}_{g_i^l\in\{p_s\}_{s\in[k]}}$ are $\frac{\theta}{n}$ and $\frac{\theta}{n}(1-\frac{\theta}{n})$, respectively. Hence, by Bernstein inequality (Uspensky, 1937), we have

$$\begin{aligned}
\text{Prob}(Y_{2,j} \ge 4\max\{1,\alpha\}\log m) &\le \text{Prob}(Y_{2,j} > \alpha\theta + 3\max\{1,\alpha\}\log m) \\
&\le \exp\left(\frac{-\frac{1}{2}(3\max\{1,\alpha\}\log m)^2}{\alpha\theta\left(1-\frac{\theta}{n}\right) + \max\{1,\alpha\}\log m}\right) \\
&\le \exp\left(\frac{-\frac{1}{2}(3\max\{1,\alpha\}\log m)^2}{2\max\{1,\alpha\}\log m}\right) \\
&\le \exp\left(-2\max\{1,\alpha\}\log m\right) \le \frac{1}{m^2},
\end{aligned} \tag{25}$$

where the first inequality is because $\max\{1,\alpha\}\log m > \alpha\theta$ and the third inequality is because $\max\{1,\alpha\}\log m \ge \alpha\theta\left(1-\frac{\theta}{n}\right)$. Suppose $\alpha = 0$. (25) holds trivially.

Combining (24) and (25) for $j = 1, 2, \ldots, m$ and using the union bound, we have

$$\text{Prob}(Y_{1,j} + Y_{2,j} < (e + 4\max\{1,\alpha\})\log m \text{ for } j = 1, 2, \ldots, m) \ge 1 - \frac{2}{m}.$$

Therefore, we have shown that, with a probability of at least $1 - \frac{2}{m}$, the multi-set $\{g_i^l | i \in S_j'\}$ contains fewer than $(e+4\max\{1,\alpha\})\log m \le k$ functions from $\{p_s\}_{s\in[k]}$ (repetition counted) for any $j \in [m]$. In other words, with a probability of at least $1 - \frac{2}{m}$, no machine in $\mathcal{A}_l$ has all of the functions in $\{p_s\}_{s\in[k]}$. If the event that "*none of the m machines has all functions in $\{p_s\}_{s\in[k]}$*" (same as the event $\mathcal{E}$ in Lemma 8) indeed happens in $\mathcal{A}_l$, we call $\mathcal{A}_l$ *bad*. Then, since $m \ge (e + 4\max\{1,\alpha\})^2$, we have actually proved

$$\text{Prob}(\mathcal{A}_l \text{ is bad}) \ge 1 - \frac{2}{m} \ge 1 - \frac{1}{e}. \tag{26}$$

By Proposition 5, after H rounds, the solutions $\hat{x} = \mathcal{A}(\{f_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H) \in \mathbb{R}^d$ and $\hat{w}^l = \mathcal{A}_l(\{g_i^l\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H) \in \mathbb{R}^{|D_l|}$ for $l = 1, 2, \ldots, n$ satisfy

$$\begin{aligned}\mathbb{E}[f(\hat{x}) - f(x^*)] &= \sum_{l=1}^{n} \mathbb{E}[\bar{g}^l(\hat{w}^l) - \bar{g}^l(x^*_{D_l})] = \sum_{l=1}^{n} \mathbb{E}[\bar{g}(\hat{w}^l) - \bar{g}(x^*_{D_l})] \\ &\geq \sum_{l=1}^{n} \mathbb{E}[\bar{g}(\hat{w}^l) - \bar{g}(x^*_{D_l})|\mathcal{A}_l \text{ is bad}]\text{Prob}(\mathcal{A}_l \text{ is bad}) \\ &\geq \sum_{l=1}^{n} \mathbb{E}[\bar{g}(\hat{w}^l) - \bar{g}(x^*_{D_l})|\mathcal{A}_l \text{ is bad}] \left(1 - \frac{1}{e}\right).\end{aligned}$$

Therefore, if $\mathbb{E}[f(\hat{x}) - f(x^*)] \leq \epsilon$, there must exist an $l \in [n]$ such that

$$\mathbb{E}[\bar{g}(\hat{w}^l) - \bar{g}(x^*_{D_l})|\mathcal{A}_l \text{ is bad}] \left(1 - \frac{1}{e}\right) \leq \frac{\epsilon}{n}. \tag{27}$$

When $\mathcal{A}_l$ is bad, none of the m machines in $\mathcal{A}_l$ has all functions in $\{p_s\}_{s\in[k]}$. By Lemma 8, there exist u and $b = uk$ such that (27) happens only if

$$\begin{aligned}H &\geq \left(\frac{\sqrt{\kappa' + k - 1} - \sqrt{k}}{4k\sqrt{k}}\right) \log\left(\left(1 - \frac{1}{e}\right) \frac{\mu n \|w^*\|^2}{4\epsilon}\right) \\ &= \left(\frac{\sqrt{\kappa' + k - 1} - \sqrt{k}}{4k\sqrt{k}}\right) \log\left(\left(1 - \frac{1}{e}\right) \frac{\mu \|x^*\|^2}{4\epsilon}\right), \\ &\geq \left(\frac{\sqrt{\kappa' - 1}}{4\sqrt{2}k\sqrt{k}}\right) \log\left(\left(1 - \frac{1}{e}\right) \frac{\mu \|x^*\|^2}{4\epsilon}\right),\end{aligned}$$

which is the desired lower bound after applying $\kappa' \geq \frac{2\kappa}{3n}$.

4.4 Proof for Theorem 3

We provide the proof for Theorem 3 in this subsection by following a similar argument to the proof of Theorem 2. Some notations in the previous sections will be used here but defined differently.

We first use the machinery developed in Section 2.1.1 by Nesterov (2013) to construct k functions on $\mathbb{R}^b$ where $b = uk$ for some integers $k, u \geq 1$. In particular, for $i, j = 1, \ldots, b$, let $\delta_{i,j}$ be an $b \times b$ matrix with its (i, j) entry being one and others being zeros. Let $M_0, M_1, \ldots, M_{b-1}$ be $b \times b$ matrices defined as

$$M_i = \begin{cases} \delta_{1,1} & \text{for } i = 0 \\ \delta_{i,i} - \delta_{i,i+1} - \delta_{i+1,i} + \delta_{i+1,i+1} & \text{for } 1 \leq i \leq b-1. \end{cases} \tag{28}$$

Here, only the M_{b-1} in (28) is different from the M_{b-1} in (8). For $s \in [k]$, let $\Sigma_s = \sum_{i=0}^{u-1} M_{ik+s-1}$ which has the same form as (9) but with a different M_{b-1}. For example,

when $u = 2$ and $k = 3$ (so $b = 6$), the matrices Σ_s's are given as follows.

$$\Sigma_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \Sigma_2 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\Sigma_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

We define k functions $p_1, \ldots, p_k : \mathbb{R}^b \to \mathbb{R}$ as follows

$$p_s(w) = \begin{cases} \frac{L}{4}\left[\frac{w^T \Sigma_1 w}{2} - e_1^\top w\right] & \text{for } s = 1 \\ \frac{L}{4}\left[\frac{w^T \Sigma_s w}{2}\right] & \text{for } s = 2, \ldots, k, \end{cases} \tag{29}$$

where $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^b$, and denote their average by

$$\bar{p} \equiv \frac{1}{k}\sum_{s=1}^{k} p_s = \frac{L}{4k}\left[\frac{1}{2} w^T \left(\sum_{s=1}^{k} \Sigma_s\right) w - e_1^T w\right]. \tag{30}$$

It is also easy to show that $\lambda_{\max}(\Sigma_s) \leq 4$ so that ∇p_s has a Lipschitz continuity constant of L so that p_s is L-smooth.

Observing that

$$\sum_{s=1}^{k} \Sigma_s = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix},$$

we note that the function $\bar{p}(w)$ defined in (30) is the one studied in Section 2.1.2 by Nesterov (2013) for establishing the lower bound for the iteration complexity for single-machine first-order methods. Following Section 2.1.2 by Nesterov (2013), the optimal solution and the optimal value of $\min_{w\in\mathbb{R}^b} \bar{p}(w)$ can be characterized as follows.

Lemma 9 (Nesterov (2013)) *The vector $w^* = (w_1^*, w_2^*, \ldots, w_b^*)^T \in \mathbb{R}^b$ with*

$$w_j^* = 1 - \frac{j}{b+1}, \text{ for } j = 1, 2, \ldots, b \tag{31}$$

is the optimal solution of $\min_{w\in\mathbb{R}^b} \bar{p}(w)$. Moreover, $\|w^*\|^2 \leq \frac{b+1}{3}$ and $\bar{p}(w^*) = \frac{L}{8k}\left(-1 + \frac{1}{b+1}\right)$.

The following lemma can be extracted from Section 2.1.2 in Nesterov (2013) which characterizes the optimal value of $\bar{p}$ when restricted in the subspace of first t coordinates.

Lemma 10 (Nesterov (2013)) *Let $\bar{p}$ defined as (30). We have*

$$\min_{(w_1,\ldots,w_t)\in\mathbb{R}^t} \bar{p}(w_1,\ldots,w_t,0,\ldots,0) = \frac{L}{8k}\left(-1+\frac{1}{t+1}\right).$$

We can show that $\{p_s\}_{s\in[k]}$ defined in (29) has the following properties.

Lemma 11 *Suppose U is a* **strict** *subset of $\{p_s\}_{s\in[k]}$ defined in (10), and q is an arbitrary linear combination of some functions p_s in U. The Hessian of q is a block diagonal matrix where each block is a square matrix of a size at most k.*

Proof Same as the proof of Lemma 7. ■

Lemma 12 *For any $b = uk$ with integers $k, u \geq 1$, let $\{g_i\}_{i\in[N]}$ be the multi-set of functions on $\mathbb{R}^b$ that consists of v copies of $\{p_s\}_{s\in[k]}$ defined as (29) and $N - vk$ zero functions. More specifically, $\{g_i\}_{i\in[N]}$ is defined as (14) but with $\{p_s\}_{s\in[k]}$ defined as (29). Suppose an algorithm $\mathcal{A} \in \mathcal{F}_\alpha$ is applied to (15) and the sets $\{S'_j\}_{j\in[m]}$ are generated as in Definition 1. Let $\mathcal{E}$ be the random event that none of the m machines has all functions in $\{p_s\}_{s\in[k]}$. Let $\hat{w} = \mathcal{A}(\{g_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H)$. Then, for any $\epsilon > 0$ and $k \geq 1$, there exist u and $b = uk$ such that $\mathbb{E}[g(\hat{w}) - g(w^*)|\mathcal{E}] \leq \epsilon$ only if $H \geq \sqrt{\frac{3L\|w^*\|^2}{128k^3n\epsilon}}$.*

Proof Let $u = 2H+1$, $b = 2Hk+k$ and $w^* \in \mathbb{R}^b$ defined as (31). Since $\bar{g} = \frac{1}{N}\sum_{i=1}^N g_i = \frac{v}{N}\sum_{s=1}^k p_s = \frac{\theta\bar{p}}{n}$, we have $w^* = \arg\min_{w\in\mathbb{R}^b} \bar{p}(w) = \arg\min_{w\in\mathbb{R}^b} \bar{g}(w)$ by Lemma 9.

Let E_t defined as in the proof of Lemma 8. Following the same argument as in the proof of Lemma 8, we can show that, if $\mathcal{E}$ happens, after H rounds in algorithm $\mathcal{A}$, we must have $\cup_j W_j \subset E_{Hk}$, where W_j is the working set of machine j by Definition 1. Let $t = Hk$. Since $\hat{w} = \mathcal{A}(\{g_i\}_{i\in[N]}, \{S'_j\}_{j\in[m]}, H)$, we must have $\hat{w} \in E_t$ if $\mathcal{E}$ happens. According to Lemma 10, we have

$$\bar{p}(\hat{w}) \geq \min_{(w_1,\ldots,w_t)\in\mathbb{R}^t} \bar{p}(w_1,\ldots,w_t,0,\ldots,0) = \frac{L}{8k}\left(-1+\frac{1}{Hk+1}\right),$$

which, together with Lemma 9, implies

$$\mathbb{E}\left[\frac{\bar{p}(\hat{w})-\bar{p}(w^*)}{\|w^*\|}\Big|\mathcal{E}\right] \geq \frac{\frac{L}{8k}\left(-1+\frac{1}{Hk+1}+1-\frac{1}{b+1}\right)}{\frac{1}{3}(b+1)} \geq \frac{3L}{32k^3(H+1)^2} \geq \frac{3L}{128k^3H^2}.$$

Based on this inequality, when $\mathbb{E}[\bar{g}(\hat{w}) - \bar{g}(w^*)|\mathcal{E}] \leq \epsilon$, or equivalently, when $\mathbb{E}[\bar{p}(\hat{w}) - \bar{p}(w^*)|\mathcal{E}] \leq \frac{n\epsilon}{\theta} \leq \frac{3n\epsilon}{2}$, we must have $H \geq \sqrt{\frac{L\|w^*\|^2}{64k^3n\epsilon}}$. ■

The rest of the proof of Theorem 3 is almost identical to that of Theorem 2 so we will only provide a brief sketch. In particular, we will construct N special non-strongly convex functions $\{f_i\}_{i\in[N]}$ on $\mathbb{R}^d$ with $d = nb$ for a sufficiently large b based on $\{p_s\}_{s\in[k]}$ defined as (29), and show that any algorithm $\mathcal{A} \in \mathcal{F}_\alpha$ will need at least the targeted amount of rounds of communication when it is applied to $\{f_i\}_{i\in[N]}$.

For a large enough u and $b = uk$, we partition the set of indices $[d] = \{1,\ldots,d\}$ into n disjoint subsets $D_1, D_2, \ldots, D_n$ with $|D_l| = b$ and $D_l = \{b(l-1)+1, b(l-1)+2, \ldots, bl\}$.

For any $l \in [n]$ and $s \in [k]$, let $q_{l,s}(x)$ be a function on $\mathbb{R}^d$ defined as (18) where $\{p_s\}_{s\in[k]}$ is defined as (29). We define $\{f_i\}_{i\in[N]}$ to be a set that consists of v copies of $\{q_{l,s}\}_{l\in[n],s\in[k]}$ and $N - vkn$ zeros functions. In other words, the set of non-zero functions in $\{f_i\}_{i\in[N]}$ is defined as (19) with $\{p_s\}_{s\in[k]}$ given as (29). It is easy to verify that the each function in $\{f_i\}_{i\in[N]}$ is L-smooth.

Because of Lemma 9 and (20), the optimal solution x^* for (1) with $\{f_i\}_{i\in[N]}$ constructed as above is $x^* = \mathbf{1}_n \otimes w^* = (w^*, w^*, \ldots, w^*)^T$ where $w^* \in \mathbb{R}^d$ is defined as (31).

By its construction, $\{f_i\}_{i\in[N]}$ are simultaneously decomposable with respect to the partition $D_1, \ldots, D_n$ with $D_l = \{b(l-1)+1, b(l-1)+2, \ldots, bl\}$. In fact, $f_i(x) = p_s(x_{D_l})$ for some $s \in [k]$ and $l \in [n]$ with $\{p_s\}_{s\in[k]}$ defined as (29) so that $f_i(x)$ can be represented as (21) where $g_i^l \in \{p_s\}_{s\in[k]}$ for exactly one $l \in [n]$ and $g_i^l \equiv 0$ for other l's. Moreover, for any $l \in [n]$, the equality (22) holds with $\{g_i\}_{i\in[N]}$ defined as (14) and $\{p_s\}_{s\in[k]}$ defined as (29).

By Proposition 5, $\mathcal{A}$ can be decomposed with respective to the partition $D_1, D_2, \ldots, D_n$ into $\mathcal{A}_1, \ldots, \mathcal{A}_n \in \mathcal{F}_\alpha$ and $\mathcal{A}_l$ is applied to (23) (the same as (15)). Let $S'_j = S_j \cup R_j$ generated as in Definition 1. When $\{f_i | i \in S'_j\}$ is allocated to machine j for algorithm $\mathcal{A}$, the set $\{g_i^l | i \in S'_j\}$ is allocated to machine j for algorithm $\mathcal{A}_l$ for $l = 1, 2, \ldots, n$. We still call $\mathcal{A}_l$ *bad* if none of the m machines in algorithm $\mathcal{A}_l$ has all functions in $\{p_s\}_{s\in[k]}$. Then, following exactly the same argument as in Theorem 2, we can obtain (26) which indicates that $\mathcal{A}_l$ can be *bad* with a constant probability for each l. As a result, if $\mathbb{E}[f(\hat{x}) - f(x^*)] \leq \epsilon$, there must exist an $l \in [n]$ such that (27) holds.

When $\mathcal{A}_l$ is bad, after the data distribution stage, none of the m machines in $\mathcal{A}_l$ has all functions in $\{p_s\}_{s\in[k]}$. According to Lemma 12, there exists b such that (27) happens only if $H \geq \sqrt{\left(1-\frac{1}{e}\right)\frac{L\|x^*\|^2}{64k^3n\epsilon}}$, which is the desired lower bound.

5. Distributed SVRG

In this section, we consider a *distributed stochastic variance reduced gradient* (DSVRG) method that is based on a parallelization of the stochastic variance reduced gradient (SVRG) method (Johnson and Zhang, 2013; Mahdavi et al., 2013; Xiao and Zhang, 2014; Konečnỳ and Richtárik, 2017). SVRG works in multiple stages and, in each stage, one batch gradient is computed using all N data points and $O(\kappa)$ iterative updates are performed with only one data point processed in each iterative update.

Our DSVRG algorithm randomly partitions the N data points into m subsets $\{S_j\}_{j\in[m]}$ with $|S_j| = n$ and S_j is allocated to machine j in order to parallelize the computation of the batch gradient in SVRG. Then, we let the m machines conduct the iterative update of SVRG in serial in a "round-robin" scheme, namely, let all machine stay idle except one machine that performs a certain steps of iterative updates of SVRG using its local data and pass the solution to the next machine. However, the only caveat in this idea is that the iterative update of SVRG requires an unbiased estimator of $\nabla f(x)$ which can be constructed by sampling over the whole data set. However, the unbiasedness will be lost if each machine can only sample over S_j. To address this issue, we let machine j sample with replacement from the whole data set to construct a second set R_j with $|R_j| = \alpha n$ for $\alpha > 0$ during the data allocation. In each iterative update, if each machine samples over R_j, the unbiased

Algorithm 1 Distributed SVRG (DSVRG)

Input: An initial solution $\tilde{x}^0 \in \mathbb{R}^d$, data $\{f_i\}_{i\in[N]}$, the number of machines m, a step length $\eta < \frac{1}{4L}$, the number of stages K, the number of iterations T in each stage, and a parameter $\alpha \geq \frac{TK}{N}$.

Output: $\tilde{x}^K$

1: Generate (a) a random partition $\{S_j\}_{j\in[m]}$ of $[N]$, (b) m multi-sets $\{R_j\}_{j\in[m]}$ with $|R_j| = \alpha n$ uniformly sampled with replacement from $[N]$, and (c) data $\{f_i \mid i \in S_j \cup R_j\}$ stored on machine j.
2: $k \leftarrow 1$
3: **for** $\ell = 0, 1, 2, \ldots, K-1$ **do**
4: Center sends $\tilde{x}^\ell$ to each machine
5: **for** machine $j = 1, 2, \ldots, m$ **in parallel do**
6: Compute $h_j^\ell = \sum_{i\in S_j} \nabla f_i(\tilde{x}^\ell)$ and send it to center
7: **end for**
8: Center computes $h^\ell = \frac{1}{N}\sum_{j=1}^m h_j^\ell$ and send it to machine k
9: $(\tilde{x}^{\ell+1}, \{R_j\}_{j\in[m]}, k) \leftarrow \text{SS-SVRG}(\tilde{x}^\ell, \{f_i\}_{i\in[N]}, h^\ell, \{R_j\}_{j\in[m]}, k, \eta, T)$
10: **end for**

estimator will be still available so that the convergence property can be inherited from the single-machine SVRG.

5.1 DSVRG Algorithm and Its Round Complexity

Let $\alpha > 0$. We first randomly partition $[N]$ into subsets $\{S_j\}_{j\in[m]}$ with $|S_j| = \frac{N}{m} = n$ and create m multi-sets $\{R_j\}_{j\in[m]}$ with $|R_j| = \alpha n$ by uniformly sampling with replacement from $[N]$. With the data $\{f_i | i \in S_j \cup R_j\}$ distributed on machine j for each $j \in [m]$, we describe formally each stage of this DSVRG in Algorithm 1 and the iterative update in Algorithm 2. From the definitions of S_j and R_j and the communication and computation performed, we can easily see that DSVRG belongs to the family $\mathcal{F}_\alpha$ ($\alpha > 0$) in Definition 1.

We start DSVRG in machine k with $k = 1$ initially at an initial solution $\tilde{x}^0 \in \mathbb{R}^d$. At the beginning of stage ℓ of DSVRG, all m machines participate in computing a batch gradient h^ℓ in parallel using the data indexed by $S_1, \ldots, S_m$. Within stage ℓ, in each iteration, machine k samples one data f_i from its local data indexed by R_k to construct a stochastic gradient $\nabla f_i(x_t) - \nabla f_i(\tilde{x}^\ell) + h^\ell$ and performs the *iterative update* (Line 3 of Algorithm 2). After this iteration, the index i is removed from R_k. Due to the construction of R_k, we can easily prove that this stochastic gradient is an unbiased estimation of the batch gradient $\nabla f(x_t)$.

Lemma 13 *Suppose $\tilde{x}$, x_t, $\{f_i\}_{i\in[N]}$, h, $\{R_j\}_{j\in[m]}$ and k are defined as in Algorithm 2 (SS-SVRG). In addition, suppose each element of R_j was i.i.d. sampled with replacement from $[N]$ and $h = \frac{1}{N}\sum_{i=1}^N \nabla f(\tilde{x})$. In the t-th iteration of Algorithm 2, we have*

$$\mathbb{E}[\nabla f_i(x_t) - \nabla f_i(\tilde{x}) + h | x_t, \tilde{x}] = \nabla f(x_t),$$

where the expectation is taken over i and $\{R_j\}_{j\in[m]}$.

Proof Since R_k is a multi-set that consists of indices uniformly sampled with replacement from $[N]$, the index i has a uniform distribution over $[N]$. Moreover, because each index

Algorithm 2 Single-Stage SVRG: SS-SVRG$(\tilde{x}, \{f_i\}_{i\in[N]}, h, \{R_j\}_{j\in[m]}, k, \eta, T)$

Input: A solution $\tilde{x} \in \mathbb{R}^d$, data $\{f_i\}_{i\in[N]}$, a batch gradient h, m multi-sets $\{R_j\}_{j\in[m]}$, the index of the active machine k, a step length $\eta < \frac{1}{4L}$, and the number of iterations T.

Output: The average solution $\bar{x}_T$, the updated multi-sets $\{R_j\}_{j\in[m]}$, and the updated index of active machine k.

1: $x_0 = \tilde{x}$ and $\bar{x}_0 = \mathbf{0}$
2: **for** $t = 0, 1, 2, \ldots, T-1$ **do**
3: Machine k uniformly samples an instance i from R_k and computes

$$x_{t+1} = x_t - \eta\left(\nabla f_i(x_t) - \nabla f_i(\tilde{x}) + h\right), \quad \bar{x}_{t+1} = \frac{x_{t+1} + t\bar{x}_t}{t+1}, \quad R_k \leftarrow R_k \backslash \{i\}$$

4: **if** $R_k = \emptyset$ **then**
5: x_{t+1} and $\bar{x}_{t+1}$ are sent to machine $k+1$
6: $k \leftarrow k+1$
7: **end if**
8: **end for**

sampled from R_k in the previous iteration has been removed from R_k, the set R_k at iteration t does not contain the indices in the previous iterations that determine x_t so that R_k is independent of x_t. With these facts, we can show that

$$\begin{aligned}
&\mathbb{E}[\nabla f_i(x_t) - \nabla f_i(\tilde{x}) + h | x_t, \tilde{x}] = \mathbb{E}\left[\mathbb{E}[\nabla f_i(x_t) - \nabla f_i(\tilde{x}) + h | R_k, x_t, \tilde{x}] \Big| x_t, \tilde{x}\right] \\
= \;& \mathbb{E}\left[\frac{1}{|R_k|}\sum_{i\in R_k}(\nabla f_i(x_t) - \nabla f_i(\tilde{x}) + h) \Big| x_t, \tilde{x}\right] = \mathbb{E}_{i\sim N}[\nabla f_i(x_t) - \nabla f_i(\tilde{x}) + h | x_t, \tilde{x}] = \nabla f(x_t),
\end{aligned}$$

where the third equality is because R_k is independent of x_t and the marginal distribution of each i in R_k is uniform distribution on $[N]$. ■

The m machines do the iterative updates in the order from machine 1 to machine m. Once the current active machine, say, machine k, has removed all of its samples in R_k (so that $R_k = \emptyset$), then it must pass the current solution and the running average of all solutions generated in the current stage to machine $k+1$. At any time during the algorithm, there is only one machine updating the solution x_t and the other $m-1$ machines only contribute in computing the batch gradient h^ℓ. We want to emphasize that it is important that machines should never use any samples in R_j's more than once since, otherwise, the stochastic gradient $\nabla f_i(x_t) - \nabla f_i(\tilde{x}^\ell) + h^\ell$ will lose its unbiasedness. Because one element in R_k is removed in each iterative update (Line 3) of Algorithm 2, this update cannot be performed any longer once each R_k becomes empty. Hence, a requirement for using Algorithm 1 is $TK \leq \sum_{j\in[m]} |R_j| = \alpha n m = \alpha N$.

The convergence of Algorithm 1 is established by the following theorem.

Theorem 14 *Suppose $0 < \eta < \frac{1}{4L}$ and $\mu > 0$. Algorithm 1 guarantees*

$$\mathbb{E}\left[f(\tilde{x}^K) - f(x^*)\right] \leq \left(\frac{1}{\mu\eta(1-4L\eta)T} + \frac{4L\eta(T+1)}{(1-4L\eta)T}\right)^K \left[f(\tilde{x}^0) - f(x^*)\right]. \tag{32}$$

When $\eta = \frac{1}{16L}$, $T = 96\kappa$ *and* $\alpha \geq \frac{96\kappa}{\log(9/8)N} \log\left(\frac{f(\tilde{x}^0)-f(x^*)}{\epsilon}\right)$, *Algorithm 1 finds an* ϵ*-optimal solution with* $K \leq \frac{1}{\log(9/8)} \log\left(\frac{f(\tilde{x}^0)-f(x^*)}{\epsilon}\right)$ *stages and at most* $\frac{\frac{96\kappa}{\alpha n}+1}{\log(9/8)} \log\left(\frac{f(\tilde{x}^0)-f(x^*)}{\epsilon}\right)$ *rounds of communication.*

In particular, if $\frac{96\kappa}{\log(9/8)} \log\left(\frac{f(\tilde{x}^0)-f(x^*)}{\epsilon}\right) \leq O(N)$ *in addition to the previous conditions, Algorithm 1 with* $\alpha = \Theta(1)$ *finds an* ϵ*-optimal solution with at most* $O\left((\frac{\kappa}{n}+1)\log\left(\frac{1}{\epsilon}\right)\right)$ *rounds of communication.*

Proof In the iterative update given in Line 3 of Algorithm 2, a stochastic gradient $\nabla f_i(x_t) - \nabla f_i(\tilde{x}) + h$ is constructed with h being the batch gradient $\nabla f(\tilde{x})$ and i sampled from R_k in the active machine k. By Lemma 13, this stochastic gradient is unbiased estimator of $\nabla f(x_t)$. Therefore, the path of solutions $\tilde{x}^0, \tilde{x}^1, \tilde{x}^2, \ldots$ generated by Algorithm 1 has the same distribution as the ones generated by single-machine SVRG so that the convergence result for the single-machine SVRG can be directly applied to Algorithm 1. The inequality (32) has been shown in Theorem 1 in Xiao and Zhang (2014) for single-machine SVRG, which now also holds for Algorithm 1.

When $\eta = \frac{1}{16L}$ and $T = 96\kappa$, it is easy to show that

$$\frac{1}{\mu\eta(1-4L\eta)T} + \frac{4L\eta(T+1)}{(1-4L\eta)T} \leq \frac{1}{\mu\eta(1-4L\eta)T} + \frac{8L\eta}{(1-4L\eta)} = \frac{2}{9} + \frac{2}{3} = \frac{8}{9}$$

so that Algorithm 1 needs $K \leq \frac{1}{\log(9/8)} \log\left(\frac{f(\tilde{x}^0)-f(x^*)}{\epsilon}\right)$ stages to find an ϵ-optimal solution. Since Algorithm 1 performs TK iterative updates with one element of R_k removed after each update, it requires $TK \leq \sum_{j\in[m]} |R_j| = \alpha N$, which is satisfied by the above upper bound of K, the choice of T and the assumption on α.

Since Algorithm 1 needs one round of communication to compute a batch gradient at each stage (one call of SS-SVRG) and one round after every αn iterative updates (when $R_k = \emptyset$), it needs $K + \frac{TK}{\alpha n} \leq \left(\frac{96\kappa}{\alpha n}+1\right)\frac{1}{\log(9/8)} \log\left(\frac{f(\tilde{x}^0)-f(x^*)}{\epsilon}\right)$ rounds of communication in total to find an ϵ-optimal solution. The second part of the conclusion is a straightforward outcome from the first part. ■

By Theorem 14, to find an ϵ-optimal solution, DSVRG requires $\alpha \geq \Omega(\frac{\kappa}{N}\log(1/\epsilon))$ so that each machine must have a minimum memory space of $|S_j| + |R_j| = n + \alpha n = \Omega(n + \frac{\kappa}{m}\log(1/\epsilon))$. For ERM problem where $\kappa = \Theta(\sqrt{N})$ and ϵ is reasonably small, for example, $\epsilon = \frac{1}{n^s}$ for a constant s, the requirement becomes $\alpha \geq \Omega(\frac{\log(n)}{\sqrt{nm}})$, which can be easily satisfied with a constant α ($\alpha = \Theta(1)$). In this case, DSVRG only asks for $\Theta(n + \sqrt{\frac{n}{m}}\log(n)) = \Theta(n)$ memory space in each machine to store $S_j \cup R_j$. Note that a memory space of n is necessary in all distributed algorithms to store S_j so the space required by DSVRG is in the same order of magnitude as other distributed algorithms in this scenario. Although we use $\kappa = \Theta(\sqrt{N})$ as an example, DSVRG still applies with $\alpha = \Theta(1)$ and $\Theta(n)$ memory space when $\epsilon = \frac{1}{n^s}$ and $\kappa = \Theta(N^b)$ with $b \in (0,1)$.

Compared to other methods which use only S_j, DSVRG and DASVRG (introduced later) require additional $\Theta(\alpha N)$ amount of communication during the data allocation stage for transmitting the second data set R_j from the center to each machine. Although the

communication during the data allocation stage is not included in the total amount of communication in the performance metrics defined in Section 1.1, it will certainly affect the efficiency of an algorithm in practice. Hence, in Algorithm 4 in Appendix, we design an efficient data allocation scheme which reuses the randomness of first randomly partitioned data set S_j to construct R_j. We show in Lemma 18 that this method helps to increase the overlap between R_j and S_j so that it requires a smaller amount of communication to allocate $\{f_i | i \in S_j \cup R_j\}$ to machine j than the direct implementation. For example, when $\kappa = \Theta(\sqrt{N})$ in a typical ERM problem, with Algorithm 4, DSVRG only needs $O((\log(1/\epsilon))^2)$ more amount of communication during the data allocation stage than other methods.

5.2 Regime Where DSVRG is Optimal

In this subsection, we consider a scenario where $\kappa = O(n^{1-2\delta})$ with a constant $0 < \delta < \frac{1}{2}$ and $\epsilon = \frac{1}{n^s}$ with a constant $s > 0$. We show that with modified values of η, T and K, DSVRG can find an ϵ-optimal solution for (1) with the optimal parallel runtime, the optimal amount of communication, and the optimal number of rounds of communication *simultaneously*.

Proposition 15 *Suppose* $0 < \kappa \leq \frac{n^{1-2\delta}}{32}$ *with a constant* $0 < \delta < \frac{1}{2}$. *When* $\eta = \frac{1}{16n^\delta L}$, $T = n$ *and* $\alpha \geq \frac{1}{\log(n^\delta/2)} \log\left(\frac{f(\tilde{x}^0)-f(x^*)}{\epsilon}\right)$, *Algorithm 1 finds an* ϵ*-optimal solution for (1) with* $K = O(\frac{\log(1/\epsilon)}{\log n})$ *stages,* $O(\left(\frac{1+\alpha}{\alpha}\right)\frac{\log(1/\epsilon)}{\log n})$ *rounds of communications,* $O(\frac{n\log(1/\epsilon)}{\log n})$ *parallel runtime, and* $O((m+\frac{1}{\alpha})\frac{\log(1/\epsilon)}{\log n})$ *amount of communication.*

If $\epsilon = \frac{1}{n^s}$ *with a positive constant* s *in addition to the conditions above, Algorithm 1 finds an* ϵ*-optimal solution for (1) with* $\alpha = \Theta(1)$, $O(1)$ *rounds of communications,* $O(n)$ *parallel runtime, and* $O(m)$ *amount of communication.*

Proof Since $\eta = \frac{1}{16n^\delta L} < \frac{1}{4L}$, Algorithm 1 guarantees (32) according to Theorem 14. With $T = n$ and $\eta = \frac{1}{16n^\delta L}$, we have

$$\begin{aligned}\frac{1}{\mu\eta(1-4L\eta)T} + \frac{4L\eta(T+1)}{(1-4L\eta)T} &\leq \frac{1}{\mu\eta(1-4L\eta)T} + \frac{8L\eta}{(1-4L\eta)} = \frac{16n^\delta\kappa}{(1-1/(4n^\delta))n} + \frac{1}{2n^\delta(1-\frac{1}{4n^\delta})} \\ &\leq \frac{1}{2n^\delta(1-1/(4n^\delta))} + \frac{1}{2n^\delta(1-1/(4n^\delta))} \leq \frac{2}{n^\delta}.\end{aligned}$$

Hence, $\mathbb{E}\left[f(\tilde{x}^K) - f(x^*)\right] \leq \epsilon$ is implied by (32) with $K \leq \frac{1}{\log(n^\delta/2)} \log\left(\frac{f(\tilde{x}^0)-f(x^*)}{\epsilon}\right) = O(\frac{\log(1/\epsilon)}{\log n})$. The requirement $TK \leq \alpha N$ is satisfied by the above upper bound of K and the assumption on α.

Since Algorithm 1 needs one round of communication to compute a batch gradient at each stage and one round after every αn iterative updates, it needs $K + \frac{TK}{\alpha n} = O(\frac{(1+\alpha)\log(1/\epsilon)}{\alpha \log n})$ rounds of communication in total to find an ϵ-optimal solution. Since one gradient is computed in each iterative update and n gradients are computed in parallel in each stage, the parallel runtime of Algorithm 1 is $(n+T)K = O(\frac{n\log(1/\epsilon)}{\log(n)})$. Finally, each batch gradient computation requires machines sending m vectors of size $O(d)$ to the center and every αn iterative updates require sending two vectors of size $O(d)$ between machines. Hence, Algo-

rithm 1 requires $(mK + \frac{2TK}{\alpha n}) = O((m + \frac{1}{\alpha})\frac{\log(1/\epsilon)}{\log n})$ amount of communication to find an ϵ-optimal solution.[5]

We can easily obtain the second part of the conclusion after substituting ϵ in the three performance metrics above by $\frac{1}{n^s}$ with a positive constant s and using the fact that $\alpha \geq \frac{1}{\log(n^s/2)} \log\left(\frac{f(\tilde{x}^0)-f(x^*)}{\epsilon}\right) = \Theta(1)$ ■

We emphasize again that the amount of communication in Proposition 15 is measured by the number of vectors of size $O(d)$ transmitted through the network after the data distribution phase so it does not include the communication for sending S_j and R_j to machines. The justification for the scenario where $\kappa = O(n^{1-2\delta})$ and $\epsilon = \frac{1}{n^s}$ and why these performance guarantees are optimal have been discussed in Section 1.2. To apply DSVRG under this scenario, the required memory space in each machine is $O(n+\alpha n) = O(n)$ (since we are able to choose $\alpha = \Theta(1)$ in this case), which is in the same order of magnitude as the memory requirement of all distributed methods.

6. Accelerated Distributed SVRG

In this section, we use the generic acceleration techniques in Frostig et al. (2015) and Lin et al. (2015) to further improve the theoretical performance of DSVRG and obtain a *distributed accelerated stochastic variance reduced gradient* (DASVRG) method. DASVRG belongs to the family $\mathcal{F}_\alpha$ ($\alpha > 0$) in Definition 1. We will show that the lower bound given in Theorem 2 for the round complexity of $\mathcal{F}_\alpha$ is matched by the rounds needed by DASVRG with $\alpha = \Theta(1)$ up to logarithmic factors for a flexible and realistic regime of parameters.

6.1 DASVRG Algorithm and Its Round Complexity

We define the *proximal function* for $f(x)$ as

$$f_\sigma(x; y) \equiv f(x) + \frac{\sigma}{2}\|x - y\|^2 = \frac{1}{N}\sum_{i=1}^{N} f_{\sigma,i}(x; y), \tag{33}$$

where $f_{\sigma,i}(x; y) = f_i(x) + \frac{\sigma}{2}\|x - y\|^2$, σ is a positive constant to be determined later and $y \in \mathbb{R}^d$ is a *proximal point*. The condition number of this proximal function is $\kappa(f_\sigma) \equiv \frac{L+\sigma}{\mu+\sigma}$ which is always smaller than κ.

Given an algorithm, denoted by $\mathcal{A}$, that can be applied to (1), the acceleration scheme developed in Frostig et al. (2015) and Lin et al. (2015) is an iterative method that involves inner and outer loops and uses $\mathcal{A}$ as a sub-routine called in each outer loop. In particular, in the p-th outer iteration of this acceleration scheme, the algorithm $\mathcal{A}$ is applied to find a solution for the p-th proximal problem defined on a proximal point $y_p \in \mathbb{R}^d$, namely,

$$\min_{x\in\mathbb{R}^d} f_\sigma(x; y_p) \text{ for } p = 0, 1, 2, \ldots, P-1. \tag{34}$$

Since $\kappa(f_\sigma) < \kappa$, solving (34) is generally easier than solving (1). Moreover, the algorithm $\mathcal{A}$ does not need to solve (34) to optimality but only needs to generate an approximate

5. As we defined in Section 1.1, the communication used to distribute S_j and R_j to machine j is not included in the amount of communication given here.

solution $\hat{x}_{p+1}$ with an accuracy ϵ_p in the sense that

$$\mathbb{E}\left[f_\sigma(\hat{x}_{p+1}; y_p) - \min_{x\in\mathbb{R}^d} f_\sigma(x; y_p)\right] \leq \epsilon_p. \tag{35}$$

Then, the acceleration scheme uses the convex combination of $\hat{x}_{p+1}$ and another auxiliary solution z_{p+1} to construct a new proximal point y_{p+1}. After that, the $(p+1)$-th proximal problem is constructed based on y_{p+1} which will be solved in the next outer iteration while the auxiliary solution z_{p+1} will be updated by a momentum step. With an appropriately chosen value for σ, it is shown by Frostig et al. (2015) and Lin et al. (2015) that, for many existing $\mathcal{A}$ including SAG (Schmidt et al., 2017; Roux et al., 2012), SAGA (Defazio et al., 2014a), SDCA (Shalev-Shwartz and Zhang, 2013), SVRG (Johnson and Zhang, 2013) and Finito/MISO (Defazio et al., 2014b; Mairal, 2015), this acceleration scheme needs a smaller runtime for finding an ϵ-optimal solution than applying algorithm $\mathcal{A}$ directly to (1).

Given the success of this acceleration scheme in the single-machine setting, it will be promising to also apply this scheme to the DSVRG to further improve its theoretical performance. Indeed, this can be done by choosing $\mathcal{A}$ in this acceleration scheme to be DSVRG. In particular, we follow the scheme in Algorithm 2 in Frostig et al. (2015) and obtain the DASVRG method as in Algorithm 3. In the p-th outer iteration of DASVRG, DSVRG is used to solve the proximal problem (34) in a distributed way up to an accuracy ϵ_p.

Theorem 16 *When* $\mu > 0$, $\sigma = \frac{L}{n}$, $\eta = \frac{1}{16L}$, $T = 96\kappa(f_\sigma)$, $K = \frac{1}{\log(9/8)} \log\left(4\left(\frac{2\sigma+\mu}{\mu}\right)^{3/2}\right)$

$$\text{and} \quad \alpha \geq \frac{384}{\log(9/8)m}\sqrt{\frac{2\kappa}{n}+1}\log\left(4\left(\frac{2\kappa}{n}+1\right)^{3/2}\right)\log\left(\left(\frac{2\kappa}{n}+1\right)\frac{f(\hat{x}_0)-f(x^*)}{\epsilon}\right), \tag{36}$$

Algorithm 3 find an ϵ-optimal solution with $P \leq \sqrt{\frac{8\sigma+4\mu}{\mu}}\log\left(\frac{(2\sigma+2\mu)(f(\hat{x}_0)-f(x^*))}{\mu\epsilon}\right)$ outer loops and at most $\frac{\frac{384}{\alpha}+2}{\log(9/8)}\sqrt{\frac{2\kappa}{n}+1}\log\left(4\left(\frac{2\kappa}{n}+1\right)^{3/2}\right)\log\left(\left(\frac{2\kappa}{n}+1\right)\frac{f(\hat{x}_0)-f(x^*)}{\epsilon}\right)$ rounds of communication.

In particular, if the right hand side of (36) is at most $O(1)$ in addition to the previous conditions, Algorithm 1 with $\alpha = \Theta(1)$ finds an ϵ-optimal solution with at most $O\left((1+\sqrt{\frac{\kappa}{n}})\log(1+\frac{\kappa}{n})\log\frac{1}{\epsilon}\right)$ rounds of communication.

Proof Due to the way $\{R_j\}_{j\in[m]}$ is generated and Lemma 13, the solution path $\tilde{x}^0, \tilde{x}^1, \tilde{x}^2, \ldots$ generated within the p-th outer loop of Algorithm 3 has same distribution as the solution path generated by applying the single-machine SVRG to (34) with an initial solution of $\tilde{x}^0 = \hat{x}_p$. Hence, according to Theorem 1 in Xiao and Zhang (2014), the following inequality holds for the p-th outer loop of Algorithm 3

$$\mathbb{E}\left[f_\sigma(\hat{x}_{p+1}; y_p) - \min_{x\in\mathbb{R}^d} f_\sigma(x; y_p)\right] \leq \left(\frac{1}{\mu\eta(1-4L\eta)T} + \frac{4L\eta(T+1)}{(1-4L\eta)T}\right)^K \left[f_\sigma(\hat{x}_p; y_p) - \min_{x\in\mathbb{R}^d} f_\sigma(x; y_p)\right].$$

Applying the values of η and T to this inequality, the p-th outer loop of Algorithm 3 ensures

$$\mathbb{E}\left[f_\sigma(\hat{x}_{p+1}; y_p) - \min_{x\in\mathbb{R}^d} f_\sigma(x; y_p)\right] \leq \left(\frac{8}{9}\right)^K \left[f_\sigma(\hat{x}_p; y_p) - \min_{x\in\mathbb{R}^d} f_\sigma(x; y_p)\right].$$

Algorithm 3 Distributed Accelerated SVRG (DASVRG)

Input: An initial solution $\hat{x}_0 \in \mathbb{R}^d$, data $\{f_i\}_{i\in[N]}$, the number of machines m, a step length $\eta < \frac{1}{4L}$, the number of stages K in DSVRG, the number of iterations T in each stage of DSVRG, the number of outer iterations P in DASVRG, a parameter $\sigma \geq 0$, and a parameter $\alpha \geq \frac{TKP}{N}$.

Output: $\hat{x}_P$

Generate (a) a random partition $\{S_j\}_{j\in[m]}$ of $[N]$, (b) m multi-sets $\{R_j\}_{j\in[m]}$ with $|R_j| = \alpha n$ uniformly sampled with replacement from $[N]$, and (c) data $\{f_i \mid i \in S_j \cup R_j\}$ stored on machine j.
$k \leftarrow 1$
Initialize $\rho = \frac{\mu+2\sigma}{\mu}$ and $z_0 = \hat{x}_0$
for $p = 0, 1, 2, \ldots, P-1$ **do**
  Center computes $y_p = \frac{1}{1+\rho^{-1/2}}\hat{x}_p + \frac{\rho^{-1/2}}{1+\rho^{-1/2}} z_p$ and $\tilde{x}^0 = \hat{x}_p$ and sends y_p to each machine
  for $\ell = 0, 1, 2, \ldots, K-1$ **do**
    Center sends $\tilde{x}^\ell$ to each machine
    for machine $j = 1, 2, \ldots, m$ **in parallel do**
      Compute $h_j^\ell = \sum_{i\in S_j} \nabla \tilde{f}_i(\tilde{x}^\ell; y_p)$ and send it to center
    end for
    Center computes $h^\ell = \frac{1}{N}\sum_{j=1}^m h_j^\ell$ and sends it to machine k
    $(\tilde{x}^{\ell+1}, \{R_j\}_{j\in[m]}, k) \leftarrow \text{SS-SVRG}(\tilde{x}^\ell, \{f_{\sigma,i}(x; y_p)\}_{i\in[N]}, h^\ell, \{R_j\}_{j\in[m]}, k, \eta, T)$ as an example, DASVRG still applies
  end for
  Machine k computes $\hat{x}_{p+1} = \tilde{x}^K$ and sends $\hat{x}_{p+1}$ to center
  Center computes $z_{p+1} = (1-\rho^{-1/2})z_p + \rho^{-1/2}[y_p - \rho(y_p - \hat{x}_{p+1})]$
end for

By this inequality and the choice of K, we can show that the solution $\hat{x}_{p+1}$ generated from the p-th outer loop of Algorithm 3 satisfies (35) with

$$\epsilon_p = \frac{1}{4}\left(\frac{\mu}{2\sigma+\mu}\right)^{3/2}\left[f_\sigma(\hat{x}_p; y_p) - \min_{x\in\mathbb{R}^d} f_\sigma(x; y_p)\right].$$

Using this result and Lemma 3.3, 3.4, 3.5 and 3.6 in Frostig et al. (2015), we can show that

$$\mathbb{E}\left[f(\hat{x}_P) - f(x^*)\right] \leq \frac{2\sigma+2\mu}{\mu}\left(1 - \frac{1}{2}\sqrt{\frac{\mu}{2\sigma+\mu}}\right)^P \left[f(\hat{x}_0) - f(x^*)\right].$$

This means Algorithm 3 finds an ϵ-optimal solution after $P \leq 2\sqrt{\frac{2\sigma+\mu}{\mu}}\log\left(\frac{(2\sigma+2\mu)(f(\hat{x}_0)-f(x^*)}{\mu\epsilon}\right)$ outer loops.

Since Algorithm 3 performs TKP iterative updates with one element of R_k removed after each update, it requires $TKP \leq \sum_{j\in[m]}|R_j| = \alpha N$, which is satisfied by the choices of T and K, the above upper bound of P and the assumption on α.

Since Algorithm 3 needs one round of communication to compute a batch gradient at each stage (one call of SS-SVRG) and one round after every αn iterative updates (when

$R_k = \emptyset$), it needs

$$\begin{aligned} & KP + \frac{TKP}{\alpha n} = \left(\frac{T}{\alpha n} + 1\right) KP \\ \leq\ & \left(\frac{96\kappa(f_\sigma)}{\alpha n} + 1\right)\frac{2}{\log(9/8)}\sqrt{\frac{2\sigma+\mu}{\mu}}\log\left(4\left(\frac{2\sigma+\mu}{\mu}\right)^{3/2}\right)\log\left(\frac{(2\sigma+2\mu)(f(\hat{x}_0)-f(x^*)}{\mu\epsilon}\right) \\ \leq\ & \left(\frac{192}{\alpha}+1\right)\frac{2}{\log(9/8)}\sqrt{\frac{2\kappa}{n}+1}\log\left(4\left(\frac{2\kappa}{n}+1\right)^{3/2}\right)\log\left(\left(\frac{2\kappa}{n}+1\right)\frac{f(\hat{x}_0)-f(x^*)}{\epsilon}\right) \end{aligned}$$

rounds of communication in total to find an ϵ-optimal solution. In the second inequality above, we use the fact that $\sigma = \frac{L}{n}$ and $\frac{\kappa(f_\sigma)}{n} = \frac{L+\sigma}{n\mu+n\sigma} \leq \frac{n+1}{n} \leq 2$. The second conclusion is a straightforward outcome of the first conclusion. ■

According to Theorem 16, DASVRG requires $\alpha \geq \Omega(\frac{1}{m}\sqrt{\frac{\kappa}{n}+1}\log(\frac{\kappa}{n}+1)\log(1/\epsilon))$ so that each machine must have a memory space of $|S_j| + |R_j| = n + \alpha n = \Omega(n + \frac{n}{m}\sqrt{\frac{\kappa}{n}+1}\log(\frac{\kappa}{n}+1)\log(1/\epsilon))$. For ERM problem where $\kappa = \Theta(\sqrt{N})$ and $\epsilon = \frac{1}{n^s}$ for a constant s, the requirement becomes $\alpha \geq \Omega(\frac{1}{m}\sqrt{\sqrt{\frac{m}{n}}+1}\log(\sqrt{\frac{m}{n}}+1)\log(n))$, which can be easily satisfied with a constant α ($\alpha = \Theta(1)$). Moreover, in this case, DASVRG only asks for $\Theta(n + \frac{n}{m}\sqrt{\sqrt{\frac{m}{n}}+1}\log(\sqrt{\frac{m}{n}}+1)\log(n)) = \Theta(n)$ memory space in each machine to store $S_j \cup R_j$ which is in the same order of magnitude as the space required by other distributed algorithms. Although we use $\kappa = \Theta(\sqrt{N})$ as an example, DASVRG still applies with $\alpha = \Theta(1)$ and $\Theta(n)$ memory space when $\epsilon = \frac{1}{n^s}$ and $\kappa = \Theta(N^b)$ with $b \in (0, 1)$.

Furthermore, another crucial insight we obtain here is that, although in the single-machine setting, the acceleration scheme of Frostig et al. (2015) and (Lin et al., 2015) only helps to reduce the time complexity when $\kappa \geq N$ by choosing $\sigma = \Theta(N)$, in the distributed setting, it helps to reduce the rounds as long as $\kappa \geq n$ by choosing $\sigma = \Theta(n)$. This shows an interesting and new application of this generic acceleration scheme.

In DSVRG and DASVRG, the subsets $\{S_j\}_{j\in[m]}$ are only used to compute the batch gradient $\nabla f(\tilde{x}^\ell)$. Therefore, both DSVRG and DASVRG can be applied with the same theoretical guarantees under the general setting where $\{S_j\}_{j\in[m]}$ is an arbitrary (not necessarily random) partition of $\{f_i\}_{i\in[N]}$ into equal-sized subsets. Because our lower bound for the round complexity is provided for the algorithms in $\mathcal{F}_\alpha$ where $\{S_j\}_{j\in[m]}$ is a random partition, we present both DSVRG and DASVRG in the same setting to show that the theoretical lower bound is almost reachable and DASVRG is nearly optimal within $\mathcal{F}_\alpha$.

7. DASVRG for Non-Strongly Convex Problem

In this section, we consider solving (1) when f is not strongly convex ($\mu = 0$) by the standard technique of applying DASVRG to the following regularized version of (1)

$$\min_{x\in\mathbb{R}^d}\left\{f^\lambda(x) \equiv f(x) + \frac{\lambda}{2}\|x\|^2 = \frac{1}{N}\sum_{i=1}^N f_i^\lambda(x)\right\}, \tag{37}$$

where $f_i^\lambda(x) \equiv f_i(x) + \frac{\lambda}{2}\|x\|^2$ and λ is a positive constant. Note that problem (37) satisfies Assumption 1 with ∇f_i^λ being $(L+\lambda)$-Lipschitz continuous and f^λ being λ-convex. The

condition number of f^λ is $\kappa(f^\lambda) \equiv \frac{L+\lambda}{\lambda}$. With λ small enough, the ϵ-solution of (37) found by DASVRG can be an ϵ-solution for (1). Although this method is well-known, we present it here just to show that the lower bound of round complexity for non-strongly convex problems in Theorem 3 can be achieved up to logarithmic terms under some scenario.

Theorem 17 *Suppose Algorithm 3 is applied to (37) where* $\lambda = \frac{\epsilon}{D^*}$ *with* $D^* \geq \|x^*\|^2$. *When* $\sigma = \frac{L+\lambda}{n}$, $\eta = \frac{1}{16(L+\lambda)}$, $T = 96\kappa(f^\lambda_\sigma) = 96\frac{L+\lambda+\sigma}{\lambda+\sigma}$, $K = \frac{1}{\log(9/8)}\log\left(4\left(\frac{2\sigma+\lambda}{\lambda}\right)^{3/2}\right)$, *and*

$$\alpha \geq \frac{384}{\log(9/8)m}\sqrt{\frac{2D^*L}{n\epsilon}+3}\log\left(4\left(\frac{2D^*L}{n\epsilon}+3\right)^{3/2}\right)\log\left(\left(\frac{2D^*L}{n\epsilon}+3\right)\frac{f(\hat{x}_0)-f(x^*)}{\epsilon/2}\right). \quad (38)$$

Algorithm 3 find an ϵ-optimal solution of (1) with $P = \sqrt{\frac{8\sigma+4\lambda}{\lambda}}\log\left(\frac{(2\sigma+2\lambda)(f(\hat{x}_0)-f(x^*)}{\lambda\epsilon/2}\right)$ *and* $\left(\frac{192}{\alpha}+1\right)\frac{2}{\log(9/8)}\sqrt{\frac{2D^*L}{n\epsilon}+3}\log\left(4\left(\frac{2D^*L}{n\epsilon}+3\right)^{3/2}\right)\log\left(\left(\frac{2D^*L}{n\epsilon}+3\right)\frac{f(\hat{x}_0)-f(x^*)}{\epsilon/2}\right)$ *rounds of communication.*

In particular, if the right hand side of (38) is at most $O(1)$ in addition to the previous conditions, Algorithm 1 with $\alpha = \Theta(1)$ finds an ϵ-optimal solution of (1) with at most $\tilde{O}\left(\left(\sqrt{\frac{L}{n\epsilon}}\right)\log^2\left(\frac{1}{\epsilon}\right)\right)$ *rounds of communication.*

Proof Let $\hat{x}_P$ be the output of Algorithm 3 when it is applied to (37) with the specific values of η, σ, T and K given in the statement of the theorem. Because ∇f_i^λ is $(L+\lambda)$-Lipschitz continuous and f^λ is λ-convex, according to Theorem 16, $\hat{x}_P$ is an $\frac{\epsilon}{2}$-optimal solution of (37) and it is found with the number of rounds of communication stated in the conclusion. Moreover, we can easily show that

$$\begin{aligned}\mathbb{E}[f(\hat{x}_P)-f(x^*)] &\leq f(\hat{x}_P)+\frac{\lambda}{2}\|\hat{x}_P\|^2-f(x^*)-\frac{\lambda}{2}\|x^*\|^2+\frac{\lambda}{2}\|x^*\|^2\\ &\leq f^\lambda(\hat{x}_P)-\min_{x\in\mathbb{R}^d}f^\lambda(x)+\frac{\lambda}{2}\|x^*\|^2 \leq \frac{\epsilon}{2}+\frac{\epsilon}{2}=\epsilon,\end{aligned}$$

where the last inequality is because $\lambda = \frac{\epsilon}{D^*}$ with $D^* \geq \|x^*\|^2$. Hence, $\hat{x}_P$ is an ϵ-optimal solution of (1). Then, the rest of the conclusion will be proved by applying Theorem 20. ■

According to (38) in Theorem 17, we acknowledge here that DASVRG can be applied to non-strongly convex problem with $\alpha = \Theta(1)$ only under a restricted condition, for example, when $\epsilon = O(\frac{1}{N^b})$ with $b \in (0,1)$. The development in this section is mostly for the theoretical interest on the tightness of the lower bound by Theorem 3.

8. Numerical Experiments

This section presents the numerical performances of our methods in comparisons with some existing distributed optimization techniques on simulated and real data.

8.1 Experiments with Simulated Data

In this section, we conduct numerical experiments with simulated data to compare our DSVRG algorithm with DISCO (Zhang and Lin, 2015) and a distributed implementation of gradient descent (GD) method. We use a single machine to simulate the distributed environment and all methods are implemented in Matlab running on a 64-bit Microsoft Windows 10 machine with a 2.70Ghz Intel(R) i7-6820HQ CPU and 8GB of memory. We consider ridge regression, $\min_{x\in\mathbb{R}^d}\frac{1}{N}\sum_{i=1}^N(a_i^Tx-b_i)^2+\frac{\lambda}{2}\|x\|^2$, where $a_i \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$ for $i \in [N]$ are the data points. We generate (a_i, b_i) following the experimental setup in Agarwal et al. (2012) by choosing a correlation parameter $\omega \in [0,1]$. Let A be the $N \times d$ matrix whose rows are a_i's. According to Agarwal et al. (2012), the eigenvalues of $\frac{1}{N}\mathbb{E}[A^TA]$ lie within the interval $\left[\frac{1}{(1+\omega)^2}, \frac{2}{(1-\omega)^2(1+\omega)}\right]$ so that the condition number κ increases when we choose ω close to 1. To make comparisons under different settings, we choose $\lambda = 10^{-4}$, $d = 50$, $N = 10^4$, $\omega \in \{0, 0.3, 0.5\}$ and $m \in \{10, 20\}$. All generated instances satisfy $\kappa = \frac{n^{1-2\delta}}{32}$ with $0 < \delta < \frac{1}{2}$ which is in the optimal regime of DSVRG (see Section 5.2).

For each instance, we compute the largest and the smallest eigenvalues of $\frac{1}{N}A^TA$ so as to compute the condition number κ and the parameters δ, η and T as in Proposition 15 for implementing DSVRG. Note that the lower bound for α given in Proposition 15 can be conservative and depends on the unknown value $f(x^*)$. In this experiment, we choose $\alpha = 1$ so that DSVRG can run for at most N iterations, which corresponds to $\frac{N}{|R_j|} = m$ rounds because $|R_j| = T = n$. In each main iteration of DISCO, an inexact Newton direction v is computed by solving a preconditioned linear system $P^{-1}\nabla^2 f(x_k)v = P^{-1}\nabla f(x_k)$ using a distributed preconditioned conjugate gradient method, where x_k in the solution in the kth main iteration. We choose $P = \nabla^2\bar{f}_1(x_k) + (\lambda + \sqrt{m}\times 0.001)I$ with $\bar{f}_1$ defined in (4) and compute P^{-1} directly. In the GD method, we use a step length of $\eta = \frac{1}{L}$.

Since DSVRG stores $|R_j| + |S_j| = (1+\alpha)n$ data points in machine j, it is meaningful to allow other methods to also store $(1+\alpha)n$ data points in each machine in the comparisons. With more local data, the round complexity of the distributed GD method is unchanged while that of DISCO will be reduced according to its round complexity $O((1+\frac{\sqrt{\kappa}}{n^{.25}})\log\frac{1}{\epsilon})$. Therefore, when implementing DISCO, we use different numbers of machines which lead to different amounts of local data. In particular, we choose $m = 2, 5$ and 10 in DISCO when $m = 10$ in DSVRG and GD and choose $m = 4, 10$ and 20 in DISCO when $m = 20$ in DSVRG and GD. Since $\alpha = 1$, DSVRG with 10 (20) machines and DISCO with 5 (10) machines store the same amount of data points in each machine.

The performance of the three methods are shown in Figure 2, where the vertical axis represents the logarithm of optimality gap (i.e., $\log(f(x_k) - f(x^*))$) and the horizon axis represents the number of rounds of communication. The values of κ and δ for each instance are shown in the figures. According to Figure 2, when δ is relatively large, DSVRG is less efficient than GD because the step length $\eta = \frac{1}{16n^\delta L}$ in DSVRG is too small and the small κ leads to a low round complexity of GD. However, when δ is relatively small, DSVRG significantly outperforms GD, which is consistent with Proposition 15. The performance of DISCO is improved when it uses more machines (and a smaller n), which is also consistent with its round complexity $O((1+\frac{\sqrt{\kappa}}{n^{.25}})\log\frac{1}{\epsilon})$. DSVRG with 10 machines is more efficient than DISCO with 10 and 5 machines for all instances here and is slightly worse than DISCO

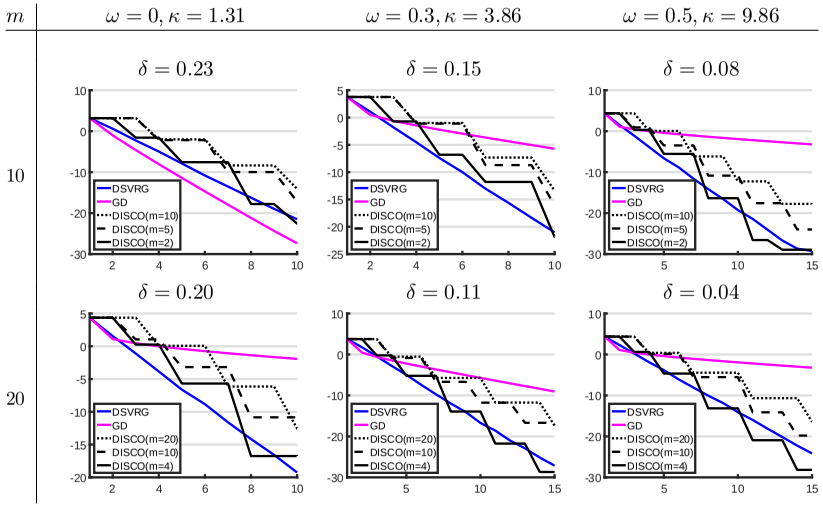


Figure 2: Comparing DSVRG implemented as in Proposition 15 with DISCO and the distributed gradient descent method (GD) in rounds. The condition number κ of each instance is in the regime where DSVRG is optimal (see Section 5.2).

with 2 machines. Recall that DSVRG with 10 machines stores the same amount of local data as DISCO with 5 machines. The results suggest that DSVRG with a small α can potentially outperform DISCO on the problems with $\kappa = O(n^{1-2\delta})$ even if the machines have access to the same amount of data in both algorithms. However, when α is large, DSVRG can be less efficient than DISCO because DISCO needs much fewer machines. The same conclusion can be made from the results where DSVRG uses 20 machines and DISCO uses 4, 10 and 20 machines.

8.2 Experiments with Real Data

In this section, we conduct numerical experiments with real data to compare our DSVRG and DASVRG algorithms with DisDSCA by Yang (2013) and a distributed implementation of the accelerated gradient method (Accel Grad) by Nesterov (2013). We apply these four algorithms to the ERM problem (2) with three data sets:[6] Covtype, Million Song and Epsilon. In particular, we solve ridge regression $\min_{x\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^N (a_i^T x - b_i)^2 + \frac{\lambda}{2}\|x\|^2$ with Million Song data and regularized logistic regression $\min_{x\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^N \log(1+\exp(-b_i a_i^T x)) + \frac{\lambda}{2}\|x\|^2$ with Covtype and Epsilon data. Following the previous work, we map the target variable of year from $1922 \sim 2011$ into $[0,1]$ for the Million Song data. To compare these methods in a challenging setting, we conduct experiments using random Fourier features

6. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

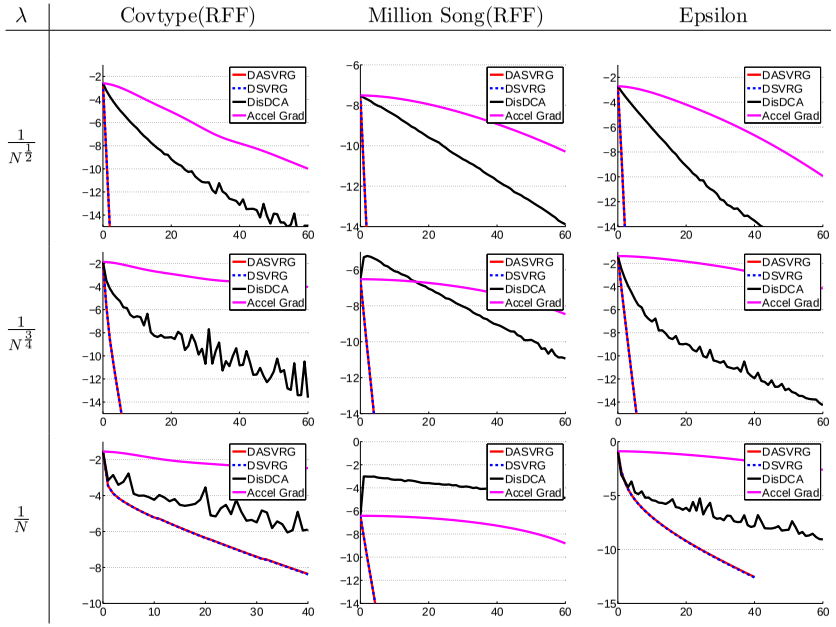


Figure 3: Comparing the DSVRG and DASVRG methods with DisDCA and the accelerated gradient method (Accel Grad) in rounds.

(RFF) (Rahimi and Recht, 2008) on Covtype and Million Song data sets. The RFF is a method for solving large-scale kernel methods by generating finite dimensional features, of which the inner product approximate the kernel similarity. We generate RFF corresponding to RBF kernel. After generating RFF, Covtype data has $N = 522,911$ examples $d = 1,000$ features and Million Song data has $N = 463,715$ examples and $d = 2,000$ features. Since the original Epsilon data is large enough (12G), we use its original features.

The experiments are conducted on a server (Intel(R) Xeon(R) CPU E5-2667 v2 3.30GHz) with multiple processes with each process simulating one machine. We choose the number of processes (machines) to be $m = 5$. To test the performances of algorithms for different condition numbers, we choose the regularization parameter λ to be $1/N^{0.5}$, $1/N^{0.75}$ and $1/N$. For each setting, L is computed as $\frac{\max_{i=1,\ldots,N}\|a_i\|^2}{\gamma} + \lambda$ where $\frac{1}{\gamma}$ is the Lipschiz continuous constant of $\nabla_x \phi(x,\xi)$ in (2), and μ is set to be λ. We implement DSVRG by choosing $\eta = \frac{1}{L}$, $T = 10,000$ and $K = \frac{N}{T}$. For DASVRG, we choose $\eta = \frac{1}{L}$, $T = 10,000$, $K = 1$ and $P = \frac{N}{T}$. In both DSVRG and DASVRG, we directly choose $R_j = S_j$ since, in practice, this implementation gives performances very similar to the performances when R_j is sampled separately. For DisDCA, we use SDCA (Shalev-Shwartz and Zhang, 2013) as

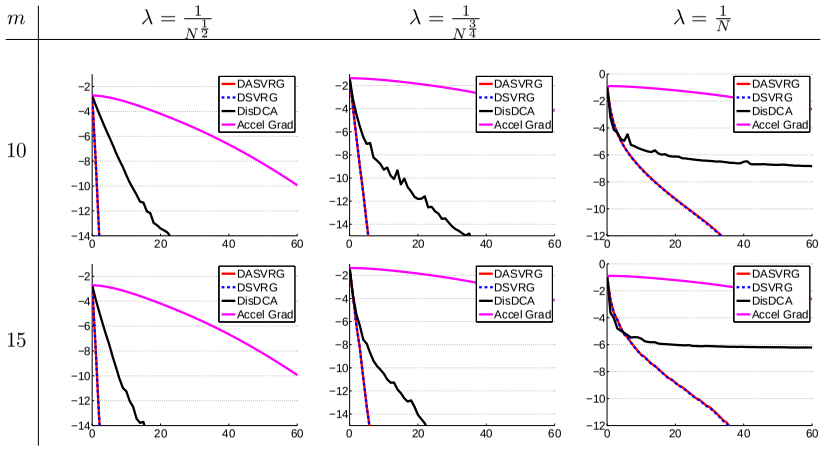


Figure 4: Comparing the DSVRG and DASVRG methods with DisDCA and the accelerated gradient method (Accel Grad) in rounds using Epsilon data.

the local solver so that it is equivalent to the implementation of CoCoA+ with $\sigma' = m$ and $\gamma = 1$ as in the experiments in Ma et al. (2015). We run SDCA for $T = 10,000$ iterations in each round of DisDCA with $\frac{N}{T}$ rounds in total.

The numerical results are presented in Figure 3, where the horizontal axis presents the number of rounds of communication conducted by algorithms and the vertical axis represents the logarithm of optimality gap. According to Figure 3, the performances of all algorithms get worse when λ decreases (so the condition number increases). We find that DSVRG and DASVRG have almost identical performances in rounds of communication and they both outperform the other two methods significantly. To compare the performances of algorithms under different values of m. We choose the $m = 10$ and 15 and repeat the same experiments on Epsilon data. The numerical results are shown in Figure 4. Similar to the case of $m = 5$, our DSVRG and DASVRG require fewer rounds to reach the same ϵ-optimal solution than the other two methods.

9. Conclusion

We study the round complexity for minimizing the average of N convex functions by distributed optimization with m machines under a new setting where each machine receives a subset of the N functions through both random partition and random sampling with replacement. Then, we propose a DSVRG algorithm under this setting which is a distributed extension of the existing SVRG algorithm. In DSVRG, the batch gradients are computed in parallel while the machines perform iterative updates in serial. DSVRG utilizes the local functions sampled with replacement to construct the unbiased stochastic gradient in each iterative update. We provide the theoretical analysis on the rounds of communication needed by DSVRG to find an ϵ-optimal solution, showing that DSVRG is optimal in terms of runtime, the amount of communication and the rounds of communication when the condition number is small. When the condition number is large, using an acceleration strategy by Frostig et al. (2015) and Lin et al. (2015), we proposed a DASVRG algorithm that requires even fewer rounds of communication than DSVRG and many existing methods that only store random partitioned data in machines, showing the advantage of the new distributed setting. Then, we define a broad family of algorithms which cover DSVRG, DASVRG and many existing methods in the literature. We provide the minimum number of rounds of communication needed by this family of algorithms for finding an ϵ-solution. The rounds of communication needed by DASVRG matches this lower bound up to logarithmic terms, and thus, is nearly optimal within this family.

Acknowledgments

We would like to thank Roy Frostig, Hongzhou Lin, Lin Xiao, and Yuchen Zhang for numerous helpful discussions throughout various stages of this work. Tianbao Yang is partially supported by National Science Foundation (IIS-1463988, IIS-1545995).

Appendix A. An Efficient Data Allocation Procedure

In this section, we design an efficient data allocation scheme which requires a smaller amount of communication to allocate $\{f_i | i \in S_j \cup R_j\}$ to machine j than the direct implementation.

We assume that a random partition $S_1, \ldots, S_m$ of $[N]$ can be constructed efficiently. Let Q be the number of samples in $\{R_j\}_{j\in[m]}$ needed for the algorithm to run the desired number of iterations. In other words, $Q = TK$ in DSVRG and $Q = TKP$ in DASVRG, both of which can be much smaller than αN, the total size $\{R_j\}_{j\in[m]}$. Therefore, in practical implementation, we only need to ensure the total size of $\{R_j\}_{j\in[m]}$ equal Q by allowing $|R_j| \leq \alpha n$ or even $R_j = \emptyset$ in order to save the communication amount.

A straightforward data allocation procedure is to prepare a partition $S_1, \ldots, S_m$ of $[N]$ and then sample a sequence of Q i.i.d. indices $r_1, \ldots, r_Q$ uniformly with replacement from $[N]$. After partitioning $r_1, \ldots, r_Q$ into m multi-sets $R_1, \ldots, R_m \subset [N]$, we allocate data $\{f_i | i \in S_j \cup R_j\}$ to machine j. Note that the amount of communication in distributing $S_1, \ldots, S_m$ is exactly N which is necessary for almost all distributed algorithms. However, in the worst case, this straightforward procedure requires an extra Q amount of communication for distributing $R_j \backslash S_j$.

To improve the efficiency of data allocation, we propose a procedure which reuses the randomness of $S_1, \ldots, S_m$ to generate the indices $r_1, \ldots, r_Q$ so that the overlap between S_j and R_j can be increased which helps reduce the additional amount of communication for distributing $R_1, \ldots, R_m$. The key observation is that the concatenation of $S_1, \ldots, S_m$ is a random permutation of $[N]$ which has already provided enough randomness needed by $R_1, \ldots, R_m$. Hence, it will be easy to build the i.i.d. samples $r_1, \ldots, r_Q$ by adding a little additional randomness on top of $S_1, \ldots, S_m$. With this observation in mind, we propose our data allocation procedure in Algorithm 4.

Algorithm 4 Data Allocation : DA(N, m, Q, α)

Input: Index set $[N]$, the number of machines m, and the constant α, and the length of target sequence Q.

Output: A random partition $S_1, \ldots, S_m$ of $[N]$, indices $r_1, \ldots, r_Q \in [N]$, multi-sets $R_1, \ldots, R_m \subset [N]$, and data $\{f_i \mid i \in S_j \cup R_j\}$ stored on machine j for each $j \in [m]$.

Center samples $r_1, \ldots, r_Q$ and $R_1, \ldots, R_m$ as follows:

1: Randomly partition $[N]$ into m disjoint sets $S_1, \ldots, S_m$ of the same size $n = \frac{N}{m}$.
2: Concatenate the subsets $S_1, \ldots, S_m$ into a random permutation $i_1, \ldots, i_N$ of $[N]$ so that $S_j = \{i_{(j-1)n+1}, \ldots, i_{jn}\}$.
3: **for** $\ell = 1$ to Q **do**
4: Let

$$r_\ell = \begin{cases} i_\ell & \text{with probability } 1 - \frac{\ell-1}{N} \\ i_{\ell'} & \text{with probability } \frac{1}{N} \text{ for } \ell' = 1, 2, \ldots, \ell - 1. \end{cases}$$

5: **end for**
6: Let

$$R_j = \begin{cases} \{r_{(j-1)\alpha n+1}, \ldots, r_{j\alpha n}\} & \text{if } j = 1, \ldots, \lceil \frac{Q}{\alpha n} \rceil - 1 \\ \{r_{(\lceil \frac{Q}{\alpha n} \rceil - 1)\alpha n+1}, \ldots, r_Q\} & \text{if } j = \lceil \frac{Q}{\alpha n} \rceil \\ \emptyset & \text{if } j = \lceil \frac{Q}{\alpha n} \rceil + 1, \ldots, m. \end{cases} \tag{39}$$

Distribute data points to machines:

7: Machine j acquires data points in $\{f_i | i \in S_j \cup R_j\}$ from the storage center.

The correctness and the expected amount of communication of Algorithm 4 are characterized as follows.

Lemma 18 *The sequence $r_1, \ldots, r_Q$ generated in Algorithm 4 has the same joint distribution as a sequence of i.i.d. indices uniformly sampled with replacement from $[N]$. Moreover, the expected amount of communication for distributing $\cup_{i=1}^m \{f_i | i \in R_j \backslash S_j\}$ is at most $\frac{Q^2}{2N}$.*

Proof Conditioned on $i_1, \ldots, i_{\ell-1}$ and $r_1, \ldots, r_{\ell-1}$, the random index i_ℓ has uniform distribution over $[N] \setminus \{i_1, \ldots, i_{\ell-1}\}$. Therefore, by Line 4 in Algorithm 4, the conditional distribution of the random index r_ℓ, conditioning on $i_1, \ldots, i_{\ell-1}$ and $r_1, \ldots, r_{\ell-1}$, is a uniform distribution over $[N]$. Hence, we complete the proof of the first claim of the lemma.

To analyze the amount of communication, we note that $i_\ell \neq r_\ell$ with probability $\frac{\ell-1}{N}$. Suppose $i_\ell \in S_j$ for some j. We know that the data point f_{r_ℓ} needs to be transmitted to machine j separately from S_j only if $i_\ell \neq r_\ell$. Therefore, the expected amount of communication for distributing $\cup_{i=1}^m \{f_i | i \in R_j \backslash S_j\}$ is upper bounded by $\sum_{\ell=1}^Q \frac{\ell-1}{N} \leq \frac{Q^2}{2N}$. ■

According to Lemma 18, besides the (necessary) N amount of communication to distribute $S_1, \ldots, S_m$, Algorithm 4 needs only $\frac{Q^2}{2N}$ additional amount of communication to distribute $R_1, \ldots, R_m$ thanks to the overlaps between S_j and R_j for each j. This additional amount is less than the Q amount required by the straightforward method when $Q \leq N$.

Note that we only need $Q = O(\kappa \log(1/\epsilon))$ in the DSVRG algorithm and $Q = O(n(1 + \sqrt{\frac{\kappa}{n}}) \log(1 + \frac{\kappa}{n}) \log \frac{1}{\epsilon})$ in the DASVRG algorithm. When $\kappa = \Theta(\sqrt{N})$ in a typical ERM problem, we have $\frac{Q^2}{2N} = O((\log(1/\epsilon))^2)$ in DSVRG and $\frac{Q^2}{2N} = O(\sqrt{\frac{n}{m}}(\log(\sqrt{\frac{m}{n}}) \log(1/\epsilon))^2)$ in DASVRG, both of which are much less than the N amount of communication for distributing $S_1, \ldots, S_m$. In other words, although DSVRG and DASVRG require additional amount of communication to allocate the data compared to other algorithms, this additional amount is marginal if κ is not too large.

Although Algorithm 4 is given for allocating S_j and R_j together to machine j, it can be used to only allocate R_j to machine j when S_j has already existed in machine j beforehand. The only requirements are that $\{S_j\}_{j \in [m]}$ still forms a random partition of $[N]$ and that, after generating R_j as (39), machine j knows which machine holds i for each $i \in R_j \backslash S_j$ so that machine j can acquire data in $R_j \backslash S_j$ from that machine in Line 7 of Algorithm 4.

References

Alekh Agarwal and Leon Bottou. A lower bound for the optimization of finite sums. In *International Conference on Machine Learning (ICML)*, 2015.

Alekh Agarwal, Sahand Negahban, and Martin Wainwright. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40 (5):2452–2482, 2012.

Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Necdet Aybat, Zi Wang, and Garud Iyengar. An asynchronous distributed proximal gradient method for composite convex optimization. In *International Conference on Machine Learning (ICML)*, 2015.

Amir Beck, Angelia Nedić, Asuman Ozdaglar, and Marc Teboulle. An $O(1/k)$ gradient method for network resource allocation problems. *Transactions on Control of Network Systems*, 1:64–73, 2014.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Mark Braverman, Ankit Garg, Tengyu Ma, Huy L Nguyen, and David P Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *The ACM Symposium on Theory of Computing*, pages 1011–1020, 2016.

Tsung-Hui Chang, Mingyi Hong, and Xiangfeng Wang. Multi-agent distributed optimization via inexact consensus ADMM. *IEEE Transactions on Signal Processing*, 63:482 – 497, 2015.

Annie I. Chen and Asuman E. Ozdaglar. A fast distributed proximal-gradient method. In *Annual Allerton Conference on Communication, Control, and Computing*, pages 601–608, 2012.

Zihao Chen, Luo Luo, and Zhihua Zhang. Communication lower bounds for distributed convex optimization: Partition data on features. In *AAAI*, pages 1812–1818, 2017.

Vaclav Chvatal. The tail of the hypergeometric distribution. *Discrete Mathematics*, 25: 285–287, 1979.

Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014a.

Aaron Defazio, Justin Domke, and Tiberio S. Caetano. Finito: A faster, permutable incremental gradient method for big data problems. In *International Conference on Machine Learning (ICML)*, 2014b.

Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66:889–916, 2016.

John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.

Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning (ICML)*, pages 2540–2548, 2015.

William Gropp, Ewing Lusk, Nathan Doss, and Anthony Skjellum. A high-performance, portable implementation of the mpi message passing interface standard. *Parallel Computing*, 22(6):789–828, 1996.

Martin Jaggi, Virginia Smith, Martin Takac, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I. Jordan. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Dusan Jakovetić, José M. F. Moura, and Joao Xavier. Distributed Nesterov-like gradient algorithms. In *IEEE Annual Conference on Decision and Control (CDC)*, 2012.

Dusan Jakovetić, Joao Xavier, and José M. F. Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59:1131–1146, 2014.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 315–323, 2013.

Jakub Konečnỳ and Peter Richtárik. Semi-stochastic gradient descent methods. *Frontiers in Applied Mathematics and Statistics*, 3:9, 2017.

Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

Guanghui Lan and Yi Zhou. An optimal randomized incremental gradient method. *Mathematical Programming*, 12:1–49, 2017.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I Jordan, Peter Richtárik, and Martin Takáč. Adding vs. averaging in distributed primal-dual optimization. In *International Conference on Machine Learning (ICML)*, 2015.

Mehrdad Mahdavi, Lijun Zhang, and Rong Jin. Mixed optimization for smooth functions. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Julien Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25:829–855, 2015.

Ali Makhdoumi and Asuman Ozdaglar. Convergence rate of distributed ADMM over networks. *IEEE Transactions on Automatic Control*, 2017.

Aryan Mokhtari, Wei Shi, Qing Ling, and Alejandro Ribeiro. Dqm: Decentralized quadratically approximated alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 64:5158 –5173, 2016.

João F. C. Mota, João M. F. Xavier, Pedro M. Q. Aguiar, and Markus Püschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. *IEEE Transaction on Signal Processing*, 61(10):2718–2723, 2013.

Angelia Nedić and Asuman Ozdaglar. On the rate of convergence of distributed subgradient methods for multi-agent optimization. In *IEEE Annual Conference on Decision and Control (CDC)*, 2007.

Angelia Nedić and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54:49–61, 2009.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87. Springer Science & Business Media, 2013.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1177–1184, 2008.

Sashank J Reddi, Jakub Konečnỳ, Peter Richtárik, Barnabás Póczós, and Alex Smola. AIDE: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

Nicolas L Roux, Mark Schmidt, and Francis R Bach. A stochastic gradient method with an exponential convergence _rate for finite training sets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2663–2671, 2012.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, 2017.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.

Shai Shalev-Shwartz, Ohad Shamir, Karthik Sridharan, and Nathan Srebro. Stochastic convex optimization. In *International Conference of Machine Learning (ICML)*, 2009.

Ohad Shamir. Without-replacement sampling for stochastic gradient methods. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

Ohad Shamir and Nathan Srebro. On distributed stochastic optimization and learning. In *Annual Allerton Conference on Communication, Control, and Computing*, 2014.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate Newton-type method. In *International Conference on Machine Learning (ICML)*, pages 1000–1008, 2014.

Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62:1750–1761, 2014.

Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing*, 63:6013–6023, 2015a.

Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25:944–966, 2015b.

J. V. Uspensky. *Introduction to Mathematical Probability*. McGraw-Hill, 1937.

Ermin Wei and Asuman Ozdaglar. Distributed alternating direction method of multipliers. In *IEEE Annual Conference on Decision and Control (CDC)*, pages 5445–5450, 2012.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Tianbao Yang. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Yuchen Zhang and Xiao Lin. Disco: Distributed optimization for self-concordant empirical loss. In *International Conference on Machine Learning (ICML)*, pages 362–370, 2015.

Journal of Machine Learning Research 18 (2017) 1-41 Submitted 6/17; Revised 10/17; Published 11/17

Kernel Partial Least Squares for Stationary Data

Marco Singer msinger@gwdg.de
Institute for Mathematical Stochastics
Georg-August-Universität
Göttingen, 37077, Germany

Tatyana Krivobokova tkrivob@gwdg.de
Institute for Mathematical Stochastics
Georg-August-Universität
Göttingen, 37077, Germany

Axel Munk amunk1@gwdg.de
Institute for Mathematical Stochastics
Georg-August-Universität
Göttingen, 37077, Germany

Editor: Sara van de Geer

Abstract

We consider the kernel partial least squares algorithm for non-parametric regression with stationary dependent data. Probabilistic convergence rates of the kernel partial least squares estimator to the true regression function are established under a source and an effective dimensionality condition. It is shown both theoretically and in simulations that long range dependence results in slower convergence rates. A protein dynamics example shows high predictive power of kernel partial least squares.

Keywords: effective dimensionality, long range dependence, nonparametric regression, source condition, protein dynamics

1. Introduction

Partial least squares (PLS) is a regularized regression technique developed by Wold et al. (1984) to deal with collinearities in the regressor matrix. It is an iterative algorithm where the covariance between response and regressor is maximized at each step, see Helland (1988) for a detailed description. Regularization in the PLS algorithm is obtained by stopping the iteration process early.

Several studies showed that partial least squares algorithm is competitive with other regression methods such as ridge regression and principal component regression, needing generally fewer iterations than the latter to achieve comparable estimation and prediction, see, e.g., Frank and Friedman (1993), Krämer and Braun (2007) and Singer et al. (2016). For an overview of further properties of PLS we refer to Rosipall and Krämer (2006).

Reproducing kernel Hilbert spaces (RKHS) have a long history in probability and statistics (see, e.g., Berlinet and Thomas-Agnan, 2004). Here we focus on the supervised kernel based learning approach for the solution of non-parametric regression problems. RKHS methods are both computationally and theoretically attractive, due to the kernel trick (Schölkopf et al., 1998) and the representer theorem (Wahba, 1999), as well as its generalization (Schölkopf et al., 2001). Within the reproducing kernel Hilbert space framework one can adapt linear regularized regression techniques like ridge regression and principal component regression to a non-parametric setting, see Saunders et al. (1998) and Rosipal et al. (2000), respectively. We refer to Schölkopf and Smola (2001) for more details on the kernel based learning approach.

Kernel PLS (KPLS) was introduced in Rosipal and Trejo (2001) by using the reformulation of the PLS algorithm of Lindgren et al. (1993). The relationship to kernel conjugate gradient (KCG) methods was highlighted in Blanchard and Krämer (2010a). It can be seen in Hanke (1995) that conjugate gradient methods are well suited for handling ill-posed problems, as they arise in kernel learning, see, e.g., De Vito et al. (2006).

Rosipal (2003) investigated the performance of kernel partial least squares for non-linear discriminant analysis. Blanchard and Krämer (2010a) proved the consistency of KPLS when the algorithm is stopped early without giving convergence rates.

Caponnetto and de Vito (2007) showed that kernel ridge regression (KRR) attains optimal probabilistic rates of convergence for independent and identically distributed data, using a source and a polynomial effective dimensionality condition. A generalization of these results to a wider class of effective dimensionality conditions and extension to kernel principal component regression can be found in Dicker et al. (2017).

For independent identically distributed data Blanchard and Krämer (2010b) obtained probabilistic convergence rates for a certain kernel conjugate gradient estimator under early stopping, while Lin and Zhou (2017) considered kernel partial least squares estimators with the cross-validation stopping rule.

In contrast to existing works, we derive probabilistic convergence rates of the kernel partial least squares estimator to the true regression function when the input data are not independent and identically distributed, but rather stationary time series. To the best of our knowledge, none of the kernel regression methods have been considered for the dependent data so far. Our results can be applied to stationary dependence structures, given that certain concentration inequalities for these data hold. The derived convergence rates depend not only on the complexity of the target function and of the data mapped into the kernel space, but also on the persistence of the dependence in the data. For measuring the complexity of the data we consider general effective dimensionality conditions. In a Gaussian setting we prove that the short range dependence still leads to optimal rates, but if the dependence is more persistent, the rates become slower. We illustrate the good predictive performance of KPLS by an application to the molecular dynamics of a bacteriophage protein.

2. Kernel Partial Least Squares

Consider the non-parametric regression problem

$$y_t = f^*(X_t) + \varepsilon_t, \quad t \in \mathbb{Z}. \tag{1}$$

Here $\{X_t\}_{t\in\mathbb{Z}}$ is a d-dimensional, $d \in \mathbb{N}$, stationary time series on a probability space $(\Omega, \mathcal{A}, \mathrm{P})$ and $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ is an independent and identically distributed sequence of real valued random variables with expectation zero and variance $\sigma^2 > 0$ that is independent of $\{X_t\}_{t\in\mathbb{Z}}$.

©2017 Marco Singer, Tatyana Krivobokova and Axel Munk.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-306.html.

Let X be a random vector that is independent of $\{X_t\}_{t\in\mathbb{Z}}$ and $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ with the same distribution as X_0. The target function we seek to estimate is $f^* \in \mathcal{L}^2\left(\mathrm{P}^X\right)$.

For the purpose of supervised learning assume that we have a training sample $\{(X_t, y_t)\}_{t=1}^n$ for some $n \in \mathbb{N}$. In the following we introduce some basic notation for the kernel based learning approach.

Define with $(\mathcal{H}, \langle\cdot,\cdot\rangle_\mathcal{H})$ the RKHS of functions on $\mathbb{R}^d$ with reproducing kernel $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, i.e., it holds

$$g(x) = \langle g, k(\cdot, x)\rangle_\mathcal{H}, \quad x \in \mathbb{R}^d, g \in \mathcal{H}. \tag{2}$$

The corresponding inner product and norm in $\mathcal{H}$ is denoted by $\langle\cdot,\cdot\rangle_\mathcal{H}$ and $\|\cdot\|_\mathcal{H}$, respectively. We refer to Berlinet and Thomas-Agnan (2004) for examples of Hilbert spaces and their reproducing kernels. In the following we deal with reproducing kernel Hilbert spaces which fulfill the following, rather standard, conditions:

(K1) $\mathcal{H}$ is separable,

(K2) There exists a $\kappa > 0$ such that $|k(x, y)| \le \kappa$ for all $x, y \in \mathbb{R}^d$ and k is measurable.

Under (K1) the Hilbert-Schmidt norm $\|\cdot\|_{\mathrm{HS}}$ for operators mapping from $\mathcal{H}$ to $\mathcal{H}$ is well defined. If condition (K2) holds, all functions in $\mathcal{H}$ are bounded, see Berlinet and Thomas-Agnan (2004), chapter 2. The conditions are satisfied for a variety of popular kernels, e.g., Gaussian or triangular.

The main principle of RKHS methods is the mapping of the data X_t into $\mathcal{H}$ via the feature maps $\phi_t = k(\cdot, X_t)$, $t = 1, \ldots, n$. This mapping can be done implicitly by using the kernel trick $\langle\phi_t, \phi_s\rangle_\mathcal{H} = k(X_t, X_s)$ and thus only the $n \times n$ dimensional kernel matrix $K_n = n^{-1}[k(X_t, X_s)]_{t,s=1}^n$ is needed in the computations. Then the task for RKHS methods is to find coefficients $\alpha_1, \ldots, \alpha_n$ such that $f_\alpha = \sum_{t=1}^n \alpha_t\phi_t$ is an adequate approximation of f^* in $\mathcal{H}$, measured in the $\mathcal{L}^2\left(\mathrm{P}^X\right)$ norm $\|\cdot\|_2$.

There are a variety of different approaches to estimate the coefficients $\alpha_1, \ldots, \alpha_n$, including kernel ridge regression, kernel principal component regression and, of course, kernel partial least squares. The latter method was introduced by Rosipal and Trejo (2001) and is the focus of the current work.

It was shown by Krämer and Braun (2007) that the KPLS algorithm solves

$$\widehat{\alpha}_i = \arg\min_{v\in\mathcal{K}_i(K_n,y)} n^{-1}\|y - K_n v\|^2, \quad i = 1, \ldots, n, \tag{3}$$

with $y = (y_1, \ldots, y_n)^\mathrm{T}$. Here $\mathcal{K}_i(K_n, y) = \operatorname{span}\left\{y, K_n y, K_n^2 y, \ldots, K_n^{i-1} y\right\}$, $i = 1, \ldots, n$, is the ith order Krylov space with respect to K_n and y and $\|\cdot\|$ denotes the Euclidean norm. The dimension i of the Krylov space is the regularization parameter for KPLS.

We will introduce several operators that will be crucial for our further analysis. Define two integral operators: the kernel integral operator $T^* : \mathcal{L}^2\left(\mathrm{P}^X\right) \to \mathcal{H}, g \mapsto \mathrm{E}\{k(\cdot, X)g(X)\}$ and the change of space operator $T : \mathcal{H} \to \mathcal{L}^2\left(\mathrm{P}^X\right), g \mapsto g$, which is well defined if (K2) holds. It is easy to see that T, T^* are adjoint, i.e., for $u \in \mathcal{H}$ and $v \in \mathcal{L}^2\left(\mathrm{P}^X\right)$ it holds $\langle T^* v, u\rangle_\mathcal{H} = \langle v, Tu\rangle_2$ with $\langle\cdot,\cdot\rangle_2$ being the inner product in $\mathcal{L}^2\left(\mathrm{P}^X\right)$.

The sample analogues of T, T^* are $T_n : \mathcal{H} \to \mathbb{R}^n, g \mapsto \{g(X_1), \ldots, g(X_n)\}^\mathrm{T}$ and $T_n^* : \mathbb{R}^n \to \mathcal{H}, (v_1, \ldots, v_n)^\mathrm{T} \mapsto n^{-1}\sum_{t=1}^n v_t k(\cdot, X_t)$, respectively. Both operators are adjoint with respect to the rescaled Euclidean product $n^{-1}u^\mathrm{T}v$, $u, v \in \mathbb{R}^d$

Finally, we define the sample kernel covariance operator $S_n = T_n^* T_n : \mathcal{H} \to \mathcal{H}$ and the population kernel covariance operator $S = T^*T : \mathcal{H} \to \mathcal{H}$. Note that it holds $K_n = T_n T_n^*$. Under (K1) and (K2) S is a self-adjoint compact operator with operator norm $\|S\|_\mathcal{L} \le \kappa$, see Caponnetto and de Vito (2007).

With this notation we can restate (3) for the function f_α

$$f_{\widehat{\alpha}_i} = \arg\min_{g\in\mathcal{K}_i(S_n, T_n^* y)} n^{-1}\|y - \{g(X_1), \ldots, g(X_n)\}^\mathrm{T}\|^2 = \arg\min_{g\in\mathcal{K}_i(S_n, T_n^* y)} n^{-1}\|y - T_n g\|^2. \tag{4}$$

Hence, we are looking for functions that minimize the squared distance to y constrained to a sequence of Krylov spaces.

In the literature of ill-posed problems it is well known that without further conditions on the target function f^* the convergence rate of the conjugate gradient algorithm can be arbitrarily slow, see Hanke (1995), chapter 3.2. One common a-priori assumption on the regression function f^* is a source condition:

(S) There exist $r \ge 0$, $R > 0$ and $u \in \mathcal{L}^2\left(\mathrm{P}^X\right)$ such that $f^* = (TT^*)^r u$ and $\|u\|_2 \le R$.

If $r \ge 1/2$, then the target function $f^* \in \mathcal{L}^2\left(\mathrm{P}^X\right)$ coincides almost surely with a function $f \in \mathcal{H}$ and we can write $f^* = Tf$, see Cucker and Smale (2002). With this the kernel partial least squares estimator $f_{\widehat{\alpha}_i}$ estimates the correct target function, not only its best approximation in $\mathcal{H}$. This case is known as the inner case.

The situation with $r < 1/2$ is referred to as the outer case. Under additional assumptions, e.g., the availability of additional unlabeled data, it is still possible that an estimator of f^* converges to the true target function in $\mathcal{L}^2\left(\mathrm{P}^X\right)$ norm with optimal rates (with respect to the number n of labeled data points). See De Vito et al. (2006) for a detailed description of this semi-supervised approach for kernel ridge regression in the independent and identically distributed case. We do not treat the case $r < 1/2$ in this work.

A source condition is often interpreted as an abstract smoothness condition, see, e.g., Bissantz et al. (2007) for several examples. This can be seen as follows. Let $\eta_1 \ge \eta_2 \ge \ldots$ be the eigenvalues and $\psi_1, \psi_2, \ldots$ the corresponding eigenfunctions of the compact operator S. Then it is easy to see that the source condition (S) is equivalent to $f = \sum_{j=1}^\infty b_j\psi_j \in \mathcal{L}^2\left(\mathrm{P}^X\right)$ with b_j such that $\sum_{j=1}^\infty \eta_j^{-2(r+1/2)} b_j^2 < \infty$. Hence, the higher r is chosen the faster the sequence $\{b_j\}_{j=1}^\infty$ must converge to zero. Therefore, the sets of functions for which source conditions hold are nested, i.e., the larger r is the smaller the corresponding set will be. The set with $r = 1/2$ is the largest one and corresponds to a zero smoothness condition, i.e., $\sum_{j=1}^\infty \eta_j^{-2} b_j^2 < \infty$, which is equivalent to $f \in \mathcal{H}$. For more details we refer to Dicker et al. (2017).

3. Consistency of Kernel Partial Least Squares

The KCG algorithm as described by Blanchard and Krämer (2010b) is consistent when stopped early and convergence rates can be obtained when a source condition (S) holds. Here we will proof the same property for KPLS. Early stopping in this context means that we stop the algorithm at some $a = a(n) \le n$ and consider the estimator $f_{\widehat{\alpha}_a}$ for f^*.

The difference between KCG and KPLS is the norm which is optimized. The kernel conjugate gradient algorithm studied in Blanchard and Krämer (2010b) estimates the coefficients $\alpha \in \mathbb{R}^n$ of f_α via $\widehat{\alpha}_i^{CG} = \arg\min_{v\in\mathcal{K}_i(K_n,y)}\langle y - K_n v, K_n(y - K_n v)\rangle$. It is easy to

see that this optimization problem can be rewritten for the function f_α as

$$\min_{g\in\mathcal{K}_i(S_n,T_n^*y)} n^{-1}\|T_n^*y - S_ng\|_{\mathcal{H}}^2 = \min_{g\in\mathcal{K}_i(S_n,T_n^*y)} n^{-1}\|T_n^*\left(y - T_ng\right)\|_{\mathcal{H}}^2,$$

compared to (4) for KPLS. Thus, KCG obtains the least squares approximation g in the $\mathcal{H}$-norm for the normal equation $T_n^*y = T_n^*T_ng$ and KPLS finds a function that minimizes the residual sum of squares. In both methods the solutions are restricted to functions $g \in \mathcal{K}_i(S_n, T_n^*y)$.

An advantage of the kernel conjugate gradient estimator is that concentration inequalities can be established for both T_n^*y and S_n and applied directly as the optimization function contains both quantities. The stopping index for the regularization can be chosen by a discrepancy principle as $a^* = \min\{1 \le i \le n : \|S_nf_{\widehat{\alpha}_i^{CG}} - T_n^*y\| \le \Lambda_n\}$ with Λ_n being a threshold sequence that goes to zero as n increases.

On the other hand, the function to be optimized for KPLS contains only y and $T_ng = \{g(X_1), \ldots, g(X_n)\}^{\mathrm{T}}$ for which statistical properties are not readily available. Thus, we need to find a way to apply the concentration inequalities for T_n^*y and S_n to this slightly different problem. This leads to complications in the proof of consistency and a rather different and more technical stopping rule for choosing the optimal regularization parameter a^* is used, as can be seen in Theorem 1. This stopping rule has its origin in Hanke (1995).

In the following $\|\cdot\|_{\mathcal{L}}$ denotes the operator norm and $\|\cdot\|_{\mathrm{HS}}$ is the Hilbert-Schmidt norm.

Theorem 1 *Assume that conditions (K1), (K2), (S) hold with $r \ge 3/2$ and there are constants $C_\delta(\nu), C_\epsilon(\nu) > 0$ and a sequence $\{\gamma_n\}_{n\in\mathbb{N}} \subset [0,\infty)$, $\gamma_n \to 0$, such that we have for $\nu \in (0,1]$*

$$\mathrm{P}\left(\|S_n - S\|_{\mathcal{L}} \le C_\delta(\nu)\gamma_n\right) \ge 1 - \nu/2,$$
$$\mathrm{P}\left(\|T_n^*y - Sf\|_{\mathcal{H}} \le C_\epsilon(\nu)\gamma_n\right) \ge 1 - \nu/2.$$

Define the stopping index a^* by

$$a^* = \min\left\{1 \le a \le n : \sum_{i=0}^{a} \|S_nf_{\widehat{\alpha}_i} - T_n^*y\|_{\mathcal{H}}^{-2} \ge \left(C\gamma_n\right)^{-2}\right\}, \tag{5}$$

with $C = C_\epsilon(\nu) + \kappa^{r-1/2}(r+1/2)R\{1 + C_\delta(\nu)\}$.

Then it holds with probability at least $1-\nu$ that

$$\|f_{\widehat{\alpha}_{a^*}} - f^*\|_2 = O\left\{\gamma_n^{2r/(2r+1)}\right\},$$
$$\|f_{\widehat{\alpha}_{a^*}} - f\|_{\mathcal{H}} = O\left\{\gamma_n^{(2r-1)/(2r+1)}\right\},$$

with $f^* = Tf$.

It can be shown that the stopping rule (5) always determines a finite index, i.e., the set the minimum is taken over is not empty, see Hanke (1995), chapter 4.3.

The theorem yields two convergence results, one in the $\mathcal{H}$-norm and one in the $\mathcal{L}^2\left(\mathrm{P}^X\right)$-norm. It holds that $\|v\|_2 = \|S^{1/2}v\|_{\mathcal{H}}$. These are the endpoints of a continuum of norms $\|v\|_\beta = \|S^\beta v\|_{\mathcal{H}}$, $\beta \in [0, 1/2]$ that were considered in Nemirovskii (1986) for the derivation of convergence rates for KCG algorithms in a deterministic setting.

The convergence rate of the kernel partial least squares estimator depends crucially on the sequence γ_n and the source parameter r. If $\gamma_n = O(n^{-1/2})$, this yields the same convergence rate as Theorem 2.1 of Blanchard and Krämer (2010b) for kernel conjugate gradient or de Vito et al. (2005) for kernel ridge regression with independent and identically distributed data. For stationary Gaussian time series we will derive concentration inequalities in the next section and obtain convergence rates depending on the source parameter r and the range of dependence. Note that Theorem 1 is rather general and it can be applied to any kind of dependence structure, as long as the necessary concentration inequalities can be established.

The next theorem derives faster convergence rates under assumptions on the effective dimensionality of operator S, which is defined as $d_\lambda = \mathrm{tr}\{(S+\lambda)^{-1}S\}$. The concept of effective dimensionality was introduced in Zhang (2003) to get sharp error bounds for general learning problems considered there. If $\mathcal{H}$ is a finite dimensional space, it was shown in Zhang (2003) that $d_\lambda \le \dim(\mathcal{H})$. For infinite dimensional spaces it describes the complexity of the interactions between data and reproducing kernel.

If $d_\lambda = O(\lambda^{-s})$ for some $s \in (0,1]$, Caponnetto and de Vito (2007) showed that the order optimal convergence rates $n^{-r/(2r+s)}$ are attained for KRR with independent and identically distributed data.

The effective dimensionality clearly depends on the behaviour of eigenvalues of S. If these converge sufficiently fast to zero, nearly parametric rates of convergence can be achieved for reproducing kernel Hilbert space methods, see, e.g., Dicker et al. (2017). In particular, the behaviour of d_λ around zero is of interest, since it determines how ill-conditioned the operator $(S+\lambda)^{-1}$ becomes. In the following theorem we set $\lambda = \lambda_n$ for a sequence $\{\lambda_n\}_{n\in\mathbb{N}} \subset (0,\infty)$ that converges to zero.

Theorem 2 *Assume that conditions (K1), (K2), (S) hold with $r \ge 1/2$ and that the effective dimensionality d_λ is known. Additionally, there are constants $C_\delta(\nu), C_\epsilon(\nu), C_\psi > 0$ and a sequence $\{\gamma_n\}_{n\in\mathbb{N}} \subset [0,\infty)$, $\gamma_n \to 0$, such that for $\nu \in (0,1]$ and n sufficiently large*

$$\mathrm{P}\left\{\|S_n - S\|_{\mathcal{L}} \le C_\delta(\nu)\gamma_n\right\} \ge 1 - \nu/3,$$
$$\mathrm{P}\left\{\|(S+\lambda_n)^{-1/2}(T_n^*y - Sf)\|_{\mathcal{H}} \le C_\epsilon(\nu)\sqrt{d_{\lambda_n}}\gamma_n\right\} \ge 1 - \nu/3,$$
$$\mathrm{P}\left\{\|(S+\lambda_n)^{1/2}(S_n+\lambda_n)^{-1/2}\|_{\mathcal{L}} \le C_\psi\right\} \ge 1 - \nu/3,$$

Here $\{\lambda_n\}_{n\in\mathbb{N}} \subset (0,\infty)$ is a sequence converging to zero such that for n large enough

$$\gamma_n \le \lambda_n^{r-1/2}. \tag{6}$$

Take $\zeta_n = \max\{\sqrt{\lambda_n d_{\lambda_n}}\gamma_n, \lambda_n^{r+1/2}\}$ Define the stopping index a^* by

$$a^* = \min\left\{1 \le a \le n : \sum_{i=0}^{a} \|S_nf_{\widehat{\alpha}_i} - T_n^*y\|_{\mathcal{H}}^{-2} \ge \left(C\zeta_n\right)^{-2}\right\}, \tag{7}$$

with $C = 4R\max\{1, C_\psi^2, (r-1/2)\kappa^{r-3/2}C_\delta(\nu), 2^{-1/2}R^{-1}C_\psi C_\epsilon(\nu)\}$.

Then it holds with probability at least $1-\nu$ *that*

$$\|f_{\widehat{\alpha}_{a^*}} - f^*\|_2 = O\left\{\lambda_n^{-1/2}\zeta_n\right\},$$
$$\|f_{\widehat{\alpha}_{a^*}} - f\|_{\mathcal{H}} = O\left\{\lambda_n^{-1}\zeta_n\right\},$$

with $f^* = Tf$.

The condition (6) holds trivially for $r = 1/2$ as γ_n converges to zero. For $r > 1/2$ the sequence λ_n must not converge to zero arbitrarily fast.

In its general form Theorem 2 does not give immediate insight in the probabilistic convergence rates of the kernel partial least squares estimator. Therefore, we state two corollaries, where the function d_λ is specified. In both corollaries we explicitly state the choice of the sequence λ_n that yield the corresponding rates.

Corollary 3 *Assume that there exists* $s \in (0,1]$ *such that* $d_\lambda = O(\lambda^{-s})$ *for* $\lambda \to 0$. *Then under conditions of Theorem 2 with* $\lambda_n = \gamma_n^{2/(2r+s)}$ *it holds with probability at least* $1-\nu$ *that*

$$\|f_{\widehat{\alpha}_{a^*}} - f^*\|_2 = O\left\{\gamma_n^{2r/(2r+s)}\right\}.$$

Polynomial decay of the effective dimensionality $d_\lambda = \operatorname{tr}\{(S+\lambda)^{-1}S\}$ occurs if the eigenvalues of S also decay polynomially fast, that is, $\mu_i = c_s i^{-1/s}$ for $s \in (0,1]$, since in this case $d_\lambda = \sum_{i=1}^{\infty}\{1+\lambda/c_s i^{1/s}\}^{-1} = O(\lambda^{-s})$. This holds, for example, for the Sobolev kernel $k(x,y) = \min(x,y)$, $x, y \in [0,1]$ and data that are uniformly distributed on $[0,1]$, see Raskutti et al. (2014).

If $\gamma_n = n^{-1/2}$, then the KPLS estimator converges in the $\mathcal{L}^2\left(\mathrm{P}^X\right)$-norm with a rate of $n^{-r/(2r+s)}$. This rate is shown to be optimal in Caponnetto and de Vito (2007) for KRR with independent identically distributed data.

Note that the rate obtained in Theorem 1 corresponds to $\gamma_n^{-2r/(2r+s)}$ with $s = 1$, i.e., the worst case rate with respect to the parameter $s \in (0,1]$.

In the next corollary to Theorem 2 we assume that the effective dimensionality behaves in a logarithmic fashion.

Corollary 4 *Let* $d_\lambda = O\{\log(1+a/\lambda)\}$ *for* $\lambda \to 0$ *and* $a > 0$. *Then under the conditions of Theorem 2 with* $\lambda_n = \gamma_n^2\log\{\gamma_n^{-2}\}$ *and* $r = 1/2$ *it holds with probability at least* $1-\nu$ *that*

$$\|f_{\widehat{\alpha}_{a^*}} - f^*\|_2 = O\left\{\gamma_n\log(1/2\gamma_n^{-2})\right\}.$$

The effective dimensionality takes the special form considered in this corollary, for example, when the eigenvalues of S decay exponentially fast. This holds, for example, if the data are Gaussian and the Gaussian kernel is used, see Section A. If $\gamma_n = O(n^{-1/2})$, then the convergence rate is of order $O\{n^{-1}\log(n)\}$, which is nearly parametric. It is noteworthy that the source condition only impacts the choice of the sequence λ_n, not the convergence rates of the estimator in the $\mathcal{L}^2\left(\mathrm{P}^X\right)$-norm. Therefore, we stated the corollary for $r = 1/2$,

which is a minimal smoothness condition on f^*, i.e., that $f^* = Tf$ almost surely for an $f \in \mathcal{H}$.

The rates obtained in Corollaries 3 and 4 for $\gamma_n = n^{-1/2}$ were derived in Dicker et al. (2017) for kernel ridge regression and kernel principal component regression under the assumption of independent and identically distributed data.

4. Concentration Inequalities for Gaussian Time Series

Crucial assumptions of Theorem 1 and 2 are the concentration inequalities for S_n and T_n^*y and convergence of the sequence $\{\gamma_n\}_{n\in\mathbb{N}}$. Here we establish such inequalities in a Gaussian setting for stationary time series. At the end of this section we will state explicit convergence rates for $f_{\widehat{\alpha}_{a^*}}$ that depend not only on the source parameter $r \geq 1/2$ and the effective dimensionality d_λ, but also on the persistence of the dependence in the data.

The Gaussian setting is summarized in the following assumptions

(D1) $(X_h, X_0)^{\mathrm{T}} \sim \mathcal{N}_{2d}(0, \Sigma_h)$, $h = 1, \ldots, n-1$, with

$$\Sigma_h = \begin{bmatrix}\tau_0 & \tau_h \\ \tau_h & \tau_0\end{bmatrix} \otimes \Sigma.$$

Here $\Sigma \in \mathbb{R}^{d\times d}$ and $V = [\tau_{|i-j|}]_{i,j=1}^n \in \mathbb{R}^{n\times n}$ are positive definite, symmetric matrices and $\otimes$ denotes the Kronecker product between matrices. Furthermore $X_0 \sim \mathcal{N}_d(0, \tau_0\Sigma)$.

(D2) For the autocorrelation function $\rho_h = \tau_0^{-1}\tau_h$ there exists a $q > 0$ such that $|\rho_h| \leq (h+1)^{-q}$ for $h = 0, \ldots, n-1$.

Condition (D1) is a separability condition for the covariance matrices Σ_h, $h = 0, \ldots, n-1$. Due to (D1) the effects (on the covariance) over time and between the different variables can be treated separately. Under condition (D2) it is easy to see that from $q > 1$ follows the absolute summability of the autocorrelation function ρ and thus $\{X_t\}_{t\in\mathbb{Z}}$ is a short memory process. Stationary short memory processes keep many of the properties of independent and identically distributed data, see, e.g., Brockwell and Davis (1991).

On the other hand $q \in (0,1]$ yields a long memory process, see, e.g., Definition 3.1.2 in Giraitis et al. (2012). Examples of long memory processes are the fractional Gaussian noise with an autocorrelation function that behaves like $(h+1)^{-2(1-H)}$, with $H \in [0,1)$ being the Hurst coefficient. Stationary long memory processes exhibit dependencies between observations that are more persistent, and many statistical results that hold for independent and identically distributed data, turn out to be false. See Samorodnitsky (2007) for more details.

The next theorem gives concentration inequalities for both estimators S_n and T_n^*y in a Gaussian setting with convergence rates depending on the parameter $q > 0$. These inequalities are the ones needed in Theorem 1 and Theorem 2. Recall that $d_\lambda = \operatorname{tr}\{(S+\lambda)^{-1}S\}$ denotes the effective dimensionality of S.

Theorem 5 *(i) Define* $\mathrm{d}\mu_h(x,y) = \mathrm{d}\mathrm{P}^{X_h,X_0}(x,y) - \mathrm{d}\mathrm{P}^{X_0}(x)\mathrm{d}\mathrm{P}^{X_0}(y)$. *Under Assumptions (K1) and (K2) it holds for* $\nu \in (0,1]$ *with probability at least* $1-\nu$ *that*

$$\|S_n - S\|_{\mathcal{L}}^2 \leq \frac{2\nu^{-1}}{n^2}\sum_{h=1}^{n-1}(n-h)\int\limits_{\mathbb{R}^{2d}} k^2(x,y)\mathrm{d}\mu_h(x,y) + \frac{\nu^{-1}}{n}\left[\mathrm{E}\{k^2(X_0,X_0)\} - \|S\|_{\mathrm{HS}}^2\right],$$

$$\|T_n^* y - Sf\|_{\mathcal{H}}^2 \leq \frac{2\nu^{-1}}{n^2}\sum_{h=1}^{n-1}(n-h)\int\limits_{\mathbb{R}^{2d}} k(x,y)f(x)f(y)\mathrm{d}\mu_h(x,y) + \frac{\nu^{-1}}{n}\left[\mathrm{E}\left\{k(X_0,X_0)f^2(X_0)\right\} - \|Sf\|_{\mathcal{H}}^2 + \sigma^2\mathrm{E}\{k^2(X_0,X_0)\}\right].$$

(ii) Assume that additionally to (K1), (K2) also (D1), (D2) for $q > 0$ *are fulfilled. Denote* $M = \sup_{x\in\mathbb{R}^d}|f(x)|$.

Then there exists a constant $C(q) > 0$ *such that*

$$\|S_n - S\|_{\mathcal{L}} \leq \nu^{-1/2}\{\gamma_n^2(q)\kappa C_\gamma + n^{-1}(\kappa^2 - \|S\|_{\mathrm{HS}}^2)\}^{1/2},$$
$$\|T_n^* y - Sf\|_{\mathcal{H}} \leq \nu^{-1/2}\left[\gamma_n^2(q)MC_\gamma + n^{-1}\left\{\kappa(M+\sigma^2) - \|Sf\|_{\mathcal{H}}^2\right\}\right]^{1/2},$$

for $C_\gamma = C(q)\{(2\pi)^d\det(\Sigma)\}^{-1/2}\kappa d^{1/2}(1-4^{-q})^{-1/4(d+2)}$. *The function* $\gamma_n(q)$, $q > 0$, *is defined as*

$$\gamma_n(q) = \begin{cases} n^{-1/2} & , \quad q > 1 \\ n^{-1/2}\log(1/2n) & , \quad q = 1 \\ n^{-q/2} & , \quad q \in (0,1). \end{cases}$$

(iii) Let (K1), (K2) and (S) hold. Let $\gamma_n(q)$ *be the function as defined in (ii). Then there exists a constant* $\tilde{C}_\epsilon > 0$ *such that it holds with probability at least* $1-\nu$ *for* $\lambda > 0$ *that*

$$\|(S+\lambda)^{-1/2}(T_n^* y - S_n f\|_{\mathcal{H}} \leq \nu^{-1/2}\tilde{C}_\epsilon\sigma\sqrt{d_\lambda}\gamma_n(q).$$

(iv) Let (K1), (K2), (S), (D1) and (D2) hold. Let $\lambda_n^{-1/2}d_{\lambda_n}^{1/2}\gamma_n(q) \to 0$ *for a sequence* $\lambda_n \to 0$ *and* $\gamma_n(q)$ *the function defined in (ii). Then there exists an* $n_0 = n_0(\nu,q) \in \mathbb{N}$ *such that with probability at least* $1-\nu$ *we have for all* $n \geq n_0$

$$\|(S+\lambda_n)^{1/2}(S_n+\lambda_n)^{-1/2}\|_{\mathcal{L}} \leq \sqrt{2}.$$

The first part of the theorem is general and can be used to derive concentration inequalities not only in the Gaussian setting and is of interest in itself. The convergence rate is controlled by the sums appearing on the right hand side. If these sums are of $O(n)$, then the mean squared error of both S_n and $T_n^* y$ will converge to zero with a rate of n^{-1}. On the other hand, if the sums are of order $O(n^{2-q})$ for some $q \in (0,1)$, the mean squared errors will converge with the reduced rate n^{-q}.

The second part derives explicit concentration inequalities in the Gaussian setting described by (D1) and (D2) with rates depending on the range of the dependence measured by $q > 0$. These inequalities appear in Theorem 1.

Parts (iii) and (iv) give the additional probabilistic bounds needed to apply Theorem 2. The condition $\lambda_n^{-1/2}d_{\lambda_n}^{1/2}\gamma_n(q) \to 0$ in Theorem 5 (iv) is fulfilled in the settings of Corollary 3 and Corollary 4.

Theorem 1, Corollary 3, Corollary 4 and Theorem 5 together imply

Corollary 6 *Let the conditions of Theorem 2 and (D1), (D2) hold.*

(i) Assume that there exists $s \in (0,1]$ *such that* $d_\lambda = O(\lambda^{-s})$ *for* $\lambda \to 0$. *Then with probability at least* $1-\nu$

$$\|f_{\widehat{\alpha}_{a^*}} - f^*\|_2 = \begin{cases} O\{n^{-r/(2r+s)}\}, & q > 1, \\ O\{n^{-qr/(2r+s)}\}, & q \in (0,1). \end{cases}$$

If instead of conditions of Theorem 2, conditions of Theorem 1 are assumed, then the convergence rates above have $s = 1$.

(ii) Assume that there exists $a > 0$ *such that* $d_\lambda = O\{\log(1+a/\lambda)\}$ *for* $\lambda \to 0$ *and* $r = 1/2$. *Then with probability at least* $1-\nu$

$$\|f_{\widehat{\alpha}_{a^*}} - f^*\|_2 = \begin{cases} O\{n^{-1/2}\log(1/2n)\}, & q > 1, \\ O\{n^{-q/2}\log(1/2n^q)\}, & q \in (0,1). \end{cases}$$

Hence, for $q > 1$ the kernel partial least squares algorithm achieves the same rates as if the data were independent and identically distributed. For $q \in (0,1)$ the convergence rates become substantially slower, highlighting that dependence structures that persist over a long time can influence the convergence rates of the algorithm.

5. Simulations

To validate the theoretical results of the previous sections, we conducted a simulation study. The reproducing kernel Hilbert space is chosen to correspond to the Gaussian kernel $k(x,y) = \exp(-l\|x-y\|^2)$, $x,y \in \mathbb{R}^d$, $l = 2$, for $d = 1$. Our data will also be normally distributed. We refer to Proposition 7 in Appendix A for the derivation of functions that fulfill the source condition in this setting. Proposition 8 shows that the the eigenvalues of S decay exponentially fast. Hence the effective dimensionality d_λ behaves as in Corollary 4 and thus we expect convergence rates as given by Corollary 6 (ii).

The source parameter is taken to be $r = 4.5$ and we consider the function

$$f(x) = 4.37^{-1}\{3L_4(x,-4) - 2L_4(x,3) + 1.5L_4(x,9)\}, \quad x \in \mathbb{R}.$$

The normalization constant is chosen such that f takes values in $[-0.35, 0.65]$ and L_4 is the exponential function given in Proposition 7. The function f is shown in Figure 1.

In condition (D1) we set $\sigma_x^2 = \Sigma = 4$ (recall that $d = 1$). For the matrix $V = [\tau_{|i-j|}]_{i,j=1}^n \in \mathbb{R}^{n\times n}$ we choose three different structures for $n \in \{200, 400, 1000\}$. In the first setting $\tau_h = \mathbb{I}(h = 0)$, which corresponds to independent data. The second setting with $\tau_h = 0.9^{-h}$ implies an autoregressive process of order one. Finally, the third setting with $\tau_h = (1+h)^{-q}$, $q = 1/4$, $h = 0, \ldots, n-1$ leads to the long range dependent case.

In a Monte Carlo simulation with $M = 1000$ repetitions the time series $\{X_t^{(j)}\}_{t=1}^n$ are generated via $X^{(j)} = VN^{(j)}$ with $N^{(j)} \sim \mathcal{N}_n(0, \sigma^2 I_n)$, $j = 1, \ldots, M$, where I_n is the $n \times n$-dimensional identity matrix.

The residuals $\varepsilon_1^{(j)}, \ldots, \varepsilon_n^{(j)}$ are generated as independent standard normally distributed random variables and independent of $\{X_t^{(j)}\}_{t=1}^n$. The response is defined as $y_t^{(j)} = f(X_t^{(j)}) + \eta\varepsilon_t^{(j)}$, $t = 1, \ldots, n$, $j = 1, \ldots, M$, with $\eta = 1/16$.

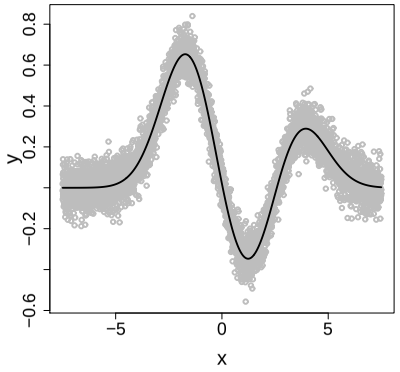


Figure 1: The function f evaluated on $[-7.5, 7.5]$ (black) and one realisation of the noisy data $y = f(x) + \varepsilon$ (grey).

The kernel partial least squares and kernel conjugate gradient algorithms are run for each sample $\{(X_t^{(j)}, y_t^{(j)})^{\mathrm{T}}\}_{t=1}^n$, $j = 1, \ldots, M$, with a maximum of 40 iteration steps. We denote the estimated coefficients with $\widehat{\alpha}_1^{(j,m)}, \ldots, \widehat{\alpha}_{40}^{(j,m)}$, $j = 1, \ldots, M$, with $m = CG$ meaning that the kernel conjugate gradient algorithm was employed and $m = PLS$ that kernel partial least squares was used to estimate $\alpha_1, \ldots, \alpha_n$.

The squared error in the $\mathcal{L}^2\left(\mathrm{P}^X\right)$-norm is calculated via

$$\widehat{e}_{n,\tau}^{(j,m)} = \min_{a=1,\ldots,40} \left[\frac{1}{\sqrt{2\pi\sigma_x^2}} \int_{-\infty}^{\infty} \left\{ f_{\widehat{\alpha}_a^{(j,m)}}(x) - f(x) \right\}^2 \exp\left(-\frac{1}{2\sigma_x^2} x^2 \right) \mathrm{d}x \right],$$

for $j = 1, \ldots, M$, $n = 200, 400, \ldots, 1000$ and $m \in \{CG, PLS\}$.

The results of the Monte-Carlo simulations are depicted in the boxplots of Figure 2. For kernel partial least squares (left panels) one observes that independent and autoregressive dependent data have roughly the same convergence rates, although the latter have a somewhat higher error. In contrast, the long range dependent data show slower convergence with the larger interquartile range, supporting the theoretical results of Corollary 6.

The $\mathcal{L}^2\left(\mathrm{P}^X\right)$-error of kernel conjugate gradient estimators is generally slightly higher than that of kernel partial least squares. Nonetheless, both of them have a similar behaviour.

We also investigated the the stopping indices $a = 1, \ldots, 40$ for which the errors $\widehat{e}_{n,\tau}^{(j,m)}$ were attained. These are shown in Figure 3 for independent and identically distributed data. It can be seen that the optimal indices for both algorithms have a rather similar behaviour. Kernel conjugate gradient stops slightly later, but overall the differences seem negligible.

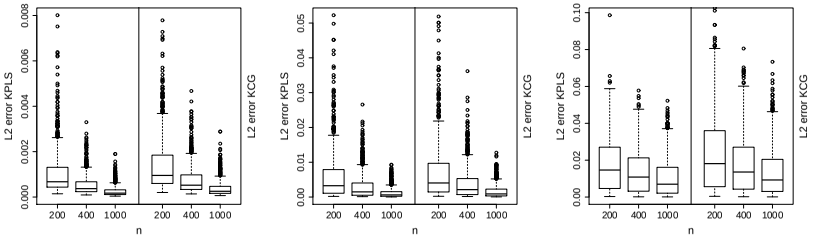


Figure 2: Boxplots of the $\mathcal{L}^2\left(\mathrm{P}^X\right)$-errors $\{\widehat{e}_{n,\tau}^{(j,m)}\}_{j=1}^M$ of kernel partial least squares (left side of each panel) and kernel conjugate gradient (right side of each panel) for different autocovariance functions τ and $n = 200, 400, 1000$. On the left is $\tau_h = \mathbb{I}(h = 0)$, in the middle $\tau_h = 0.9^{-h}$ and on the right $\tau_h = (h+1)^{-1/4}$.

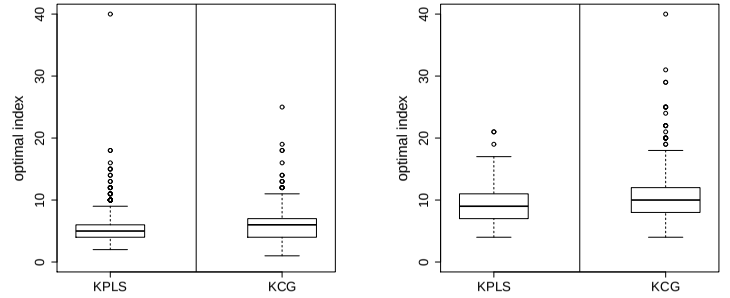


Figure 3: Boxplots of the optimal indices $a \in \{1, \ldots, 40\}$ for which the $\mathcal{L}^2\left(\mathrm{P}^X\right)$-errors $\{\widehat{e}_{n,\tau}^{(j,m)}\}_{j=1}^M$ were attained. Kernel partial least squares is on the left of each panel and kernel conjugate gradient on the right. On the left is $n = 200$, on the right $n = 1000$. The data were assumed to be independent and identically distributed.

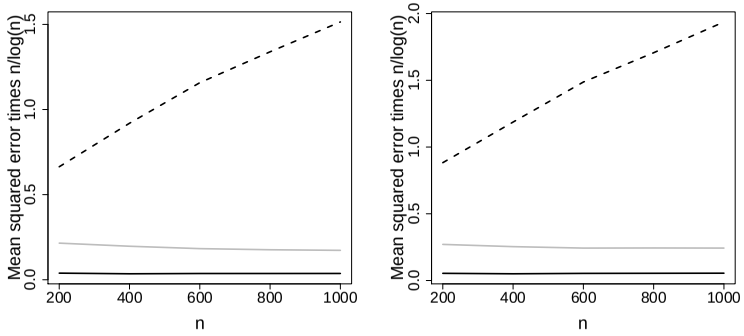


Figure 4: Mean of the $\mathcal{L}^2\left(\mathrm{P}^X\right)$-errors $\{\widehat{e}_{n,\tau}^{(j,m)}\}_{j=1}^M$ of kernel partial least squares (left) and kernel conjugate gradient (right) for $n = 200, 400, \ldots, 1000$ multiplied by $n/\log(n)$. The solid black line is for $\tau_h = \mathbb{I}(h = 0)$, the grey line for $\tau_h = 0.9^{-h}$ and the dashed black line for $\tau_h = (h+1)^{-1/4}$.

Figure 4 shows the mean (over j) of the estimated $\mathcal{L}^2\left(\mathrm{P}^X\right)$ errors $\{\widehat{e}_{n,\tau}^{(j,m)}\}_{j=1}^M$ for different n, τ and $m \in \{CG, PLS\}$. The errors were multiplied by $n/\log(n)$ to illustrate the convergence rates. According to Proposition 8 and Corollary 6 (ii) we expect the rates for the independent and autoregressive cases to be $n^{-1}\log(n)$, which is verified by the fact that the solid black and grey lines are roughly constant. For the long range dependent case we expect worse convergence rates which are also illustrated by the divergence of the dashed black line.

6. Application to Molecular Dynamics Simulations

The collective motions of protein atoms are responsible for its biological function and molecular dynamics simulations is a popular tool to explore this (Henzler-Wildman and Kern, 2007).

Typically, the $p \in \mathbb{N}$ backbone atoms of a protein are considered for the analysis with the relevant dynamics happening in time frames of nanoseconds. Although the dynamics are available exactly, the high dimensionality of the data and large number of observations can be cumbersome for regression analysis, e.g., due to the high collinearity in the columns of the covariates matrix. Many function-dynamic relationships are also non-linear (Hub and de Groot, 2009). A further complication is the fact that the motions of different backbone atoms are highly correlated, making additive non-parametric models for the target function f^* less suitable.

We consider T4 Lysozyme (T4L) of the bacteriophage T4, a protein responsible for the hydrolisis of 1,4-beta-linkages in peptidoglycans and chitodextrins from bacterial cell walls.

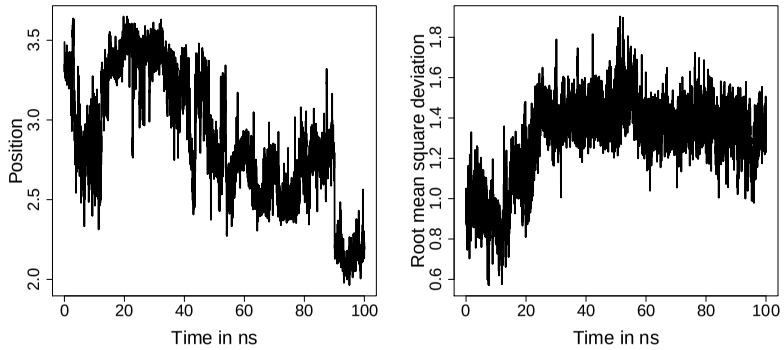


Figure 5: Time series of $X_{t,1}$, i.e., the first coordinate of the first atom T4L consists of (left) and the root mean squared deviation y_t between the protein configuration at time t and the (apo) crystal structure.

The number of available observations is $n = 4601$ and T4L consists of $p = 486$ backbone atoms.

Denote with $A_{t,i} \in \mathbb{R}^3$ the ith atom, $i = 1, \ldots, p$, at time $t = 1, \ldots, n$ and $c_i \in \mathbb{R}^3$ the ith atom in the (apo) crystal structure of T4L. A usual representation of the protein in a regression setting is the Cartesian one, i.e., we take as the covariate $X_t = (A_{1,t}^{\mathrm{T}}, \ldots, A_{p,t}^{\mathrm{T}})^{\mathrm{T}}$, $t = 1, \ldots, n$, see Brooks and Karplus (1983). The functional quantity to predict is the root mean square deviation of the protein configuration X_t at time $t = 1, \ldots, n$ from the (apo) crystal structure $C = (c_1^{\mathrm{T}}, \ldots, c_d^{\mathrm{T}})^{\mathrm{T}}$, i.e.,

$$y_t = \left\{ p^{-1} \sum_{i=1}^{p} \|X_{i,t} - C_i\|^2 \right\}^{1/2}.$$

This nonlinear function was previously considered in Hub and de Groot (2009), where it was established that linear models are insufficient for the prediction.

Figure 5 shows the time series corresponding to $X_{t,1}$ (i.e., the first coordinate of the first atom of T4L) on the left and the functional quantity y_t on the right. These plots reveal certain persistent dependence over time.

Fitting autoregressive moving average models of order $(3, 2)$ ($ARMA(3, 2)$) to y_t and $ARMA(5, 2)$ to $X_{t,1}$ shows that the smallest root of their respective characteristic polynomial is close to one (1.009 for y_t and 1.003 for $X_{t,1}$), highlighting that we are on the border of stationarity, see, e.g., Brockwell and Davis (1991).

Next, we performed an augmented Dickey Fuller test for the null hypothesis of non-stationarity against stationarity. Calculating the test statistics with 16 lags yields a p-value smaller than 0.01 for y_t and 0.0122 for $X_{t,1}$. Hence, both time series are likely stationary.

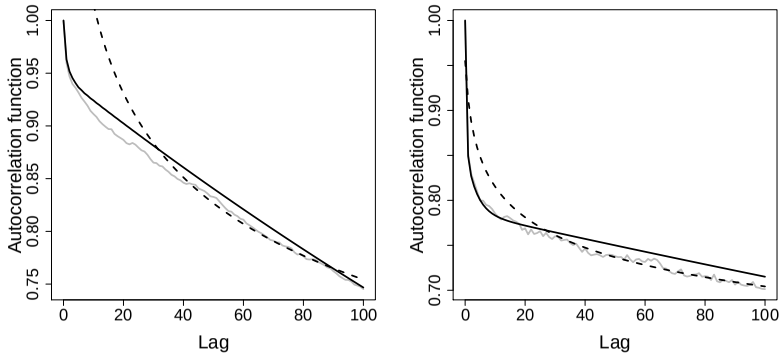


Figure 6: Autocorrelation plots of $X_{t,1}$ (left) and y_t (right). The estimated autocorrelation function is grey, the theoretical one of a fitted $ARMA(3,2)$ process is solid black and $\rho_h \propto (h+1)^{-q}$ for a suitable choice of $q > 0$ is dashed black.

Finally, to test for long range dependence, we employed the rescaled variance test of Giraitis et al. (2003). The null hypothesis of this test is the short range dependence in the data, while the alternative is the long range dependence. Calculating the test statistics with 16 lags gives the p-value for both y_t and $X_{t,1}$ smaller than 0.01, suggesting the long range dependence.

Figure 6 depicts the autocorrelation functions of $X_{t,1}$ and y_t, the theoretical autocorrelation function of the corresponding autoregressive moving average process and $\rho_h \propto (h+1)^{-q}$ for $q = 0.134$ for $X_{t,1}$ and $q = 0.066$ for y_t. The latter, as highlighted in Section 4, is an autocorrelation function for a stationary long range dependent process. These plots together with the above findings suggest that $X_{t,1}$ and y_t are stationary, long range dependent processes.

We apply kernel partial least squares to this data set with the Gaussian kernel $k(x,y) = \exp(-l\|x-y\|^2)$, $x, y \in \mathbb{R}^{3p}$, $l > 0$. The function f we aim to estimate is a distance between protein configurations, so using a distance based kernel seems reasonable. Moreover, we also investigated the impact of other bounded kernels such as triangular and Epanechnikov and obtained similar results. The first 50% of the data form a training set to calculate the kernel partial least squares estimator and the remaining data are used for testing.

The parameter $l > 0$ is calculated via cross validation on the training set. In our evaluation we obtained $l = 10.22$.

Figure 7 compares the observed response in the test set with the prediction on the test set obtained by kernel partial least squares, kernel principal component regression and linear partial least squares. Apparently, kernel partial least squares show the best performance and the kernel principal components algorithm is able to achieve comparable prediction with

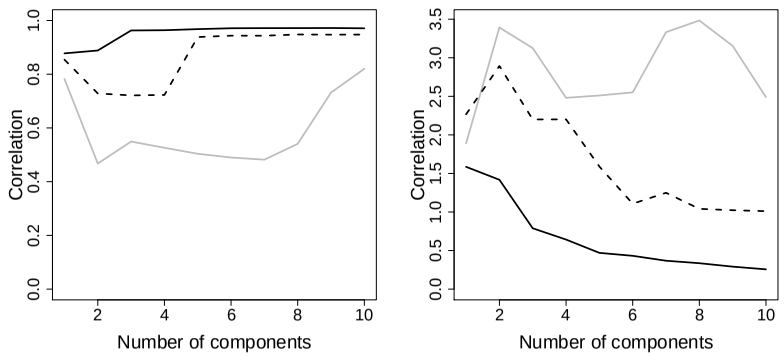


Figure 7: Correlation (left) and residual sum of squares (right) between predicted values and the observed response on the test set depending on the number of used components for kernel partial least squares (solid black), partial least squares (grey) and kernel principal component regression (dashed black).

more components. Obviously, linear partial least squares can not cope with the non-linearity of the problem.

This application highlights that kernel partial least squares still deliver a robust prediction even for long range dependent data, if enough observations are available.

Acknowledgements

The authors are grateful to the action editor and two reviewers for their valuable comments which helped to improve the article. The support of the German Research Foundation via FOR 916 (Project B5) and CRC 803 (Proejct Z2) is gratefully acknowledged.

Appendix A. Source Condition and Effective Dimensionality for Gaussian Kernels

The source condition (S) and the effective dimensionality d_λ are of great importance in the convergence rates derived in Section 3. Here we investigate these conditions for the reproducing kernel Hilbert space corresponding to the Gaussian kernel $k(x,y) = \exp(-l\|x-y\|^2)$, $x, y \in \mathbb{R}^d$, $l > 0$, for $d = 1$. Hence, the space $\mathcal{H}$ is the space of all analytic functions that decay exponentially fast, see Steinwart et al. (2005).

We also impose the normality conditions (D1) and (D2) on $\{X_t\}_{t\in\mathbb{Z}}$, where now $\sigma_x^2 = \Sigma \in \mathbb{R}$ due to $d = 1$. The following proposition derives a more explicit representation for $f \in \mathcal{H}$.

Proposition 7 *Assume that (K1),(K2) and (S) hold for $r \geq 1/2$. Let $d = 1$, $X_0 \sim \mathcal{N}(0, \sigma_x^2), \sigma_x^2 > 0$ and consider the Gaussian kernel $k(x,y) = \exp\{-l(x-y)^2\}$ for $x, y \in \mathbb{R}$, $l > 0$. Then f can be expressed for $\mu = r - 1/2 \in \mathbb{N}$ via $f(x) = \sum_{i=1}^{\infty} c_i L_\mu(x, z_i)$ for fixed $\{z_i\}_{i=1}^{\infty}, \{c_i\}_{i=1}^{\infty} \subset \mathbb{R}$ such that $\sum_{i,j=1}^{\infty} c_i c_j k(z_i, z_j) \leq R^2$, $R > 0$. Here we have for $x, z \in \mathbb{R}$*

$$L_\mu(x,z) = \exp\left[-1/2\left\{\frac{\det(\Lambda)(x^2+z^2) - 2l^{\mu+1}xz}{\det(\Lambda_{1:\mu})}\right\}\right],$$

with $\Lambda \in \mathbb{R}^{(\mu+1)\times(\mu+1)}$ being a tridiagonal matrix with elements

$$\Lambda_{i,j} = \begin{cases} \sigma_x^{-2} + 2l & , \quad i = j < \mu + 1 \\ l & , \quad i = j = \mu + 1 \\ -l & , \quad |i-j| = 1 \\ 0 & , \quad else \end{cases}$$

for $i, j = 1, \ldots, \mu + 1$ and $\Lambda_{1:\mu}$ is the $\mu \times \mu$-dimensional sub-matrix of Λ including the first μ columns and rows.

Conversely any function $f^* = Tf$ with f of the above form fulfills a source condition (S) with $r = \mu + 1/2$, $\mu \in \mathbb{N}$.

Hence if we fix an $r \geq 1/2$ with $r - 1/2 \in \mathbb{N}$ this theorem gives us a way to construct functions $f \in \mathcal{H}$ with $f^* = Tf$ that fulfill (S).

The next proposition derives the effective dimensionality d_λ in this setting:

Proposition 8 *Let $d = 1$, $X_0 \sim \mathcal{N}(0, \sigma_x^2)$ for some $\sigma_x^2 > 0$ and consider the Gaussian kernel $k(x,y) = \exp\{-l(x-y)^2\}$, $x, y \in \mathbb{R}$, $l > 0$.*

Then there is a constant $D > 0$ such that it holds for any $\lambda \in (0, 1]$

$$d_\lambda = \mathrm{tr}\{(S+\lambda)^{-1}S\} \leq D \log(1 + a/\lambda),$$

with $a = \sqrt{2}(1 + \beta + \sqrt{1+\beta})^{-1/2}$, $\beta = 4l\sigma_x^2$.

With the latter result Corollary 4 is applicable and we expect convergence rates for the kernel partial least squares algorithm of order $O\{\gamma_n \log(1/2\gamma_n^{-2})\}$ for a sequence $\{\gamma_n\}_n$ as in Theorem 2.

Appendix B. Proofs

B.1 Proof of Theorem 1

The proof of Theorem 1 makes use of the connection between the partial least squares and the conjugate gradient algorithm. This section is structured as follows: First we will introduce the link between kernel partial least squares and kernel conjugate gradient. We will state some key facts about orthogonal polynomials and their relationship to the algorithm in Lemma 9. Then the consistency of kernel partial least squares is shown with the help of three error bounds that are obtained in Lemmas 11 — 13.

With a slight abuse of notation we define $f_i = f_{\widehat{\alpha}_i}$ for $i = 1, \ldots, n$.

B.1.1 Orthogonal Polynomials and Some Notation

Denote with $\mathcal{P}_i$ the set of polynomials of degree at most $i = 0, \ldots, n$. For functions $\psi, \phi : \mathbb{R} \to \mathbb{R}$ and $r \in \mathbb{N}_0$ define the inner products $[\psi, \phi]_r = \langle \psi(S_n) T_n^* y, S_n^r \phi(S_n) T_n^* y \rangle_{\mathcal{H}}$. From the definition of the Krylov space it is immediate that every element $v \in \mathcal{K}_i(S_n, T_n^* y)$, $i = 1, \ldots, n$, can be represented by a polynomial $q \in \mathcal{P}_{i-1}$ via $v = q(S_n) T_n^* y$.

The following discussion is based on Hanke (1995), chapter 2. There exist two sequences of polynomials $\{p_i\}_{i=0}^n, \{q_i\}_{i=0}^n \subset \mathcal{P}_i$, such that $f_i = q_{i-1}(S_n) T_n^* y$ with $q_{-1} = 0$ and $T_n^* y - S_n f_i = p_i(S_n) T_n^* y$. Both sequences are connected by the equation $p_i(x) = 1 - x q_{i-1}(x)$, $x \in \mathbb{R}$, and the polynomials $\{p_i\}_{i=0}^n$ are orthogonal with respect to $[\cdot, \cdot]_0$.

We will also consider other sequences of polynomials, namely $\{q_i^{[r]}\}_{i=0}^n, \{p_i^{[r]}\}_{i=0}^n \subset \mathcal{P}_i$, $q_{-1}^{[r]} = 0$, such that $p_i^{[r]}(x) = 1 - x q_{i-1}^{[r]}(x)$, $x \in \mathbb{R}$, and the sequence $\{p_i^{[r]}\}_{i=0}^n$ is orthogonal with respect to $[\cdot, \cdot]_r$. This yields for every $r \in \mathbb{N}_0$ a separate conjugate gradient algorithm with solution $f_i^{[r]} = q_{i-1}^{[r]}(S_n) T_n^* y \in \mathcal{K}_i(S_n, T_n^* y)$ and residuals $T_n^* y - S_n f_i^{[r]} = p_i^{[r]}(S_n) T_n^* y$, $i = 1, \ldots, n$. The $p_i^{[r]}$, $i = 0, \ldots, n$, $r \in \mathbb{N}_0$, are called residual polynomials.

As S_n is self-adjoint, positive semi-definite and the kernel is bounded by κ we know that its spectrum is a subset of $[0, \kappa]$, see Caponnetto and de Vito (2007). This also implies that $\max\{\|S\|_{\mathcal{L}}, \|S_n\|_{\mathcal{L}}\} \leq \kappa$, with $\|\cdot\|_{\mathcal{L}}$ denoting the operator norm. The i distinct roots of $p_i^{[r]}$ will be denoted by $0 < x_{1,i}^{[r]} < \ldots x_{i,i}^{[r]} < \kappa$, $i = 1, \ldots, n$.

We will summarize some key facts about the orthogonal polynomials in the next lemma.

Lemma 9 *Let $r, s \in \mathbb{N}_0$ and $i = 1, \ldots, n$. Then we have:*

(i) The roots of consecutive orthogonal polynomials interlace, i.e., for $j = 1, \ldots, i$ it holds

$$0 < x_{j,i+1}^{[r]} < x_{j,i}^{[r]} < x_{j,i}^{[r+1]} < x_{j+1,i+1}^{[r]} < x_{j+1,i}^{[r]} < \cdots < x_{i,i}^{[r+1]} < x_{i+1,i+1}^{[r]} < \kappa,$$

(ii) the optimality property $[p_i^{[1]}, p_i^{[1]}]_0^{1/2} = \|T_n^* y - S_n f_i^{[1]}\|_{\mathcal{H}} \leq \|T_n^* y - S_n h\|_{\mathcal{H}}$ holds for all $h \in \mathcal{K}_i(S_n, T_n^* y)$,

(iii) on $x \in [0, x_{1,i}^{[r]}]$ it holds $0 \leq p_i^{[r]}(x) \leq 1$ and $q_i^{[r]}(x) \leq \left|\left(p_i^{[r]}\right)'(0)\right|$,

(iv) $p_n^{[r]} = p_n^{[s]}$,

(v) $\left(p_i^{[r]}\right)'(0) = -\sum_{j=1}^{i} \left(x_{j,i}^{[r]}\right)^{-1}$,

(vi) for $r \geq 1$ define $\phi_i(x) = p_i^{[r]}(x) \left(x_{1,i}^{[r]}\right)^{1/2} \left(x_{1,i}^{[r]} - x\right)^{-1/2}$, $x \in [0, x_{1,i}^{[r]}]$, $i = 1, \ldots, n$. Then it holds for $u \geq 0$ that $x^u \phi_i^2(x) \leq u^u \left|\left(p_i^{[r]}\right)'(0)\right|^{-u}$ with the convention $0^0 = 1$.

Proof: (i) See Hanke (1995), Corollary 2.7.

(ii) See Hanke (1995), Proposition 2.1.

(iii) Due to part (i) we know that all i roots of the polynomial $p_i^{[r]}$ are contained in $(0, \kappa)$. Furthermore $p_i^{[r]}(0) = 1 - 0 q_i^{[r]} = 1$. Thus $p_i^{[r]}$ is convex and falling in $[0, x_{1,i}^{[r]}]$ and the first assertion follows.

Because of the convexity of $p_i^{[r]}$ on $[0, x_{1,i}^{[r]}]$ we get $q_i^{[r]}(x) = x^{-1}\{1-p_i^{[r]}(x)\} \leq \left|\left(p_i^{[r]}\right)'(0)\right|$.

(iv) See the discussion in Hanke (1995) preceding Proposition 2.1 and use the facts that $T_n^* y \in \text{range}(S_n)$ and S_n is an operator of rank n.

(v) Write $p_i^{[r]}(x) = \prod_{j=1}^{i}(1 - x/x_{j,i}^{[r]})$, $x \in [0, \kappa]$, and the result is immediate.

(vi) See equation (3.10) in Hanke (1995). ■

We denote for $x \geq 0$ by P_x the orthogonal projection operator on the eigenspace corresponding to the eigenvalues of S_n that are smaller or equal x and $P_x^{\perp} = I_{\mathcal{H}} - P_x$ with $I_{\mathcal{H}} : \mathcal{H} \to \mathcal{H}$ being the identity operator.

B.1.2 Preparation for the Proof

We consider the kernel partial least squares algorithm as an optimization problem

$$f_i = \arg \min_{g \in \mathcal{K}_i(S_n, T_n^* y)} n^{-1}\|y - T_n g\|^2, \quad i = 1, \ldots, n. \tag{8}$$

This is the conjugate gradient algorithm CGNE discussed in chapter 2.2 of Hanke (1995), to which we refer for more details. Note that, for example,

$$n^{-1}\langle \phi(T_n T_n^*)y, T_n T_n^* \psi(T_n T_n^*)y\rangle = \langle \phi(S_n)T_n^* y, \psi(S_n)T_n^* y\rangle_{\mathcal{H}} = [\phi, \psi]_0,$$

for polynomials ϕ, ψ.

In the upcoming proofs we will make use of the following operator inequality:

Lemma 10 *Let $B, C : \mathcal{H} \to \mathcal{H}$ be two positive semi-definite, self-adjoint operators with $\max\{\|B\|_{\mathcal{L}}, \|C\|_{\mathcal{L}}\} \leq \kappa$. Then it holds for any $r \geq 0$ with $\zeta = \max\{r - 1, 0\}$*

$$\|B^r - C^r\|_{\mathcal{L}} \leq (\zeta + 1)\kappa^{\zeta}\|B - C\|_{\mathcal{L}}^{r-\zeta}.$$

Proof: See Blanchard and Krämer (2010b), Lemma A.6. ■

For the remainder of the proof we assume that we are on the set where it holds with probability at least $1-\nu$, $\nu \in (0, 1]$, that $\|S_n - S\|_{\mathcal{L}} \leq C_\delta(\nu)\gamma_n$ and $\|T_n^* y - Sf\|_{\mathcal{H}} \leq C_\epsilon(\nu)\gamma_n$ for a sequence $\{\gamma_n\}_n$ converging to zero and constants $C_\delta = C_\delta(\nu), C_\epsilon = C_\epsilon(\nu) > 0$.

With Lemma 2.4 in Hanke (1995) we see that the stopping iteration (5) can also be expressed as

$$a^* = \min\left\{1 \leq a \leq n : \|S_n f_a^{[1]} - T_n^* y\|_{\mathcal{H}} \leq C\gamma_n\right\}, \tag{9}$$

i.e., we stop the kernel partial least squares algorithm when a discrepancy principle for $f_a^{[1]}$ holds.

It is easy to see that from (S) it follows for $r \geq 1/2$ that

(SH) There exist $\mu \geq 0$, $R > 0$ and $u \in \mathcal{H}$ such that $f = S^{\mu}u$ and $\|u\|_{\mathcal{H}} \leq R$.

This condition is known as the Hölder source condition with $\mu = r - 1/2$.

Recall that $\mathcal{H} \subseteq \mathcal{L}^2\left(\mathrm{P}^X\right)$ and $T : \mathcal{H} \to \mathcal{L}^2\left(\mathrm{P}^X\right)$ is the change of space operator. Using the fact that T, T^* are adjoint operators, $f_{a^*} = Tf_{a^*}$ and $f^* = Tf$ for $r \geq 1/2$ we see

$$\|f_{a^*} - f^*\|_2 = \|T(f_{a^*} - f)\|_2 = \langle S(f_{a^*} - f), f_{a^*} - f\rangle_{\mathcal{H}} = \|S^{1/2}(f_{a^*} - f)\|_{\mathcal{H}}.$$

An application of Lemma 10 yields

$$\begin{aligned}\|f_{a^*} - f^*\|_2 &= \|S^{1/2}(f_{a^*} - f)\|_{\mathcal{H}} \leq \|S^{1/2}(f_{a^*} - f_{a^*}^{[1]})\|_{\mathcal{H}} + \|S^{1/2}(f_{a^*}^{[1]} - f)\|_{\mathcal{H}} \\ &\leq C_\delta^{1/2}\gamma_n^{1/2}\left(\|f_{a^*} - f_{a^*}^{[1]}\|_{\mathcal{H}} + \|f_{a^*}^{[1]} - f\|_{\mathcal{H}}\right) + \|S_n^{1/2}(f_{a^*} - f_{a^*}^{[1]})\|_{\mathcal{H}} + \|S_n^{1/2}(f_{a^*}^{[1]} - f)\|_{\mathcal{H}}.\end{aligned} \tag{10}$$

The following lemmas will deal with bounding the quantities in (10) in terms of the source parameter $r = \mu + 1/2 \geq 1/2$ and the sequence γ_n. First we will derive upper bounds for the quantities containing the difference of the KPLS estimator f_{a^*} and the estimator $f_{a^*}^{[1]}$:

Lemma 11 *Assume $C_x \in (0, 1]$ such that $x_* = (C_x\gamma_n)^{1/(\mu+1)} < x_{1,a^*-1}^{[1]}$ and $C > C_\epsilon + C_x R + C_\delta(\mu + 1)\kappa^\mu R$. Under the conditions of the theorem it holds $\mu \geq 0$*

$$\begin{aligned}\|f_{a^*} - f_{a^*}^{[1]}\|_{\mathcal{H}} &\leq \gamma_n^{\mu/(\mu+1)} \frac{C}{C_x^{1/(\mu+1)}\left[1 - C^{-1}\{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R\}\right]^2} \\ \|S_n^{1/2}(f_{a^*} - f_{a^*}^{[1]})\|_{\mathcal{H}} &\leq \gamma_n^{(2\mu+1)/(2\mu+2)} \frac{C}{C_x^{1/(2\mu+2)}\left[1 - C^{-1}\{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R\}\right]}.\end{aligned}$$

Proof: If the inner products $[\cdot, \cdot]_0$ and $[\cdot, \cdot]_1$ are the same the proof is done because both polynomial sequences are identical.

We now observe that we have for $a^* = n$ due to Lemma 9 (iv) $q_{n-1}(x) - q_{n-1}^{[1]}(x) = x^{-1}\{p_n^{[1]}(x) - p_n(x)\} = 0$, i.e., $\|f_{a^*} - f_{a^*}^{[1]}\|_{\mathcal{H}} = 0$ and $\|S_n^{1/2}(f_{a^*} - f_{a^*}^{[1]})\|_{\mathcal{H}} = 0$ and the proof is done.

If the inner products differ and we have $0 < a^* < n$ it holds $f_{a^*} \neq f_{a^*}^{[1]}$.

Proposition 2.8 in Hanke (1995) can now be applied for $0 < a^* < n$ and yields $q_{a^*-1}(x) - q_{a^*-1}^{[1]}(x) = x^{-1}\{p_{a^*}^{[1]}(x) - p_{a^*}(x)\} = \theta_{a^*}p_{a^*-1}^{[2]}(x)$, $x \geq 0$, with $\theta_{a^*} = (p_{a^*}^{[1]})'(0) - (p_{a^*}^{[0]})'(0) > 0$. We get $f_{a^*} - f_{a^*}^{[1]} = q_{a^*-1}(S_n)T_n^* y - q_{a^*-1}^{[1]}(S_n)T_n^* y = \theta_{a^*}p_{a^*-1}^{[2]}(S_n)T_n^* y$.

Proposition 2.9 in Hanke (1995) yields $\theta_{a^*} = \left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1^{-1}\left[p_{a^*}^{[1]}, p_{a^*}^{[1]}\right]_0$. The optimality property of $f_{a^*}^{[1]}$ in Lemma 9 (ii) shows that

$$\|T_n^* y - S_n f_{a^*}^{[1]}\|_{\mathcal{H}} = \|p_{a^*}^{[1]}(S_n)T_n^* y\|_{\mathcal{H}} = \left[p_{a^*}^{[1]}, p_{a^*}^{[1]}\right]_0^{1/2} \leq \left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_0^{1/2}. \tag{11}$$

Combining these results yields

$$\|f_{a^*} - f_{a^*}^{[1]}\|_{\mathcal{H}} = \frac{\left[p_{a^*}^{[1]}, p_{a^*}^{[1]}\right]_0}{\left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1}\left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_0^{1/2} \leq \frac{\left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_0}{\left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1}\|p_{a^*}^{[1]}(S_n)T_n^* y\|_{\mathcal{H}}. \tag{12}$$

Recall that $x^{[2]}_{1,a^*-1}$ denotes the first root of $p^{[2]}_{a^*-1}$. It holds for any $0 \le x \le x^{[2]}_{1,a^*-1}$ that $0 \le p^{[2]}_{a^*-1}(x) \le 1$, see Lemma 9 (iii), and thus

$$\begin{aligned}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2} &\le \|P_x p^{[2]}_{a^*-1}(S_n)\{T_n^* y - Sf + Sf\}\|_{\mathcal{H}} + \|P_x^{\perp} p^{[2]}_{a^*-1}(S_n) T_n^* y\|_{\mathcal{H}} \\ &\le C_\epsilon \gamma_n + \|P_x p^{[2]}_{a^*-1}(S_n) S^{\mu+1} u\|_{\mathcal{H}} + x^{-1/2}\|P_x^{\perp} S_n^{1/2} p^{[2]}_{a^*-1}(S_n) T_n^* y\|_{\mathcal{H}} \\ &\le C_\epsilon \gamma_n + x^{\mu+1} R + \|P_x p^{[2]}_{a^*-1}(S^{\mu+1} - S_n^{\mu+1}) u\|_{\mathcal{H}} + \frac{1}{\sqrt{x}}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2}.\end{aligned}$$

In the second inequality (SH) with $\mu \ge 0$ was applied.

By assumption $x_* = (C_x\gamma)^{1/(\mu+1)} \le x^{[1]}_{1,a^*-1} < x^{[2]}_{1,a^*-1}$ due to the interlacing property of the roots of the polynomials $p_i^{[r]}$, $i = 1, \dots, n$, $r \in \mathbb{N}_0$, see Lemma 9 (i).

Using Lemma 10 we get $\|S^{\mu+1} - S_n^{\mu+1}\|_{\mathcal{L}} \le (\mu+1)\kappa^\mu C_\delta \gamma_n$ and setting $x = x_*$ we get

$$\begin{aligned}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2} &\le C_\epsilon\gamma_n + x_*^{\mu+1} R + C_\delta \gamma_n (\mu+1)\kappa^\mu R + x_*^{-1/2}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2} \\ &= \gamma_n\left\{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R\right\} + x_*^{-1/2}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2}. \qquad (13)\end{aligned}$$

Due to (9) and (11) we have additionally $C\gamma_n \le \|S_n f^{[1]}_{a^*-1} - T_n^* y\|_{\mathcal{H}} = \|p^{[1]}_{a^*-1}(S_n) T_n^* y\|_{\mathcal{H}} \le \left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2}$.

Plugging this into (13) yields

$$\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2} \le \frac{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R}{C}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2} + \frac{1}{\sqrt{x_*}}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2},$$

or equivalently with $x_* = (C_x\gamma_n)^{1/(\mu+1)}$

$$\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2} \le \gamma_n^{-1/(2\mu+2)} \frac{\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2}}{C_x^{1/(2\mu+2)}\left[1 - C^{-1}\{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R\}\right]}, \qquad (14)$$

where by assumption $C > C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R$ and $x_* = (C_x\gamma)^{1/(\mu+1)}$.

Combining (12), (14) and $\|p^{[1]}_{a^*}(S_n) T_n^* y\|_{\mathcal{H}} \le C\gamma_n$ due to the stopping index (9) yields

$$\begin{aligned}\|f_{a^*} - f^{[1]}_{a^*}\|_{\mathcal{H}} &\le \gamma_n^{-1/(\mu+1)} \frac{\|p^{[1]}_{a^*}(S_n) T_n^* y\|_{\mathcal{H}}}{C_x^{1/(\mu+1)}\left[1 - C^{-1}\{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R\}\right]^2} \\ &\le \gamma_n^{\mu/(\mu+1)} \frac{C}{C_x^{1/(\mu+1)}\left[1 - C^{-1}\{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R\}\right]^2}.\end{aligned}$$

For the second part of the proof we derive in the same way as (12)

$$\|S_n^{1/2}(f_{a^*} - f^{[1]}_{a^*})\|_{\mathcal{H}} \le \frac{\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2}}{\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2}} \|p^{[1]}_{a^*}(S_n) T_n^* y\|_{\mathcal{H}}.$$

Using (14) and $\|p^{[1]}_{a^*}(S_n) T_n^* y\|_{\mathcal{H}} \le C\gamma_n$ gives

$$\|S_n^{1/2}(f_{a^*} - f^{[1]}_{a^*})\|_{\mathcal{H}} \le \gamma_n^{(2\mu+1)/(2\mu+2)} \frac{C}{C_x^{1/(2\mu+2)}\left[1 - C^{-1}\{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R\}\right]},$$

finishing the proof. ■

We now derive an upper bound on the quantities that contain $f^{[1]}_{a^*}$ and f. These contain $\|S_n f_i^{[1]} - T_n^* y\|_{\mathcal{H}}$, which can be controlled for $i = a^*$ by the discrepancy principle (9), and $(p_i^{[1]})'(0)$, for which we have to derive a separate bound later on.

Lemma 12 *For any $i = 1, \dots, n$ and any $0 < x \le x^{[1]}_{1,i}$ we have under the conditions of the theorem for $\mu \ge 1$*

$$\begin{aligned}\|f - f_i^{[1]}\|_{\mathcal{H}} &\le R\left\{x^\mu + C_\delta \mu \kappa^{\mu-1}\gamma_n\right\} + x^{-1}\left\{\|S_n f_i^{[1]} - T_n^* y\|_{\mathcal{H}} + (C_\epsilon + C_\delta\kappa^\mu R)\gamma_n\right\} \\ &\quad + (C_\epsilon + C_\delta\kappa^\mu R)\gamma_n |(p_i^{[1]})'(0)|, \\ \|S_n^{1/2}(f - f_i^{[1]})\|_{\mathcal{H}} &\le R\left\{x^{\mu+1/2} + x^{1/2} C_\delta \mu \kappa^{\mu-1}\gamma_n\right\} \\ &\quad + x^{-1/2}\left\{\|S_n f_i^{[1]} - T_n^* y\|_{\mathcal{H}} + (C_\epsilon + C_\delta\kappa^\mu R)\gamma_n\right\} \\ &\quad + x^{1/2}(C_\epsilon + C_\delta\kappa^\mu R)\gamma_n |(p_i^{[1]})'(0)|.\end{aligned}$$

Proof: Denote $\bar{f}_i = q^{[1]}_{i-1}(S_n) S_n f$ and consider

$$\|f - f_i^{[1]}\|_{\mathcal{H}} \le \|P_x(f - \bar{f}_i)\|_{\mathcal{H}} + \|P_x(\bar{f}_i - f_i^{[1]})\|_{\mathcal{H}} + \|P_x^{\perp}(f - f_i^{[1]})\|_{\mathcal{H}}. \qquad (15)$$

The first term of (15) can be bound by an application of Lemma 10 and (SH) with $\mu \ge 1$

$$\begin{aligned}\|P_x(f - \bar{f}_i)\|_{\mathcal{H}} &= \|P_x\{I - q^{[1]}_{i-1}(S_n) S_n\} f\|_{\mathcal{H}} = \|P_x p_i^{[1]}(S_n) f\|_{\mathcal{H}} = \|P_x p_i^{[1]}(S_n) S^\mu u\|_{\mathcal{H}} \\ &\le \|P_x p_i^{[1]}(S_n) S_n^\mu u\|_{\mathcal{H}} + \|P_x p_i^{[1]}(S_n)(S^\mu - S_n^\mu) u\|_{\mathcal{H}} \\ &\le R\left\{x^\mu + C_\delta \mu \kappa^{\mu-1}\gamma_n\right\}.\end{aligned}$$

In the last inequality we used that on $0 \le x \le x^{[1]}_{1,i}$ we have $0 \le p_i^{[1]}(x) \le 1$.

For the second term of (15) we use Lemma 9 (iii) $q_i^{[1]}(x) \le |(p_i^{[1]})'(0)|$ on $x \in [0, x^{[1]}_{1,i}]$. This yields

$$\begin{aligned}\|P_x(f_i^{[1]} - \bar{f}_i)\|_{\mathcal{H}} &= \|P_x q_i^{[1]}(S_n)(S_n f - T_n^* y)\|_{\mathcal{H}} \\ &\le \|P_x q_i^{[1]}(S_n)(Sf - T_n^* y)\|_{\mathcal{H}} + \|P_x q_i^{[1]}(S_n)(S_n - S) f\|_{\mathcal{H}} \\ &\le (C_\epsilon + C_\delta\kappa^\mu R)\gamma_n \left|\left(p_i^{[1]}\right)'(0)\right|.\end{aligned}$$

Finally, we have

$$\begin{aligned}\|P_x^{\perp}(f - f_i^{[1]})\|_{\mathcal{H}} &\le x^{-1}\|P_x^{\perp} S_n (f - f_i^{[1]})\|_{\mathcal{H}} \\ &\le x^{-1}\left\{\|S_n f_i^{[1]} - T_n^* y\|_{\mathcal{H}} + \|P_x(T_n^* y - S_n f)\|_{\mathcal{H}}\right\} \\ &\le x^{-1}\left\{\|S_n f_i^{[1]} - T_n^* y\|_{\mathcal{H}} + (C_\epsilon + C_\delta\kappa^\mu R)\gamma_n\right\}\end{aligned}$$

and thus the first inequality is proven.

For the second inequality we use

$$\|S_n^{1/2}(f-f_i^{[1]})\|_{\mathcal{H}} \le \|P_x S_n^{1/2}(f-\bar{f}_i)\|_{\mathcal{H}} + \|P_x S_n^{1/2}(\bar{f}_i - f_i^{[1]})\|_{\mathcal{H}} + \|P_x^{\perp} S_n^{1/2}(f-f_i^{[1]})\|_{\mathcal{H}}.$$

In the same way as before we derive bounds for the three terms:

$$\|P_x S_n^{1/2}(f-\bar{f}_i)\|_{\mathcal{H}} \le x^{1/2} C_\delta \mu \kappa^{\mu-1} R\gamma_n + x^{\mu+1/2} R,$$

$$\|P_x S_n^{1/2}(\bar{f}_i - f_i^{[1]})\|_{\mathcal{H}} \le x^{1/2}(C_\epsilon + C_\delta R\kappa^\mu)\gamma_n \left|\left(p_i^{[1]}\right)'(0)\right|,$$

$$\|P_x^{\perp} S_n^{1/2}(f-f_i^{[1]})\|_{\mathcal{H}} \le x^{-1/2}\{\|S_n f_i^{[1]} - T_n^* y\|_{\mathcal{H}} + (C_\epsilon + C_\delta \kappa^\mu R)\gamma_n\},$$

completing the proof. ■

Lemma 12 depends on $(p_i^{[1]})'(0)$ and hence an upper bound for this term is needed:

Lemma 13 *Assume that $C_x \in (0,1]$ is such that $x_* = (C_x\gamma)^{1/(\mu+1)} < x_{1,a^*-1}^{[1]}$ and $C > C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R$. Under the conditions of the theorem it holds for $\mu \ge 0$*

$$\left|\left(p_{a^*}^{[1]}\right)'(0)\right| \le \gamma_n^{-1/(\mu+1)} \left[C_x^{-1/(\mu+1)} \left\{ 1 - \frac{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R}{C} \right\}^{-2} + \left\{ \frac{(2\mu+2)^{\mu+1} R}{C - C_\delta(\mu+1)\kappa^\mu R + C_\epsilon} \right\}^{1/(\mu+1)} \right]$$

Proof: The proof is done in two steps by using the inequality $\left|\left(p_{a^*}^{[1]}\right)'(0)\right| \le \left|\left(p_{a^*-1}^{[1]}\right)'(0)\right| + \left|\left(p_{a^*}^{[1]}\right)'(0) - \left(p_{a^*-1}^{[1]}\right)'(0)\right|$.

Consider first $a^* > 1$.

We will bound $\|S_n f_{a^*-1}^{[1]} - T_n^* y\|_{\mathcal{H}}$ from above. Define $z = x_{1,a^*-1}^{[1]}$ and $\phi_i(x) = p_i^{[1]}(x)(z-x)^{-1/2} z^{1/2}$, $0 \le x \le z$. Due to Lemma 9 (vi) it holds that $\sup_{0\le x\le z} x^\nu \phi_{a^*-1}^2(x) \le \nu^\nu |(p_{a^*-1}^{[1]})'(0)|^{-\nu}$, $\nu \ge 0$. The proof of Lemma 3.7 in Hanke (1995) shows that

$$\left[p_{a^*-1}^{[1]}, p_{a^*-1}^{[1]}\right]_0^{1/2} \le \|P_z \phi_{a^*-1}(S_n) T_n^* y\|_{\mathcal{H}}.$$

This yields with (SH)

$$\begin{aligned}
&\|S_n f_{a^*-1}^{[1]} - T_n^* y\|_{\mathcal{H}} = \left[p_{a^*-1}^{[1]}, p_{a^*-1}^{[1]}\right]_0^{1/2} \le \|P_z \phi_{a^*-1}(S_n) T_n^* y\|_{\mathcal{H}} \\
&\le \|P_z \phi_{a^*-1}(S_n) S f\|_{\mathcal{H}} + \|P_z \phi_{a^*-1}(S_n)(T_n^* y - Sf)\|_{\mathcal{H}} \\
&\le \|P_z \phi_{a^*-1}(S_n) S f\|_{\mathcal{H}} + C_\epsilon \gamma_n \left(\sup_{0\le x\le z} \phi_{a^*-1}^2 \right)^{1/2} \\
&\le \|P_z \phi_{a^*-1}(S_n) S_n^{\mu+1} u\|_{\mathcal{H}} + \|P_z \phi_{a^*-1}(S_n)(S_n^{\mu+1} - S^{\mu+1})u\|_{\mathcal{H}} + C_\epsilon \gamma_n \\
&\le R \left\{ \left(\sup_{0\le x\le z} x^{2\mu+2} \phi_{a^*-1}^2 \right)^{1/2} + C_\delta(\mu+1)\kappa^\mu \gamma_n \left(\sup_{0\le x\le z} \phi_{a^*-1}^2 \right)^{1/2} \right\} + C_\epsilon \gamma_n \\
&\le \left| \left(p_{a^*-1}^{[1]}\right)'(0) \right|^{-\mu-1} (2\mu+2)^{\mu+1} R + \{C_\delta(\mu+1)\kappa^\mu R + C_\epsilon\}\gamma_n.
\end{aligned}$$

This gives together with $C\gamma_n \le \|S_n f_{a^*-1}^{[1]} - T_n^* y\|_{\mathcal{H}}$

$$C\gamma_n \le \left| \left(p_{a^*-1}^{[1]}\right)'(0) \right|^{-\mu-1} (2\mu+2)^{\mu+1} R + \{C_\delta(\mu+1)\kappa^\mu R + C_\epsilon\}\gamma_n.$$

If $C > C_\delta(\mu+1)\kappa^\mu R + C_\epsilon$ we finally have

$$\left|\left(p_{a^*-1}^{[1]}\right)'(0)\right| \le \gamma_n^{-1/(\mu+1)} \left\{ \frac{(2\mu+2)^{\mu+1} R}{C - C_\delta(\mu+1)\kappa^\mu R + C_\epsilon} \right\}^{1/(\mu+1)}. \tag{16}$$

If $a^* = 1$ it holds $p_{a^*-1}^{[1]} = 1$ and thus $\left|\left(p_{a^*-1}^{[1]}\right)'(0)\right| = 0$ and the inequality (16) is true as well.

We will derive an upper bound on $\left|\left(p_{a^*}^{[1]}\right)'(0) - \left(p_{a^*-1}^{[1]}\right)'(0)\right|$. Due to Corollary 2.6 of Hanke (1995) we have

$$\left|\left(p_{a^*-1}^{[1]}\right)'(0) - \left(p_{a^*}^{[1]}\right)'(0)\right| \le \frac{\left[p_{a^*-1}^{[1]}, p_{a^*-1}^{[1]}\right]_0}{\left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1}. \tag{17}$$

We have $0 \le x \le x_{1,a^*-1}^{[1]} < x_{1,a^*-1}^{[2]}$ due to the interlacing property of the roots in Lemma 9 (i) and thus $0 \le p_{a^*-1}^{[2]}(x) \le 1$ for $0 \le x \le x_{1,a^*-1}^{[2]}$. With that we get with (SH)

$$\begin{aligned}
&\|p_{a^*-1}^{[1]}(S_n) T_n^* y\|_{\mathcal{H}} \le \left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_0^{1/2} \\
&\le \|P_x p_{a^*-1}^{[2]}(S_n) T_n^* y\|_{\mathcal{H}} + x^{-1/2} \|P_x^{\perp} S_n^{1/2} p_{a^*-1}^{[2]}(S_n) T_n^* y\|_{\mathcal{H}} \\
&\le \|P_x p_{a^*-1}^{[2]}(S_n)(T_n^* y - Sf)\|_{\mathcal{H}} + \|P_x p_{a^*-1}^{[2]}(S_n) S^{\mu+1} u\|_{\mathcal{H}} + x^{-1/2} \left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1^{1/2} \\
&\le C_\epsilon \gamma_n + R\left\{C_\delta(\mu+1)\kappa^\mu \gamma_n + x^{\mu+1}\right\} + x^{-1/2} \left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1^{1/2}.
\end{aligned}$$

For the choice $x_* = (C_x\gamma)^{1/(\mu+1)}$ we get

$$\left[p_{a^*-1}^{[1]}, p_{a^*-1}^{[1]}\right]_0^{1/2} \le \gamma_n \{C_\epsilon + C_\delta(\mu+1)\kappa^\mu R + C_x\} + x_*^{-1/2} \left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1^{1/2}.$$

It holds $\left[p_{a^*-1}^{[1]}, p_{a^*-1}^{[1]}\right]_0^{1/2} = \|S_n f_{a^*-1}^{[1]} - T_n^* y\|_{\mathcal{H}} \ge C\gamma_n$. This yields with $C > C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R$

$$\left[p_{a^*-1}^{[1]}, p_{a^*-1}^{[1]}\right]_0 \le \gamma_n^{-1/(\mu+1)} C_x^{-1/(\mu+1)} \left\{ 1 - \frac{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R}{C} \right\}^{-2} \left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1.$$

Together with (17) we have

$$\left|\left(p_{a^*-1}^{[1]}\right)'(0) - \left(p_{a^*}^{[1]}\right)'(0)\right| \le \gamma_n^{-1/(\mu+1)} C_x^{-1/(\mu+1)} \left\{ 1 - \frac{C_\epsilon + C_x R + C_\delta(\mu+1)\kappa^\mu R}{C} \right\}^{-2}.$$

Combining this with (16) completes the proof. ■

B.1.3 Proof of Theorem 1

The proof is an application of Lemmas 11 — 13 to (10). First note that $r \geq 3/2$ implies $\mu \geq 1$ and thus this condition in Lemma 12 holds.

Let us choose $x_* = (C_x\gamma_n)^{1/(\mu+1)}$. Lemma 9 (v) shows that $\left|\left(p_i^{[r]}\right)'(0)\right| = \sum_{j=1}^{i}(x_{j,i}^{[r]})^{-1}$ for $i = 1, \ldots, n$, $r \in \mathbb{N}_0$. Thus it holds $\left|\left(p_i^{[1]}\right)'(0)\right|^{-1} \leq x_{1,i}^{[1]}$.

Equation (16) thus shows that C_x can be chosen small enough such that

$$x_* \leq \left|\left(p_{a^*-1}^{[1]}\right)'(0)\right|^{-1} \leq x_{1,a^*-1}^{[1]}$$

and $C_x < 1$, which makes the first condition in Lemma 11 and 13 hold true. The choice $C = C_\epsilon + (\mu+1)\kappa^\mu R(1+C_\delta)$ gives the second condition.

Now we need to check the remaining condition of Lemma 12, namely that a C_z can be chosen such that $(C_z\gamma_n)^{1/(\mu+1)} \leq x_{1,a^*}^{[1]}$ is true. Lemma 13 yields a $C_z > 0$ such that $C_z\gamma_n^{1/(\mu+1)} \leq \left|\left(p_{a^*}^{[1]}\right)'(0)\right|^{-1} \leq x_{1,a^*}^{[1]}$. Denote $z_* = (C_z\gamma_n)^{1/(\mu+1)}$ and Lemma 12 can be applied.

To ease notation we will denote everything in the derived bounds that does not depend on γ_n as a constant c_j, $j \in \mathbb{N}$. Thus we get by combining Lemmas 12 and 13 that with probability at least $1-\nu$

$$\begin{aligned}\|f - f_{a^*}^{[1]}\|_{\mathcal{H}}^2 &\leq c_1\gamma_n^{\mu/(\mu+1)} + c_2\gamma_n + c_3\gamma_n^{1-1/(\mu+1)} + c_4\gamma_n\left|\left(p_{a^*}^{[1]}\right)'(0)\right| \\ &\leq c_1\gamma_n^{\mu/(\mu+1)} + c_2\gamma_n + c_3\gamma_n^{\mu/(\mu+1)} + c_5\gamma_n^{1-1/(\mu+1)} = O\{\gamma_n^{\mu/(\mu+1)}\}\end{aligned}$$

and

$$\begin{aligned}&\|S_n^{1/2}(f - f_{a^*}^{[1]})\|_{\mathcal{H}}^2 \\ &\leq c_6\gamma_n^{(\mu+1/2)/(\mu+1)} + c_7\gamma_n^{1/(2\mu+2)}\gamma_n + c_8\gamma_n^{-1/(2\mu+2)}\gamma_n + c_9\gamma_n^{1/(2\mu+1)}\gamma_n\left|\left(p_{a^*}^{[1]}\right)'(0)\right| \\ &\leq c_6\gamma_n^{(\mu+1/2)/(\mu+1)} + c_7\gamma_n^{(2\mu+3)/(2\mu+2)} + c_8\gamma_n^{(2\mu+1)/(2\mu+2)} + c_10\gamma_n^{1+1/(2\mu+2)-1/(\mu+1)} \\ &= O\{\gamma_n^{(2\mu+1)/(2\mu+2)}\}.\end{aligned}$$

Finally Lemma 11 gives

$$\|f_{a^*} - f_{a^*}^{[1]}\|_{\mathcal{H}}^2 = O\{\gamma_n^{\mu/(\mu+1)}\}, \qquad \|S_n^{1/2}(f_{a^*} - f_{a^*}^{[1]})\|_{\mathcal{H}} = O\{\gamma_n^{(2\mu+1)/(2\mu+2)}\}.$$

Combining the above with (10) yields

$$\begin{aligned}&\|f - f_{a^*}\|_{\mathcal{H}}^2 = O\{\gamma_n^{\mu/(\mu+1)}\}, \\ &\|f^* - f_{a^*}\|_2^2 = O\{\gamma^{1/2}\gamma_n^{\mu/(\mu+1)}\} + O\{\gamma_n^{(2\mu+1)/(2\mu+2)}\} = O\{\gamma_n^{(2\mu+1)/(2\mu+2)}\},\end{aligned}$$

completing the proof with $\mu = r - 1/2$. ■

B.2 Proof of Theorem 2

The overall design of this proof is similar to the one of Theorem 1 and makes heavy use of results obtained in Blanchard and Krämer (2010b).

B.2.1 Preparation for the Proof

The stopping index (7) can be reformulated with $\mu = r - 1/2$ as

$$a^* = \min\{1 \leq a \leq n : \|S_n f_a^{[1]} - T_n^* y\|_{\mathcal{H}} \leq C\zeta_n\}, \tag{18}$$

with $\zeta_n = \max\{\sqrt{\lambda_n d_\lambda}\gamma_n, \lambda_n^{\mu+1}\}$.

We will derive the result in a similar way to Theorem 1. First it holds due to (10)

$$\begin{aligned}\|f_{a^*} - f^*\|_2 &= \|S^{1/2}(f_{a^*} - f)\|_{\mathcal{H}} \leq \|S^{1/2}(f_{a^*} - f_{a^*}^{[1]})\|_{\mathcal{H}} + \|S^{1/2}(f_{a^*}^{[1]} - f)\|_{\mathcal{H}} \\ &\leq C_\psi\lambda^{1/2}\|f_{a^*} - f_{a^*}^{[1]}\|_{\mathcal{H}} + C_\psi\|S_n^{1/2}(f_{a^*} - f_{a^*}^{[1]})\|_{\mathcal{H}} + \|S^{1/2}(f_{a^*}^{[1]} - f)\|_{\mathcal{H}}.\end{aligned} \tag{19}$$

Now we prove the analogue versions of Lemma 11 — 13. The comments regarding those lemmas also hold here.

Lemma 14 *Let $x = C_x\lambda_n$, $C_x > 0$ such that $0 < x < x_{1,a^*-1}^{[2]}$. Choose $C > \tilde{c}_2$, with $\tilde{c}_1 = R\max\{1, C_\psi^2, \mu\kappa^{\mu-1}C_\delta\}$ and $\tilde{c}_2 = 2\max\{C_\psi C_\epsilon\sqrt{C_x+1}, \tilde{c}_1 C_x(C_x^\mu + 1)\}$. Then it holds*

$$\begin{aligned}\|f_{a^*} - f_{a^*}^{[1]}\|_{\mathcal{H}} &\leq \frac{C^3}{C_x(C-\tilde{c}_2)^2}\lambda_n^{-1}\zeta_n, \\ \|S_n^{1/2}(f_{a^*} - f_{a^*}^{[1]})\|_{\mathcal{H}} &\leq \frac{C^2}{C_x^{1/2}(C-\tilde{c}_2)}\lambda_n^{-1/2}\zeta_n.\end{aligned}$$

Proof: According to the proof of Lemma 14 we can focus on the case $0 < a^* < n$. Furthermore we have due to (12)

$$\|f_{a^*} - f_{a^*}^{[1]}\|_{\mathcal{H}} \leq \frac{\left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_0}{\left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1}\|p_{a^*}^{[1]}(S_n)T_n^* y\|_{\mathcal{H}}. \tag{20}$$

Using Lemma A.3 in Blanchard and Krämer (2010b) (and the first line of its proof) we have for $0 < x < x_{1,a^*-1}^{[2]}$ with $\tilde{c}_1 = R\max\{1, C_\psi^2, \mu\kappa^{\mu-1}C_\delta\}$

$$\left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_0^{1/2} \leq C_\psi C_\epsilon\sqrt{x+\lambda_n}\sqrt{d_{\lambda_n}}\gamma_n + \tilde{c}_1 x\{x^\mu + Z_\mu(\lambda_n)\} + x^{-1/2}\left[p_{a^*-1}^{[2]}, p_{a^*-1}^{[2]}\right]_1^{1/2}. \tag{21}$$

Here we define $Z_\mu(\lambda) = \lambda^\mu\mathbb{I}(\mu \leq 1) + \gamma_n\mathbb{I}(\mu > 1)$. Note that under the assumptions of the theorem it holds $Z_\mu(\lambda_n) \leq \lambda_n^\mu$.

Choosing $x = C_x\lambda_n$ yields in (21)

$$\begin{aligned}
&\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2} \\
&\leq C_\psi C_\epsilon\sqrt{C_x+1}\sqrt{\lambda_n d_{\lambda_n}}\gamma_n + \tilde{c}_1 C_x(C_x^\mu+1)\lambda_n^{\mu+1} + C_x^{-1/2}\lambda_n^{-1/2}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2} \\
&\leq \tilde{c}_2\max\{\sqrt{\lambda_n d_{\lambda_n}}\gamma_n, \lambda_n^{\mu+1}\} + C_x^{-1/2}\lambda_n^{-1/2}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2} \\
&= \tilde{c}_2\zeta_n + C_x^{-1/2}\lambda_n^{-1/2}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2},
\end{aligned}$$

with $\tilde{c}_2 = 2\max\{C_\psi C_\epsilon\sqrt{C_x+1}, \tilde{c}_1 C_x(C_x^\mu+1)\}$. Due to the stopping condition (18) we know that

$$\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2} \geq \left[p^{[1]}_{a^*-1}, p^{[1]}_{a^*-1}\right]_0^{1/2} = \|S_n f_a^{[1]} - T_n^* y\|_{\mathcal{H}} \geq C\zeta_n.$$

This gives

$$\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2} \leq \frac{C}{\sqrt{C_x}(C-\tilde{c}_2)}\lambda_n^{-1/2}\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2}. \tag{22}$$

Plugging this into (20) yields together with the definition of the stopping index a^*

$$\|f_{a^*} - f^{[1]}_{a^*}\|_{\mathcal{H}} \leq \frac{C^3}{C_x(C-\tilde{c}_2)^2}\lambda_n^{-1}\zeta_n.$$

In a similar way we derive for the second case

$$\|S_n^{1/2}(f_{a^*} - f^{[1]}_{a^*})\|_{\mathcal{H}} = \frac{\left[p^{[1]}_{a^*}, p^{[1]}_{a^*}\right]_0}{\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2}} \leq \frac{\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0^{1/2}}{\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1^{1/2}}\left[p^{[1]}_{a^*}, p^{[1]}_{a^*}\right]_0.$$

An application of (22) yields

$$\|S_n^{1/2}(f_{a^*} - f^{[1]}_{a^*})\|_{\mathcal{H}} \leq \frac{C^2}{\sqrt{C_x}(C-\tilde{c}_2)}\lambda_n^{-1/2}\zeta_n.$$

■

Lemma 15 *Denote $\tilde{c}_1 = R\max\{1, C_\psi^2, \mu\kappa^{\mu-1}C_\delta\}$. For any $i = 1, \ldots, n$ and $0 < x < x^{[1]}_{1,i}$ we have under the conditions of the theorem*

$$\begin{aligned}
\|S^{1/2}(f_i^{[1]} - f)\|_{\mathcal{H}} \leq C_\psi &\left[C_\delta + \sqrt{2}C_\epsilon + \lambda_n\left\{C_\delta\left|\left(p_i^{[1]}\right)'(0)\right| + \sqrt{2}C_\epsilon x^{-1}\right\}\right]\sqrt{d_{\lambda_n}}\gamma_n \\
&+ \tilde{c}_1(\sqrt{x}+\sqrt{\lambda_n})(x^\mu + \lambda_n^\mu) + (1+\sqrt{x^{-1}\lambda_n})x^{-1/2}\|S_n f_i^{[1]} - T_n^* y\|_{\mathcal{H}}.
\end{aligned}$$

Proof: Follow the proof of Lemma A.2 in Blanchard and Krämer (2010b). Note that $Z_\mu(\lambda_n) \leq \lambda_n^\mu$. ■

Lemma 16 *Let $C > \max\{C_\psi C_\epsilon, \tilde{c}_2\}$, where $\tilde{c}_2$ is given in Lemma 14. Choose $x = C_x\lambda_n$ such that $0 < x \leq x^{[1]}_{1,a^*-1}$. Then there exists a constant $c^* > 0$ such that*

$$\left|\left(p^{[1]}_{a^*}\right)'(0)\right| \leq c^*\lambda_n^{-1}. \tag{23}$$

Proof: In analogue to Lemma 13 we will first derive an upper bound on $\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|$. Lemma A.1 in Blanchard and Krämer (2010b) yields

$$\begin{aligned}
\|S_n f^{[1]}_{a^*-1} - T_n^* y\|_{\mathcal{H}} \leq\ & R(2\mu+2)^{\mu+1}\max\{1, C_\psi^{2\mu}\}\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|^{-\mu-1} \\
&+ 2R\mu\kappa^{\mu-1}\max\{1, C_\delta, C_\psi^{2\mu}, C_\psi^{2\mu}C_\delta\}\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|^{-1} Z_\mu(\lambda_n) \\
&+ C_\epsilon C_\psi\left\{\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|^{-1/2} + \sqrt{\lambda_n}\right\}\sqrt{d_{\lambda_n}}\gamma_n.
\end{aligned}$$

Denote $\tilde{c}_3 = R(2\mu+2)^{\mu+1}\max\{1, C_\psi^{2\mu}\}$ and $\tilde{c}_4 = 2R\mu\kappa^{\mu-1}\max\{1, C_\delta, C_\psi^{2\mu}, C_\psi^{2\mu}C_\delta\}$.

The definition of a^* gives $C\zeta_n \leq \|S_n f^{[1]}_{a^*-1} - T_n^* y\|_{\mathcal{H}}$. Combining both inequalities, setting $x = C_x\lambda_n$ and keeping $\sqrt{\lambda_n d_{\lambda_n}}\gamma_n \leq \zeta_n$ in mind gives

$$\begin{aligned}
(C - C_\psi C_\epsilon C_\lambda)\zeta_n &\leq \tilde{c}_3\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|^{-\mu-1} + \tilde{c}_4\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|^{-1}\lambda_n^\mu \\
&\quad + C_\epsilon C_\psi\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|^{-1/2}\sqrt{d_{\lambda_n}}\gamma_n \\
&\leq 3\max\left\{\tilde{c}_3\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|^{-\mu-1}, \tilde{c}_4\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|^{-1}\lambda_n^\mu,\right. \\
&\quad \left. C_\epsilon C_\psi\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right|^{-1/2}\sqrt{d_{\lambda_n}}\gamma_n\right\}.
\end{aligned}$$

Now we assume that the maximum on the right hand side is attained in each of the three possible cases

$$\begin{aligned}
\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right| &\leq \{3(C - C_\epsilon C_\psi)^{-1}\tilde{c}_3\}^{1/(\mu+1)}\zeta_n^{-1/(\mu+1)}, \\
\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right| &\leq 3(C - C_\epsilon C_\psi)^{-1}\tilde{c}_4\zeta_n^{-1}\lambda_n^\mu, \\
\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right| &\leq 9(C - C_\epsilon C_\psi)^{-2}C_\epsilon^2 C_\psi^2\zeta_n^{-2}d_{\lambda_n}\gamma_n^2.
\end{aligned}$$

Take $\tilde{c}_5 = \max[\{3(C - C_\epsilon C_\psi)^{-1}\tilde{c}_3\}^{1/(\mu+1)}, 3(C - C_\epsilon C_\psi)^{-1}\tilde{c}_4, 9(C - C_\epsilon C_\psi)^{-2}C_\epsilon^2 C_\psi^2]$.

It is easy to see that $\zeta_n^{-1/(\mu+1)}, \zeta_n^{-1}\lambda_n^{\mu}$ and $\zeta_n^{-2} d_{\lambda_n}\gamma_n^2$ are all bound from above by λ_n^{-1}. Hence we get

$$\left|\left(p^{[1]}_{a^*-1}\right)'(0)\right| \leq \tilde{c}_5 \lambda_n^{-1}. \tag{24}$$

For the final step in the proof we have due to (17)

$$\left|\left(p^{[1]}_{a^*-1}\right)'(0) - \left(p^{[1]}_{a^*}\right)'(0)\right| \leq \frac{\left[p^{[1]}_{a^*-1}, p^{[1]}_{a^*-1}\right]_0}{\left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_1}.$$

It holds $\left[p^{[1]}_{a^*-1}, p^{[1]}_{a^*-1}\right]_0 \leq \left[p^{[2]}_{a^*-1}, p^{[2]}_{a^*-1}\right]_0$ and hence (22) yields

$$\left|\left(p^{[1]}_{a^*-1}\right)'(0) - \left(p^{[1]}_{a^*}\right)'(0)\right| \leq \frac{C^2}{C_x(C-\tilde{c}_2)^2}\lambda_n^{-1}. \tag{25}$$

The proof is complete by combining (24) and (25). ■

B.2.2 Proof of Theorem 2

We first restrict ourselves to the set where all concentration inequalities stated in the theorem hold simultaneously with probability at least $1-\nu$, $\nu \in (0,1]$. We only proof the convergence rates in the $\mathcal{L}^2$-norm, the corresponding rates in the $\mathcal{H}$-norm are done in the same way.

The theorem is proven by an application of Lemmas 14—16. To that end we need to check the conditions of those. Equation (24) and the proof of Theorem 1 show that we can take $C_x = \min\{1/2, \tilde{c}_5\}$ to fulfill $0 < x \leq x^{[1]}_{1,a^*-1}$. Furthermore we can take $C = 4R\max\{1, C_\psi^2(\nu), (r-1/2)\kappa^{r-3/2}C_\delta(\nu), 2^{-1/2}R^{-1}C_\psi(\nu)C_\epsilon(\nu)\}$ and the conditions of Lemma 14 and 16 hold. Note that $x^{[1]}_{1,a^*-1} \leq x^{[2]}_{1,a^*-1}$ due to the interlacing property of the roots, see Lemma 9 (i).

For Lemma 15 we need to find a $0 < z < x^{[1]}_{1,a^*}$. By Lemma 16 there exists a constant $c^* > 0$ such that

$$(c_*)^{-1}\lambda_n \leq \left|\left(p^{[1]}_{a^*}\right)'(0)\right|^{-1} \leq x^{[1]}_{1,a^*},$$

hence we choose $C_z = \min\{1/2, 1/c^*\}$. Now, applying Lemmas 14—16 to (19) gives the result (we again denote any constant that does not depend on n with C_i, $i \in \mathbb{N}$)

$$\begin{aligned}
&\|f_{a^*} - f^*\|_2 \leq C_\psi \lambda_n^{1/2}\|f_{a^*} - f^{[1]}_{a^*}\|_{\mathcal{H}} + C_\psi \|S_n^{1/2}(f_{a^*} - f^{[1]}_{a^*})\|_{\mathcal{H}} + \|S^{1/2}(f^{[1]}_{a^*} - f)\|_{\mathcal{H}} \\
\leq& C_1\lambda_n^{-1/2}\zeta_n + C_2\lambda_n\left|\left(p^{[1]}_i\right)'(0)\right|\sqrt{d_{\lambda_n}}\gamma_n + C_3\lambda_n^{\mu+1/2} + C_4\lambda_n^{-1/2}\|S_n f^{[1]}_{a^*} - T_n^* y\|_{\mathcal{H}}. \\
\leq& C_5\lambda_n^{-1/2}\zeta_n + C_6\sqrt{d_{\lambda_n}}\gamma_n + C_4\lambda_n^{\mu+1/2} \leq \max\{C_4, C_5, C_6\}\lambda_n^{-1/2}\zeta_n.
\end{aligned}$$

The error bound in the $\mathcal{H}$-norm is proven in an analogue fashion. ■

B.3 Proof of Corollary 3

Take $\lambda_n = \gamma_n^{2/(2r+s)}$. It is immediate that $\lambda_n^{r-1/2} = \gamma_n^{(2r-1)/(2r+s)} \geq \gamma_n$ for n sufficiently large, hence the inequality (6) holds as soon as $\gamma_n \leq 1$. Then we have by Theorem 2 that

$$\|f_{\widehat{\alpha}_{a^*}} - f^*\|_2 = O\left\{\lambda_n^{-1/2}\zeta_n(\lambda_n)\right\} = O\left\{\gamma_n^{2r/(2r+s)}\right\}.$$

■

B.4 Proof of Corollary 4

Set $\lambda_n = \gamma_n^{1/r}\log\{1/(2r)\gamma_n^{-2}\}$. It is immediate that $\lambda_n \to 0$ as γ_n converges to zero. For $r = 1/2$ condition (6) holds trivially. Let $r > 1/2$, then we have

$$\lambda_n^{r-1/2} = \gamma_n^{(r-1)/r}\log\{(r-1/2)/(2r)\gamma_n^{-2}\} \geq \gamma_n,$$

This is equivalent to $2r - 1 \geq 2\exp(\gamma_n)\gamma_n^2$, which holds for n sufficiently large and $r > 1/2$.

For the convergence rate we first show that $d_{\lambda_n}\gamma_n^2 \leq \lambda_n^{2r}$. We have

$$d_{\lambda_n}\gamma_n^2 = \log\left\{1 + \frac{a}{\gamma_n^{1/r}\log^{1/r}(1/2\gamma_n^{-1})}\right\}\gamma_n^2 \leq \log\left(\gamma_n^{-2}\right)\gamma_n^2.$$

Equivalently we need $a^r \leq \gamma_n(\gamma_n^{-2}-1)^r\log(1/2\gamma_n^{-2})$. As γ_n converges to zero $\gamma_n(\gamma_n^{-2}-1)^r$ goes to infinity for any $r > 1/2$. Hence for suitably large n it holds $\lambda_n^r \geq \sqrt{d_{\lambda_n}}\gamma_n$. Then the convergence rate is $\lambda_n^{-1/2}\zeta_n(\lambda_n) = \lambda_n^r = \gamma_n\log(1/2\gamma^{-2})$.

Because the convergence rate does not depend on $r \geq 1/2$ we can set $r = 1/2$. ■

B.5 Proof of Theorem 5

B.5.1 Preparation for the Proof

We denote with $\mathrm{tr}(A)$ the trace of a trace class operator $A : \mathcal{H} \to \mathcal{H}$ and the tensor product $(f_1 \otimes f_2)h = \langle f_1, h\rangle_{\mathcal{H}} f_2$ for functions $f_1, f_2, h \in \mathcal{H}$. We use the notation $k_t = k(\cdot, X_t)$. Note that it holds $\|A\|^2_{\mathrm{HS}} = \mathrm{tr}(A^*A)$ for a Hilbert-Schmidt operator A.

The general structure for the proof is as follows: To derive the concentration inequalities we need $\mathrm{E}\{\|S_n - S\|^2_{\mathcal{L}}\}$ as well as $\mathrm{E}\{\|T_n^* y - Sf\|^2_{\mathcal{H}}\}$ for Markov's inequality. With the help of Lemma 17 we derive representations of these expectations as sums by using the stationarity of X_t. Under normality assumptions Lemma 18 gives upper bounds for the summands in terms of the autocorrelation function of X_t using (D1). Assuming the special structure implied by (D2) Corollary 19 shows how these upper bound depend on the long range dependence coefficient $q > 0$. To get explicit convergence rates Lemma 20 gives a simple result stating how sums of these autocorrelation functions depend on q. An analogue result is given in Lemma 21 for the expectation of a warped norm.

The next technical lemma derives some simple identities for the operator S under the Hilbert-Schmidt norm, its trace and the kernel k. These will be used to find alternate representations of $\mathrm{E}\{\|S_n - S\|^2_{\mathcal{L}}\}$ and $\mathrm{E}\{\|T_n^* y - Sf\|^2_{\mathcal{H}}\}$ under stationarity.

Lemma 17 *Under the assumptions (K1) and (K2) the following hold*

(i) $\mathrm{tr}\{(k_t \otimes k_t)(k_s \otimes k_s)\} = k^2(X_t, X_s)$,

(ii) $\|S\|^2_{\mathrm{HS}} = \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} k^2(x,y)\mathrm{dP}^{X_0}(x)\mathrm{dP}^{X_0}(y)$,

(iii) $\mathrm{E}[\mathrm{tr}\{(k_0 \otimes k_0)S\}] = \|S\|^2_{\mathrm{HS}}$.

(iv) Let X' and X'' be independent and identically distributed and denote $k' = k(\cdot, X')$, $k'' = k(\cdot, X'')$. It holds for $\nu = 1, 2$ and $\lambda > 0$

$$\mathrm{E}\left[\mathrm{tr}^\nu\left\{(S+\lambda)^{-1}k' \otimes k'k'' \otimes k''\right\}\right] = \mathrm{tr}^\nu\{(S+\lambda)^{-1}S^2\}.$$

Proof: (i) Let $\{v_i\}_{i\in\mathbb{N}}$ denote an orthonormal base of $\mathcal{H}$. Then it holds due to the reproducing property (2)

$$\mathrm{tr}\left\{(k_t \otimes k_t)(k_s \otimes k_s)\right\} = \sum_{i=1}^\infty \langle v_i, k_t\rangle_{\mathcal{H}}\langle v_i, k_s\rangle_{\mathcal{H}} k(X_t, X_s) = \left\langle \sum_{i=1}^\infty \langle v_i, k_s\rangle_{\mathcal{H}} v_i, k_t \right\rangle_{\mathcal{H}} k(X_t, X_s).$$

(ii)

$$\begin{aligned}\|S\|^2_{\mathrm{HS}} &= \sum_{i=1}^\infty \langle Sv_i, Sv_i\rangle_{\mathcal{H}} = \sum_{i=1}^\infty \int_{\mathbb{R}^d} \langle Sv_i, k(\cdot,x)\rangle_{\mathcal{H}}\langle v_i, k(\cdot,x)\rangle_{\mathcal{H}}\mathrm{dP}^X(x) \\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} \left\langle \sum_{i=1}^\infty \langle v_i, k(\cdot,x)\rangle_{\mathcal{H}} v_i, k(\cdot,y)\right\rangle_{\mathcal{H}} k(x,y)\mathrm{dP}^X(x)\mathrm{dP}^X(y).\end{aligned}$$

The assertion follows because $\mathrm{P}^X = \mathrm{P}^{X_0}$.

(iii)

$$\begin{aligned}\mathrm{E}[\mathrm{tr}\{(k_0 \otimes k_0)S\}] &= \mathrm{E}(\langle Sk_0, k_0\rangle_{\mathcal{H}}) = \mathrm{E}\left(\int_{\mathbb{R}^d} \langle k_0, k(\cdot,x)\rangle^2_{\mathcal{H}}\mathrm{dP}^X(x)\right) \\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} k^2(x,y)\mathrm{dP}^X(x)\mathrm{dP}^{X_0}(y) = \|S\|^2_{\mathrm{HS}}.\end{aligned}$$

(iv) Because S is a compact operator the spectral decomposition $S = \sum_{i=1}^\infty \mu_i \psi_i \otimes \psi_i$ holds (recall that $\{\mu_i, \psi_i\}_{i=1}^\infty$ is the eigensystem of S). Let $k(\cdot, x) = \sum_{i=1}^\infty \alpha_i(x)\psi_i$, $x \in \mathbb{R}^d$. For $\nu = 1$ we have

$$\begin{aligned}\mathrm{E}\left[\mathrm{tr}\{(S+\lambda)^{-1}k' \otimes k'k'' \otimes k''\}\right] &= \mathrm{E}\left\{k(X', X'')\sum_{i=1}^\infty \langle \psi_i, k''\rangle_{\mathcal{H}}\langle (S+\lambda)^{-1}k', \psi_i\rangle_{\mathcal{H}}\right\} \\ &= \sum_{i=1}^\infty \frac{1}{\mu_i + \lambda}\mathrm{E}\left\{k(X', X'')\langle \psi_i, k''\rangle_{\mathcal{H}}\langle \psi_i, k'\rangle_{\mathcal{H}}\right\} \\ &= \sum_{i=1}^\infty \frac{1}{\mu_i + \lambda}\sum_{j=1}^\infty \mathrm{E}\left\{\alpha_j(X')\alpha_i(X')\alpha_j(X'')\alpha_i(X'')\right\}\end{aligned}$$

On the other hand

$$\begin{aligned}\mathrm{tr}\left\{(S+\lambda)^{-1}S^2\right\} &= \mathrm{tr}\left\{(S+\lambda)^{-1}\mathrm{E}(k' \otimes k'k'' \otimes k'')\right\} \\ &= \sum_{i=1}^\infty \langle (S+\lambda)^{-1}\mathrm{E}\{k(X', X'')\langle \psi_i, k''\rangle_{\mathcal{H}}k'\}, \psi_i\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^\infty \frac{1}{\mu_i + \lambda}\langle \mathrm{E}\{k(X', X'')\langle \psi_i, k''\rangle_{\mathcal{H}}k'\}, \psi_i\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^\infty \frac{1}{\mu_i + \lambda}\sum_{j=1}^\infty \mathrm{E}\{\alpha_j(X')\alpha_i(X')\alpha_j(X'')\alpha_i(X'')\}\end{aligned}$$

and we are done. The proof for $\nu = 2$ is along the same lines. ■

Denote with g_h the common density of $(X_h, X_0)^{\mathrm{T}}$ and g_0 the density of X_0. The next lemma and the subsequent corollary will be used to show that the summands appearing in Theorem 5 (i) can be linked to the autocorrelation function ρ under the assumptions of normality (D1):

Lemma 18 *Under the assumptions (K1), (K2) and (D1) it holds for $h > 0$ with $\rho_h = \tau_0^{-1}\tau_h$*

$$\begin{aligned}\int_{\mathbb{R}^{2d}} k^2(x,y)\{g_h(x,y) - g_0(x)g_0(y)\}\mathrm{d}(x,y) &\le \frac{\kappa^2}{\{(4\pi\tau_0)^d \det(\Sigma)\}^{1/2}}\theta^{1/2}(\rho_h), \\ \int_{\mathbb{R}^{2d}} k(x,y)f(x)f(y)\{g_h(x,y) - g_0(x)g_0(y)\}\mathrm{d}(x,y) &\le \frac{\kappa M}{\{(4\pi\tau_0)^d \det(\Sigma)\}^{1/2}}\theta^{1/2}(\rho_h),\end{aligned}$$

with $\theta(\rho) = 1 + (1-\rho^2)^{-d/2} - 2^{d+1}(4-\rho^2)^{-d/2}$, $\rho \in [0,1)$.

Proof: We will only proof the first inequality, the second one follows in the same way.

By Jensen's inequality and (K2) we know

$$\begin{aligned}&\int_{\mathbb{R}^{2d}} k^2(x,y)\{g_h(x,y) - g_0(x)g_0(y)\}\mathrm{d}(x,y) \\ &\le \kappa^2\left[\int_{\mathbb{R}^{2d}} \left\{g_h^2(x,y) - 2g_h(x,y)g_0(x)g_0(y) + g_0^2(x)g_0^2(y)\right\}\mathrm{d}(x,y)\right]^{1/2}.\end{aligned}$$

The first and third integral term can readily be calculated as

$$\begin{aligned}\int_{\mathbb{R}^{2d}} g_h^2(x,y)\mathrm{d}(x,y) &= [(4\pi)^d(\tau_0^2 - \tau_h^2)^{d/2}\det(\Sigma)]^{-1} \\ \left\{\int_{\mathbb{R}^d} g_0^2(x)\mathrm{d}x\right\}^2 &= \{(4\pi)^d\tau_0^d\det(\Sigma)\}^{-1}.\end{aligned}$$

For the first equality we use $\det(A \otimes \Sigma) = \det(A)^d \det(\Sigma)^2$ for $A \in \mathbb{R}^{2\times 2}$ and thus

$$\int_{\mathbb{R}^{2d}} g_h(x,y) g_0(x) g_0(y) \mathrm{d}(x,y) = \frac{\int_{\mathbb{R}^{2d}} \exp\left(-1/2 z^{\mathrm{T}} G^{-1} z\right) \mathrm{d}z}{(2\pi)^{2d} \det(\Sigma)^2 \tau_0^d (\tau_0^2 - \tau_h^2)^{d/2}}, \tag{26}$$

with

$$G^{-1} = \left\{ \begin{pmatrix} \tau_0 & \tau_h \\ \tau_h & \tau_0 \end{pmatrix}^{-1} + \begin{pmatrix} \tau_0^{-1} & 0 \\ 0 & \tau_0^{-1} \end{pmatrix} \right\} \otimes \Sigma^{-1}.$$

It holds $\det(G) = (4\tau_0^2 - \tau_h^2)^{-d} (\tau_0^4 - \tau_0^2 \tau_h^2)^d \det(\Sigma)^2$. Thus we get with (26)

$$\begin{aligned} \int_{\mathbb{R}^{2d}} g_h(x,y) g_0(x) g_0(y) \mathrm{d}(x,y) &= \frac{(2\pi)^d \tau_0^d (\tau_0^2 - \tau_h^2)^{d/2} \det(\Sigma)}{(2\pi)^{2d} \det(\Sigma)^2 (4\tau_0^2 - \tau_h^2)^{d/2} \tau_0^d (\tau_0^2 - \tau_h^2)^{d/2}} \\ &= \left\{ (2\pi)^d (4\tau_0^2 - \tau_h^2)^{d/2} \det(\Sigma) \right\}^{-1}, \end{aligned}$$

completing the proof by multiplying all terms with $\tau_0^{-d} \tau_0^d$. ■

Using the special structure imposed by (D2) we can link the bounds derived in Lemma 18 to the long range coefficient $q > 0$:

Corollary 19 *Under the assumptions (K1), (K2), (D1) and (D2) it holds for all $h > 0$ and $q > 0$*

$$\int_{\mathbb{R}^{2d}} k^2(x,y) \{g_h(x,y) - g_0(x) g_0(y)\} \mathrm{d}(x,y) \leq \frac{\kappa^2 d^{1/2}}{\sqrt{(2\pi)^d \det(\Sigma)}} (1 - 4^{-q})^{-1/4(d-2)} |\rho_h|$$

$$\int_{\mathbb{R}^{2d}} k(x,y) f(x) f(y) \{g_h(x,y) - g_0(x) g_0(y)\} \mathrm{d}(x,y) \leq \frac{\kappa M d^{1/2}}{\sqrt{(2\pi)^d \det(\Sigma)}} (1 - 4^{-q})^{-1/4(d-2)} |\rho_h|.$$

Proof: Recall that $\theta(\rho) = 1 + \{1-\rho^2\}^{-d/2} - 2^{d+1}\{4-\rho^2\}^{-d/2}$ for $\rho \in [0,1)$. We seek to find bounds on θ and the corollary can be proven by an application of Lemma 18.

By assumption (D2) we know there is a ρ_* such that $\rho_h^2 \leq \rho_*^2 < 1$ for all $h > 0$. Thus consider $\rho \in [0, \rho_*]$. We start by finding a constant $C > 0$ with

$$\theta'(\rho) = \rho \left\{ (1-\rho^2)^{-d/2-1} - 2^{d+1}(4-\rho^2)^{-d/2-1} \right\} d \leq C\rho^2.$$

Thus C can be taken as $C = d\left\{(1-\rho_*^2)^{-d/2-1} - 2^{d+1}(4-\rho_*^2)^{-d/2-1}\right\}$.

Thus we know that the slope of θ is always less than that of $C\rho^2$. Finally it holds that $\theta(0) = 0$ and thus $0 \leq \theta(\rho) \leq C\rho^2$, $\rho \in [0, \rho_*]$.

Under condition (D2) it holds $\{1-\rho_*^2\}^{-d/2} \leq \{1 - 2^{-2q}\}^{-d/2}$, completing the proof by using Lemma 18. ■

Having derived bounds for the summands in Theorem 5 (i) in terms of ρ the overall behavior of the sums is still open. The next simple lemma gives insight into this and shows how the convergence rate of the sums crucially depends on $q > 0$:

Lemma 20 *Assume that condition (D2) holds. Then we have*

$$n^{-2} \sum_{h=1}^{n-1} (n-h) |\rho_h| \leq C(q) \begin{cases} n^{-1} & , \quad q > 1 \\ n^{-1} \log(n) & , \quad q = 1 \\ n^{-q} & , \quad q \in (0,1). \end{cases} \tag{27}$$

with $C(q) = \zeta(q)\mathbb{I}(q>1) + \{5 - \log(4)\}\mathbb{I}(q=1) + \{2(1-q)^{-1} - (2-q)^{-1} + (2-q)^{-1}2^{2-q}\}\mathbb{I}\{q \in (0,1)\}$. Here ζ denotes the Riemann zeta function.

Proof: Recall that by condition (D2) we have $|\rho_h| \leq (h+1)^{-q}$, $h = 0, \ldots, n-1$ for some $q > 0$.

First assume $q \in (0,1]$. The integral test for series convergence gives lower and upper bounds for the hyperharmonic series as

$$(1-q)^{-1}\{(n+1)^{1-q} - 2^{1-q}\} \leq \sum_{h=2}^{n} h^{-q} \leq 2^{-q} + (1-q)^{-1}\{n^{1-q} - 2^{1-q}\}.$$

This yields

$$\begin{aligned} n^{-2} \sum_{h=1}^{n-1} (n-h)(h+1)^{-q} &= n^{-2} \sum_{h=2}^{n} (n+1-h) h^{-q} = n^{-2} \left\{ (n+1) \sum_{h=2}^{n} h^{-q} - \sum_{h=2}^{n} h^{-(q-1)} \right\} \\ &\leq n^{-2} \left[(n+1) \left\{ 2^{-q} + (1-q)^{-1}(n^{1-q} - 2^{1-q}) \right\} - (2-q)^{-1} \left\{ (n+1)^{2-q} - 2^{2-q} \right\} \right]. \end{aligned} \tag{28}$$

Now let $q \in (0,1)$, then it holds from (28) and the fact that $n^{-2} \leq n^{-1} \leq n^{-q}$

$$\begin{aligned} & n^{-2} \sum_{h=1}^{n-1} (n-h)(h+1)^{-q} \\ \leq & \frac{n+1}{n^2} \left\{ \frac{2^{-q}(1-q) - 2^{1-q}}{1-q} \right\} + \frac{n+1}{n^{1+q}} (1-q)^{-1} - \frac{(n+1)^{2-q}}{n^2} (2-q)^{-1} + \frac{1}{n^2} \frac{2^{2-q}}{2-q} \\ \leq & n^{-q} [\{2(1-q)^{-1} - (2-q)^{-1}\} + (2-q)^{-1} 2^{2-q}], \end{aligned}$$

due to $2^{-q}(1-q) - 2^{1-q} < 0$.

For $q = 1$ we evaluate the limit

$$\begin{aligned} & \lim_{q \to 1\pm} n^{-2} \left[(n+1) \left\{ 2^{-q} + (1-q)^{-1}(n^{1-q} - 2^{1-q}) \right\} - (2-q)^{-1} \left\{ (n+1)^{2-q} - 2^{2-q} \right\} \right] \\ &= (2n^2)^{-1} [3 - \log(4) - n\{1 + \log(4)\}] + n^{-2}(n+1)\log(n) \\ &\leq \frac{\log(n)}{n} \left[5 - \log(4) \right]. \end{aligned}$$

The case $q > 1$ is clear because the zeta-function $\zeta(q)$ is defined as the hyperharmonic series with coefficient q. ■

The final preparatory result is used to derive the probablistic bound in Theorem 5 (iv) and is similar to Corollary 19:

Lemma 21 *Under the assumptions (K1), (K2), (D1) and (D2) it holds for $\lambda > 0$*

$$\int_{\mathbb{R}^{2d}} k(x,y)\langle (S+\lambda)^{-1}k(\cdot,x), k(\cdot,y)\rangle_{\mathcal{H}} \left\{f_h(x,y) - f_0(x)f_0(y)\right\} \mathrm{d}(x,y) \leq \tilde{c}|\rho_h|d_\lambda,$$

with $\tilde{c} = \sqrt{d\{1-4^{-q}\}^{-d-1}}\kappa$.

Proof: Denote $\beta(x,y) = \langle (S+\lambda)^{-1}k(\cdot,x), k(\cdot,y)\rangle_{\mathcal{H}}$. By the Cauchy-Schwarz inequality

$$\begin{aligned}\phi_h &= \int_{\mathbb{R}^{2d}} k(x,y)\beta(x,y)\left\{f_h(x,y) - f_0(x)f_0(y)\right\}\mathrm{d}(x,y) \\ &\leq \left[\int k^2(x,y)\beta^2(x,y)f_0(x)f_0(y)\mathrm{d}(x,y)\int\left\{\frac{f_h(x,y)}{\sqrt{f_0(x)f_0(y)}} - \sqrt{f_0(x)f_0(y)}\right\}^2 \mathrm{d}(x,y)\right]^{1/2}. \end{aligned} \tag{29}$$

Denote by X' and X'' two independent copies of X_0 and $k' = k(\cdot, X')$, $k'' = k(\cdot, X'')$. We start by bounding the first integral term in the product:

$$\begin{aligned}\int_{\mathbb{R}^{2d}} k^2(x,y)\beta^2(x,y)f_0(x)f_0(y)\mathrm{d}(x,y) &= \mathrm{E}\left\{k^2(X',X'')\langle (S+\lambda)^{-1}k', k''\rangle_{\mathcal{H}}^2\right\} \\ &= \mathrm{E}\left[\mathrm{tr}^2\left\{(S+\lambda)^{-1}k'\otimes k'k''\otimes k''\right\}\right] \\ &\leq \kappa^2\mathrm{tr}^2\{(S+\lambda)^{-1}S\} = \kappa^2 d_\lambda^2.\end{aligned}$$

In the second to last inequality we used Lemma 17 (iv) and the definition of d_λ.

The second integral in the product in (29) is

$$\int_{\mathbb{R}^{2d}}\left\{\frac{f_h(x,y)}{\sqrt{f_0(x)f_0(y)}} - \sqrt{f_0(x)f_0(y)}\right\}^2\mathrm{d}(x,y) = \int_{\mathbb{R}^{2d}}\frac{f_h(x,y)}{\sqrt{f_0(x)f_0(y)}}\mathrm{d}(x,y) - 1.$$

We proceed in the same way as in the proof of Lemma 18 by using properties of the Gaussian distributions at hand.

First we have

$$F_h(x,y) = \frac{f_h^2(x,y)}{f_0(x)f_0(y)} = \frac{(2\pi)^d\tau_0^d\det(\Sigma)}{(2\pi)^{2d}\det(\Sigma_{\mathrm{h}})}\exp\left[-\frac{1}{2}(x^{\mathrm{T}},y^{\mathrm{T}})\left\{2\Sigma_h^{-1} - \frac{1}{\tau_0}\begin{pmatrix}\Sigma & 0\\ 0 & \Sigma\end{pmatrix}\right\}^{-1}\begin{pmatrix}x\\ y\end{pmatrix}\right].$$

Denote $G^{-1} = 2\Sigma_h^{-1} - \tau_0^{-1}\begin{pmatrix}\Sigma & 0\\ 0 & \Sigma\end{pmatrix}$. It follows in a similar way as in the proof of Lemma 18 that $\det(G) = \tau_0^{2d}\det^2(\Sigma)$ and $\det(\Sigma_h) = (\tau_0^2-\tau_h^2)^d\det^2(\Sigma)$. Hence we have

$$\begin{aligned}\int_{\mathbb{R}^{2d}} F_h(x,y)\mathrm{d}(x,y) &= \frac{(2\pi)^d\tau_0^d\det(\Sigma)}{(2\pi)^{2d}(\tau_0^2-\tau_h^2)^d\det^2(\Sigma)}(2\pi)^d\tau_0^d\det(\Sigma) \\ &= \frac{\tau_0^{2d}}{(\tau_0^2-\tau_h^2)^d} = \frac{1}{(1-\rho_h^2)^d}.\end{aligned}$$

Under (D2) we have $\rho_h < 1$ for all $h > 0$ and there is a $\tilde{\rho} = \max_h|\rho_h| \leq 2^{-q} < 1$ and hence it holds $(1-\rho_h^2)^{-d} \leq d(1-4^{-q})^{-d-1}\rho_h^2$ and we are done. ■

B.5.2 Proof of the Theorem

First note that the the operator norm is dominated by the Hilbert-Schmidt norm. By Markov's inequality we have for $\nu \in (0,1]$

$$\begin{aligned}\mathrm{P}\left(\|S_n - S\|_{\mathrm{HS}}^2 \leq \nu^{-1}\mathrm{E}\|S_n - S\|_{\mathrm{HS}}^2\right) &\geq 1-\nu, \\ \mathrm{P}\left(\|T_n^*y - Sf\|_{\mathcal{H}}^2 \leq \nu^{-1}\mathrm{E}\|T_n^*y - Sf\|_{\mathcal{H}}^2\right) &\geq 1-\nu.\end{aligned}$$

(i) It holds due to $S_n = n^{-1}\sum_{t=1}^n k_t\otimes k_t$

$$\mathrm{E}\left(\|S_n - S\|_{\mathrm{HS}}^2\right) = \frac{1}{n^2}\sum_{t,s=1}^n\left(\mathrm{E}[\mathrm{tr}\{(k_t\otimes k_t)(k_s\otimes k_s)\}] - 2\mathrm{E}[\mathrm{tr}\{(k_0\otimes k_0)S\}] + \|S\|_{\mathrm{HS}}^2\right).$$

For the first summand we get $\mathrm{E}[\mathrm{tr}\{(k_t\otimes k_t)(k_s\otimes k_s)\}] = \mathrm{E}\{k^2(X_t,X_s)\}$, due to Lemma 17(i). Using the stationarity of $\{X_t\}_{t=1}^n$ and Lemma 17(iii) we get

$$\mathrm{E}\left(\|S_n - S\|_{\mathrm{HS}}^2\right) = \frac{1}{n}\left\{\mathrm{E}\{k^2(X_0,X_0)\} - \|S\|_{\mathrm{HS}}^2\right\} + 2\sum_{h=1}^{n-1}\frac{n-h}{n^2}\left[\mathrm{E}\{k^2(X_h,X_0)\} - \|S\|_{\mathrm{HS}}^2\right],$$

yielding the first result by an application of Lemma 17 (ii).
For the second equation we see due to the independence of $\{X_t\}_{t=1}^n$ and $\{\varepsilon_t\}_{t=1}^n$ that

$$\|T_n^*y - Sf\|_{\mathcal{H}}^2 = \sigma^2 n^{-1}\mathrm{E}\{k(X_0,X_0)\} + \mathrm{E}\left(\|S_nf - Sf\|_{\mathcal{H}}^2\right).$$

The rest follows along the same lines as the first part of the proof.

(ii) An application of part (i) of this theorem, Corollary 19 and Lemma 20 yields this result.

(iii) Because the $\{\varepsilon_t\}_{t\in\mathbb{Z}}$ are independent and identically distributed and $\{X_t\}_{t\in\mathbb{Z}}$ is stationary it holds

$$\begin{aligned}&\mathrm{E}\left\{\left\|(S+\lambda)^{-1/2}(S_nf - T_n^*y)\right\|_{\mathcal{H}}^2\right\} \\ &= \mathrm{E}\left\{\left\|(S+\lambda)^{-1/2}T_n^*\varepsilon\right\|_{\mathcal{H}}^2\right\} = n^{-2}\sum_{t,s=1}^n\mathrm{E}\left\{\left\langle\varepsilon_t(S+\lambda)^{-1}k_t, \varepsilon_sk_s\right\rangle_{\mathcal{H}}\right\} \\ &= n^{-1}\sigma^2\mathrm{E}\left\{\left\langle(S+\lambda)^{-1}k_0, k_0\right\rangle_{\mathcal{H}}\right\} = n^{-1}\sigma^2\mathrm{E}\left\{\left\|(S+\lambda)^{-1/2}k_0\right\|_{\mathcal{H}}^2\right\}.\end{aligned}$$

By the definition of d_λ we get

$$\mathrm{E}\{\|(S+\lambda)^{-1/2}k_0\|_{\mathcal{H}}^2\} = \mathrm{E}[\mathrm{tr}\{(S+\lambda)^{-1}k_0\otimes k_0\}] = \mathrm{tr}\{(S+\lambda)^{-1}S\} = d_\lambda.$$

Using $n^{-1/2} \leq \gamma_n(q)$ proves the result.

(iv) Consider first

$$\begin{aligned}\mathrm{E}\left\{\|(S+\lambda)^{-1/2}(S_n - S)\|_{\mathrm{HS}}^2\right\} &= n^{-2}\sum_{t,s=1}^n\mathrm{E}\left[\mathrm{tr}\{(S+\lambda)^{-1}(k_t\otimes k_t - S)(k_s\otimes k_s - S)\}\right] \\ &= n^{-1}\mathrm{E}\|(S+\lambda)^{-1/2}(k_0\otimes k_0 - S)\|_{\mathrm{HS}}^2 \\ &\quad + \sum_{h=1}^{n-1}\mathrm{E}\left[\mathrm{tr}\{(S+\lambda)^{-1}(k_0\otimes k_0 - S)(k_h\otimes k_h - S)\}\right].\end{aligned}$$

Continuing with the expression inside the sums we expand

$$\begin{aligned}\phi_h &= \mathrm{E}\left[\mathrm{tr}\{(S+\lambda)^{-1}(k_0\otimes k_0 - S)(k_h\otimes k_h - S)\}\right]\\ &= \mathrm{E}\left[\mathrm{tr}\{(S+\lambda)^{-1}k_0\otimes k_0 k_h\otimes k_h\}\right] - \mathrm{tr}\{(S+\lambda)^{-1}S^2\}\\ &= \mathrm{E}\left\{k(X_0,X_h)\langle(S+\lambda)^{-1}k_0,k_h\rangle_{\mathcal{H}}\right\} - \mathrm{tr}\{(S+\lambda)^{-1}S^2\}.\end{aligned}$$

Using Lemma 17 (iv) we see that

$$\mathrm{tr}\{(S+\lambda)^{-1}S^2\} = \int\limits_{\mathbb{R}^{2d}} k(x,y)\langle(S+\lambda)^{-1}k(\cdot,x),k(\cdot,y)\rangle_{\mathcal{H}}\mathrm{dP}^X(x)\mathrm{dP}^X(y).$$

Hence we have

$$\phi_h = \int\limits_{\mathbb{R}^{2d}} k(x,y)\langle(S+\lambda)^{-1}k(\cdot,x),k(\cdot,y)\rangle_{\mathcal{H}}\left\{\mathrm{dP}^{X_0,X_h}(x,y) - \mathrm{dP}^{X_0}(x)\mathrm{dP}^{X_0}(y)\right\}.$$

This can be bound by the results of Lemma 21 and together with Lemma 20 there exists a constant $C(q)>0$ such that with probability at least $1-\nu$

$$\|(S+\lambda)^{-1/2}(S_n - S)\|_{\mathcal{L}}^2 \leq \nu^{-1}C^2(q)\gamma_n^2(q)d_\lambda,$$

with $\gamma_n^2(q) = \begin{cases} n^{-1}, & q>1\\ n^{-1}\log(n), & q=1\\ n^{-q}, & q\in(0,1).\end{cases}$

This implies $\|(S+\lambda)^{-1/2}(S_n-S)(S+\lambda)^{-1/2}\|_{\mathcal{L}} \leq \nu^{-1/2}C(q)\lambda^{-1/2}\sqrt{d_\lambda}\gamma_n(q)$. Let $\lambda=\lambda_n$ be a sequence converging to zero such that $\lambda_n^{-1/2}\sqrt{d_{\lambda_n}}\gamma_n(q)\to 0$. Let n be large enough such that $\nu^{-1/2}C(q)\lambda_n^{-1/2}\sqrt{d_{\lambda_n}}\gamma_n(q) < 1$. Using Lemma A.5 in Blanchard and Krämer (2010b) we obtain

$$\|(S+\lambda)^{1/2}(S_n+\lambda)^{-1/2}\| \leq [1-\nu^{-1/2}C(q)\lambda_n^{-1/2}\sqrt{d_{\lambda_n}}\gamma_n(q)]^{-1/2} \leq \sqrt{2}.$$

The latter inequality can be fulfilled for n large enough such that $\nu^{-1/2}C(q)\lambda_n^{-1/2}\sqrt{d_{\lambda_n}}\gamma_n(q) \leq 1/2$. ■

B.6 Proof of Proposition 7

Recall that $Su = \mathrm{E}\{u(X_0)k(\cdot,X_0)\}$ for $u\in\mathcal{H}$. Define the independent random variables $Y_1,\ldots,Y_\mu$ that are all distributed as X_0.

First consider the following observation for $\mu\in\mathbb{N}$:

$$\begin{aligned}S^\mu u = S(S^{\mu-1}u) &= \mathrm{E}_{Y_1}\{(S^{\mu-1}u)(Y_1)k(\cdot,Y_1)\} = \mathrm{E}_{Y_2}\mathrm{E}_{Y_1}\{(S^{\mu-2}u)(Y_2)k(Y_1,Y_2)k(\cdot,Y_1)\}\\ &= \mathrm{E}_{Y_\mu}\cdots\mathrm{E}_{Y_1}\left\{\prod_{\nu=1}^{\mu-1}k(Y_\nu,Y_{\nu+1})u(Y_\mu)k(\cdot,Y_1)\right\}. \qquad (30)\end{aligned}$$

We take $u=\sum_{i=1}^\infty c_i k(\cdot,z_i)$ for $\{z_i\}_{i\in\mathbb{N}},\{c_i\}_{i\in\mathbb{N}}\subset\mathbb{R}$ such that $\|u\|_{\mathcal{H}}^2 = \sum_{i,j=1}^\infty c_i c_j k(z_i,z_j) \leq R^2$. The fact that a function $u\in\mathcal{H}$ can be represented as a linear combination of kernel functions is due to the Moore-Aronszajn Theorem, see Berlinet and Thomas-Agnan (2004).

Define the matrix $\Gamma = [\Gamma_{i,j}]_{i,j=1}^{\mu+2}\in\mathbb{R}^{(\mu+2)\times(\mu+2)}$ via

$$\Gamma_{i,j} = \begin{cases}\sigma_x^{-2}+2l & , \quad i=j=2,\ldots,\mu+1\\ l & , \quad i=j=1,\mu+2\\ -l & , \quad |i-j|=1\\ 0 & , \quad else\end{cases}.$$

Then we have via the integration of Gaussian functions and (30)

$$\begin{aligned}f(x) &= \frac{1}{(2\pi\sigma_x^2)^{\mu/2}}\sum_{i=1}^\infty c_i\int\limits_{\mathbb{R}^\mu}\exp\left\{-1/2(x,x_1,\ldots,x_\mu,z_i)\Gamma(x,x_1,\ldots,x_\mu,z_i)^{\mathrm{T}}\right\}\mathrm{d}(x_1,\ldots,x_\mu)\\ &= \frac{1}{\sigma_x^\mu\det(\Gamma_{2:\mu+1})^{1/2}}\sum_{i=1}^\infty c_i\exp\left[-1/2\frac{\det(\Lambda_{1:\mu+1})(x^2+z_i^2)-2l^{\mu+1}xz_i}{\det(\Gamma_{2:\mu+1})}\right].\end{aligned}$$

Here we used the symmetry property $\det(\Gamma_{2:\mu+2}) = \det(\Gamma_{1:\mu+1})$ as the first and last rows and columns of Γ are identical. This concludes the proof. ■

B.7 Proof of Proposition 8

In Shi et al. (2008) it was shown that the eigenvalues of S have the form $\mu_i = ab^{i-1}$, $i=1,2,\ldots$ with

$$a = \sqrt{2}(1+\beta+\sqrt{1+\beta})^{-1/2},\quad b = (1+\beta+\sqrt{1+2\beta})^{-1}\beta.$$

and $\beta = 4l\sigma_x^2$. It is clear that $0<b<1$ and hence $0<\mu_i\leq a$. We have $d_\lambda = \sum_{i=0}^\infty\{1+a^{-1}b^{-i}\lambda\}^{-1}$. Denote $f(x) = \{1+a^{-1}b^{-x}\lambda\}^{-1}$. We want to apply the integral test to the sum. We have $\int_0^\infty f(x)\mathrm{d}x = \log^{-1}(b^{-1})\log(1+a\lambda^{-1})$. This yields the bounds

$$\frac{\log(1+a/\lambda)}{\log(b^{-1})} \leq d_\lambda \leq \frac{1}{1+\lambda/a} + \frac{\log(1+a/\lambda)}{\log(b^{-1})}.$$

On $\lambda\in(0,1]$ we get $d_\lambda \leq D\log(1+a/\lambda)$ for a constant $D>0$. This can be seen as follows: The function $g_1(\lambda) = (1+\lambda/a)^{-1}$ is bounded from above by $C_1=1$ and the function $g_2(\lambda) = (b^{-1})\log(1+a/\lambda)$ is lower bounded by $c_2 = \log(1+a)$ and has no upper bound.

Hence on the set $I = \{\lambda\in(0,1] : g_2(\lambda)\geq C_1\}$ we can choose $C=2$. On the set I^c we have on the other hand $Cg_2(x)\geq c_2C\geq g_1(x)+g_2(x)$, hence we need $C = 2c_2^{-1}C_1 = 2\log^{-1}(1+a)$. The choice $D = 2\log^{-1}(b^{-1})\max\{1,\log^{-1}(1+a)\}$ is sufficient and we have $d_\lambda \leq D\log(1+a/\lambda)$, $\lambda\in(0,1]$. ■

References

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer Academic, Boston, 2004.

N. Bissantz, T. Hohage, A. Munk, and F. Ruymgaart. Convergence rates of general regularization methods for statistical inverse problems. *SIAM J. Numer. Anal.*, 45:2610–2636, 2007.

G. Blanchard and N. Krämer. Kernel partial least squares is universally consistent. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings*, volume 9, pages 57–64. JMLR, 2010a.

G. Blanchard and N. Krämer. Optimal learning rates for kernel conjugate gradient regression. *Adv. Neural Inf. Process. Syst.*, 23:226–234, 2010b.

P.J. Brockwell and R.A. Davis. *Time Series: Theory and Methods*. Springer, New York, 2 edition, 1991.

B. Brooks and M. Karplus. Harmonic dynamics of proteins: Normal modes and fluctuations in bovine pancreatic trypsin inhibitor. *Proc. Natl. Acad. Sci.*, 80:6571–6575, 1983.

A. Caponnetto and E. de Vito. Optimal rates for regularized least-squares algorithm. *Found. Comp. Math.*, 7:331–368, 2007.

F. Cucker and S. Smale. On the mathematical foundations of learning. *Bull. Amer. Math. Soc.*, 39:1–49, 2002.

E. de Vito, L. Rosasco, A. Caponnetto, U. de Giovanni, and F. Odone. Learning from examples as an inverse problem. *J. Mach. Learn. Res.*, 6:883–904, 2005.

E. De Vito, A. Caponnetto, and L. Rosasco. Discretization error analysis for Tikhonov regularization in learning theory. *Anal. Appl.*, 4:81–99, 2006.

L.H. Dicker, D.P. Foster, and D. Hsu. Kernel ridge vs. principal component regression: Minimax bounds and the qualification of regularization operators. *Electron. J. Stat.*, 11: 1022–1047, 2017.

I. Frank and J.H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

L. Giraitis, P. Kokoszka, R. Leipus, and G. Teyssière. Rescaled variance and related tests for long memory in volatility and levels. *J. Econometr.*, 112:265–294, 2003.

L. Giraitis, L.K. Hira, and D. Surgailis. *Large Sample Inference for Long Memory Processes*. Imperial College Press, London, 1 edition, 2012.

M. Hanke. *Conjugate Gradient Type Methods for Ill-posed Problems*. Wiley, New York, 1 edition, 1995.

I.S. Helland. On the structure of partial least squares regression. *Commun. Stat. Simul. Comput.*, 17(2):581–607, 1988.

K. Henzler-Wildman and D. Kern. Dynamic personalities of proteins. *Nature*, 450:964–972, 2007.

J.S. Hub and B.L. de Groot. Detection of functional modes in protein dynamics. *PLoS Comput. Biol.*, 5:1029–1044, 2009.

N. Krämer and M. L. Braun. Kernelizing PLS, degrees of freedom, and efficient model selection. In *Proceedings of the 24th International Conference on Machine Learning*, pages 441–448. ACM, 2007.

SB. Lin and DX. Zhou. Optimal learning rates for kernel partial least squares. *J. Fourier Anal. Appl.*, 2017.

F. Lindgren, P. Geladi, and S. Wold. The kernel algorithm for PLS. *J. Chemometr.*, 7: 45–59, 1993.

A.S. Nemirovskii. The regularizing properties of the adjoint gradient method in ill-posed problems. *Comput. Math. Math. Phys.*, 26:7–16, 1986.

G. Raskutti, M.J. Wainwright, and B. Yu. Early stopping and non-parametric regression: An optimal data-dependent stopping rule. *J. Mach. Learn. Res.*, 15:335–366, 2014.

R. Rosipal. Kernel partial least squares for nonlinear regression and discrimination. *Neural Netw. World*, 13:291–300, 2003.

R. Rosipal and L.J. Trejo. Kernel partial least squares regression in reproducing kernel Hilbert space. *J. Mach. Learn. Res.*, 2:97–123, 2001.

R. Rosipal, M. Girolami, and L.J. Trejo. Kernel PCA for feature extraction of event-related potentials for human signal detection performance. In *Proceedings of ANNIMAB-1 Conference*, pages 321–326. Springer, 2000.

R. Rosipall and N. Krämer. Overview and recent advances in partial least squares. In *Lecture Notes in Computer Science*, volume 3940, pages 34–51. Springer, 2006.

G. Samorodnitsky. *Long Range Dependence*. now Publisher, Hanover, 1 edition, 2007.

C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann Publishers, 1998.

B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, 1 edition, 2001.

B. Schölkopf, A. J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10:1299–1319, 1998.

B. Schölkopf, R. Herbrich, and A.J. Smola. A generalized representer theorem. In *International Conference on Computational Learning Theory*, pages 416–426. Springer, 2001.

T. Shi, M. Belkin, and B. Yu. Data spectroscopy: Learning mixture models using eigenspaces of convolution operators. In *Proceedings of the 25th International Conference on Machine Learning*, pages 936–943. Omnipress, 2008.

M. Singer, T. Krivobokova, B.L. de Groot, and A. Munk. Partial least squares for dependent data. *Biometrika*, 103:351–362, 2016.

I. Steinwart, D. Hush, and C. Scovel. An explicit description of the reproducing kernel Hilbert spaces of Gaussian RBF kernels. Technical report, IEEE Trans. Inform. Theory, 2005.

G. Wahba. Support vector machines, reproducing kernel Hilbert spaces and randomized GACV. In *Advances in Kernel Methods - Support Vector Learning*, pages 69–88. MIT Press, 1999.

S. Wold, A. Ruhe, H. Wold, and W.J. Dunn. The collinearity problem in linear regression. The partial least squares (PLS) approach to generalized inverses. *SIAM J. Sci. Comput.*, 5:735–743, 1984.

T. Zhang. Effective dimension and generalization of kernel learning. In *Advances in Neural Information Processing Systems 15*, pages 471–478. MIT Press, 2003.

Journal of Machine Learning Research 18 (2017) 1-40 Submitted 12/16; Revised 11/17; Published 12/17

Robust and Scalable Bayes via a Median of Subset Posterior Measures

Stanislav Minsker minsker@usc.edu
Department of Mathematics
University of Southern California
Los Angeles, CA 90089, USA

Sanvesh Srivastava sanvesh-srivastava@uiowa.edu
Department of Statistics and Actuarial Science
University of Iowa
Iowa City, IA 52242, USA

Lizhen Lin lizhen.lin@nd.edu
Department of Applied and Computational Mathematics and Statistics
The University of Notre Dame
Notre Dame, IN 46556, USA

David B. Dunson dunson@duke.edu
Departments of Statistical Science, Mathematics, and ECE
Duke University
Durham, NC 27708, USA

Editor: David Blei

Abstract

We propose a novel approach to Bayesian analysis that is provably robust to outliers in the data and often has computational advantages over standard methods. Our technique is based on splitting the data into non-overlapping subgroups, evaluating the posterior distribution given each independent subgroup, and then combining the resulting measures. The main novelty of our approach is the proposed aggregation step, which is based on the evaluation of a median in the space of probability measures equipped with a suitable collection of distances that can be quickly and efficiently evaluated in practice. We present both theoretical and numerical evidence illustrating the improvements achieved by our method.

Keywords: Big data, geometric median, distributed computing, parallel MCMC, Wasserstein distance

1. Introduction

Contemporary data analysis problems pose several general challenges. One is resource limitations: massive data require computer clusters for storage and processing. Another problem occurs when data are severely contaminated by "outliers" that are not easily identified and removed. Following Box and Tiao (1968), an outlier can be defined as *"being an observation which is suspected to be partially or wholly irrelevant because it is not generated by the stochastic model assumed."* While the topic of robust estimation has occupied an important place in the statistical literature for several decades and significant progress has been made in the theory of point estimation, robust Bayesian methods are not sufficiently well-understood.

Our main goal is to take a step towards solving these problems, proposing a general Bayesian approach that is

(i) provably robust to the presence of outliers in the data without any specific assumptions on their distribution or reliance on preprocessing;

(ii) scalable to big data sets through allowing computational algorithms to be implemented in parallel for different data subsets prior to an efficient aggregation step.

The proposed approach consists in splitting the sample into disjoint parts, implementing Markov chain Monte Carlo (MCMC) or another posterior sampling method to obtain draws from each "subset posterior" in parallel, and then using these draws to obtain weighted samples from the *median posterior (or M-Posterior)*, a new probability measure which is a (properly defined) median of a collection of subset posterior distributions. We show that, despite the loss of "interactions" among the data in different groups, the final result still admits strong guarantees; moreover, splitting the data gives certain advantages in terms of robustness to outliers.

In particular, we demonstrate that the M-Posterior is a probability measure centered at the "robust" estimator of the unknown parameter, the associated credible sets are often of the same "width" as the credible sets obtained from the usual posterior distribution and admit strong "frequentist" coverage guarantees (see Section 3.2 for exact statements).

The paper is organized as follows: Section 1.1 contains an overview of the existing literature and explains the goals that we aim to achieve in this work. Section 2 introduces the mathematical background and key facts used throughout the paper. Section 3 describes the main theoretical results for the median posterior. Section 4 presents details of algorithms, implementation, and numerical performance of the median posterior for several models. The simulation study and analysis of data examples convincingly show the robustness properties of the median posterior. In particular, we have used M-Posterior for scalable nonparametric Bayesian modeling of joint dependence in the multivariate categorical responses collected by the General Social Survey (`gss.norc.org`). Proofs that are omitted in the main text are contained in the supplementary material.

1.1 Discussion of Related Work

A. Dasgupta remarks that (see the discussion following the work by Berger, 1994): "Exactly what constitutes a study of Bayesian robustness is of course impossible to define." The popular definition which also indicates the main directions of research in this area is due to Berger (1994): "Robust Bayesian analysis is the study of the sensitivity of Bayesian answers

©2017 Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin and David B. Dunson.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v18/16-655.html`.

to uncertain inputs. These uncertain inputs are typically the model, prior distribution, or utility function, or some combination thereof." Notable works on robustness of Bayesian procedures to model misspecification include Doksum and Lo (1990) and Hoff (2007) that investigate methods based on conditioning on partial information, as well as a more recent paper by Miller and Dunson (2015) who introduced the notion of the "coarsened" posterior; however, its behavior in the presence of outliers is not explicitly addressed. Outliers are typically accommodated by either employing heavy-tailed likelihoods (e.g., Svensen and Bishop, 2005) or by attempting to identify and remove them as a first step (as in Box and Tiao, 1968 or Bayarri and Berger, 1994). The common assumption in the Bayesian literature is that the distribution of the outliers can be modeled (e.g., using a t-distribution, contamination by a larger variance parametric distribution, etc). In this paper, we instead bypass the need to place a model on the outliers and do not require their removal prior to analysis; similar approach has previously been advocated by Hooker and Vidyashankar (2014). We base inference on the *median posterior*, whose robustness can be formally and precisely quantified in terms of concentration properties around the true delta measure under the potential influence of outliers and contaminations of arbitrary nature.

Also relevant is the recent progress in scalable Bayesian algorithms. Most methods designed for distributed computing share a common feature: they efficiently use the data subset available to a single machine and combine the "local" results for "global" learning, while minimizing communication among cluster machines (Smola and Narayanamurthy, 2010). A wide variety of optimization-based approaches are available for distributed learning (Boyd et al., 2011); however, the number of similar Bayesian methods is limited. One of the reasons for this limitation is that Bayesian approaches typically require an approximation to the full posterior distribution instead of just a point estimate of parameters.

Several major approaches exist for scalable Bayesian learning in a distributed setting. The first approach independently evaluates the likelihood for each data subset across multiple machines and returns the likelihoods to a "master" machine, where they are appropriately combined with the prior using conditional independence assumptions of the probabilistic model. These two steps are repeated at every MCMC iteration (see Smola and Narayanamurthy, 2010; Agarwal and Duchi, 2012). This approach is problem-specific and involves extensive communication among machines. The second approach uses a so-called stochastic approximation (SA) and successively learns "noisy" approximations to the full posterior distribution using data in small mini-batches. A group of methods based on this approach uses sampling-based techniques to explore the posterior distribution through modified Hamiltonian or Langevin dynamics (e.g., Welling and Teh, 2011; Ahn et al., 2012; Korattikara et al., 2013). Unfortunately, these methods fail to accommodate discrete-valued parameters and multimodality. Another subgroup of methods uses deterministic variational approximations and learns the variational parameters of the approximated posterior through an optimization-based approach (see Wang et al., 2011; Hoffman et al., 2013; Broderick et al., 2013). Although these techniques often have excellent predictive performance, it is well known (Bishop, 2006) that variational methods tend to substantially underestimate posterior uncertainty and provide a poor characterization of posterior dependence, while lacking theoretical guarantees.

Our approach instead falls in a class of methods which avoid extensive communication among machines by running independent MCMC chains for each data subset and obtaining draws from subset posteriors. These subset posteriors can be combined in a variety of ways. Some of these methods simply average draws from each subset (Scott et al., 2013). Other alternatives use an approximation to the full posterior distribution based on kernel density estimates (Neiswanger et al., 2013) or the so-called Weierstrass transform (Wang and Dunson, 2013). These methods have limitations related to the dimension of the parameter, moreover, their applicability and theoretical justification are restricted to parametric models. Unlike the method proposed below, none of the aforementioned algorithms are provably robust. Another closely related method is the so-called WASP (Srivastava et al., 2015b). It is scalable but still lacks robustness guarantees.

Our work was inspired by recent multivariate median-based techniques for robust estimation developed in Minsker (2015) (see also Hsu and Sabato, 2013; Alon et al., 1996; Lerasle and Oliveira, 2011; Nemirovskiĭ and David, 1983 where similar ideas were applied in different frameworks).

2. Preliminaries

We proceed by recalling key definitions and facts which will be used throughout the paper, followed by the definition of the M-Posterior distribution in Section 2.4.

2.1 Notation

In what follows, $\|\cdot\|_2$ denotes the standard Euclidean distance in $\mathbb{R}^p$ and $\langle\cdot,\cdot\rangle_{\mathbb{R}^p}$ the associated dot product.

Given a totally bounded metric space $(\mathbb{Y}, d)$, the packing number $M(\varepsilon, \mathbb{Y}, d)$ is the maximal number N such that there exist N *disjoint* d-balls $B_1, \dots, B_N$ of radius ε contained in $\mathbb{Y}$, i.e., $\bigcup_{j=1}^{N} B_j \subseteq \mathbb{Y}$.

Let $\{p_\theta,\ \theta \in \Theta\}$ be a family of probability density functions on $\mathbb{R}^p$. Let $l, u : \mathbb{R}^p \mapsto \mathbb{R}_+$ be two functions such that $l(x) \le u(x)$ for every $x \in \mathbb{R}^p$ and $d^2(l,u) := \int_{\mathbb{R}^p} (\sqrt{u} - \sqrt{l})^2(x)dx < \infty$. A bracket $[l, u]$ consists of all functions $g : \mathbb{R}^p \mapsto \mathbb{R}$ such that $l(x) \le g(x) \le u(x)$ for all $x \in \mathbb{R}^p$. For $A \subseteq \Theta$, the bracketing number $N_{[]}(\varepsilon, A, d)$ is defined as the smallest number N such that there exist N brackets $[l_i, u_i],\ i = 1, \dots, N$ satisfying $\{p_\theta,\ \theta \in A\} \subseteq \bigcup_{i=1}^{N} [l_i, u_i]$ and $d(l_i, u_i) \le \varepsilon$ for all $1 \le i \le N$.

For $y \in \mathbb{Y}$, δ_y denotes the Dirac measure concentrated at y. In other words, for any Borel-measurable B, $\delta_y(B) = I\{y \in B\}$, where $I\{\cdot\}$ is the indicator function.

We will say that $k : \mathbb{Y} \times \mathbb{Y} \mapsto \mathbb{R}$ is a *kernel* if it is a symmetric, positive definite function. Assume that $(\mathbb{H}, \langle\cdot,\cdot\rangle_{\mathbb{H}})$ is a reproducing kernel Hilbert space (RKHS) of functions $f : \mathbb{Y} \mapsto$

$\mathbb{R}$. Then k is a *reproducing kernel* for $\mathbb{H}$ if for any $f \in \mathbb{H}$ and $y \in \mathbb{Y}$, $\langle f, k(\cdot, y)\rangle_{\mathbb{H}} = f(y)$ (see Aronszajn, 1950 for details).

For a square-integrable function $f \in L_2(\mathbb{R}^p)$, $\hat{f}$ stands for its Fourier transform. For $x \in \mathbb{R}$, $\lfloor x \rfloor$ denotes the largest integer not greater than x.

Finally, given two nonnegative sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n \lesssim b_n$ if $a_n \leq Cb_n$ for some $C > 0$ and all n. Other objects and definitions are introduced in the course of exposition when necessity arises.

2.2 Generalizations of the Univariate Median

Let $\mathbb{Y}$ be a normed space with norm $\|\cdot\|$, and let μ be a probability measure on $(\mathbb{Y}, \|\cdot\|)$ equipped with Borel σ-algebra. Define the *geometric median* of μ by

$$x_* = \operatorname*{argmin}_{y \in \mathbb{Y}} \int_{\mathbb{Y}} (\|y - x\| - \|x\|)\, \mu(dx).$$

In this paper, we focus on the special case when μ is a uniform distribution on a finite collection of atoms $x_1, \ldots, x_m \in \mathbb{Y}$, so that

$$x_* = \operatorname{med}_g(x_1, \ldots, x_m) := \operatorname*{argmin}_{y \in \mathbb{Y}} \sum_{j=1}^{m} \|y - x_j\|. \tag{1}$$

The geometric median exists under rather general assumptions; for example, if $\mathbb{Y}$ is a Hilbert space (this case will be our main focus; more general conditions were obtained by Kemperman, 1987). Moreover, it is well-known that in this situation $x_* \in \operatorname{co}(x_1, \ldots, x_m)$—the convex hull of $x_1, \ldots, x_m$ (meaning that there exist nonnegative α_j, $j = 1 \ldots m$, $\sum_{j=1}^{m} \alpha_j = 1$ such that $x_* = \sum_{j=1}^{m} \alpha_j x_j$).

Another useful generalization of the univariate median is defined as follows. Let $(\mathbb{Y}, d)$ be a metric space with metric d, and $x_1, \ldots, x_k \in \mathbb{Y}$. Define B_* to be the d-ball of minimal radius such that it is centered at one of $\{x_1, \ldots, x_m\}$ and contains at least half of these points. Then the median $\operatorname{med}_0(x_1, \ldots, x_m)$ of $x_1, \ldots, x_m$ is the center of B_*. In other words, let

$$\varepsilon_* := \inf\Big\{\varepsilon > 0 :\ \exists j = j(\varepsilon) \in \{1, \ldots, m\} \text{ and } I(j) \subset \{1, \ldots, m\} \text{ such that } |I(j)| > \frac{m}{2} \text{ and } \forall i \in I(j),\ d(x_i, x_j) \leq 2\varepsilon\Big\}, \tag{2}$$

$j_* := j(\varepsilon_*)$, where ties are broken arbitrarily, and set

$$x_* = \operatorname{med}_0(x_1, \ldots, x_m) := x_{j_*}. \tag{3}$$

We will say that x_* is the *metric median* of $x_1, \ldots, x_m$. Note that x_* always belongs to $\{x_1, \ldots, x_m\}$ by definition. Advantages of this definition are its generality (only metric space structure is assumed) and simplicity of numerical evaluation since only the pairwise

distances $d(x_i, x_j)$, $i, j = 1, \ldots, m$ are required to compute the median. This construction was previously employed by Nemirovskiĭ and David (1983) in the context of stochastic optimization and is further studied by Hsu and Sabato (2013). A closely related notion of the median was used by Lopuhaa and Rousseeuw (1991) under the name of the "minimal volume ellipsoid" estimator.

Finally, we recall an important property of the median (shared both by med_g and med_0) which states that it transforms a collection of independent, "weakly concentrated" estimators into a single estimator with significantly stronger concentration properties. Given q, α such that $0 < q < \alpha < 1/2$, define a nonnegative function $\psi(\alpha, q)$ via

$$\psi(\alpha, q) := (1 - \alpha) \log \frac{1 - \alpha}{1 - q} + \alpha \log \frac{\alpha}{q}. \tag{4}$$

The following result is an adaptation of Theorem 3.1 in (Minsker, 2015):

Theorem 1

a *Assume that $(\mathbb{H}, \|\cdot\|)$ is a Hilbert space and $\theta_0 \in \mathbb{H}$. Let $\hat{\theta}_1, \ldots, \hat{\theta}_m \in \mathbb{H}$ be a collection of independent random variables. Let κ be a constant satisfying $0 \leq \kappa < \frac{1}{3}$. Suppose $\varepsilon > 0$ is such that for all j, $1 \leq j \leq \lfloor (1 - \kappa)m \rfloor + 1$,*

$$\Pr\Big(\|\hat{\theta}_j - \theta_0\| > \varepsilon\Big) \leq \frac{1}{7}. \tag{5}$$

Let $\hat{\theta}_* = \operatorname{med}_g(\hat{\theta}_1, \ldots, \hat{\theta}_m)$ be the geometric median of $\{\hat{\theta}_1, \ldots, \hat{\theta}_m\}$. Then

$$\Pr\Big(\|\hat{\theta}_* - \theta_0\| > 1.52\varepsilon\Big) \leq \left[e^{(1-\kappa)\psi\left(\frac{3/7 - \kappa}{1 - \kappa}, 1/7\right)}\right]^{-m}.$$

b *Assume that $(\mathbb{Y}, d)$ is a metric space and $\theta_0 \in \mathbb{Y}$. Let $\hat{\theta}_1, \ldots, \hat{\theta}_m \in \mathbb{Y}$ be a collection of independent random variables. Let κ be a constant satisfying $0 \leq \kappa < \frac{1}{3}$. Suppose $\varepsilon > 0$ is such that for all j, $1 \leq j \leq \lfloor (1 - \kappa)m \rfloor + 1$,*

$$\Pr\Big(d(\hat{\theta}_j, \theta_0) > \varepsilon\Big) \leq \frac{1}{4}. \tag{6}$$

Let $\hat{\theta}_* = \operatorname{med}_0(\hat{\theta}_1, \ldots, \hat{\theta}_m)$. Then

$$\Pr\Big(d(\hat{\theta}_*, \theta_0) > 3\varepsilon\Big) \leq \left[e^{(1-\kappa)\psi\left(\frac{1/2 - \kappa}{1 - \kappa}, 1/4\right)}\right]^{-m}.$$

Proof See Section A.1 in the appendix. ■

Remark 2 *While we require $\kappa < 1/3$ above for clarity and to keep the constants small, we prove a slightly more general result that holds for any $\kappa < 1/2$.*

Theorem 1 implies that the concentration of the geometric median of independent estimators around the "true" parameter value improves geometrically fast with respect to the number of such estimators, while the estimation rate is preserved, up to a constant. In our case, the role of $\hat{\theta}_j$'s will be played by posterior distributions based on disjoint subsets of observations, viewed as elements of the space of signed measures equipped with a suitable distance.

Parameter κ allows taking corrupted observations into account: if the initial sample contains not more than $\lfloor \kappa m \rfloor$ outliers (of arbitrary nature), then at most $\lfloor \kappa m \rfloor$ estimators amongst $\{\theta_1, \dots, \theta_m\}$ can be affected but their median remains stable, still being close to the unknown θ_0 with high probability. To clarify the notion of "robustness" that such a statement provides, assume that $\hat{\theta}_1, \dots, \hat{\theta}_m$ are consistent estimators of θ_0 based on disjoint samples of size n/m each. If $\frac{n}{m} \to \infty$, then $\frac{\kappa m}{n} \to 0$, hence the breakdown point of the estimator $\hat{\theta}_*$ is 0 is general. However, it is able to handle a number of outliers that grows like $o(n)$ while preserving consistency, which is the best one can hope for without imposing any additional assumptions on the underlying distribution, parameter of interest or nature of the outliers.

Let us also mention that the the geometric median of a collection of points in a Hilbert space belongs to the convex hull of these points. Thus, one can think about "downweighing" some observations (potential outliers) and increasing the weight of others, and geometric median gives a way to formalize this approach. The median med_0 defined in (3) corresponds to the extreme case when all but one weight are equal to 0. Its potential advantage lies in the fact that its evaluation requires only the knowledge of pairwise distances $d(\hat{\theta}_i, \hat{\theta}_j),\ i, j = 1, \dots, m$, see (2).

2.3 Distances Between Probability Measures

Next, we discuss the special family of distances between probability measures that will be used throughout the paper. These distances provide the necessary structure to define and evaluate medians in the space of measures, as discussed above. Since one of our goals was to develop computationally efficient techniques, we focus on distances that admit accurate numerical approximation.

Assume that $(\mathbb{X}, \rho)$ is a separable metric space, and let $\mathcal{F} = \{f : \mathbb{X} \mapsto \mathbb{R}\}$ be a collection of real-valued functions. Given two Borel probability measures P, Q on $\mathbb{X}$, define

$$\|P - Q\|_{\mathcal{F}} := \sup_{f \in \mathcal{F}} \left| \int_{\mathbb{X}} f(x) d(P - Q)(x) \right|. \tag{7}$$

Important special cases include the situation when

$$\mathcal{F} = \mathcal{F}_L := \{f : \Theta \mapsto \mathbb{R} \text{ s.t. } \|f\|_L \le 1\}, \tag{8}$$

where $\|f\|_L := \sup_{x_1 \neq x_2} \frac{|f(x_1) - f(x_2)|}{\rho(x_1, x_2)}$ is the Lipschitz constant of f.

It is well-known (Dudley, 2002, Theorem 11.8.2) that in this case $\|P - Q\|_{\mathcal{F}_L}$ is equal to the Wasserstein distance (also known as the Kantorovich-Rubinstein distance)

$$d_{W_{1,\rho}}(P, Q) = \inf \Big\{ \mathbb{E}\rho(\boldsymbol{X}, \boldsymbol{Y}) : \mathcal{L}(\boldsymbol{X}) = P,\ \mathcal{L}(\boldsymbol{Y}) = Q \Big\}, \tag{9}$$

where $\mathcal{L}(\boldsymbol{Z})$ denotes the law of a random variable $\boldsymbol{Z}$ and the infimum on the right is taken over the set of all joint distributions of $(\boldsymbol{X}, \boldsymbol{Y})$ with marginals P and Q.

Another fruitful structure emerges when $\mathcal{F}$ is a unit ball in a Reproducing Kernel Hilbert Space $(\mathbb{H}, \langle \cdot, \cdot \rangle_{\mathbb{H}})$ with a reproducing kernel $k : \mathbb{X} \times \mathbb{X} \mapsto \mathbb{R}$. That is,

$$\mathcal{F} = \mathcal{F}_k := \{f : \mathbb{X} \mapsto \mathbb{R},\ \|f\|_{\mathbb{H}} := \sqrt{\langle f, f \rangle_{\mathbb{H}}} \le 1\}. \tag{10}$$

Let $\mathcal{P}_k := \{P \text{ is a probability measure, } \int_{\mathbb{X}} \sqrt{k(x,x)} dP(x) < \infty\}$, and assume that $P, Q \in \mathcal{P}_k$. Theorem 1 proven by Sriperumbudur et al. (2010) implies that the corresponding distance between measures P and Q takes the form

$$\|P - Q\|_{\mathcal{F}_k} = \left\| \int_{\mathbb{X}} k(x, \cdot) d(P - Q)(x) \right\|_{\mathbb{H}}. \tag{11}$$

It follows that $P \mapsto \int_{\mathbb{X}} k(x, \cdot) dP(x)$ is an embedding of $\mathcal{P}_k$ into the Hilbert space $\mathbb{H}$ which can be seen as an application of the "kernel trick" in our setting. The Hilbert space structure allows one to use fast numerical methods to approximate the geometric median, see Section 4 below.

Remark 3 *Note that when P and Q are discrete measures (e.g., $P = \sum_{j=1}^{N_1} \beta_j \delta_{z_j}$ and $Q = \sum_{j=1}^{N_2} \gamma_j \delta_{y_j}$), then*

$$\|P - Q\|^2_{\mathcal{F}_k} = \sum_{i,j=1}^{N_1} \beta_i \beta_j k(z_i, z_j) + \sum_{i,j=1}^{N_2} \gamma_i \gamma_j k(y_i, y_j) - 2 \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \beta_i \gamma_j k(z_i, y_j). \tag{12}$$

In this paper, we will only consider *characteristic* kernels, which means that $\|P - Q\|_{\mathcal{F}_k} = 0$ if and only if $P = Q$. It follows from Theorem 7 in Sriperumbudur et al. (2010) that a sufficient condition for k to be characteristic is its *strict positive definiteness*: we say that k is *strictly positive definite* if it is bounded, measurable, and such that for all non-zero signed Borel measures ν

$$\iint_{\mathbb{X} \times \mathbb{X}} k(x, y) d\nu(x) d\nu(y) > 0.$$

When $\mathbb{X} = \mathbb{R}^p$, a simple sufficient criterion for the kernel k to be characteristic follows from Theorem 9 in Sriperumbudur et al. (2010):

Proposition 4 *Let $\mathbb{X} = \mathbb{R}^p,\ p \ge 1$. Assume that $k(x, y) = \phi(x - y)$ for some bounded, continuous, integrable, positive-definite function $\phi : \mathbb{R}^p \mapsto \mathbb{R}$.*

1. *Let $\widehat{\phi}$ be the Fourier transform of ϕ. If $|\widehat{\phi}(x)| > 0$ for all $x \in \mathbb{R}^p$, then k is characteristic;*

2. If ϕ is compactly supported, then k is characteristic.

Remark 5 *It is important to mention that in practical applications, we often deal with empirical measures based on a collection of MCMC samples from the posterior distribution. A natural question is the following: if P and Q are probability measures on $\mathbb{R}^D$ and P_m, Q_n are their empirical versions, what is the size of the error*

$$e_{m,n} := \left| \|P - Q\|_{\mathcal{F}_k} - \|P_m - Q_n\|_{\mathcal{F}_k} \right| ?$$

For i.i.d samples, a useful and favorable fact is that $e_{m,n}$ often does not depend on D: under weak assumptions on kernel k, $e_{m,n}$ has an upper bound of order $m^{-1/2} + n^{-1/2}$ (that is, $\lim_{m,n\to\infty} \Pr\left(e_{m,n} \geq C(m^{-1/2} + n^{-1/2})\right)$ can be made arbitrarily small by choosing C big enough, see Corollary 12 in Sriperumbudur et al., 2009). On the other hand, the bound for the (stronger) Wasserstein distance is not dimension-free and is of order $m^{-1/(D+1)} + n^{-1/(D+1)}$. Similar error rates hold for empirical measures based on samples from Markov chains used to approximate invariant distributions, including MCMC samples (see Boissard and Le Gouic, 2014 and Fournier and Guillin, 2015).

If $\mathbb{X}$ is a separable Hilbert space with dot product $\langle \cdot, \cdot \rangle_{\mathbb{X}}$ and P_1, P_2 are probability measures with

$$\int_{\mathbb{X}} \|x\|_{\mathbb{X}} dP_i(x) < \infty, \ i = 1, 2,$$

it will be useful to assume that the class $\mathcal{F}$ is chosen such that the distance between the measures is lower bounded by the distance between their means, namely

$$\left\| \int_{\mathbb{X}} x dP_1(x) - \int_{\mathbb{X}} x dP_2(x) \right\|_{\mathbb{X}} \leq C \|P_1 - P_2\|_{\mathcal{F}} \tag{13}$$

for some absolute constant $C > 0$. Clearly, this holds if $\mathcal{F}$ contains the set of continuous linear functionals $\mathbb{L} = \{x \mapsto \langle u, x \rangle_{\mathbb{X}}, \ u \in \mathbb{X}, \ \|u\|_{\mathbb{X}} \leq 1/C\}$, since

$$\left\| \int_{\mathbb{X}} x dP_i(x) \right\|_{\mathbb{X}} = \sup_{\|u\|_{\mathbb{X}} \leq 1} \int_{\mathbb{X}} \langle x, u \rangle_{\mathbb{X}} \, dP_i(x), \ i = 1, 2.$$

In particular, this is true for the Wasserstein distance $d_{W_{1,\rho}}(\cdot, \cdot)$ defined with respect to the metric ρ such that $\rho(x, y) \geq c_1 \|x - y\|_{\mathbb{X}}$. Next, we will state a simple sufficient condition on the kernel $k(\cdot, \cdot)$ for (13) to hold for the unit ball $\mathcal{F}_k$.

Proposition 6 *Let $\mathbb{X}$ be a separable Hilbert space, $k_0 : \mathbb{X} \times \mathbb{X} \mapsto \mathbb{R}$ - a characteristic kernel, and define*

$$k(x, y) := k_0(x, y) + \langle x, y \rangle_{\mathbb{X}} .$$

Then k is characteristic and satisfies (13) with $C = 1$.

Proof Let $\mathbb{H}_1$ and $\mathbb{H}_2$ be two reproducing kernel Hilbert spaces with kernels k_1 and k_2 respectively. It is well-known (e.g., Aronszajn, 1950) that the space corresponding to kernel $k = k_1 + k_2$ is

$$\mathbb{H} = \{f = f_1 + f_2, \ f_1 \in \mathbb{H}_1, \ f_2 \in \mathbb{H}_2\}$$

with the norm $\|f\|^2_{\mathbb{H}} = \inf\{\|f_1\|^2_{\mathbb{H}} + \|f_2\|^2_{\mathbb{H}}, \ f_1 + f_2 = f\}$. Hence, the unit ball of $\mathbb{H}$ contains the unit balls of $\mathbb{H}_1$ and $\mathbb{H}_2$, so that for any probability measures P, Q

$$\|P - Q\|_{\mathcal{F}_k} \geq \max\left(\|P - Q\|_{\mathcal{F}_{k_1}}, \|P - Q\|_{\mathcal{F}_{k_2}} \right),$$

which easily implies the result. ■

The kernels of the form $k(x, y) = k_0(x, y) + \langle x, y \rangle_{\mathbb{X}}$ will prove especially useful in the situation when the parameter of interest is finite-dimensional (see Section 3.2 for details).

Finally, we recall the definition of the well-known Hellinger and total variation distances. Assume that P and Q are probability measures on $\mathbb{R}^D$ which are absolutely continuous with respect to Lebesgue measure with densities p and q respectively. Then the Hellinger distance between P and Q is given by

$$h(P, Q) := \sqrt{\frac{1}{2} \int_{\mathbb{R}^D} \left(\sqrt{p(x)} - \sqrt{q(x)} \right)^2 dx}.$$

The total variation distance between two probability measures defined on a σ-algebra $\mathfrak{B}$ is

$$\|P - Q\|_{\mathrm{TV}} = \sup_{B \in \mathfrak{B}} |P(B) - Q(B)|.$$

We are ready to introduce the *median posterior* (or M-Posterior) distribution.

2.4 Construction of the M-Posterior Distribution

Let $\{P_\theta, \ \theta \in \Theta\}$ be a family of probability distributions over $\mathbb{R}^D$ indexed by Θ. Suppose that for all $\theta \in \Theta$, P_θ is absolutely continuous with respect to Lebesgue measure dx on $\mathbb{R}^D$ with $dP_\theta(\cdot) = p_\theta(\cdot) dx$. In what follows, we equip Θ with a metric (that we will refer to as the "Hellinger metric")

$$\rho(\theta_1, \theta_2) := h(P_{\theta_1}, P_{\theta_2}), \tag{14}$$

and assume that the metric space (Θ, ρ) is separable.

Let k be a characteristic kernel defined on $\Theta \times \Theta$. Kernel k defines a metric on Θ via

$$\rho_k(\theta_1, \theta_2) := \|k(\cdot, \theta_1) - k(\cdot, \theta_2)\|_{\mathbb{H}} = \Big(k(\theta_1, \theta_1) + k(\theta_2, \theta_2) - 2k(\theta_1, \theta_2) \Big)^{1/2}, \tag{15}$$

where $\mathbb{H}$ is the RKHS associated to k. We will assume that (Θ, ρ_k) is separable. Note that the Hellinger metric $\rho(\theta_1, \theta_2)$ is a particular case corresponding to the kernel

$$k_H(\theta_1, \theta_2) := \left\langle \sqrt{p_{\theta_1}}, \sqrt{p_{\theta_2}} \right\rangle_{L_2(dx)} .$$

All subsequent results apply to this special case. While this is a natural metric for the problem, the disadvantage of $k_H(\cdot, \cdot)$ is that it is often difficult to evaluate numerically. Instead, we will consider metrics ρ_k that are dominated by ρ (this is formalized in assumption 1).

Let $X_1, \dots, X_n$ be i.i.d. $\mathbb{R}^D$-valued random vectors defined on a probability space $(\Omega, \mathcal{B}, P)$ with unknown distribution $P_0 := P_{\theta_0}$ for some $\theta_0 \in \Theta$. Bayesian inference of P_0 requires specifying a prior distribution Π over Θ (equipped with the Borel σ-algebra induced by ρ). The posterior distribution given the observations $\mathcal{X}_n := \{X_1, \dots, X_n\}$ is a random probability measure on Θ defined by

$$\Pi_n(B|\mathcal{X}_n) := \frac{\int\limits_B \prod_{i=1}^n p_\theta(X_i) d\Pi(\theta)}{\int\limits_\Theta \prod_{i=1}^n p_\theta(X_i) d\Pi(\theta)}$$

for all Borel measurable sets $B \subseteq \Theta$. It is known (see Ghosal et al., 2000) that under rather general assumptions the posterior distribution Π_n *contracts* towards θ_0, meaning that

$$\Pi_n(\theta \in \Theta : \rho(\theta, \theta_0) \geq \varepsilon_n | \mathcal{X}_n) \to 0$$

almost surely or in probability as $n \to \infty$ for a suitable sequence $\varepsilon_n \to 0$.

One of the questions that we address can be formulated as follows: what happens if *some* observations in $\mathcal{X}_n$ are corrupted, e.g., if $\mathcal{X}_n$ contains outliers of arbitrary nature and magnitude? Even if there is only one outlier, the usual posterior distribution might concentrate most of its mass far from the true value θ_0.

We proceed with a general description of our proposed algorithm for constructing a robust version of the posterior distribution. Let $1 \leq m \leq n/2$ be an integer. Divide the sample $\mathcal{X}_n$ into m disjoint groups $G_1, \dots, G_m$ of size $|G_j| \geq \lfloor n/m \rfloor$ each:

$$\{X_1, \dots, X_n\} = \bigcup_{j=1}^m G_j, \ G_i \cap G_l = \emptyset \text{ for } i \neq j, \ |G_j| \geq \lfloor n/m \rfloor, \ j = 1 \dots m.$$

A good choice of m efficiently exploits the available computational resource while ensuring that the groups G_js are sufficiently large.

Let Π be a prior distribution over Θ, and let

$$\left\{ \Pi^{(j)}(\cdot) := \Pi_{|G_j|}(\cdot|G_j), \ j = 1, \dots, m \right\}$$

be the family of subset posterior distributions depending on disjoint subgroups G_j, $j = 1, \dots, m$:

$$\Pi_{|G_j|}(B|G_j) := \frac{\int\limits_B \prod_{i \in G_j} p_\theta(X_i) d\Pi(\theta)}{\int\limits_\Theta \prod_{i \in G_j} p_\theta(X_i) d\Pi(\theta)}.$$

Define the M-Posterior as

$$\hat{\Pi}_{n,g} := \text{med}_g(\Pi^{(1)}, \dots, \Pi^{(m)}), \tag{16}$$

or

$$\hat{\Pi}_{n,0} := \text{med}_0(\Pi^{(1)}, \dots, \Pi^{(m)}), \tag{17}$$

where the medians $\text{med}_g(\cdot)$ and $\text{med}_0(\cdot)$ are evaluated with respect to $\|\cdot\|_{\mathcal{F}_L}$ or $\|\cdot\|_{\mathcal{F}_k}$ introduced in Section 2.2 above. Note that $\hat{\Pi}_{n,g}$ and $\hat{\Pi}_{n,0}$ are always probability measures: indeed, due to the aforementioned properties of a geometric median, there exists $\alpha_1 \geq 0, \dots, \alpha_m \geq 0, \ \sum\limits_{j=1}^m \alpha_j = 1$ such that $\hat{\Pi}_{n,g} = \sum\limits_{j=1}^m \alpha_j \Pi^{(j)}$, and $\hat{\Pi}_{n,0} \in \{\Pi^{(1)}(\cdot), \dots, \Pi^{(m)}(\cdot)\}$ by definition.

While $\hat{\Pi}_{n,g}$ and $\hat{\Pi}_{n,0}$ possess several nice properties (such as robustness to outliers), in practice they often overestimate the uncertainty about θ_0, especially when the number of groups m is large: indeed, if for example $\theta \in \mathbb{R}$ and Bernstein-von Mises theorem holds, then each $\Pi_{|G_j|}(\cdot|G_j)$ is "approximately normal" with covariance $\frac{m}{n} I^{-1}(\theta_0)$ (here, $I(\theta_0)$ is the Fisher information). However, the asymptotic covariance of the posterior distribution based on the whole sample is $\frac{1}{n} I^{-1}(\theta_0)$.

To overcome this difficulty, we propose a modification of our approach where the random measures $\Pi_n^{(j)}$ are replaced by the *stochastic approximations* $\Pi_{|G_j|,m}(\cdot|G_j), \ j = 1, \dots, m$ of the full posterior distribution. To this end, define the "stochastic approximation" based on the subsample G_j as

$$\Pi_{|G_j|,m}(B|G_j) := \frac{\int\limits_B \left(\prod_{i \in G_j} p_\theta(X_i) \right)^m d\Pi(\theta)}{\int\limits_\Theta \left(\prod_{i \in G_j} p_\theta(X_i) \right)^m d\Pi(\theta)}, \tag{18}$$

where we assume that $p_\theta^m(\cdot)$ is an integrable function for all θ. In other words, $\Pi_{|G_j|,m}(\cdot|G_j)$ is obtained as a posterior distribution given that each data point from G_j is observed m times. While each of $\Pi_{|G_j|,k}(\cdot|G_j)$ might underestimate uncertainly, the median $\hat{\Pi}_{n,g}^{\text{st}}$ (or $\hat{\Pi}_{n,0}^{\text{st}}$) of these random measures yields credible sets with much better coverage. This approach shows good performance in numerical experiments. One of our main results (see Section 3.2) provides a justification for this observation, albeit, under rather strong assumptions and for the parametric case.

3. Theoretical Analysis

We proceed with a discussion of the theoretical guarantees for the M-Posterior, starting with the contraction rates and robustness properties.

3.1 Convergence of Posterior Distribution and Robust Bayesian Inference

Our first result establishes the "weak concentration" property of the posterior distribution around the true parameter. Let $\delta_0 := \delta_{\theta_0}$ be the Dirac measure supported on $\theta_0 \in \Theta$. Recall the following version of Theorem 2.1 in Ghosal et al. (2000) (we state the result for the Wasserstein distance $d_{W_{1,\rho}}(\Pi_n(\cdot|\mathcal{X}_l), \delta_0)$ rather than the closely related contraction rate of the posterior distribution). Here, the Wasserstein distance is evaluated with respect to the "Hellinger metric" $\rho(\cdot, \cdot)$ defined in (14).

Theorem 7 *Let $\mathcal{X}_l = \{X_1, \dots, X_l\}$ be an i.i.d. sample from P_0. Assume that $\varepsilon_l > 0$ and $\Theta_l \subset \Theta$ are such that for some constant $C > 0$*

(1) *the packing number satisfies* $\log M(\varepsilon_l, \Theta_l, \rho) \le l\varepsilon_l^2$,

(2) $\Pi(\Theta \setminus \Theta_l) \le \exp(-l\varepsilon_l^2(C+4))$,

(3) $\Pi\left(\theta : \ -P_0\left(\log \frac{p_\theta}{p_0}\right) \le \varepsilon_l^2, \ P_0\left(\log \frac{p_\theta}{p_0}\right)^2 \le \varepsilon_l^2\right) \ge \exp(-Cl\varepsilon_l^2)$.

Then there exists $R = R(C)$ and a universal constant $\tilde{K}$ such that

$$\Pr\left(d_{W_{1,\rho}}(\delta_0, \Pi_l(\cdot|\mathcal{X}_l)) \ge R\varepsilon_l + e^{-\tilde{K}l\varepsilon_l^2}\right) \le \frac{1}{l\varepsilon_l^2} + 4e^{-\tilde{K}l\varepsilon_l^2}. \tag{19}$$

Proof The proof closely mimics the argument behind Theorem 2.1 in (Ghosal et al., 2000). For reader's convenience, details are outlined in Section A.2 of the appendix. ■

Conditions of Theorem 7 are standard assumptions guaranteeing that the resulting posterior distribution contracts to the true parameter θ_0 at the rate ε_n. Note that the bounds for the distance $d_{W_{1,\rho}}(\delta_0, \Pi_l(\cdot|\mathcal{X}_l)$ slightly differ from the contraction rate itself: indeed, we have

$$d_{W_{1,\rho}}(\delta_0, \Pi_l(\cdot|\mathcal{X}_l)) \le \varepsilon_l + \int\limits_{h(P_\theta, P_0) \ge \varepsilon_l} d\Pi_l(\cdot|\mathcal{X}_l),$$

hence to obtain the inequality $d_{W_{1,\rho}}(\delta_0, \Pi_l(\cdot|\mathcal{X}_l)) \lesssim \varepsilon_l$, we usually require $\int\limits_{h(P_\theta, P_0) \ge \varepsilon_l} d\Pi_l(\cdot|\mathcal{X}_l) \lesssim \varepsilon_l$, which adds an extra logarithmic factor in the parametric case.

Combination of Theorems 7 and 1 immediately yields the corollary for $\hat{\Pi}_{n,0}$. Let $\mathbb{H}$ be the reproducing kernel Hilbert space with the reproducing kernel

$$k_H(\theta_1, \theta_2) = \frac{1}{2}\left\langle \sqrt{p_{\theta_1}}, \sqrt{p_{\theta_2}} \right\rangle_{L_2(dx)}.$$

Let $f \in \mathbb{H}$ and note that, due to the reproducing property and Cauchy-Schwarz inequality, we have

$$\begin{aligned} f(\theta_1) - f(\theta_2) &= \langle f, k_H(\cdot, \theta_1) - k_H(\cdot, \theta_2)\rangle_{\mathbb{H}} \\ &\le \|f\|_{\mathbb{H}} \left\|k_H(\cdot, \theta_1) - k_H(\cdot, \theta_2)\right\|_{\mathbb{H}} = \|f\|_{\mathbb{H}} \, \rho(\theta_1, \theta_2). \end{aligned} \tag{20}$$

Therefore, $\mathcal{F}_k \subseteq \mathcal{F}_L$ and $\|P - Q\|_{\mathcal{F}_k} \le \|P - Q\|_{\mathcal{F}_L}$, where $\mathcal{F}_k$ and $\mathcal{F}_L$ were defined in (10) and (8) respectively, and the underlying metric structure is given by ρ. In particular, convergence with respect to $\|\cdot\|_{\mathcal{F}_L}$ implies convergence with respect to $\|\cdot\|_{\mathcal{F}_k}$.

Corollary 8 *Let $X_1, \dots, X_n$ be an i.i.d. sample from P_0, and assume that $\hat{\Pi}_{n,g}$ is defined with respect to the norm $\|\cdot\|_{\mathcal{F}_L}$ as in (17) above. Set $l := \lfloor n/m \rfloor$, assume that conditions of Theorem 7 hold, and, moreover, that ε_l satisfies*

$$\frac{1}{l\varepsilon_l^2} + 4e^{-(1+K/2)l\varepsilon_l^2/2} < \frac{1}{7}.$$

Then

$$\Pr\left(\left\|\delta_0 - \hat{\Pi}_{n,g}\right\|_{\mathcal{F}_{k_H}} \ge 1.52\left(R\varepsilon_l + e^{-\tilde{K}l\varepsilon_l^2}\right)\right) \le \left[e^{\psi(3/7,1/7)}\right]^{-m} < 1.27^{-m}.$$

Proof It is enough to apply part (a) of Theorem 1 with $\kappa = 0$ to the independent random measures $\Pi_n(\cdot|G_j)$, $j = 1, \dots, m$. Note that the "weak concentration" assumption (29) is implied by (19). ■

Once again, note the exponential improvement of concentration as compared to Theorem 7. It is easy to see that a similar statement holds for the median $\hat{\Pi}_{n,0}(\cdot)$ defined in (17) (even for the stronger Wasserstein distance $d_{W_{1,\rho}}(\delta_0, \hat{\Pi}_{n,0})$), modulo changes in constants.

Remark 9 *The case when the sample $\mathcal{X}_n = \{X_1, \dots, X_n\}$ contains $\lfloor \kappa m \rfloor$ outliers (which can be completely arbitrary vectors in $\mathbb{R}^D$) for some $\kappa < 1/3$ can be handled similarly. In most examples throughout the paper, we state the results for the case $\kappa = 0$ for simplicity, keeping in mind that the generalization is a trivial corollary of Theorem 1. For example, if we allow $\lfloor \kappa m \rfloor$ outliers in the setup of Corollary 8, the resulting bounds becomes*

$$\Pr\left(\left\|\delta_0 - \hat{\Pi}_{n,g}\right\|_{\mathcal{F}_{k_H}} \ge 1.52\left(R\varepsilon_l + e^{-\tilde{K}l\varepsilon_l^2}\right)\right) \le \left[e^{(1-\kappa)\psi\left(\frac{3/7-\kappa}{1-\kappa}, 1/7\right)}\right]^{-m}.$$

While the result of the previous statement is promising, numerical approximation and sampling from the "robust posterior" $\hat{\Pi}_{n,g}$ is often problematic due to the underlying geometry defined by the Hellinger metric, and the associated distance $\|\cdot\|_{\mathcal{F}_{k_H}}$ is hard to estimate in practice. Our next goal is to derive similar guarantees for the M-Posterior evaluated with respect to the computationally tractable family of distances discussed in Section 2.3 above.

To transfer the conclusions of Theorem 7 and Corollary 8 to the case of other kernels $k(\cdot, \cdot)$ and associated metrics $\rho_k(\cdot, \cdot)$, we need to guarantee the existence of tests versus the complements of the balls in these distances. Such tests can be obtained from comparison inequalities between distances.

Assumption 1 *There exists $\gamma > 0$, $r(\theta_0) > 0$ and $\tilde{C}(\theta_0) > 0$ satisfying*

$$d(\theta, \theta_0) \ge \tilde{C}(\theta_0)\rho_k^\gamma(\theta, \theta_0) \text{ whenever } d(\theta, \theta_0) \le r(\theta_0),$$

where d is the Hellinger distance or the Euclidean distance (in the parametric case).

Remark 10 *When d is the Euclidean distance, we will impose an additional mild assumption guaranteeing existence of test versus the complements of the balls (for the Hellinger distance, this is always true, see Ghosal et al., 2000). Namely, we will assume that for every n and every pair $\theta_1, \theta_2 \in \Theta$, there exists a test $\phi_n := \phi_n(X_1, \dots, X_n)$ such that for some $\gamma > 0$ and a universal constant $K > 0$*

$$\begin{aligned} \mathbb{E}_{P_{\theta_1}} \phi_n &\le e^{-Knd^2(\theta_1, \theta_2)}, \\ \sup_{d(\theta, \theta_2) < d(\theta_1, \theta_2)/2} \mathbb{E}_{P_\theta}(1 - \phi_n) &\le e^{-Knd^2(\theta_1, \theta_2)}. \end{aligned} \tag{21}$$

Below, we provide examples of kernels satisfying the stated assumption.

Example 1 (Exponential families) *Let $\{P_\theta,\ \theta \in \Theta \subseteq \mathbb{R}^p\}$ be of the form*

$$\frac{dP_\theta}{dx}(x) := p_\theta(x) = \exp\Big(\langle T(x), \Theta\rangle_{\mathbb{R}^p} - G(\theta) + q(x)\Big),$$

where $\langle \cdot, \cdot \rangle_{\mathbb{R}^p}$ is the standard Euclidean dot product. Then the Hellinger distance can be expressed as (Nielsen and Garcia, 2011)

$$h^2(P_{\theta_1}, P_{\theta_2}) = 1 - \exp\Big(-\frac{1}{2}\Big(G(\theta_1) + G(\theta_2) - 2G\Big(\frac{\theta_1+\theta_2}{2}\Big)\Big)\Big).$$

If $G(\theta)$ is convex and its Hessian $D^2G(\theta)$ satisfies $D^2G(\theta) \succeq A$ uniformly for all $\theta \in \Theta$ and some symmetric positive definite operator $A : \mathbb{R}^p \mapsto \mathbb{R}^p$, then

$$h^2(P_{\theta_1}, P_{\theta_2}) \geq 1 - \exp\Big(-\frac{1}{8}(\theta_1-\theta_2)^T A(\theta_1-\theta_2)\Big),$$

hence assumption 1 holds with d being the Hellinger distance, $\gamma = 1$, $\tilde{C} = \frac{1}{\sqrt{2}}$ and $r(\theta_0) \equiv 1$ for

$$k(\theta_1, \theta_2) := \exp\left(-\frac{1}{8}(\theta_1-\theta_2)^T A(\theta_1-\theta_2)\right).$$

For finite-dimensional models, we will be especially interested in kernels $k(\cdot,\cdot)$ such that the associated metric $\rho_k(\cdot,\cdot)$ is bounded by the Euclidean distance. The following proposition gives a sufficient condition for this to hold.

Assume that kernel $k(\cdot,\cdot)$ satisfies conditions of Proposition 4 (in particular, k is characteristic). Recall that by Bochner's theorem, there exists a finite nonnegative Borel measure ν such that $k(\theta) = \int\limits_{\mathbb{R}^p} e^{i\langle x,\theta\rangle} d\nu(x)$.

Proposition 11 *Assume that $\int\limits_{\mathbb{R}^p} \|x\|_2^2 d\nu(x) < \infty$. Then there exists $D_k > 0$ depending only on k such that for all θ_1, θ_2,*

$$\rho_k(\theta_1, \theta_2) \leq D_k \left\|\theta_1 - \theta_2\right\|_2.$$

Proof For all $z \in \mathbb{R}$, $|e^{iz} - 1 - iz| \leq \frac{|z|^2}{2}$, implying that

$$\begin{aligned}
\rho_k^2(\theta_1,\theta_2) &= \|k(\cdot,\theta_1) - k(\cdot,\theta_2)\|_{\mathbb{H}}^2 \\
&= 2k(0) - 2k(\theta_1-\theta_2) = 2\int\limits_{\mathbb{R}^p} (1 - e^{i\langle x, \theta_1-\theta_2\rangle}) d\nu(x) \\
&\leq \int\limits_{\mathbb{R}^p} \langle x, \theta_1-\theta_2\rangle_{\mathbb{R}^p}^2 \, d\nu(x) \leq \|\theta_1-\theta_2\|_2^2 \int\limits_{\mathbb{R}^p} \|x\|_2^2 d\nu(x).
\end{aligned}$$

■

Moreover, the result of the previous proposition clearly remains valid for kernels of the form

$$\tilde{k}(\theta_1,\theta_2) = k(\theta_1-\theta_2) + c\,\langle \theta_1, \theta_2\rangle_{\mathbb{R}^p}, \tag{22}$$

where $c > 0$ and k satisfies the assumptions of proposition 11. For such a kernel, we have the obvious lower bound $\rho_{\tilde{k}}(\theta_1,\theta_2) \geq \sqrt{c}\,\|\theta_1-\theta_2\|_2$, hence $\rho_{\tilde{k}}$ is equivalent (in the strong sense) to the Euclidean distance.

We are ready to state our main result for convergence with respect to the RKHS-induced distance $\|\cdot\|_{\mathcal{F}_k}$.

Theorem 12 *Assume that conditions of Theorem 7 hold with ρ being the Hellinger or the Euclidean distance, and that assumption 1 is satisfied. In addition, let prior Π be such that*

$$D_k^W := \int_\Theta \rho_k^W(\theta,\theta_0) d\Pi(\theta) < \infty$$

for a sufficiently large W (it follows from the proof that $W = \frac{4}{3} + \frac{4+2C}{3\tilde{K}}$ is sufficient, with C and $\tilde{K}$ being the constants from the statement of Theorem 7). Then there exists a sufficiently large $R = R(\theta_0,\gamma) > 0$ and an absolute constant $\tilde{K}$ such that

$$\Pr\Big(\|\delta_0 - \Pi_l(\cdot|\mathcal{X}_l)\|_{\mathcal{F}_k} \geq R\varepsilon_l^{1/\gamma} + D_k e^{-\tilde{K}l\varepsilon_l^2/2}\Big) \leq \frac{1}{l\varepsilon_l^2} + 4e^{-\tilde{K}l\varepsilon_l^2}. \tag{23}$$

Proof The result essentially follows from the combination of Theorem 7 and assumption 1, see Section A.3 for details. ■

Theorem 12 yields the "weak" estimate that is needed to obtain the stronger bound for the M-Posterior distribution $\hat{\Pi}_{n,g}$. This is summarized in the following corollary:

Corollary 13 *Let $X_1, \ldots, X_n$ be an i.i.d. sample from P_0, and assume that $\hat{\Pi}_{n,g}$ is defined with respect to the distance $\|\cdot\|_{\mathcal{F}_k}$ as in (17) above. Let $l := \lfloor n/m \rfloor$. Assume that conditions of Theorem 12 hold, and, moreover, ε_l is such that*

$$\frac{1}{l\varepsilon_l^2} + 4e^{-\tilde{K}l\varepsilon_l^2} < \frac{1}{7}.$$

Then

$$\Pr\Big(\big\|\delta_0 - \hat{\Pi}_{n,g}\big\|_{\mathcal{F}_k} \geq 1.52\left(R\varepsilon_l^{1/\gamma} + D_k e^{-\tilde{K}l\varepsilon_l^2/2}\right)\Big) \leq 1.27^{-m}. \tag{24}$$

Proof It is enough to apply parts (a) and (b) of Theorem 1 with $\kappa = 0$ to the independent random measures $\Pi_{|G_j|}(\cdot|G_j)$, $j = 1, \ldots, m$. Note that the "weak concentration" assumption (28) is implied by (23). ■

Note that if $\Theta \subseteq \mathbb{R}^p$ and kernel $k(\cdot,\cdot)$ is of the form (22), the previous corollary together with proposition 6 imply that

$$\Pr\Big(\big\|\theta_* - \theta_0\big\|_2 \geq \frac{1.52R}{c}\varepsilon_l\Big) \leq 1.27^{-m},$$

where $\theta_* = \int_\Theta \theta d\hat{\Pi}_{n,g}(\theta)$ is the mean of $\hat{\Pi}_{n,g}$. In other words, this shows that the M-Posterior mean is the robust estimator of θ_0.

3.2 Bayesian Inference Based on Stochastic Approximation of the Posterior Distribution

As we have already mentioned in Section 2.4, when the number of disjoint subgroups m is large, the resulting M-Posterior distribution is "too flat", which results in large credible sets and overestimation of uncertainty. Clearly, the source of the problem is the fact that each individual random measure $\Pi_{|G_j|}(\cdot|G_j),\ j=1,\ldots,m$ is based on sample of size $l \simeq \frac{n}{m}$ which can be much smaller than n.

One way to reduce the variance of each subset posterior distribution is to repeat each observation in G_j m times (although other alternatives, such as bootstrap, are possible; for instance, bootstrap-related techniques have been investigated by Kleiner et al. (2014) in the frequentist setting), $\tilde{G}_j = \underbrace{\{G_j,\ldots,G_j\}}_{m \text{ times}}$. Formal application of the Bayes rule in this situation yields a collection of new measures on the parameter space:

$$\Pi_{|G_j|,m}(B|G_j) := \frac{\int\limits_B \left(\prod_{i\in G_j} p_\theta(X_i)\right)^m d\Pi(\theta)}{\int\limits_\Theta \left(\prod_{i\in G_j} p_\theta(X_i)\right)^m d\Pi(\theta)},$$

where we have assumed that $p_\theta(\cdot)$ is integrable. Here, $\left(\prod_{i\in G_j} p_\theta(X_i)\right)^m$ can be viewed as an approximation of the full data likelihood. We call the random measure $\Pi_{|G_j|,m}(\cdot|G_j)$ the *j-th stochastic approximation* to the full posterior distribution.

Of course, such a "correction" negatively affects coverage properties of the credible sets associated with each measure $\Pi_{|G_j|}(\cdot|G_j)$. However, taking the median of stochastic approximations yields improved coverage of the resulting M-Posterior distribution. The main goal of this section is to establish an asymptotic statement in spirit of a Bernstein-von Mises theorem for the M-Posterior based on stochastic approximations $\Pi_{|G_j|,m}(B|G_j),\ j=1,\ldots,m$.

We will start by showing that under certain assumptions the upper bounds for the convergence rates of $\Pi_{|G_j|,m}(\cdot|G_j)$ towards δ_0 are the same as for $\Pi_{|G_j|}(\cdot|G_j)$, the "standard" posterior distribution given G_j.

For $A \subseteq \Theta$, let $N_{[]}(u,A,d)$ be the bracketing number of $\{p_\theta,\ \theta\in A\}$ with respect to the distance $d(l,u) := \int\limits_{\mathbb{R}^D} \left(\sqrt{l(x)} - \sqrt{u(x)}\right)^2 dx$, and let

$$H_{[]}(u;A) := \log N_{[]}(u,A,d)$$

be the *bracketing entropy*. In what follows, $B(\theta_0,r) := \{\theta\in\Theta:\ h(P_\theta,P_{\theta_0}) \le r\}$ denotes the "Hellinger ball" of radius r centered at θ_0.

Theorem 14 (Wong et al., 1995, Theorem 1) *There exist constants $c_j,\ j=1,\ldots,4$ and $\zeta>0$ such that if*

$$\int\limits_{\zeta^2/2^8}^{\sqrt{2}\zeta} H_{[]}^{1/2}\left(u/c_3; B(\theta_0,\zeta\sqrt{2})\right) du \le c_4\sqrt{l}\zeta^2,$$

then

$$P\left(\sup_{\theta:h(P_\theta,P_0)\ge\zeta} \prod_{j=1}^{l} \frac{p_\theta}{p_0}(X_j) \ge e^{-c_1 l\zeta^2}\right) \le 4e^{-c_2 l\zeta^2}.$$

In particular, one can choose $c_1 = 1/24,\ c_2 = (4/27)(1/1926),\ c_3 = 10$ and $c_4 = (2/3)^{5/2}/512$.

In "typical" parametric problems ($\Theta\subseteq\mathbb{R}^p$), the bracketing entropy can be bounded as $H_{[]}(u;B(\theta_0,r)) \le C_1\log(C_2 r/u)$, whence the minimal ζ that satisfies conditions of Theorem 14 is of order $\zeta \simeq \sqrt{\frac{1}{l}}$. In particular, it is easy to check (e.g., using Theorem 2.7.11 in van der Vaart and Wellner, 1996) that this is the case when

(a) there exists $r_0>0$ such that

$$h\left(P_\theta,P_{\theta_0}\right) \ge K_1\|\theta-\theta_0\|_2$$

whenever $h\left(P_\theta,P_{\theta_0}\right)\le r_0$, and

(b) there exists $\alpha>0$ such that for $\theta_1,\theta_2\in B(\theta_0,r_0)$,

$$|p_{\theta_1}(x) - p_{\theta_2}(x)| \le F(x)\,\|\theta_1-\theta_2\|_2^\alpha$$

with $\int_{\mathbb{R}^D} F(x)dx < \infty$.

Application of theorem 14 to the analysis of "stochastic approximations" yields the following result.

Theorem 15
Let $\varepsilon_l>0$ be such that conditions of Theorem 14 hold with $\zeta := \varepsilon_l$, and

(a) for some $C>0$

$$\Pi\left(\theta:\ -P_0\left(\log\frac{p_\theta}{p_0}\right) \le \varepsilon_l^2,\ P_0\left(\log\frac{p_\theta}{p_0}\right)^2 \le \varepsilon_l^2\right) \ge \exp(-Cl\varepsilon_l^2),$$

(b) k is a positive-definite kernel that satisfies assumption 1 for the Hellinger distance for some $\tilde{C}(\theta_0)$ and $\gamma>0$.

Then there exists $\tilde{R} = \tilde{R}(C,\tilde{C},\gamma) > 0$ such that

$$\Pr\left(\|\delta_0 - \Pi_{l,m}(\cdot|\mathcal{X}_l)\|_{\mathcal{F}_k} \ge \tilde{R}\varepsilon_l^{1/\gamma} + e^{-ml\varepsilon_l^2}\right) \le \frac{1}{l\varepsilon_l^2} + 4e^{-c_2\tilde{C}^2\tilde{R}^{2\gamma}l\varepsilon_l^2}.$$

Proof See Section A.4 in the appendix. ■

Remark 16 *Note that for the kernel $k(\cdot,\cdot)$ of the form (22), assumption 1 reduces to the inequality between the Hellinger and Euclidean distances.*

As before, Theorem 1 combined with the "weak concentration" inequality of Theorem 15 gives stronger guarantees for the median $\hat{\Pi}^{\text{st}}_{n,g}$ (or its alternative $\hat{\Pi}^{\text{st}}_{n,0}$) of

$$\Pi_{|G_1|,m}(\cdot|G_1), \ldots, \Pi_{|G_m|,m}(\cdot|G_m).$$

Exact statement is very similar in spirit to Corollary 13.

Our next goal is to obtain the result describing the asymptotic behavior of the M-Posterior distribution $\hat{\Pi}^{\text{st}}_{n,0}$ in the parametric case. We start with a result that addresses each individual stochastic approximation $\Pi_{|G_j|,m}(\cdot|G_j),\ j = 1, \ldots, m$. Assume that $\Theta \subseteq \mathbb{R}^p$ has non-empty interior. For $\theta \in \Theta$, let

$$I(\theta) := \mathbb{E}_{\theta_0}\left[\frac{\partial}{\partial\theta}\log p_\theta(X)\left(\frac{\partial}{\partial\theta}\log p_\theta(X)\right)^T\right]$$

be the Fisher information matrix (we are assuming that it is well-defined). We will say that the family $\{P_\theta,\ \theta \in \Theta\}$ is *differentiable in quadratic mean* (see Chapter 7 in van der Vaart, 2000 for details) if there exists $\dot{\ell}_{\theta_0} : \mathbb{R}^D \mapsto \mathbb{R}^p$ such that

$$\int_{\mathbb{R}^D}\left(\sqrt{p_{\theta_0+h}} - \sqrt{p_{\theta_0}} - \frac{1}{2}h^T\dot{\ell}_{\theta_0}\sqrt{p_{\theta_0}}\right)^2 = o(\|h\|_2^2)$$

as $h \to 0$; usually, $\dot{\ell}_\theta(x) = \frac{\partial}{\partial\theta}\log p_\theta(x)$. Next, define

$$\Delta_{l,\theta_0} := \frac{1}{\sqrt{l}}\sum_{j=1}^{l} I^{-1}(\theta_0)\dot{\ell}_{\theta_0}(X_j).$$

We will first state a preliminary result for each individual "subset posterior" distribution:

Proposition 17 *Let $X_1, \ldots, X_l$ be an i.i.d. sample from P_{θ_0} for some θ_0 in the interior of Θ. Assume that*

(a) the family $\{P_\theta,\ \theta \in \Theta\}$ is differentiable in quadratic mean;

(b) the prior Π has a density (with respect to the Lebesgue measure) that is continuous and positive in the neighborhood of θ_0;

(c) conditions of Theorem 14 hold with $\zeta = \frac{C}{\sqrt{l}}$ for some $C > 0$ and l large enough.

Then for any integer $m \geq 1$,

$$\left\|\Pi_{l,m}(\cdot|X_1, \ldots, X_l) - N\left(\theta_0 + \frac{\Delta_{l,\theta_0}}{\sqrt{l}}, \frac{1}{l\cdot m}I^{-1}(\theta_0)\right)\right\|_{\text{TV}} \to 0$$

in P_{θ_0}-probability as $l \to \infty$.

Proof The proof follows standard steps (e.g., Theorem 10.1 in van der Vaart, 2000), where the existence of tests is substituted by the inequality of Theorem 14. See Section A.5 in the appendix for details. ■

The implication of this result for the M-Posterior is the following: if k is the kernel of

type (22), for sufficiently regular parametric families (differentiable in quadratic mean, with "well-behaved" bracketing numbers, satisfying assumption 1 for the Euclidean distance with $\gamma = 1$) and regular priors, then

(a) the M-Posterior is well approximated by a normal distribution centered at the "robust" estimator θ^* of unknown θ_0;

(b) the estimator θ^* is a center of the confidence set of level 1.15^{-m} and diameter of order $\sqrt{\frac{m}{n}}$ (same as we would expect for this level for the usual posterior distribution - however, the bound for the M-Posterior holds for finite sample sizes).

This is formalized below:

Theorem 18

(a) Let k be the kernel of type (22), and suppose that the assumptions of Proposition 17 hold. Moreover, let the prior Π be such that $\int_{\mathbb{R}^p}\|\theta\|_2^2 d\Pi(\theta) < \infty$. Then for any fixed $m \geq 1$,

$$\left\|\hat{\Pi}^{\text{st}}_{n,0} - N\left(\theta^*, \frac{1}{n}I^{-1}(\theta_0)\right)\right\|_{\text{TV}} \to 0 \text{ as } n \to \infty,$$

in P_{θ_0}-probability when $n \to \infty$, where θ^* is the mean of $\hat{\Pi}^{\text{st}}_{n,0}$.

(b) Assume that conditions (a), (b) of Theorem 15 hold with

$$\varepsilon_l \gtrsim \frac{1}{\sqrt{l}} \simeq \sqrt{\frac{m}{n}}$$

and $\gamma = 1$. Then for all $n \geq n_0$ and $\bar{R}$ large enough,

$$\Pr\left(\|\theta^* - \theta_0\|_2 \geq \bar{R}\left(\varepsilon_l + e^{-ml\varepsilon_l^2}\right)\right) \leq 1.15^{-m}.$$

Proof (a) It is easy to see that convergence in total variation norm, together with an assumption that the prior distribution satisfies

$$\int_{\mathbb{R}^p}\|\theta\|_2^2 d\Pi(\theta) < \infty,$$

implies that the expectations converge in P_{θ_0}-probability as well:

$$\left\|\int_\Theta \theta\left(d\Pi_{l,m}(\theta|\mathcal{X}_l) - dN\left(\theta_0 + \frac{\Delta_{l,\theta_0}}{\sqrt{l}}, \frac{1}{l\cdot m}I^{-1}(\theta_0)\right)(\theta)\right)\right\|_2 \to 0 \text{ as } l \to \infty.$$

Together with an observation that the total variation distance between $N(\mu_1, \Sigma)$ and $N(\mu_2, \Sigma)$ is bounded by the multiple of $\|\mu_1 - \mu_2\|_2$, it implies that we can replace $\theta_0 + \frac{\Delta_{l,\theta_0}}{\sqrt{l}}$ by the mean

$$\bar{\theta}_{l,m}(X_1, \ldots, X_l) := \int_\Theta \theta d\Pi_{l,m}(\theta|\mathcal{X}_l),$$

in other words, the conclusion of Proposition 17 can be stated as

$$\left\| \Pi_{l,m}(\cdot|\mathcal{X}_l) - N\left(\bar{\theta}_{l,m}, \frac{1}{l\cdot m} I^{-1}(\theta_0)\right)\right\|_{\mathrm{TV}} \to 0 \text{ as } l \to \infty,$$

in P_{θ_0}-probability. Now assume that $m = \lfloor \frac{n}{l} \rfloor$ is fixed, and let $n, l \to \infty$. As before, let $G_1, \dots, G_m$ be disjoint groups of i.i.d. observations from P_{θ_0} of cardinality l each. Recall that, by the definition (3) of $\mathrm{med}_0(\cdot)$, $\Pi^{\mathrm{st}}_{n,0} = \Pi_{l,m}(\cdot|\mathcal{X}_{l_*})$ for some $l_* \le m$, and $\theta^* := \theta_{l_*,m}$ is the mean of $\Pi^{\mathrm{st}}_{n,0}$. Clearly, we have

$$\left\| \Pi^{\mathrm{st}}_{n,0} - N\left(\theta^*, \frac{1}{l\cdot m} I^{-1}(\theta_0)\right)\right\|_{\mathrm{TV}} \le$$
$$\max_{j=1,\dots,m} \left\| \Pi_{l,m}(\cdot|G_j) - N\left(\bar{\theta}_{l,m}(G_j), \frac{1}{l\cdot m} I^{-1}(\theta_0)\right)\right\|_{\mathrm{TV}} \to 0 \text{ as } n \to \infty. \tag{25}$$

(b) Let $\varepsilon_l \ge C\sqrt{\frac{1}{l}}$ where C large enough so that

$$\frac{1}{l\varepsilon_l^2} + 4e^{-c_2\tilde{C}^2\tilde{R}^2 l\varepsilon_l^2} \le \frac{1}{4},$$

where c_2, $\tilde{R}$ are the same as in Theorem 15. Applying Theorem 15, we get

$$\Pr\left(\|\delta_{\theta_0} - \Pi_{l,m}(\cdot|\mathcal{X}_l)\|_{\mathcal{F}_k} \ge \tilde{R}\varepsilon_l + e^{-ml\varepsilon_l^2}\right) \le \frac{1}{4}.$$

By part (b) of Theorem 1,

$$\left\| \hat{\Pi}^{\mathrm{st}}_{n,0} - \delta_{\theta_0}\right\|_{\mathcal{F}_{\tilde{k}}} \le 3\left(\tilde{R}\varepsilon_l + e^{-ml\varepsilon_l^2}\right)$$

with probability $\ge 1 - 1.15^{-m}$. Since kernel $\tilde{k}$ is of the type (22), proposition (6) implies that

$$\|\theta^* - \theta_0\|_2 \le \left\| \hat{\Pi}^{\mathrm{st}}_{n,0} - \delta_{\theta_0}\right\|_{\mathcal{F}_{\tilde{k}}}, \tag{26}$$

and the result follows.

■

In particular, for $m = A\log(n)$ and $\varepsilon_l \simeq \sqrt{\frac{m}{n}}$, we obtain the bound

$$\Pr\left(\|\theta^* - \theta_0\|_2 \ge \bar{R}\sqrt{\frac{A\log n}{n}}\right) \le n^{-A}.$$

for some constant $\bar{R}$ *independent* of m. Note that θ^* itself depends on m, hence this bound is *not uniform*, and holds only for a given confidence level $1 - n^{-A}$.

It is convenient to interpret this (informally) in terms of the credible sets: to obtain the credible set with "frequentist" coverage level $\ge 1 - n^{-A}$, pick $m = A\log n$ and use the $(1 - n^{-A})$ - credible set of the M-Posterior $\hat{\Pi}^{\mathrm{st}}_{n,0}$.

Algorithm 1 Evaluating the geometric median of probability distributions via Weiszfeld's algorithm

Input:

1. Discrete measures $Q_1, \dots, Q_m$;
2. The kernel $k(\cdot,\cdot) : \mathbb{R}^p \times \mathbb{R}^p \mapsto \mathbb{R}$;
3. Threshold $\varepsilon > 0$;

Initialize:

1. Set $w_j^{(0)} := \frac{1}{m}$, $j = 1 \dots m$;
2. Set $Q_*^{(0)} := \frac{1}{m}\sum_{j=1}^m Q_j$;

repeat

Starting from $t = 0$, for each $j = 1, \dots, m$:

1. Update $w_j^{(t+1)} = \frac{\|Q_*^{(t)} - Q_j\|_{\mathcal{F}_k}^{-1}}{\sum_{i=1}^m \|Q_*^{(t)} - Q_i\|_{\mathcal{F}_k}^{-1}}$; (apply (12) to evaluate $\|Q_*^{(t)} - Q_i\|_{\mathcal{F}_k}$);
2. Update $Q_*^{(t+1)} = \sum_{j=1}^m w_j^{(t+1)} Q_j$;

until $\|Q_*^{(t+1)} - Q_*^{(t)}\|_{\mathcal{F}_k} \le \varepsilon$;
Return: $w_* := (w_1^{(t+1)}, \dots, w_m^{(t+1)})$.

4. Numerical Algorithms and Examples

In this section, we consider examples and applications in which comparisons are made for the inference based on the usual posterior distribution and on the M-Posterior. One of the well-known and computationally efficient ways to find the geometric median in Hilbert spaces is the famous *Weiszfeld's algorithm*, introduced in Weiszfeld (1936). Details of implementation are described in Algorithms 1 and 2. Algorithm 1 is a particular case of Weiszfeld's algorithm applied to subset posterior distributions and distance $\|\cdot\|_{\mathcal{F}_k}$, while Algorithm 2 shows how to obtain an approximation to M-Posterior given the samples from $\Pi_{n,m}(\cdot|G_j)$, $j = 1 \dots m$. Note that the subset posteriors $\Pi_{n,m}(\cdot|G_j)$ whose "weights" $w_{*,j}$ in the expression of the M-Posterior are small (in our case, smaller than $1/(2m)$) are excluded from the analysis. Our extensive simulations show the empirical evidence in favor of this additional thresholding step.

Detailed discussion of convergence rates and acceleration techniques for Weiszfeld's method from the viewpoint of modern optimization can be found in Beck and Sabach (2013). For alternative approaches and extensions of Weiszfeld's algorithm, see Bose et al. (2003), Ostresh (1978), Overton (1983), Chandrasekaran and Tamir (1990), Cardot et al. (2012), Cardot et al. (2013), among other works.

In all numerical simulations below, we use stochastic approximations and the corresponding M-Posterior $\hat{\Pi}^{\mathrm{st}}_{n,g}$, unless noted otherwise. The posterior $\hat{\Pi}^{\mathrm{st}}_{n,0}$ based on the metric median is mainly of theoretical interest, while numerical results obtained for it are typically inferior compared to the geometric median, so we do not discuss them further.

Algorithm 2 Approximating the M-Posterior distribution

Input:

1. Samples $\{Z_{j,i}\}_{i=1}^{S_j} \sim \Pi_{n,m}(\cdot|G_j),\ j = 1 \dots m$ (see equation (18));

Do:

1. $Q_j := \frac{1}{S_j} \sum_{i=1}^{S_j} \delta_{Z_{j,i}}$, $j = 1 \dots m$ - empirical approximations of $\Pi_{n,m}(\cdot|G_j)$.
2. Apply Algorithm 1 to $Q_1, \dots, Q_m$; return $w_* = (w_{*,1} \dots w_{*,m})$;
3. For $j = 1, \dots, m$, set $\bar{w}_j := w_{*,j} I\{w_{*,j} \geq \frac{1}{2m}\}$; define $\hat{w}_j^* := \bar{w}_j / \sum_{i=1}^m \bar{w}_i$.

Return: $\hat{\Pi}_{n,g}^{\text{st}} := \sum_{i=1}^m \hat{w}_i^* Q_i$.

Before presenting the results of numerical analysis, let us remark on two important computational aspects.

Remark 19

(a) The number of subsets m appears as a "free parameter" entering the theoretical guarantees for M-Posterior. One interpretation of m (in terms of the credible sets) is given in the end of Section 3.2. Our results also imply that partitioning the data into $m = 2k + 1$ subsets guarantees robustness to the presence of k outliers of arbitrary nature.

In many applications, m is dictated by the sample size and computational resources (e.g., the number of available machines). In Section A.6, we describe a heuristic approach to selection of m that shows good practical performance. As a rule of a thumb, we recommend choosing $m \lesssim \sqrt{n}$ as larger values of m lead to an M-Posterior that overestimates uncertainty. This heuristic is supported by the numerical results presented below.

(b) It is easy to get a general idea regarding the potential improvement in computational time complexity achieved by the M-Posterior. Given the data set $\mathcal{X}_n = \{X_1, \dots, X_n\}$ of size n, let $t(n)$ be the running time of the algorithm (e.g., MCMC) that outputs a single observation from the posterior distribution $\Pi_n(\cdot|\mathcal{X}_n)$. If the goal is to obtain S samples from the posterior, then the total running time is $O(S \cdot t(n))$. Let us compare this time with the running time needed to obtain S samples from the M-posterior given that the algorithm is running on m machines in parallel. In this case, we need to generate $O(S)$ samples from each of m subset posteriors, which is done in time $O\left(S \cdot t\left(\frac{n}{m}\right)\right)$, where S is typically large and $m \ll n$. According to Theorem 7.1 in Beck and Sabach (2013), Weiszfeld's algorithm approximates the M-Posterior to degree of accuracy ε in at most $O(1/\varepsilon)$ steps, and each of these steps has complexity $O(S^2)$ (which follows from (12)), so that the total running time is

$$O\left(S \cdot t\left(\frac{n}{m}\right) + \frac{S^2}{\varepsilon}\right). \tag{27}$$

The term $\frac{S^2}{\varepsilon}$ can be refined in several ways via application of more advanced optimization techniques (see the aforementioned references). If, for example, $t(n) \simeq n^r$ for some $r \geq 1$, then $\frac{S}{m} \cdot t\left(\frac{n}{m}\right) \simeq \frac{1}{m^{1+r}} S n^r$ which should be compared to $S \cdot n^r$ required by the standard approach.

To give a specific example, consider an application of (27) in the context of Gaussian process (GP) regression. If n is the number of training samples, then GP regression has $O(n^3) + O(Sn^2)$ asymptotic time complexity to obtain S samples from the posterior distribution of GP (Rasmussen and Williams, 2006, Algorithm 2.1). Assuming we have access to m machines, the time complexity to obtain S samples from M-Posterior in GP regression is $O\left(\left(\frac{n}{m}\right)^3 + S\left(\frac{n}{m}\right)^2 + \frac{S^2}{\varepsilon}\right)$. If for example $S = cn$ for some $c > 0$ and $m^2 < n\varepsilon$, we get $O(m^2)$ improvement in running time.

(c) In many cases, replacing the "subset posterior" by the stochastic approximation does not result in increased sampling complexity: indeed, the log-likelihood in the sampling algorithm for the subset posterior is simply multiplied by m to obtain the sampler for the stochastic approximation. We have included the description of a modified Dirichlet mixture model in the supplementary material as an illustration.

4.1 Numerical Analysis: Simulated Data

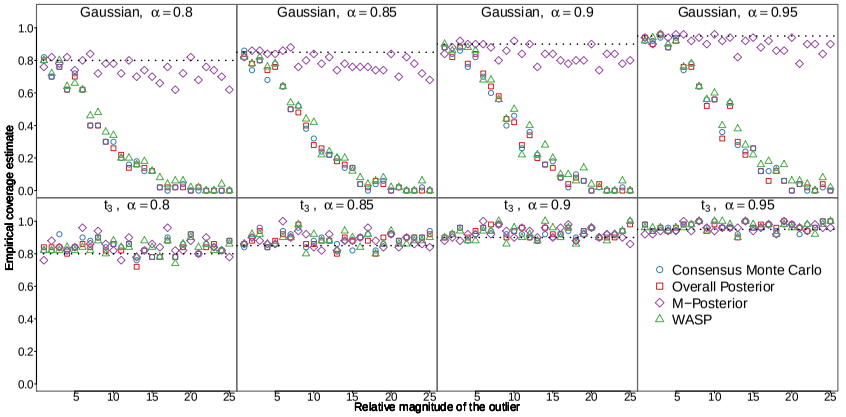


Figure 1: Effect of the outlier on empirical coverage of (1-α)100% credible intervals (CIs). The x-axis represents the outlier magnitude. The y-axis represents the empirical coverage computed from 50 simulation replications. The dotted horizontal lines show the theoretical frequentist coverage.

This section demonstrates the effect of magnitude of an outlier on the posterior distribution of the mean parameter μ. We empirically show that M-Posterior of μ is a robust alternative to the overall posterior. To this end, we used the simplest univariate Gaussian model $\{P_\mu = \mathcal{N}(\mu, 1),\ \mu \in \mathbb{R}\}$.

We simulated 25 data sets containing 200 observations each. Each data set $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,200})$ contained 199 independent observations from the standard Gaussian distribution ($x_{i,j} \sim$

$\mathcal{N}(0,1)$ for $i = 1,\ldots,25$ and $j = 1,\ldots,199$). The last entry in each data set, $x_{i,200}$, was an outlier, and its value increased linearly for $i = 1,\ldots,25$: $x_{i,200} = i\max(|x_{i,1}|,\ldots,|x_{i,199}|)$. The index of outlier was unknown to every sampling algorithm. The true variance of observations was fixed at 1 and was assumed to be known. The algorithms for sampling from the subset and overall posterior distributions of μ were implemented using Stan language (Carpenter et al., 2016) based on Stan's default Gaussian prior on μ. We used two likelihoods for the data, one was the standard Gaussian likelihood and the other was a (more robust) Student's t-distribution likelihood with 3 degrees of freedom (t_3). We generated 1000 samples from each posterior distribution $\Pi_{200}(\cdot|\,\mathbf{x}_i)$ for $i = 1,\ldots,25$. Setting $m = 10$ in Algorithm 1, we generated 1000 samples from every subset posterior $\Pi_{200,10}(\cdot|G_{j,i})$, $j = 1,\ldots,10$ to form the empirical measures $Q_{j,i}$; here, $\cup_{j=1}^{10} G_{j,i} = \mathbf{x}_i$. Using these $Q_{j,i}$s, Algorithm 2 generated 10000 samples from the M-Posterior $\hat{\Pi}^{\text{st}}_{2000,g}(\cdot|\,\mathbf{x}_i)$ for each $i = 1,\ldots,25$. This process was replicated 50 times.

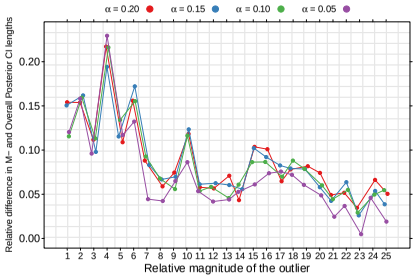


Figure 2: Calibration of uncertainty quantification of M-Posterior. The x-axis represents the outlier magnitude that increases from 1 to 25. The y-axis represents the relative difference between M-Posterior and overall posterior CI lengths. A value close to 0 represents that the M-Posterior CIs are well-calibrated.

We used Consensus MCMC (Scott et al., 2013) and WASP (Srivastava et al., 2015a) as representative methods for scalable MCMC approach, and compared their performance with M-Posterior. Across 50 simulation replications with the Gaussian and t_3 likelihood, we estimated the empirical coverage of (1-α)100% credible intervals (CIs) for the "consensus posterior", WASP, the overall posterior, and the M-Posterior for $\alpha = 0.2, 0.15, 0.10$, and 0.05. The empirical coverages of M-Posterior's CIs showed robustness to magnitude of the outlier in the Gaussian likelihood model. On the contrary, performance of the WASP, consensus posterior and overall posterior deteriorated fairly quickly across all α's leading to 0% empirical coverage as magnitude of the outlier increased from $i = 1$ to $i = 25$ (Figure 1). When the more robust t_3 likelihood was used, empirical coverages of CIs computed using the four methods were close to their theoretical values for all four α's. We compared uncertainty quantification of the M-Posterior with that of the overall posterior using relative lengths of their CIs, with zero value corresponding to identical lengths and a positive value to wider CIs of the M-Posterior. We found that widths of CIs for both posteriors were

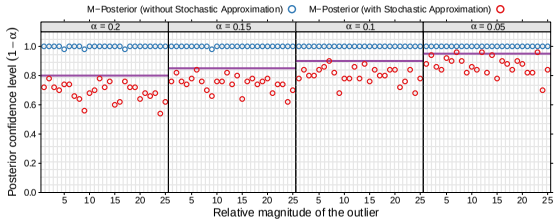


Figure 3: Effect of stochastic approximation on empirical coverage of (1-α)100% CIs. The x-axis represents the outlier magnitude that increases from 1 to 25. The y-axis represents the fraction of times the CIs of M-Posteriors with and without stochastic approximation include the true mean over 50 replications. The horizontal lines (in violet) show the theoretical frequentist coverage.

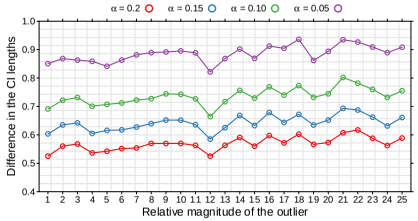
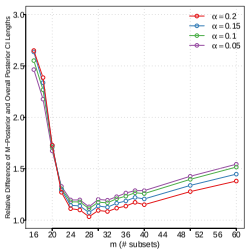


Figure 4: Effect of stochastic approximation on the length of (1-α)100% CIs. The x-axis represents the outlier magnitude that increases from 1 to 25. The y-axis represents the differences in the lengths of the CIs of M-Posteriors without and with stochastic approximation.

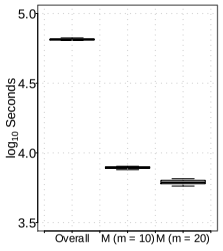
fairly similar for $i = 1,\ldots,25$, with M-Posterior's CIs being slightly wider in absence of large outliers (Figure 2).

Stochastic approximation was important for proper calibration of uncertainty quantification. The empirical coverages of (1-α)100% CIs of the M-Posterior without stochastic approximation overcompensated for uncertainty at all levels of α (Figure 3). Similarly, lengths of the CIs of M-Posterior without stochastic approximation are wider than those with stochastic approximation (Figure 4). Both these observations showed that stochastic approximation led to shorter CIs for M-Posterior that had empirical coverages close to their theoretical values.

The number of subsets m had an effect on credible interval length of the M-Posterior. We modified the simulation above and generated 1000 observations $\mathbf{x}$, with the last 10 observations in $\mathbf{x}$ being outliers with value $x_j = 25\max(|x_1|,\ldots,|x_{990}|)$, $j = 991,\ldots,1000$. The simulation setup was replicated 50 times. M-Posteriors were obtained for $m = 16, 18,\ldots,40, 50, 60$. Across all values of m, M-Posterior's CI was compared to the CI of the overall posterior



(a) Sensitivity to the choice of m



(b) Time

Figure 5: (a) Effect of m on the length of (1-α)100% CIs. The x-axis represents different choices of m. The y-axis represents the relative difference between M-Posterior and overall posterior CI lengths (median across all 25 outlier magnitudes and 50 replications). (b) Computation time to estimate overall posterior and M-Posterior (M) with $m = 10$ and $m = 20$ in real data analysis.

after removing the outliers; the relative difference of M-Posterior and the overall posterior CI lengths decreases for $m \geq 22 > 2k$, where $k = 10$ is the number of outliers, remains stable as m increases to $m = 38$, and grows for larger values of m (Figure 5a). This demonstrates that inference based on M-Posterior was not too sensitive to the choice of m for a wide range of values.

4.2 Real Data Analysis: General Social Survey

The General Social Survey (GSS; `gss.norc.org`) has collected responses to questions about evolution of American society since 1972. We selected data for 9 questions from different social topics: "happy" (happy), "Bible is a word of God" (bible), "support capital punishment" (cap), "support legalization of marijuana" (grass), "support premarital sex" (pre), "approve bible prayer in public schools" (prayer), "expect US to be in world war in 10 years" (uswar), "approve homosexual sex relations" (homo), "support abortion" (abort). These questions were in the survey since 1988 and their answers were converted to two levels: *yes* or *no*. Missing data were imputed based on the average, resulting in a data set with approximately 28,000 respondents.

We use a Dirichlet process (DP) mixture of product multinomial distributions, probabilistic parafac (p-parafac), to model the multivariate dependence among the 9 questions. Let $c_k \in \{\text{yes}, \text{no}\}$ represent the response to kth question, $k = 1, \ldots, 9$, then $\boldsymbol{\pi}_{c_1,\ldots,c_9}$ is the joint probability of response $c = (c_1, \ldots, c_9)$ for the 9 questions. Using $\boldsymbol{\pi}_{c_1,\ldots,c_9}$, we estimated the joint probability of response to two questions $\boldsymbol{\pi}_{c_i,c_j}$ for every i and j in $\{1, \ldots, 9\}$; see Section 6 of the supplementary material for the p-parafac generative model and sampling algorithms. The GSS data were randomly split into 10 test and training data sets. Samples from the overall posteriors of $\boldsymbol{\pi}_{c_i,c_j}$s were obtained using the Gibbs sampling algorithm of Dunson and Xing (2009). We chose m as 10 and 20 and estimated M-Posteriors for $\boldsymbol{\pi}_{c_i,c_j}$s in

four steps: training data were randomly split into m subsets, samples from subset posteriors were obtained after modifying the original sampler using stochastic approximation, weights of subsets posteriors were estimated using Algorithm 2, and atoms with estimated weights below $\frac{1}{2m}$ were removed.

M-Posterior had similar uncertainty quantification as the overall posterior while being more efficient. M-Posterior was at least 10 ($m = 20$) and 8 times ($m = 10$) faster than the overall posterior and it used less than 25% of the memory resources required by the overall posterior (Figure 5b). The lengths of credible intervals for $\boldsymbol{\pi}_{c_i,c_j}$s obtained using the overall posterior and the two M-Posteriors were very similar, but the overall posterior's coverage of the maximum likelihood estimators for $\boldsymbol{\pi}_{c_i,c_j}$s obtained from the test data was worse than that of the two M-Posteriors (Table 1).

Empirical Coverage of (1-α)100% Credible Intervals				
α	0.05	0.10	0.15	0.20
Overall Posterior	0.56 (0.10)	0.52 (0.01)	0.49 (0.02)	0.46 (0.02)
M-Posterior ($m = 10$)	0.71 (0.02)	0.65 (0.04)	0.62 (0.04)	0.60 (0.04)
M-Posterior ($m = 20$)	0.74 (0.03)	0.70 (0.03)	0.66 (0.03)	0.63 (0.03)
Length of (1-α)100% Credible Intervals (in 10^{-2})				
α	0.05	0.10	0.15	0.20
Overall Posterior	1.20 (0.12)	1.08 (0.11)	1.00 (0.10)	0.95 (0.10)
M-Posterior ($m = 10$)	1.70 (0.13)	1.54 (0.12)	1.43 (0.11)	1.35 (0.10)
M-Posterior ($m = 20$)	1.88 (0.11)	1.70 (0.10)	1.59 (0.09)	1.49 (0.09)

Table 1: Empirical coverage and lengths of (1-α)100% credible intervals. The results are averaged across all joint probabilities $\boldsymbol{\pi}_{c_i,c_j}$s and 10 folds of cross-validation. Monte Carlo errors are in parentheses.

Acknowledgments

Authors were partially supported by grant R01-ES-017436 from the National Institute of Environmental Health Sciences (NIEHS) of the National Institutes of Health (NIH). Stanislav Minsker acknowledges support from NSF grants DMS-1712956, FODAVA CCF-0808847, DMS-0847388, ATD-1222567. Lizhen Lin was partially supported by NSF CAREER DMS-1654579 and IIS-1663870.

Appendix A. Proofs.

A.1 Proof of Theorem 1

We will prove a slightly more general result:

Theorem 20

a *Assume that $(\mathbb{H}, \|\cdot\|)$ is a Hilbert space and $\theta_0 \in \mathbb{H}$. Let $\hat\theta_1, \dots, \hat\theta_m \in \mathbb{H}$ be a collection of independent random variables. Let the constants α, q, γ be such that $0 < q < \alpha < 1/2$, and $0 \le \gamma < \frac{\alpha - q}{1-q}$. Suppose $\varepsilon > 0$ is such that for all $j,\ 1 \le j \le \lfloor (1-\gamma)m \rfloor + 1$,*

$$\Pr\left(\|\hat\theta_j - \theta_0\| > \varepsilon\right) \le q. \tag{28}$$

Let $\hat\theta_* = \mathrm{med}_g(\hat\theta_1, \dots, \hat\theta_m)$ be the geometric median of $\{\hat\theta_1, \dots, \hat\theta_m\}$. Then

$$\Pr\left(\|\hat\theta_* - \theta_0\| > C_\alpha \varepsilon\right) \le \left[e^{(1-\gamma)\psi\left(\frac{\alpha-\gamma}{1-\gamma}, q\right)}\right]^{-m},$$

where $C_\alpha = (1-\alpha)\sqrt{\frac{1}{1-2\alpha}}$.

b *Assume that $(\mathbb{Y}, d)$ is a metric space and $\theta_0 \in \mathbb{Y}$. Let $\hat\theta_1, \dots, \hat\theta_m \in \mathbb{Y}$ be a collection of independent random variables. Let the constants q, γ be such that $0 < q < \frac{1}{2}$ and $0 \le \gamma < \frac{1/2 - q}{1-q}$. Suppose $\varepsilon > 0$ are such that for all $j,\ 1 \le j \le \lfloor (1-\gamma)m \rfloor + 1$,*

$$\Pr\left(d(\hat\theta_j, \theta_0) > \varepsilon\right) \le q. \tag{29}$$

Let $\hat\theta_* = \mathrm{med}_0(\hat\theta_1, \dots, \hat\theta_m)$. Then

$$\Pr\left(d(\hat\theta_*, \theta_0) > 3\varepsilon\right) \le e^{-m(1-\gamma)\psi\left(\frac{1/2-\gamma}{1-\gamma}, q\right)}.$$

To get the bound stated in the paper, take $q = \frac{1}{7}$ and $\alpha = \frac{3}{7}$ in part **(a)** and $q = \frac{1}{4}$ in part **b**.

We start by proving part **a**. To this end, we will need the following lemma (see lemma 2.1 in Minsker, 2015):

Lemma 21 *Let $\mathbb{H}$ be a Hilbert space, $x_1, \dots, x_m \in \mathbb{H}$ and let x_* be their geometric median. Fix $\alpha \in \left(0, \frac{1}{2}\right)$ and assume that $z \in \mathbb{H}$ is such that $\|x_* - z\| > C_\alpha r$, where*

$$C_\alpha = (1-\alpha)\sqrt{\frac{1}{1-2\alpha}}$$

and $r > 0$. Then there exists a subset $J \subseteq \{1, \dots, m\}$ of cardinality $|J| > \alpha m$ such that for all $j \in J$, $\|x_j - z\| > r$.

Assume that event $\mathcal{E} := \left\{\|\hat\theta_* - \theta_0\| > C_\alpha \varepsilon\right\}$ occurs. Lemma 21 implies that there exists a subset $J \subseteq \{1, \dots, m\}$ of cardinality $|J| \ge \alpha k$ such that $\|\hat\theta_j - \theta_0\| > \varepsilon$ for all $j \in J$, hence

$$\begin{aligned}\Pr(\mathcal{E}) \le &\Pr\left(\sum_{j=1}^m I\left\{\|\hat\theta_j - \theta_0\| > \varepsilon\right\} > \alpha m\right) \le \\ &\Pr\left(\sum_{j=1}^{\lfloor (1-\gamma)m\rfloor + 1} I\left\{\|\hat\theta_j - \theta_0\| > \varepsilon\right\} > (\alpha - \gamma)m \frac{\lfloor (1-\gamma)m\rfloor + 1}{\lfloor (1-\gamma)m\rfloor + 1}\right) \le \\ &\Pr\left(\sum_{j=1}^{\lfloor (1-\gamma)m\rfloor + 1} I\left\{\|\hat\theta_j - \theta_0\| > \varepsilon\right\} > \frac{\alpha - \gamma}{1-\gamma}\left(\lfloor (1-\gamma)m\rfloor + 1\right)\right).\end{aligned}$$

If W has Binomial distribution $W \sim B(\lfloor (1-\gamma)m\rfloor + 1, q)$, then

$$\begin{aligned}\Pr\left(\sum_{j=1}^{\lfloor (1-\gamma)m\rfloor + 1} I\left\{\|\hat\theta_j - \theta_0\| > \varepsilon\right\} > \frac{\alpha - \gamma}{1-\gamma}\left(\lfloor (1-\gamma)m\rfloor + 1\right)\right) \le \\ \Pr\left(W > \frac{\alpha - \gamma}{1-\gamma}\left(\lfloor (1-\gamma)m\rfloor + 1\right)\right)\end{aligned}$$

(see Lemma 23 in Lerasle and Oliveira, 2011 for a rigorous proof of this fact). Chernoff bound (e.g., Proposition A.6.1 in van der Vaart and Wellner, 1996), together with an obvious bound $\lfloor (1-\gamma)m\rfloor + 1 > (1-\gamma)m$, implies that

$$\Pr\left(W > \frac{\alpha - \gamma}{1-\gamma}\left(\lfloor (1-\gamma)m\rfloor + 1\right)\right) \le \exp\left(-m(1-\gamma)\psi\left(\frac{\alpha-\gamma}{1-\gamma}, q\right)\right).$$

To establish part **b**, we proceed as follows: let $\mathcal{E}_1$ be the event

$$\mathcal{E}_1 = \{\text{more than a half of events } d(\hat\theta_j, \theta_0) \le \varepsilon,\ j = 1 \dots m \text{ occur}\}.$$

Assume that $\mathcal{E}_1$ occurs. Then we clearly have $\varepsilon_* \le \varepsilon$, where ε_* is defined in equation (2.2) of the paper: indeed, for any $\theta_{j_1},\ \theta_{j_2}$ such that $d(\hat\theta_{j_i}, \theta_0) \le \varepsilon,\ i = 1, 2$, triangle inequality gives $d(\theta_{j_1}, \theta_{j_2}) \le 2\varepsilon$. By the definition of $\hat\theta_*$, inequality $d(\hat\theta_*, \hat\theta_j) \le 2\varepsilon_* \le 2\varepsilon$ holds for at least a half of $\{\hat\theta_1, \dots, \hat\theta_m\}$, hence, it holds for some $\hat\theta_{\tilde j}$ with $d(\hat\theta_{\tilde j}, \theta_0) \le \varepsilon$. In turn, this implies (by triangle inequality) $d(\hat\theta_*, \theta_0) \le 3\varepsilon$. We conclude that

$$\Pr\left(d(\hat\theta_*, \theta_0) > 3\varepsilon\right) \le \Pr(\mathcal{E}_1).$$

The rest of the proof repeats the argument of part **a** since

$$\Pr(\mathcal{E}_1^c) = \Pr\left(\sum_{j=1}^m I\left\{d(\hat\theta_j, \theta_0) > \varepsilon\right\} \ge \frac{m}{2}\right),$$

where $\mathcal{E}_1^c$ is the complement of $\mathcal{E}_1$.

A.2 Proof of Theorem 7

By the definition of Wasserstein distance d_{W_1},

$$d_{W_1,\rho}(\delta_0, \Pi_l(\cdot|\mathcal{X}_l)) = \int_\Theta \rho(\theta,\theta_0) d\Pi_l(\theta|X_1,\dots,X_l). \tag{30}$$

(recall that ρ is the Hellinger distance). Let R be a large enough constant to be determined later. Note that the Hellinger distance is uniformly bounded by 1. Using (30), it is easy to see that

$$d_{W_{1,\rho}}(\delta_0, \Pi_l(\cdot|\mathcal{X}_l)) \le R\varepsilon_l + \int\limits_{\rho(\theta,\theta_0)\ge R\varepsilon_l} d\Pi_l(\cdot|\mathcal{X}_l). \tag{31}$$

To this end, it remains to estimate the second term in the sum above. We will follow the proof of Theorem 2.1 in Ghosal et al. (2000). Bayes formula implies that

$$\Pi_l(\theta:\rho(\theta,\theta_0)\ge R\varepsilon_l|\mathcal{X}_l) = \int\limits_{\rho(\theta,\theta_0)\ge R\varepsilon_l} \frac{\prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta)}{\int\limits_\Theta \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta)}.$$

Let

$$A_l = \left\{\theta : -P_0\left(\log\frac{p_\theta}{p_0}\right) \le \varepsilon_l^2, P_0\left(\log\frac{p_\theta}{p_0}\right)^2 \le \varepsilon_l^2\right\}.$$

For any $C_1 > 0$, Lemma 8.1 Ghosal et al. (2000) yields

$$\Pr\left\{\int\limits_\Theta \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) dQ(\theta) \le \exp\big(-(1+C_1)l\varepsilon_l^2\big)\right\} \le \frac{1}{C_1^2 l\varepsilon_l^2}.$$

for every probability measure Q on the set A_l. Moreover, by the assumption on the prior Π,

$$\Pi(A_l) \ge \exp\left(-Cl\varepsilon_l^2\right).$$

Consequently, with probability at least $1 - \frac{1}{C_1^2 l\varepsilon_l^2}$,

$$\int\limits_\Theta \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta) \ge \exp\Big(-(1+C_1)l\varepsilon_l^2\Big)\Pi(A_l) \ge \exp(-(1+C_1+C)l\varepsilon_l^2).$$

Define the event $B_l = \left\{\int\limits_\Theta \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta) \le \exp\Big(-(1+C_1+C)l\varepsilon_l^2\Big)\right\}$.

Let Θ_l be the set satisfying conditions of Theorem 3.1. Then by Theorem 7.1 in Ghosal et al. (2000), there exist test functions $\phi_l := \phi_l(X_1,\dots,X_l)$ and a universal constant K such that

$$\begin{aligned} &\mathbb{E}_{P_0}\phi_l \le 2e^{-Kl\varepsilon_l^2}, \\ \sup_{\theta\in\Theta_l, h(P_\theta,P_0)\ge R\varepsilon_l} &\mathbb{E}_{P_\theta}(1-\phi_l) \le e^{-KR^2\cdot l\varepsilon_l^2}, \end{aligned} \tag{32}$$

where $KR^2 - 1 > K$.

Note that

$$\Pi_l(\theta:\rho(\theta,\theta_0)\ge R\varepsilon_l|X_1,\dots,X_l) = \Pi_l(\theta:\rho(\theta,\theta_0)\ge R\varepsilon_l|X_1,\dots,X_l)(\phi_l + 1 - \phi_l).$$

For the first term,

$$\mathbb{E}_{P_0}\Big[\Pi_l(\theta:\rho(\theta,\theta_0)\ge R\varepsilon_l|X_1,\dots,X_l)\cdot\phi_l\Big] \le \mathbb{E}_{P_0}\phi_l \le 2e^{-Kl\varepsilon_l^2}. \tag{33}$$

Next, by the definition of B_l, we have

$$\begin{aligned} &\Pi_l(\theta:\rho(\theta,\theta_0)\ge R\varepsilon_l|\mathcal{X}_l)(1-\phi_l) = \\ &\frac{\int\limits_{\rho(\theta,\theta_0)\ge R\varepsilon_l} \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta)(1-\phi_l)}{\int\limits_\Theta \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta)} (I\{B_l\} + I\{B_l^c\}) \\ &\le I\{B_l\} + e^{(1+C_1+C)l\varepsilon_l^2} \int\limits_{\rho(\theta,\theta_0)\ge R\varepsilon_l} \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta)(1-\phi_l). \end{aligned} \tag{34}$$

To estimate the second term of last equation, note that

$$\begin{aligned} &\mathbb{E}_{P_0} \int\limits_{\rho(\theta,\theta_0)\ge R\varepsilon_l} \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta)(1-\phi_l) \le \\ &\mathbb{E}_{P_0}\left(\int\limits_{\theta\in\Theta\setminus\Theta_l} \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta)(1-\phi_l) + \int\limits_{\{\Theta_l\cap\rho(\theta,\theta_0)\ge R\varepsilon_l\}} \prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta)(1-\phi_l)\right) \le \\ &\Pi(\Theta\setminus\Theta_l) + \int\limits_{\{\Theta_l\cap\rho(\theta,\theta_0)\}\ge R\varepsilon_l} \mathbb{E}_{P_0}\left(\prod_{i=1}^l \frac{p_\theta}{p_0}(X_i) d\Pi(\theta)(1-\phi_l)\right) \le \\ &e^{-l\varepsilon_l^2(C+4)} + e^{-KR^2\cdot l\varepsilon_l^2} \le 2e^{-l\varepsilon_l^2(C+4)} \end{aligned} \tag{35}$$

for $R \ge \sqrt{(C+4)/K}$. Set $C_1 = 1$ and note that $I\{B_l\} = 1$ with probability $P(B_l) \le 1/l\varepsilon_l^2$. It follows from (33), (34) and (35) and Chebyshev's inequality that for any $t > 0$

$$\begin{aligned} \Pr\Big(\Pi_l(\theta:\rho(\theta,\theta_0)\ge R\varepsilon_l|\mathcal{X}_l) \ge t\Big) &\le \Pr(B_l) + \frac{2e^{-Kl\varepsilon_l^2}}{t} + \frac{2\exp\left(-2l\varepsilon_l^2\right)}{t} \\ &\le \frac{1}{l\varepsilon_l^2} + \frac{2e^{-Kl\varepsilon_l^2}}{t} + \frac{2\exp\left(-2l\varepsilon_l^2\right)}{t}. \end{aligned}$$

Finally, for a constant $\tilde{K} = \min(K/2, 1)$ and $t = e^{-\tilde{K}l\varepsilon_l^2}$, we obtain

$$\begin{aligned} \Pr\Big(\Pi_l(\theta:\rho(\theta,\theta_0)\ge R\varepsilon_l|\mathcal{X}_l) \ge t\Big) &\le \frac{1}{l\varepsilon_l^2} + 2e^{-Kl\varepsilon_l^2/2} + 2\exp\left(-l\varepsilon_l^2\right) \\ &\le \frac{1}{l\varepsilon_l^2} + 4e^{-\tilde{K}l\varepsilon_l^2}, \end{aligned}$$

which yields the result.

A.3 Proof of Theorem 12

From equation (20) in the paper and proceeding as in proof of Theorem 7, we get that

$$\|\delta_0 - \Pi_l(\cdot|\mathcal{X}_l)\|_{\mathcal{F}_k} \leq R\varepsilon_l^{1/\gamma} + \int\limits_{\rho_k(\theta,\theta_0)\geq R\varepsilon_l^{1/\gamma}} \rho_k(\theta,\theta_0)d\Pi_l(\cdot|\mathcal{X}_l).$$

By Hölder's inequality,

$$\int\limits_{\rho_k(\theta,\theta_0)\geq R\varepsilon_l^{1/\gamma}} \rho_k(\theta,\theta_0)d\Pi_l(\cdot|\mathcal{X}_l) \leq$$

$$\left[\int\limits_{\Theta} \rho_k^w(\theta,\theta_0)d\Pi_l(\cdot|\mathcal{X}_l)\right]^{1/w} \left[\int\limits_{\rho_k(\theta,\theta_0)\geq R\varepsilon_l^{1/\gamma}} d\Pi_l(\cdot|\mathcal{X}_l)\right]^{1/q}$$

with $w > 1$ and $q = \frac{w}{w-1}$.

Define the event

$$B_l = \left\{\int\limits_{\Theta} \prod_{i=1}^{l} \frac{p_\theta}{p_0}(X_i)d\Pi(\theta) \geq \exp\Big(-(1+C_1+C)l\varepsilon_l^2\Big)\right\}.$$

Following in the proof of Theorem 3.1, we note that $\Pr(B_l) \geq 1 - \frac{1}{C_1^2 l\varepsilon_l^2}$ (where constants C, C_1 are the same as in the proof of Theorem 7). Also, note that

$$\mathbb{E}_{P_0}\left[\int_\Theta \rho_k^w(\theta,\theta_0)\prod_{i=1}^{l}\frac{p_\theta}{p_0}(X_i)d\Pi(\theta)\right] = \int_\Theta \rho_k^w(\theta,\theta_0)d\Pi(\theta)$$

Hence, with probability $\geq 1 - e^{-\tilde{K}l\varepsilon_l^2}$,

$$\int_\Theta \rho_k^w(\theta,\theta_0)\prod_{i=1}^{l}\frac{p_\theta}{p_0}(X_i)d\Pi(\theta) \leq e^{\tilde{K}l\varepsilon_l^2}\int_\Theta \rho_k^w(\theta,\theta_0)d\Pi(\theta),$$

where $\tilde{K} = \min(K/2, 1)$ is the same universal constant as in the proof of Theorem 7. Writing

$$\left[\int\limits_{\Theta} \rho_k^w(\theta,\theta_0)d\Pi_l(\cdot|\mathcal{X}_l)\right]^{1/w} = \left[\frac{\int_\Theta \rho_k^w(\theta,\theta_0)\prod_{i=1}^{l}\frac{p_\theta}{p_0}(X_i)d\Pi(\theta)}{\int_\Theta \prod_{i=1}^{l}\frac{p_\theta}{p_0}(X_i)d\Pi(\theta)}\right]^{1/w},$$

we deduce that with probability $\geq 1 - e^{-\tilde{K}l\varepsilon_l^2} - \frac{1}{l\varepsilon_l^2}$ (where we set $C_1 := 1$),

$$\left[\int\limits_{\Theta} \rho_k^w(\theta,\theta_0)d\Pi_l(\cdot|\mathcal{X}_l)\right]^{1/w} \leq e^{\frac{2+C+\tilde{K}}{w}l\varepsilon_l^2}\left[\int_\Theta \rho_k^w(\theta,\theta_0)d\Pi(\theta)\right]^{1/w}.$$

By Theorem 7.1 in Ghosal et al. (2000), there exist test functions $\phi_l := \phi_l(X_1,\ldots,X_l)$ and a universal constant K such that

$$\mathbb{E}_{P_0}\phi_l \leq 2e^{-Kl\varepsilon_l^2},$$
$$\sup_{\theta\in\Theta_l, d(\theta,\theta_0)\geq\tilde{R}\varepsilon_l} \mathbb{E}_{P_\theta}(1-\phi_l) \leq e^{-K\tilde{R}^2\cdot l\varepsilon_l^2},$$

where $K\tilde{R}^2 - 1 > K$ and $d(\cdot,\cdot)$ is the Hellinger or Euclidean distance. It immediately follows from Assumption 3.4 that

$$\Big\{\theta : \rho(\theta,\theta_0) \geq \tilde{C}_0(\theta_0)R^\gamma\varepsilon_l\Big\} \supseteq \Big\{\theta : \rho_k(\theta,\theta_0) \geq R\varepsilon_l^{1/\gamma}\Big\},$$

hence the test functions ϕ_l (for $\tilde{R} := \tilde{C}_0(\theta_0)R^\gamma$) satisfy

$$\mathbb{E}_{P_0}\phi_l \leq 2e^{-Kl\varepsilon_l^2}, \tag{36}$$
$$\sup_{\theta\in\Theta_l, \rho_k(\theta,\theta_0)\geq R\varepsilon_l^{1/\gamma}} \mathbb{E}_{P_\theta}(1-\phi_l) \leq e^{-KR^2\cdot l\varepsilon_l^2}.$$

Repeating the steps of the proof of Theorem 7, we see that if R is chosen large enough, then

$$\int\limits_{\rho_k(\theta,\theta_0)\geq R\varepsilon_l^{1/\gamma}} d\Pi_l(\cdot|\mathcal{X}_l) \leq e^{-\tilde{K}l\varepsilon_l^2}$$

with probability $\geq 1 - \frac{1}{l\varepsilon_l^2} - 4e^{-\tilde{K}l\varepsilon_l^2}$. Combining the bounds, we obtain that, for $w := \frac{4}{3} + \frac{4+2C}{3K}$, with probability $\geq 1 - \frac{1}{l\varepsilon_l^2} - 5e^{-\tilde{K}l\varepsilon_l^2}$,

$$\int\limits_{\rho(\theta,\theta_0)\geq R\varepsilon_l^{1/\gamma}} \rho_k(\theta,\theta_0)d\Pi_l(\cdot|\mathcal{X}_l) \leq e^{-\tilde{K}l\varepsilon_l^2/2}\left[\int_\Theta \rho_k^w(\theta,\theta_0)d\Pi(\theta)\right]^{1/w},$$

hence the result follows.

A.4 Proof of Theorem 15

The proof strategy is similar to Theorem 7. Note that

$$d_{W_{1,\rho}}(\delta_0, \Pi_{n,m}(\cdot|\mathcal{X}_l)) \leq R\varepsilon_l + \int\limits_{h(P_\theta,P_0)\geq R\varepsilon_l} d\Pi_l(\cdot|\mathcal{X}_l), \tag{37}$$

where $h(\cdot,\cdot)$ is the Hellinger distance.

Let $\mathcal{E}_l := \{\theta : h(P_\theta, P_0) \geq R\varepsilon_l\}$. By the definition of $\Pi_{n,m}$, we have

$$\Pi_{n,m}(\mathcal{E}_l|\mathcal{X}_l) = \frac{\int\limits_{\mathcal{E}_l}\Big(\prod_{j=1}^{l}\frac{p_\theta}{p_0}(X_j)\Big)^m d\Pi(\theta)}{\int\limits_{\Theta}\Big(\prod_{j=1}^{l}\frac{p_\theta}{p_0}(X_j)\Big)^m d\Pi(\theta)}. \tag{38}$$

To bound the denominator from below, we proceed as before. Let

$$\Theta_l = \left\{ \theta : -P_0 \left(\log \frac{p_\theta}{p_0} \right) \leq \varepsilon_l^2, P_0 \left(\log \frac{p_\theta}{p_0} \right)^2 \leq \varepsilon_l^2 \right\}.$$

Let B_l be the event defined by

$$B_l := \left\{ \int\limits_{\Theta_l} \left(\prod_{i=1}^{l} \frac{p_\theta}{p_0}(X_i) \right)^m dQ(\theta) \leq \exp(-2ml\varepsilon_l^2) \right\},$$

where Q is a probability measure supported on Θ_l. Lemma 8.1 in Ghosal et al. (2000) yields that $\Pr(B_l) \leq \frac{1}{l\varepsilon_l^2}$ for any Q, in particular, for the conditional distribution $\Pi(\cdot|\Theta_l)$. We conclude that

$$\int\limits_{\Theta} \left(\prod_{j=1}^{l} \frac{p_\theta}{p_0}(X_j) \right)^m d\Pi(\theta) \geq \Pi(\Theta_l) \exp(-2ml\varepsilon_l^2) \geq \exp(-(2m + C)l\varepsilon_l^2).$$

To estimate the numerator in (38), note that if Theorem 3.10 holds for $\gamma = \varepsilon_l$, then it also holds for $\gamma = L\varepsilon_l$ for any $L \geq 1$. This observation implies that

$$\sup_{\theta \in \mathcal{E}_l} \left(\prod_{j=1}^{l} \frac{p_\theta}{p_0}(X_j) \right)^m \leq e^{-c_1 R^2 ml\varepsilon_l^2}$$

with probability $\geq 1 - 4e^{-c_2 R^2 l\varepsilon_l^2}$, hence

$$\int\limits_{\mathcal{E}_l} \left(\prod_{j=1}^{l} \frac{p_\theta}{p_0}(X_j) \right)^m d\Pi(\theta) \leq e^{-c_1 R^2 ml\varepsilon_l^2}$$

with the same probability. Choose $R = R(C)$ large enough so that $c_1 m R^2 \geq 3m + C$. Putting the bounds for the numerator and denominator of (38) together, we get that with probability $\geq 1 - \frac{1}{l\varepsilon_l^2} - 4e^{-c_2 R^2 l\varepsilon_l^2}$,

$$\Pi_{n,m}(\mathcal{E}_l|\mathcal{X}_l) \leq e^{-ml\varepsilon_l^2}.$$

The result now follows from (37).

A.5 Proof of Proposition 17

The proof follows a standard pattern: on the first step, we show that it is enough to consider the posterior obtained from the prior restricted to a large compact set, and then proving the theorem for the prior with compact support. The second part mimics the classical argument exactly (e.g., see van der Vaart, 2000).

To show that one can restrict the prior to the compact set, it is enough to establish that for R large enough and $\mathcal{E}_l := \{\theta : \|\theta - \theta_0\| \geq \frac{R}{\sqrt{l}}\}$,

$$\Pi_{n,m}(\mathcal{E}_l|\mathcal{X}_l) = \frac{\int\limits_{\mathcal{E}_l} \left(\prod_{j=1}^{l} \frac{p_\theta}{p_0}(X_j) \right)^m d\Pi(\theta)}{\int\limits_{\Theta} \left(\prod_{j=1}^{l} \frac{p_\theta}{p_0}(X_j) \right)^m d\Pi(\theta)} \tag{39}$$

can be made arbitrarily small. This follows from the inclusion

$$\mathcal{E}_l \subseteq \left\{ \theta : h(P_\theta, P_{\theta_0}) \geq \tilde{C} \frac{R}{\sqrt{l}} \right\}$$

(due to the assumed inequality between Hellinger and Euclidean distances) and the bounds for the numerator and the denominator of (39) established in the proof of Theorem 15.

A.6 Selection of the optimal number of subsets m

The following heuristic approach picks the median among the candidate M-posteriors. Namely, start by evaluating the M-Posterior for each m in the range of candidate values $[m_1, m_2]$:

$$\begin{aligned} \hat{\Pi}^g_{n,m_1} &:= \mathrm{med}_g(\Pi_n^{(1)}, \ldots, \Pi_n^{(m_1)}), \\ \hat{\Pi}^g_{n,m_1+1} &:= \mathrm{med}_g(\Pi_n^{(1)}, \ldots, \Pi_n^{(m_1+1)}), \\ &\vdots \\ \hat{\Pi}^g_{n,m_2} &:= \mathrm{med}_g(\Pi_n^{(1)}, \ldots, \Pi_n^{(m_2)}) \end{aligned}$$

and choose $m_* \in [m_1, m_2]$ such that

$$\hat{\Pi}^g_{n,m_*} = \mathrm{med}_0 \left(\hat{\Pi}^g_{n,m_1}, \hat{\Pi}^g_{n,m_1+1}, \ldots, \hat{\Pi}^g_{n,m_2} \right), \tag{40}$$

where med_0 is the *metric median* defined in (3) in Section 2.2.

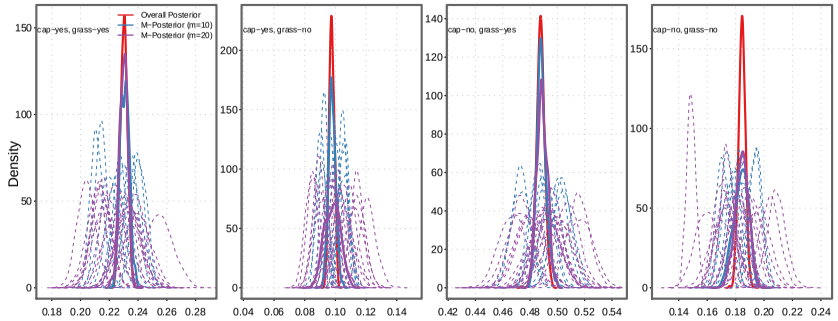
References

Alekh Agarwal and John C Duchi. Distributed delayed stochastic optimization. In *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 5451–5452, 2012.

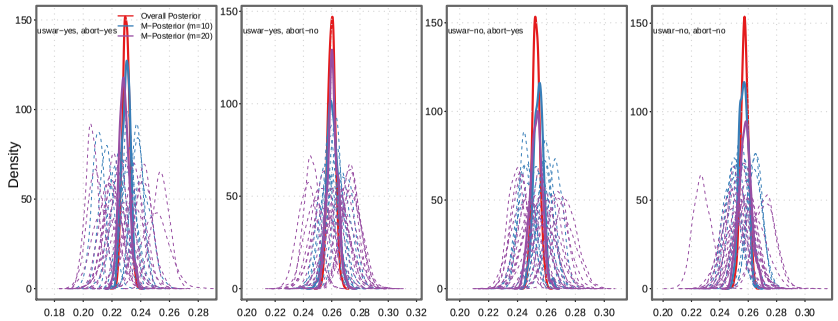
Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012.

Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 20–29. ACM, 1996.

Nachman Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.



(a) Support capital punishment (cap) and legalization of marijuana (grass)



(b) Expect US to be in world war in 10 years (uswar) and support abortion (abort)

Figure 6: Kernel density estimators of subset posteriors (dashed), M-Posterior with $m = 10$ (blue) and $m = 20$ (violet), and the overall posterior (red) for the four responses. In particular, the model "recognizes" that "expect US to be in world war in 10 years" (uswar) and "support abortion" (abort) should be independent.

M. J. Bayarri and James O. Berger. Robust Bayesian bounds for outlier detection. *Recent Advances in Statistics and Probability*, pages 175–190, 1994.

Amir Beck and Shoham Sabach. Weiszfeld's method: old and new results. *Preprint*, 2013. Available at `https://iew3.technion.ac.il/Home/Users/becka/Weiszfeld_review-v3.pdf`.

James O. Berger. An overview of robust Bayesian analysis. *Test*, 3(1):5–124, 1994.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.

Emmanuel Boissard and Thibaut Le Gouic. On the mean speed of convergence of empirical and occupation measures in wasserstein distance. *Ann. Inst. H. Poincaré Probab. Statist.*, 50(2):539–563, 2014.

Prosenjit Bose, Anil Maheshwari, and Pat Morin. Fast approximations for sums of distances, clustering, and the Fermat–Weber problem. *Computational Geometry*, 24(3):135–146, 2003.

George EP Box and George C Tiao. A Bayesian approach to some outlier problems. *Biometrika*, 55(1):119–129, 1968.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael Jordan. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, pages 1727–1735, 2013.

Hervé Cardot, Peggy Cénac, and Pierre-André Zitt. Recursive estimation of the conditional geometric median in Hilbert spaces. *Electronic Journal of Statistics*, 6:2535–2562, 2012.

Hervé Cardot, Peggy Cénac, and Pierre-André Zitt. Efficient and fast estimation of the geometric median in Hilbert spaces with an averaged stochastic gradient algorithm. *Bernoulli*, 19(1):18–43, 2013.

Bob Carpenter, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 20:1–37, 2016.

Ramaswamy Chandrasekaran and Arie Tamir. Algebraic optimization: the Fermat-Weber location problem. *Mathematical Programming*, 46(1-3):219–224, 1990.

Kjell A Doksum and Albert Y Lo. Consistent and robust Bayes procedures for location based on partial information. *The Annals of Statistics*, pages 443–453, 1990.

Richard M Dudley. *Real Analysis and Probability*, volume 74. Cambridge University Press, 2002.

David B Dunson and Chuanhua Xing. Nonparametric Bayes modeling of multivariate categorical data. *Journal of the American Statistical Association*, 104(487):1042–1051, 2009.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3):707–738, Aug 2015.

Subhashis Ghosal, Jayanta K Ghosh, and Aad W Van Der Vaart. Convergence rates of posterior distributions. *Annals of Statistics*, 28(2):500–531, 2000.

Peter D Hoff. Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied Statistics*, pages 265–283, 2007.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

Giles Hooker and Anand N Vidyashankar. Bayesian model robustness via disparities. *Test*, 23(3):556–584, 2014.

Daniel Hsu and Sivan Sabato. Loss minimization and parameter estimation with heavy tails. *arXiv preprint arXiv:1307.1827*, 2013.

Peter J. Huber and Elvezio M. Ronchetti. *Robust Statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons Inc., second edition, 2009.

J. H. B. Kemperman. The median of a finite measure on a Banach space. *Statistical Data Analysis Based on the L_1-norm and Related Methods, North-Holland, Amesterdam*, pages 217–230, 1987.

Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, and Michael I Jordan. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):795–816, 2014.

Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in MCMC land: cutting the Metropolis-Hastings budget. *arXiv preprint arXiv:1304.5299*, 2013.

Matthieu Lerasle and Roberto I. Oliveira. Robust empirical mean estimators. *arXiv preprint arXiv:1112.3914*, 2011.

Hendrik P Lopuhaa and Peter J Rousseeuw. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, pages 229–248, 1991.

Jeffrey W Miller and David B Dunson. Robust Bayesian inference via coarsening. *arXiv preprint arXiv:1506.06101*, 2015.

Stanislav Minsker. Geometric median and robust estimation in Banach spaces. *Bernoulli*, 21(4):2308–2335, 2015.

Willie Neiswanger, Chong Wang, and Eric Xing. Asymptotically exact, embarrassingly parallel MCMC. *arXiv preprint arXiv:1311.4780*, 2013.

Arkadi S. Nemirovskiĭ and B. Yudin David. *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley, 1983.

Frank Nielsen and Vincent Garcia. Statistical exponential families: a digest with flash cards. *arXiv preprint arXiv:0911.4863*, 2011.

Lawrence M. Ostresh. On the convergence of a class of iterative methods for solving the Weber location problem. *Operations Research*, 26(4):597–609, 1978.

Michael L. Overton. A quadratically convergent method for minimizing a sum of Euclidean norms. *Mathematical Programming*, 27(1):34–63, 1983.

Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. *the MIT Press*, 2, 2006.

Steven L Scott, Alexander W Blocker, Fernando V Bonassi, Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayes and big data: the consensus Monte Carlo algorithm, 2013.

Alexander Smola and Shravan Narayanamurthy. An Architecture for Parallel Topic Models. In *Very Large Databases (VLDB)*, 2010.

Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On integral probability metrics, ϕ-divergences and binary classification. *arXiv:0901.2698*, 2009.

Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert RG Lanckriet. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 99:1517–1561, 2010.

Sanvesh Srivastava, Volkan Cevher, Quoc Dinh, and David Dunson. WASP: Scalable Bayes via barycenters of subset posteriors. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 912–920, 2015a.

Sanvesh Srivastava, Cheng Li, and David B Dunson. Scalable Bayes via barycenter in Wasserstein space. *arXiv preprint arXiv:1508.05880*, 2015b.

Markus Svensen and Christopher M Bishop. Robust Bayesian mixture modelling. *Neurocomputing*, 64:235–252, 2005.

Aad van der Vaart. *Asymptotic Statistics*. Cambridge university press, 2000.

Aad van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes*. Springer Series in Statistics. Springer-Verlag, New York, 1996.

Chong Wang, John W Paisley, and David M Blei. Online variational inference for the hierarchical Dirichlet process. In *International Conference on Artificial Intelligence and Statistics*, pages 752–760, 2011.

Xiangyu Wang and David B Dunson. Parallel MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605*, 2013.

Endre Weiszfeld. Sur un problème de minimum dans l'espace. *Tohoku Mathematical Journal*, 1936.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

Wing Hung Wong, Xiaotong Shen, et al. Probability inequalities for likelihood ratios and convergence rates of sieve MLEs. *The Annals of Statistics*, 23(2):339–362, 1995.

Journal of Machine Learning Research 18 (2017) 1-53 Submitted 3/16; Revised 8/17; Published 12/17

Statistical and Computational Guarantees for the Baum-Welch Algorithm

Fanny Yang FANNY-YANG@BERKELEY.EDU
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720-1776, USA

Sivaraman Balakrishnan SIVA@STAT.CMU.EDU
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Martin J. Wainwright WAINWRIG@BERKELEY.EDU
Department of Statistics
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, CA 94720-1776, USA

Editor: Sanjoy Dasgupta

Abstract

The Hidden Markov Model (HMM) is one of the mainstays of statistical modeling of discrete time series, with applications including speech recognition, computational biology, computer vision and econometrics. Estimating an HMM from its observation process is often addressed via the Baum-Welch algorithm, which is known to be susceptible to local optima. In this paper, we first give a general characterization of the basin of attraction associated with any global optimum of the population likelihood. By exploiting this characterization, we provide non-asymptotic finite sample guarantees on the Baum-Welch updates and show geometric convergence to a small ball of radius on the order of the minimax rate around a global optimum. As a concrete example, we prove a linear rate of convergence for a hidden Markov mixture of two isotropic Gaussians given a suitable mean separation and an initialization within a ball of large radius around (one of) the true parameters. To our knowledge, these are the first rigorous local convergence guarantees to global optima for the Baum-Welch algorithm in a setting where the likelihood function is nonconvex. We complement our theoretical results with thorough numerical simulations studying the convergence of the Baum-Welch algorithm and illustrating the accuracy of our predictions.

Keywords: Hidden Markov Models, Baum-Welch algorithm, EM algorithm, non-convex optimization, graphical models

1. Introduction

Hidden Markov models (HMMs) are one of the most widely applied statistical models of the last 50 years, with major success stories in computational biology (Durbin, 1998), signal processing and speech recognition (Rabiner and Juang, 1993), control theory (Elliott et al., 1995), and econometrics (Kim and Nelson, 1999) among other disciplines. At a high level, a hidden Markov model is a Markov process split into an observable component and an unobserved or latent component. From a statistical standpoint, the use of latent states makes the HMM generic enough to model a variety of complex real-world time series, while the Markovian structure enables relatively simple computational procedures.

In applications of HMMs, an important problem is to estimate the state transition probabilities and the parameterized output densities based on samples of the observable component. From classical theory, it is known that under suitable regularity conditions, the maximum likelihood estimate (MLE) in an HMM has good statistical properties (Bickel et al., 1998). However, given the potentially nonconvex nature of the likelihood surface, computing the global maximum that defines the MLE is not a straightforward task. In fact, the HMM estimation problem in full generality is known to be computationally intractable under cryptographic assumptions (Terwijn, 2002). In practice, however, the Baum-Welch algorithm (Baum et al., 1970) is frequently applied and leads to good results. It can be understood as the specialization of the EM algorithm (Dempster et al., 1977) to the maximum likelihood estimation problem associated with the HMM. Despite its wide use in many applications, the Baum-Welch algorithm can get trapped in local optima of the likelihood function. Understanding when this undesirable behavior occurs—or does not occur—has remained an open question for several decades.

A more recent line of work (Mossel and Roch, 2006; Siddiqi et al., 2010; Hsu et al., 2012) has focused on developing tractable estimators for HMMs, via approaches that are distinct from the Baum-Welch algorithm. Nonetheless, it has been observed that the practical performance of such methods can be significantly improved by running the Baum-Welch algorithm using their estimators as the initial point; see, for instance, the detailed empirical study in Kontorovich et al. (2013). This curious phenomenon has been observed in other contexts (Chaganty and Liang, 2013), but has not been explained to date. Obtaining a theoretical characterization of when and why the Baum-Welch algorithm behaves well is the main objective of this paper.

1.1 Related work and our contributions

Our work builds upon a framework for analysis of EM, as previously introduced by a subset of the current authors (Balakrishnan et al., 2014); see also the follow-up work to regularized EM algorithms (Yi and Caramanis, 2015; Wang et al., 2014). All of this past work applies to models based on i.i.d. samples, and as we show in this paper, there are a number of non-trivial steps required to derive analogous theory for the dependent variables that arise for HMMs. Before doing so, let us put the results of this paper in context relative to older and more classical work on Baum-Welch and related algorithms.

Under mild regularity conditions, it is well-known that the maximum likelihood estimate (MLE) for an HMM is a consistent and asymptotically normal estimator; for instance, see Bickel et al. (1998), as well as the expository works (Cappé et al., 2004; van Handel, 2008). On the algorithmic level, the original papers of Baum and co-authors (Baum et al., 1970; Baum and Petrie, 1966) showed that the Baum-Welch algorithm converges to a stationary point of the sample likelihood; these results are in the spirit of the classical convergence analysis of the EM algorithm (Wu, 1983; Dempster et al., 1977). These classical convergence results only provide a relatively weak guarantee—namely, that if the algorithm is initialized

©2017 Fanny Yang, Sivaraman Balakrishnan and Martin J. Wainwright.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-093.html.

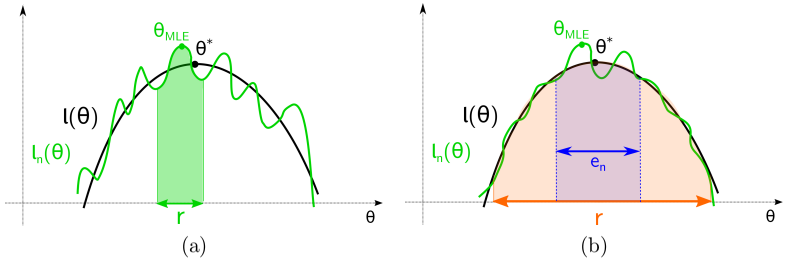


Figure 1: (a) A poorly behaved sample likelihood, for which there are many local optima at varying distances from the MLE. It would require an initialization extremely close to the MLE in order to ensure that the Baum-Welch algorithm would not be trapped at a sub-optimal fixed point. (b) A well-behaved sample likelihood, for which all local optima lie within an e_n-ball of the MLE, as well as the true parameter θ^*. In this case, the Baum-Welch algorithm, when initialized within a ball of large radius r, will converge to the ball of much smaller radius e_n. The goal of this paper is to give sufficient conditions for when the sample likelihood exhibits this favorable structure.

sufficiently close to the MLE, then it will converge to it. However, the classical analysis does not quantify the size of this neighborhood, and as a critical consequence, it *does not* rule out the pathological type of behavior illustrated in panel (a) of Figure 1. Here the sample likelihood has multiple optima, both a global optimum corresponding to the MLE as well as many local optima *far away from the MLE* that are also fixed points of the Baum-Welch algorithm. In such a setting, the Baum-Welch algorithm will only converge to the MLE if it is initialized in an extremely small neighborhood.

In contrast, the goal of this paper is to give sufficient conditions under which the sample likelihood has the more favorable structure shown in panel (b) of Figure 1. Here, even though the MLE does not have a large basin of attraction, the sample likelihood has all of its optima (including the MLE) localized to a small region around the true parameter θ^*. Our strategy to reveal this structure, as in our past work (Balakrishnan et al., 2014), is to shift perspective: instead of studying convergence of Baum-Welch updates to the MLE, we study their convergence to an ϵ_n-ball of the true parameter θ^*, and moreover, instead of focusing exclusively on the sample likelihood, we first study the structure of the population likelihood, corresponding to the idealized limit of infinite data. Our first main result (Theorem 1) provides sufficient conditions under which there is a large ball of radius r, over which the population version of the Baum-Welch updates converge at a geometric rate to θ^*. Our second main result (Theorem 2) uses empirical process theory to analyze the finite-sample version of the Baum-Welch algorithm, corresponding to what is actually implemented in practice. In this finite sample setting, we guarantee that over the ball of radius r, the Baum-Welch updates will converge to an ϵ_n-ball with $\epsilon_n \ll r$, and most importantly, this ϵ_n-ball contains the true parameter θ^*. Typically this ball also contains the MLE with high-probability, but our theory does *not* guarantee convergence to the MLE,

but rather to a point that is close to both the MLE and the true parameter θ^* and whose statistical risk is equivalent to that of the MLE upto logarithmic factors.

These latter two results are abstract, applicable to a broad class of HMMs. We then specialize them to the case of a hidden Markov mixture consisting of two isotropic components, with means separated by a constant distance, and obtain concrete guarantees for this model. It is worth comparing these results to past work in the i.i.d. setting, for which the problem of Gaussian mixture estimation under various separation assumptions has been extensively studied (e.g. (Dasgupta, 1999; Vempala and Wang, 2004; Belkin and Sinha, 2010; Moitra and Valiant, 2010)). The constant distance separation required in our work is much weaker than the separation assumptions imposed in most papers that focus on correctly labeling samples in a mixture model. Our separation condition is related to, but in general incomparable with the non-degeneracy requirements in other work (Hsu et al., 2012; Hsu and Kakade, 2013; Moitra and Valiant, 2010).

Finally, let us discuss the various challenges that arise in studying the dependent data setting of hidden Markov models, and highlight some important differences with the i.i.d. setting (Balakrishnan et al., 2014). In the non-i.i.d. setting, arguments passing from the population-based to sample-based updates are significantly more delicate. First of all, it is not even obvious that the population version of the Q-function—a central object in the Baum-Welch updates— exists. From a technical standpoint, various gradient smoothness conditions are much more difficult to establish, since the gradient of the likelihood no longer decomposes over the samples as in the i.i.d. setting. In particular, each term in the gradient of the likelihood is a function of all observations. Finally, in order to establish the finite-sample behavior of the Baum-Welch algorithm, we can no longer appeal to standard i.i.d. concentration and empirical process techniques. Nor do we pursue the approach of some past work on HMM estimation (e.g. (Hsu et al., 2012)), in which it is assumed that there are multiple independent samples of the HMM.[1] Instead, we directly analyze the Baum-Welch algorithm that practioners actually use—namely, one that applies to a single sample of an n-length HMM. In order to make the argument rigorous, we need to make use of more sophisticated techniques for proving concentration for dependent data (Yu, 1994; Nobel and Dembo, 1993).

The remainder of this paper is organized as follows. In Section 2, we introduce basic background on hidden Markov models and the Baum-Welch algorithm. Section 3 is devoted to the statement of our main results in the general setting, whereas Section 4 contains the more concrete consequences for the Gaussian output HMM. The main parts of our proofs are given in Section 5, with the more technical details deferred to the appendices.

2. Background and problem set-up

In this section, we introduce some standard background on hidden Markov models and the Baum-Welch algorithm.

1. The rough argument here is that it is possible to reduce an i.i.d. sampling model by cutting the original sample into many pieces, but this is not an algorithm that one would implement in practice.

2.1 Standard HMM notation and assumptions

We begin by defining a discrete-time hidden Markov model with hidden states taking values in a discrete space. Letting $\mathbb{Z}$ denote the integers, suppose that the observed random variables $\{X_i\}_{i\in\mathbb{Z}}$ take values in $\mathbb{R}^d$, and the latent random variables $\{Z_i\}_{i\in\mathbb{Z}}$ take values in the discrete space $[s] := \{1, \ldots, s\}$. The Markov structure is imposed on the sequence of latent variables. In particular, if the variable Z_1 has some initial distribution π_1, then the joint probability of a particular sequence $(z_1, \ldots, z_n)$ is given by

$$p(z_1, \ldots, z_n; \beta) = \pi_1(z_1; \beta) \prod_{i=1}^{n} p(z_i \mid z_{i-1}; \beta), \tag{1}$$

where the vector β is a particular parameterization of the initial distribution and Markov chain transition probabilities. We restrict our attention to the homogeneous case, meaning that the transition probabilities for step $(t-1) \to t$ are independent of the index t. Consequently, if we define the transition matrix $A \in \mathbb{R}^{s\times s}$ with entries

$$A(j, k; \beta) := p(z_2 = k \mid z_1 = j; \beta),$$

then the marginal distribution π_i of Z_i can be described by the matrix vector equation

$$\pi_i^T = \pi_1^T A^{i-1},$$

where π_i and π_1 denote vectors belonging to the s-dimensional probability simplex.

We assume throughout that the Markov chain is aperiodic and recurrent, whence it has a unique stationary distribution $\bar{\pi}$, defined by the eigenvector equation $\bar{\pi}^T = \bar{\pi}^T A$. To be clear, both $\bar{\pi}$ and the matrix A depend on β, but we omit this dependence so as to simplify notation. We assume throughout that the Markov chain begins in its stationary state, so that $\pi_1 = \bar{\pi}$, and moreover, that it is reversible, meaning that

$$\bar{\pi}(j)A(j, k) = \bar{\pi}(k)A(k, j) \tag{2}$$

for all pairs $j, k \in [s]$.

A key quantity in our analysis is the mixing rate of the Markov chain. In particular, we assume the existence of *mixing constant* $\epsilon_{\text{mix}} \in (0, 1]$ such that

$$\epsilon_{\text{mix}} \leq \frac{p(z_i|z_{i-1}; \beta)}{\bar{\pi}(z_i)} \leq \epsilon_{\text{mix}}^{-1} \tag{3}$$

for all $(z_i, z_{i-1}) \in [s] \times [s]$. This condition implies that the dependence on the initial distribution decays geometrically. More precisely, some simple algebra shows that

$$\sup_{\pi_1} \left\|\pi_1^T A^t - \pi_1^T\right\|_{\text{TV}} \leq c_0 \rho_{\text{mix}}^t \qquad \text{for all } t = 1, 2, \ldots, \tag{4}$$

where $\rho_{\text{mix}} = 1 - \epsilon_{\text{mix}}$ denotes the *mixing rate* of the process, and c_0 is a universal constant. Note that as $\epsilon_{\text{mix}} \to 1^-$, the Markov chain has behavior approaching that of an i.i.d. sequence, whereas as $\epsilon_{\text{mix}} \to 0^+$, its behavior becomes increasingly "sticky".

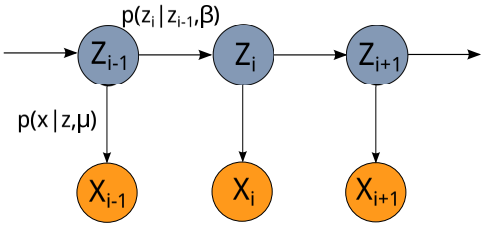


Figure 2: The hidden Markov model as a graphical model. The blue circles indicate observed variables Z_i, whereas the orange circles indicate latent variables X_i.

Associated with each latent variable Z_i is an observation $X_i \in \mathbb{R}^d$. We use $p(x_i|z_i; \mu)$ to denote the density of X_i given that $Z_i = z_i$, an object that we assume to be parameterized by a vector μ. Introducing the shorthand $x_1^n = (x_1, \ldots, x_n)$ and $z_1^n = (z_1, \ldots, z_n)$, the joint probability of the sequence (x_1^n, z_1^n) (also known as the complete likelihood) can be written in the form

$$p(z_1^n, x_1^n; \theta) = \pi_1(z_1) \prod_{i=2}^{n} p(z_i \mid z_{i-1}; \beta) \prod_{i=1}^{n} p(x_i|z_i; \mu), \tag{5}$$

where the pair $\theta := (\beta, \mu)$ parameterizes the transition and observation functions. The likelihood then reads

$$p(x_1^n; \theta) = \sum_{z_1^n} p(z_1^n, x_1^n; \theta).$$

For our convenience in subsequent analysis, we also define a form of complete likelihood including an additional hidden variable z_0 which is not associated to any observation x_0

$$p(z_0^n, x_1^n; \theta) = \pi_0(z_0) \prod_{i=1}^{n} p(z_i \mid z_{i-1}; \beta) \prod_{i=1}^{n} p(x_i|z_i; \mu), \tag{6}$$

where $\pi_0 = \bar{\pi}$. Note that it preserves the usual relationship $\sum_{z_0^n} p(z_0^n, x_1^n; \theta) = p(x_1^n; \theta)$ between the ordinary and complete likelihoods in EM problems.

A simple example: A special case helps to illustrate these definitions. In particular, suppose that we have a Markov chain with $s = 2$ states. Consider a matrix of transition probabilities $A \in \mathbb{R}^{2\times 2}$ of the form

$$A = \frac{1}{e^\beta + e^{-\beta}} \begin{bmatrix} e^\beta & e^{-\beta} \\ e^{-\beta} & e^\beta \end{bmatrix} = \begin{bmatrix} \zeta & 1-\zeta \\ 1-\zeta & \zeta \end{bmatrix}, \tag{7}$$

where $\zeta := \frac{e^\beta}{e^\beta + e^{-\beta}}$. By construction, this Markov chain is recurrent and aperiodic with the unique stationary distribution $\bar{\pi} = \begin{bmatrix}\frac{1}{2} & \frac{1}{2}\end{bmatrix}^T$. Moreover, by calculating the

eigenvalues of the transition matrix, we find that the mixing condition (4) holds with $\rho_{\mathrm{mix}} := |2\zeta - 1| = |\tanh(\beta)|$.

Suppose moreover that the observed variables in $\mathbb{R}^d$ are conditionally Gaussian, say with

$$p(x_t|z_t;\mu) = \begin{cases} \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\big\{ -\frac{1}{2\sigma^2}\|x-\mu\|_2^2\big\} & \text{if } z_t = 1 \\ \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\big\{ -\frac{1}{2\sigma^2}\|x+\mu\|_2^2\big\} & \text{if } z_t = 2. \end{cases} \tag{8}$$

With this choice, the marginal distribution of each X_t is a two-state Gaussian mixture with mean vectors μ and $-\mu$, and covariance matrices $\sigma^2 I_d$. We provide specific consequences of our general theory for this special case in the sequel.

2.2 Baum-Welch updates for HMMs

We now describe the Baum-Welch updates for a general discrete-state hidden Markov model. As a special case of the EM algorithm, the Baum-Welch algorithm is guaranteed to ascend on the likelihood function of the hidden Markov model. It does so indirectly, by first computing a lower bound on the likelihood (E-step) and then maximizing this lower bound (M-step).

For a given integer $n \geq 1$, suppose that we observe a sequence $x_1^n = (x_1, \ldots, x_n)$ drawn from the marginal distribution over X_1^n defined by the model (5). The rescaled log likelihood of the sample path x_1^n is given by

$$\ell_n(\theta) = \frac{1}{n} \log \Big(\sum_{z_0^n} p(z_0^n, x_1^n; \theta) \Big)$$

The EM likelihood is based on lower bounding the likelihood via Jensen's inequality. For any choice of parameter θ' and positive integers $i \leq j$ and $a < b$, let $\mathbb{E}_{Z_i^j | x_a^b, \theta'}$ denote the expectation under the conditional distribution $p(Z_i^j \mid x_a^b; \theta')$. With this notation, the concavity of the logarithm and Jensen's inequality imply that for any choice of θ', we have the lower bound

$$\begin{aligned} \ell_n(\theta) &= \frac{1}{n} \log \big[\mathbb{E}_{Z_0^n | x_1^n, \theta'} \frac{p(Z_0^n, x_1^n; \theta)}{p(Z_0^n \mid x_1^n; \theta')} \big] \\ &\geq \underbrace{\frac{1}{n} \mathbb{E}_{Z_0^n | x_1^n, \theta'} \big[\log p(Z_0^n, x_1^n; \theta) \big]}_{Q_n(\theta \,|\, \theta')} + \underbrace{\frac{1}{n} \mathbb{E}_{Z_0^n | x_1^n, \theta'} \big[-\log p(Z_0^n \mid x_1^n; \theta') \big]}_{H_n(\theta')}. \end{aligned}$$

For a given choice of θ', the E-step corresponds to the computation of the function $\theta \mapsto Q_n(\theta \mid \theta')$. The M-step is defined by the EM operator $M_n : \widetilde{\Omega} \mapsto \widetilde{\Omega}$

$$M_n(\theta') = \arg\max_{\theta \in \widetilde{\Omega}} Q_n(\theta \mid \theta'), \tag{9}$$

where $\widetilde{\Omega}$ is the set of feasible parameter vectors. Overall, given an initial vector $\theta^0 = (\beta^0, \mu^0)$, the EM algorithm generates a sequence $\{\theta^t\}_{t=0}^{\infty}$ according to the recursion $\theta^{t+1} = M_n(\theta^t)$.

This description can be made more concrete for an HMM, in which case the Q-function takes the form

$$\begin{aligned} Q_n(\theta \mid \theta') = \frac{1}{n} \mathbb{E}_{Z_0 | x_1^n, \theta'} \big[\log \pi_0(Z_0; \beta) \big] + \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{Z_{i-1}, Z_i | x_1^n, \theta'} \big[\log p(Z_i \mid Z_{i-1}; \beta) \big] \\ + \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{Z_i | x_1^n, \theta'} \big[\log p(x_i \mid Z_i; \mu) \big], \end{aligned} \tag{10}$$

where the dependence of π_0 on β comes from the assumption that $\pi_0 = \bar{\pi}$. Note that the Q-function can be decomposed as the sum of a term which is solely dependent on μ, and another one which only depends on β—that is

$$Q_n(\theta \mid \theta') = Q_{1,n}(\mu \mid \theta') + Q_{2,n}(\beta \mid \theta') \tag{11}$$

where $Q_{1,n}(\mu \mid \theta') = \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{Z_i | x_1^n, \theta'} \big[\log p(x_i \mid Z_i, \mu) \big]$, and $Q_{2,n}(\beta \mid \theta')$ collects the remaining terms. In order to compute the expectations defining this function (E-step), we need to determine the marginal distributions over the singletons Z_i and pairs (Z_i, Z_{i+1}) under the joint distribution $p(Z_0^n \mid x_1^n; \theta')$. These marginals can be obtained efficiently using a recursive message-passing algorithm, known either as the forward-backward or sum-product algorithm (Kschischang et al., 2001; Wainwright and Jordan, 2008).

In the M-step, the decomposition (11) suggests that the maximization over the two components (β, μ) can also be decoupled. Accordingly, with a slight abuse of notation, we often write

$$M_n^{\mu}(\theta') = \arg\max_{\mu \in \Omega_\mu} Q_{1,n}(\mu \mid \theta'), \quad \text{and} \quad M_n^{\beta}(\theta') = \arg\max_{\beta \in \Omega_\beta} Q_{2,n}(\beta \mid \theta')$$

for these two decoupled maximization steps, where Ω_β and Ω_μ denote the feasible set of transition and observation parameters respectively and $\widetilde{\Omega} := \Omega_\beta \times \Omega_\mu$. In the following, unless otherwise stated, $\Omega_\mu = \mathbb{R}^d$, so that the maximization over the observation parameters is unconstrained.

3. Main results

We now turn to the statement of our main results, along with a discussion of some of their consequences. The first step is to establish the existence of an appropriate population analog of the Q-function. Although the existence of such an object is a straightforward consequence of the law of large numbers in the case of i.i.d. data, it requires some technical effort to establish existence for the case of dependent data; in particular, we do so using a k-truncated version of the full Q-function (see Proposition 1). This truncated object plays a central role in the remainder of our analysis. In particular, we first analyze a version of the Baum-Welch updates on the expected k-truncated Q-function for an extended sequence of observations x_{1-k}^{n+k}, and provide sufficient conditions for these population-level updates to be contractive (see Theorem 1). We then use non-asymptotic forms of empirical process theory to show that under suitable conditions, the actual sample-based EM updates—i.e., the updates that are actually implemented in practice—are also well-behaved in this region with high probability (see Theorem 2). In subsequent analysis to follow in Section 4, we show that this initialization radius is suitably large for an HMM with Gaussian outputs.

3.1 Existence of population Q-function

In the analysis of Balakrishnan et al. (2014), the central object is the notion of a population Q-function—namely, the function that underlies the EM algorithm in the idealized limit of infinite data. In their setting of i.i.d. data, the standard law of large numbers ensures that as the sample size n increases, the sample-based Q-function approaches its expectation, namely the function

$$\bar{Q}(\theta \mid \theta') = \mathbb{E}\big[Q_n(\theta \mid \theta')\big] \;=\; \mathbb{E}\big[\mathbb{E}_{Z_1|X_1,\theta'}\big[\log p(X_1, Z_1; \theta)\big]\big].$$

Here we use the shorthand $\mathbb{E}$ for the expectation over all samples X that are drawn from the joint distribution (in this case $\mathbb{E} := \mathbb{E}_{X_1^n|\theta^*}$).

When the samples are dependent, the quantity $\mathbb{E}\big[Q_n(\theta \mid \theta')\big]$ is no longer independent of n, and so an additional step is required. A reasonable candidate for a general definition of the population Q-function is given by

$$\bar{Q}(\theta \mid \theta') := \lim_{n \to +\infty} [\mathbb{E} Q_n(\theta \mid \theta')]. \tag{12}$$

Although it is clear that this definition is sensible in the i.i.d. case, it is necessary for dependent sampling schemes to prove that the limit given in definition (12) actually exists.

In this paper, we do so by considering a suitably truncated version of the sample-based Q-function. Similar arguments have been used in past work (e.g., (Cappé et al., 2004; van Handel, 2008)) to establish consistency of the MLE; here our focus is instead on the behavior of the Baum-Welch algorithm. Let us consider a sequence $\{(X_i, Z_i)\}_{i=1-k}^{n+k}$, assumed to be drawn from the stationary distribution of the overall chain. Recall that $\mathbb{E}_{Z_i^j|x_a^b,\theta}$ denotes expectations taken over the distribution $p(Z_i^j \mid x_a^b, \theta)$. Then, for a positive integer k to be chosen, we define

$$Q_n^k(\theta \mid \theta') = \frac{1}{n}\Big[\mathbb{E}_{Z_0|x_{-k}^k,\theta'} \log p(Z_1; \beta) + \sum_{i=1}^{n} \mathbb{E}_{Z_{i-1}^i|x_{i-k}^{i+k},\theta'} \log p(Z_i \mid Z_{i-1}; \beta) + \sum_{i=1}^{n} \mathbb{E}_{Z_i|x_{i-k}^{i+k},\theta'} \log p(x_i \mid Z_i; \mu)\Big]. \tag{13}$$

In an analogous fashion to the decomposition in equation (10), we can decompose Q_n^k in the form

$$Q_n^k(\theta \mid \theta') = Q_{1,n}^k(\mu \mid \theta') + Q_{2,n}^k(\beta \mid \theta').$$

We associate with this triplet of Q-functions the corresponding EM operators $M_n^k(\theta')$, $M_n^{\mu,k}(\theta')$ and $M_n^{\beta,k}(\theta')$ as in Equation (9). Note that as opposed to the function Q_n from equation (10), the definition of Q_n^k involves variables Z_i, Z_{i-1} that are not conditioned on the full observation sequence x_1^n, but instead only on a $2k$ window centered around the index i. By construction, we are guaranteed that the k-truncated population function and its decomposed analogs given by

$$\begin{aligned}\bar{Q}^k(\theta \mid \theta') &:= \lim_{n\to\infty} \mathbb{E} Q_n^k(\theta \mid \theta') = \mathbb{E} Q_{1,n}^k(\mu \mid \theta') + \lim_{n\to\infty} \mathbb{E} Q_{2,n}^k(\beta \mid \theta') \\ &:= \bar{Q}_1^k(\mu \mid \theta') + \bar{Q}_2^k(\beta \mid \theta')\end{aligned} \tag{14}$$

are well-defined. In particular, due to stationarity of the random sequences $\{p(z_i \mid X_{i-k}^{i+k})\}_{i=1}^n$ and $\{p(z_{i-1}^i \mid X_{i-k}^{i+k})\}_{i=1}^n$, the expectation over $\{(X_i, Z_i)\}_{i=1-k}^{n+k}$ is independent of the sample size n. Notice that the Baum-Welch algorithm in practice essentially corresponds to using $k = n$.

Our first result uses the existence of this truncated population object in order to show that the standard population Q-function from equation (12) is indeed well-defined. In doing so, we make use of the sup-norm

$$\|Q_1 - Q_2\|_\infty := \sup_{\theta,\theta' \in \widetilde{\Omega}} \Big|Q_1(\theta \mid \theta') - Q_2(\theta \mid \theta')\Big|. \tag{15}$$

We require in the following that the observation densities satisfy the following boundedness condition

$$\sup_{\theta \in \widetilde{\Omega}} \mathbb{E}\Big[\max_{z_i \in [s]} \big|\log p(X_i \mid z_i, \theta)\big|\Big] < \infty. \tag{16}$$

Proposition 1 *Under the previously stated assumptions, the population function $\bar{Q}$ defined in equation* (12) *exists.*

The proof of this claim is given in Appendix A. It hinges on the following auxiliary claim, which bounds the difference between $\mathbb{E} Q_n$ and the k-truncated Q-function as

$$\|\mathbb{E} Q_n - \bar{Q}^k\|_\infty \leq \frac{c\, s^4}{\epsilon_{\mathrm{mix}}^9 \overline{\pi}_{\min}^2}\big(1 - \epsilon_{\mathrm{mix}} \overline{\pi}_{\min}\big)^k + \frac{c(\overline{\pi}_{\min}, s, \epsilon_{\mathrm{mix}})}{n}, \tag{17}$$

where $\overline{\pi}_{\min} := \min_{\beta \in \Omega_\beta, j \in [s]} \overline{\pi}(j \mid \beta)$ is the minimum probability in the stationary distribution, ϵ_{mix} is the mixing constant from equation (3), and $c(\cdot)$ is a constant dependent only on the inherent model parameters. The dependencies on ϵ_{mix} and $\overline{\pi}_{\min}$ are not optimized here. Since this bound holds for all n, it shows that the population function $\bar{Q}$ can be uniformly approximated by $\bar{Q}^k$, with the approximation error decreasing geometrically as the truncation level k grows. This fact plays an important role in the analysis to follow.

3.2 Analysis of updates based on $\bar{Q}^k$

Our ultimate goal is to establish a bound on the difference between the sample-based Baum-Welch estimate and θ^*, in particular showing contraction of the Baum-Welch update towards the true parameter. Our strategy for doing so involves first analyzing the Baum-Welch iterates at the population level, which is the focus of this section.

The quantity $\bar{Q}$ is significant for the EM updates because the parameter θ^* satisfies the self-consistency property $\theta^* = \arg\max_\theta \bar{Q}(\theta \mid \theta^*)$. In the i.i.d. setting, the function $\bar{Q}$ can often be computed in closed form, and hence directly analyzed, as was done in past work Balakrishnan et al. (2014). In the HMM case, this function $\bar{Q}$ no longer has a closed form, so an alternative route is needed. Here we analyze the population version via the truncated function $\bar{Q}^k$ (14) instead, where k is a given truncation level (to be chosen in the sequel). Although θ^* is no longer a fixed point of $\bar{Q}^k$, the bound (17) combined with

the assumption of strong concavity of $\overline{Q}^k$ imply an upper bound on the distance of the maximizers of $\overline{Q}^k$ and $\overline{Q}$.

With this setup, we consider an idealized population-level algorithm that, based on some initialization $\widetilde{\theta}^0 \in \Omega = \mathbb{B}_2\big(r; \mu^*\big) \times \Omega_\beta$, generates the sequence of iterates

$$\widetilde{\theta}^{t+1} = \overline{M}^k(\widetilde{\theta}^t) := \arg\max_{\theta \in \widetilde{\Omega}} \overline{Q}^k(\theta \mid \widetilde{\theta}^t). \tag{18}$$

where $\widetilde{\Omega} = \Omega_\beta \times \Omega_\mu$ is a larger set than Ω, especially $\Omega_\mu = \mathbb{R}^d$. Since $\overline{Q}^k$ is an approximate version of $\overline{Q}$, the update operator $\overline{M}^k$ should be understood as an approximation to the idealized population EM operator $\overline{M}$ where the maximum is taken with respect to $\overline{Q}$. As part (a) of the following theorem shows, the approximation error is well-controlled under suitable conditions. We analyze the convergence of the sequence $\{\widetilde{\theta}^t\}_{t=0}^\infty$ in terms of the norm $\|\cdot\|_\star : \Omega_\mu \times \Omega_\beta \to \mathbb{R}^+$ given by

$$\|\theta - \theta^*\|_\star = \|(\mu, \beta) - (\mu^*, \beta^*)\|_\star := \|\mu - \mu^*\|_2 + \|\beta - \beta^*\|_2. \tag{19}$$

Contraction in this norm implies that both parameters μ, β converge linearly to the true parameter.

Conditions on $\overline{Q}^k$: Let us now introduce the conditions on the truncated function $\overline{Q}^k$ that underlie our analysis. For this purpose, we concentrate on a potentially smaller set

$$\Omega := \mathbb{B}_2(r; \mu^*) \times \Omega_\beta$$

with radius $r > 0$, where Ω_β is the set of allowable HMM transition parameters. The goal is to find the largest $\Omega \subset \widetilde{\Omega}$, in which said conditions are fulfilled. This set Ω is then equivalent to the basin of attraction, i.e. the set in which we can initialize the algorithm and obtain linear convergence to a good optimum.

First, let us say that the function $\overline{Q}^k(\cdot \mid \theta')$ is $(\lambda_\mu, \lambda_\beta)$-*strongly concave* in Ω if for all $\theta' \in \Omega$ we have

$$\overline{Q}_1^k(\mu_1 \mid \theta') - \overline{Q}_1^k(\mu_2 \mid \theta') - \langle \nabla_\mu \overline{Q}_1^k(\mu_2 \mid \theta'), \mu_1 - \mu_2 \rangle \leq -\frac{\lambda_\mu}{2}\|\mu_1 - \mu_2\|_2^2 \tag{20a}$$

and

$$\overline{Q}_2^k(\beta_1 \mid \theta') - \overline{Q}_2^k(\beta_2 \mid \theta') - \langle \nabla_\beta \overline{Q}_2^k(\beta_2 \mid \theta'), \beta_1 - \beta_2 \rangle \leq -\frac{\lambda_\beta}{2}\|\beta_1 - \beta_2\|_2^2 \tag{20b}$$

for all $(\mu_1, \beta_1), (\mu_2, \beta_2) \in \Omega$.

Second, we impose *first-order stability* conditions on the gradients of each component of $\overline{Q}^k$:

- For each $\mu \in \Omega_\mu, \theta' \in \Omega$, we have

$$\|\nabla_\mu \overline{Q}_1^k(\mu \mid \mu', \beta') - \nabla_\mu \overline{Q}_1^k(\mu \mid \mu^*, \beta')\|_2 \leq L_{\mu,1}\|\mu' - \mu^*\|_2 \tag{21a}$$
$$\|\nabla_\mu \overline{Q}_1^k(\mu \mid \mu', \beta') - \nabla_\mu \overline{Q}_1^k(\mu \mid \mu', \beta^*)\|_2 \leq L_{\mu,2}\|\beta' - \beta^*\|_2, \tag{21b}$$

  We refer to this condition as L_μ-FOS for short.
- Secondly, for all $\beta \in \Omega_\beta, \theta' \in \Omega$, we require that

$$\|\nabla_\beta \overline{Q}_2^k(\beta \mid \mu', \beta') - \nabla_\beta \overline{Q}_2^k(\beta \mid \mu^*, \beta')\|_2 \leq L_{\beta,1}\|\mu' - \mu^*\|_2 \tag{22a}$$
$$\|\nabla_\beta \overline{Q}_2^k(\beta \mid \mu', \beta') - \nabla_\beta \overline{Q}_2^k(\beta \mid \mu', \beta^*)\|_2 \leq L_{\beta,2}\|\beta' - \beta^*\|_2. \tag{22b}$$

We refer to this condition as L_β-FOS for short. The experienced reader may find that the (L_μ, L_β)-FOS conditions look intriguingly similar to the Lipschitz gradient conditions often encountered when proving geometric convergence for gradient descent methods. On a high level, smoothness requires function values of one function to be close for any pair of arguments that are close. Although our conditions seem to invoke Lipschitz gradients as well, it is actually of a completely different nature. The important difference arises from the existence of two parameters, as we now clarify.

As opposed to gradient descent, the EM updates optimize over the first parameter θ of a function $\overline{Q}^k(\cdot \mid \theta')$ defined by the second parameter θ' at every time step. If we could access $\overline{Q}^k(\cdot \mid \theta^*)$, EM would converge in one step to the true optimum. Therefore, if we can guarantee that $\overline{Q}^k(\cdot \mid \theta')$ and $\overline{Q}^k(\cdot \mid \theta^*)$ are close in some sense, there should be good reasons to hope that under some more regularity assumptions the maximizers are close as well, i.e. that $\overline{M}^k(\theta')$ is close to θ^*.

The (L_μ, L_β)-FOS conditions are precisely encouraging closeness of these two functions in a first-order sense. In particular, we require the gradients (with respect to the first argument θ) to be Lipschitz in the second argument θ'. Typical smoothness however is a property with respect to a *fixed* function (i.e. a fixed θ' in our case) and thus requires gradients to be Lipschitz in the first argument. Loosely speaking it upper bounds the curvature of said function, and thus is more like a second-order condition by nature. This distinction also explains why (L_μ, L_β)-FOS conditions require to be uniformly satisfied only over the first argument, while one of the second arguments can be fixed at μ^* or β^* respectively. Finally, as we show in Section 4, these conditions hold for concrete models.

Convergence guarantee for $\overline{Q}^k$-updates: We are now equipped to state our main convergence guarantee for the updates. It involves the quantities

$$L := \max\{L_{\mu_1}, L_{\mu_2}\} + \max\{L_{\beta_1}, L_{\beta_2}\}, \quad \lambda := \min\{\lambda_\mu, \lambda_\beta\} \quad \text{and} \quad \kappa := \frac{L}{\lambda}, \tag{23}$$

with κ generally required to be smaller than one, as well as the additive norm $\|\cdot\|_\star$ from equation (19).

Part (a) of the theorem controls the *approximation error* induced by using the k-truncated function $\overline{Q}^k$ as opposed to the exact population function $\overline{Q}$, whereas part (b) guarantees a *geometric rate of convergence* in terms of κ defined above in equation (23).

Theorem 1 *(a)* Approximation guarantee: *Under the mixing condition* (4)*, density boundedness condition* (16)*, and $(\lambda_\mu, \lambda_\beta)$-strong concavity condition* (20)*, there is a universal constant c_0 such that*

$$\|\overline{M}^k(\theta) - \overline{M}(\theta)\|_\star^2 \leq \underbrace{\frac{Cs^4}{\lambda\, \epsilon_{\text{mix}}^9 \overline{\pi}_{\min}^2}\big(1 - \epsilon_{\text{mix}}\overline{\pi}_{\min}\big)^k}_{=:\varphi^2(k)} \qquad \text{for all } \theta \in \Omega, \tag{24}$$

where s is the number of states, and $\overline{\pi}_{\min} := \min_{\beta \in \Omega_\beta} \min_{j \in [s]} \overline{\pi}(j; \beta)$.

(b) Convergence guarantee: *Suppose in addition that the (L_μ, L_β)-FOS conditions* (21),(22) *holds with parameter $\kappa \in (0, 1)$ as defined in* (23) *for $\theta, \theta' \in \Omega = \mathbb{B}_2\big(r; \mu^*\big) \times \Omega_\beta$, and*

that the truncation parameter k is sufficiently large to ensure that

$$\varphi(k) \leq \big(1-\kappa\big) r - \kappa \max_{\beta \in \Omega_\beta} \|\beta - \beta^*\|_2.$$

Then, given an initialization $\widetilde{\theta}^0 \in \Omega$, the iterates $\{\widetilde{\theta}^t\}_{t=0}^{\infty}$ generated by the $\overline{M}^k$ operator satisfy the bound

$$\|\widetilde{\theta}^t - \theta^*\|_\star \leq \kappa^t \|\widetilde{\theta}^0 - \theta^*\|_\star + \frac{1}{1-\kappa}\varphi(k). \tag{25}$$

Note that the subtlety here is that θ^* is no longer a fixed point of the operator $\overline{M}^k$, due to the error induced by the kth-order truncation. Nonetheless, under the mixing condition, as the bounds (24) and (25) show, this approximation error is controlled, and decays exponentially in k. The proof of the recursive bound (25) is based on showing that

$$\|\overline{M}^k(\theta) - \overline{M}^k(\theta^*)\|_\star \leq \kappa \|\theta - \theta^*\|_\star \tag{26}$$

for any $\theta \in \Omega$. Inequality (26) is equivalent to stating that the operator $\overline{M}^k$ is contractive, i.e. that applying $\overline{M}^k$ to the pair θ and θ^* always decreases the distance.

Finally, when Theorem 1 is applied to a concrete model, the task is to find a big r and Ω_β such that the conditions in the theorem are satisfied, and we do so for the Gaussian output HMM in Section 4.

3.3 Sample-based results

We now turn to a result that applies to the sample-based form of the Baum-Welch algorithm—that is, corresponding to the updates that are actually applied in practice. For a tolerance parameter $\delta \in (0,1)$, we let $\varphi_n(\delta, k)$ be the smallest positive scalar such that

$$\mathbb{P}\Big[\sup_{\theta \in \Omega} \|M_n(\theta) - M_n^k(\theta)\|_\star \geq \varphi_n(\delta, k)\Big] \leq \delta. \tag{27a}$$

This quantity bounds the approximation error induced by the k-truncation, and is the sample-based analogue of the quantity $\varphi(k)$ appearing in Theorem 1(a). For each $\delta \in (0,1)$, we let $\epsilon_n^\mu(\delta, k)$ and $\epsilon_n^\beta(\delta, k)$ denote the smallest positive scalars such that

$$\begin{aligned} &\mathbb{P}\big[\sup_{\theta \in \Omega} \|M_n^{\mu,k}(\theta) - \overline{M}^{\mu,k}(\theta)\|_2 \geq \epsilon_n^\mu(\delta, k)\Big] \leq \delta, \quad \text{and} \\ &\mathbb{P}\big[\sup_{\theta \in \Omega} \|M_n^{\beta,k}(\theta) - \overline{M}^{\beta,k}(\theta)\|_2 \geq \epsilon_n^\beta(\delta, k)\Big] \leq \delta, \end{aligned} \tag{27b}$$

where $M_n^{\mu,k}(\cdot)$ and $M_n^{\beta,k}(\cdot)$ correspond to the truncated versions of $M_n^\mu(\cdot)$ and $M_n^\beta(\cdot)$. Furthermore we define $\epsilon_n(\delta, k) := \epsilon_n^\mu(\delta, k) + \epsilon_n^\beta(\delta, k)$. For a given truncation level k, these values give an upper bound on the difference between the population and sample-based M-operators, as induced by having only a finite number n of samples.

Theorem 2 (Sample Baum-Welch) *Suppose that the truncated population EM operator $\overline{M}^k$ satisfies the local contraction bound (26) with parameter $\kappa \in (0,1)$ in Ω. For a given sample size n, suppose that (k, n) are sufficiently large to ensure that*

$$\varphi_n(\delta, k) + \varphi(k) + \epsilon_n^\mu\big(\delta, k\big) \leq (1-\kappa)\, r - \kappa \max_{\beta \in \Omega_\beta} \|\beta - \beta^*\|_2. \tag{28a}$$

Then given any initialization $\widehat{\theta}^0 \in \Omega$, with probability at least $1 - 2\delta$, the Baum-Welch sequence $\{\widehat{\theta}^t\}_{t=0}^{\infty}$ satisfies the bound

$$\|\widehat{\theta}^t - \theta^*\|_\star \leq \underbrace{\kappa^t \|\widehat{\theta}^0 - \theta^*\|_\star}_{\text{Geometric decay}} + \underbrace{\frac{1}{1-\kappa}\Big\{\varphi_n\big(\delta, k\big) + \varphi(k) + \epsilon_n\big(\delta, k\big)\Big\}}_{\text{Residual error } e_n}. \tag{28b}$$

The bound (28b) shows that the distance between $\widehat{\theta}^t$ and θ^* is bounded by two terms: the first decays geometrically as t increases, and the second term corresponds to a residual error term that remains independent of t. Thus, by choosing the iteration number T larger than $\frac{\log(2r/\epsilon)}{\log \kappa}$, we can ensure that the first term is at most ϵ. The residual error term can be controlled by requiring that the sample size n is sufficiently large, and then choosing the truncation level k appropriately. We provide a concrete illustration of this procedure in the following section, where we analyze the case of Gaussian output HMMs. In particular, we can see that the residual error is of the same order as for the MLE and that the required initialization radius is optimal up to constants. Let us emphasize here that k as well as the truncated operators are purely theoretical objects which were introduced for the analysis.

4. Concrete results for the Gaussian output HMM

We now return to the concrete example of a Gaussian output HMM, as first introduced in Section 2.1, and specialize our general theory to it. Before doing so, let us make some preliminary comments about our notation and assumptions. Recall that our Gaussian output HMM is based on $s = 2$ hidden states, using the transition matrix from equation (7), and the Gaussian output densities from equation (8). For convenience of analysis, we let the hidden variables Z_i take values in $\{-1, 1\}$. In addition, we require that the mixing coefficient $\rho_{\text{mix}} = 1 - \epsilon_{\text{mix}}$ is bounded away from 1 in order to ensure that the mixing condition (3) is fulfilled. We denote the upper bound for ρ_{mix} as $b < 1$ so that $\rho_{\text{mix}} \leq b$ and $\epsilon_{\text{mix}} \geq 1 - b$. The feasible set of the probability parameter ζ and its log odds analog $\beta = \frac{1}{2}\log\big(\frac{\zeta}{1-\zeta}\big)$ are then given by

$$\Omega_\zeta = \Big\{\zeta \in \mathbb{R} \mid \frac{1-b}{2} \leq \zeta \leq \frac{1+b}{2}\Big\}, \quad \text{and} \quad \Omega_\beta = \Big\{\beta \in \mathbb{R} \;\mid\; |\beta| < \underbrace{\frac{1}{2}\log\big(\frac{1+b}{1-b}\big)}_{\beta_B}\Big\}. \tag{29}$$

4.1 Explicit form of Baum-Welch updates

We begin by deriving an explicit form of the Baum-Welch updates for this model. Using this notation, the Baum-Welch updates take the form

$$\widehat{\mu}^{t+1} = \frac{1}{n}\sum_{i=1}^{n}(2p(Z_i = 1 \mid x_1^n; \widehat{\theta}^t) - 1)x_i, \tag{30a}$$

$$\widehat{\zeta}^{t+1} = \Pi_{\Omega_\zeta}\left(\frac{1}{n}\sum_{i=1}^{n}\sum_{Z_i} p(Z_i = Z_{i+1} \mid x_1^n; \widehat{\theta}^t)\right), \text{ and} \tag{30b}$$

$$\widehat{\beta}^{t+1} = \frac{1}{2}\log\left(\frac{\widehat{\zeta}^{t+1}}{1-\widehat{\zeta}^{t+1}}\right), \tag{30c}$$

where Π_{Ω_ζ} denotes the Euclidean projection onto the set Ω_ζ. Note that the maximization steps are carried out on the decomposed Q-functions $Q_{1,n}(\cdot \mid \theta^t), Q_{2,n}(\cdot \mid \theta^t)$. In addition, since we are dealing with a one-dimensional quantity β, the projection of the unconstrained maximizer onto the interval Ω_ζ is equivalent to the constrained maximizer over the feasible set Ω_ζ. This step is in general not valid for higher dimensional transition parameters.

4.2 Population and sample guarantees

We now use the results from Section 3 to show that the population and sample-based version of the Baum-Welch updates are linearly convergent in a ball around θ^* of fixed radius. In establishing the population-level guarantee, the key conditions which need to be fulfilled—and the one that are the most technically challenging to establish— are the (L_μ, L_β)-FOS conditions (21), (22). In particular, we want to show that these conditions hold with Lipschitz constants L_μ, L_β that decrease exponentially with the separation of the mixtures. As a consequence, we obtain that for large enough separation $\frac{L}{\lambda} < 1$, i.e. the EM operator is contractive towards the true parameter.

In order to ease notation, our explicit tracking of parameter dependence is limited to the standard deviation σ and Euclidean norm $\|\mu^*\|_2$, which together determine the signal-to-noise ratio $\eta^2 := \frac{\|\mu^*\|_2^2}{\sigma^2}$ of the mixture model. Throughout this section, we therefore use c_0, c_1 to denote universal constants and C_0, C_1 for quantities that do not depend on $(\|\mu^*\|_2, \sigma)$, but may depend on other parameters such as $\overline{\pi}_{\min}$, ρ_{mix}, b, and so on.

We begin by stating a result for the sequence $\{\widetilde{\theta}^t\}_{t=0}^{\infty}$ obtained by repeatedly applying the k-truncated population-level Baum-Welch update operator $\overline{M}^k$. Our first corollary establishes that this sequence is linearly convergent, with a convergence rate $\kappa = \kappa(\eta)$ that is given by

$$\kappa(\eta) := \frac{C_1\eta^2(\eta^2+1)\,\mathrm{e}^{-c_2\eta^2}}{1-b^2}. \tag{31}$$

Corollary 1 (Population Baum-Welch) *Consider a two-state Gaussian output HMM that is mixing (i.e. satisfies equation* (3)*), and with its SNR lower bounded as $\eta^2 \geq C$ for a sufficiently large constant C. Given the radius $r = \frac{\|\mu^*\|_2}{4}$, suppose that the truncation parameter k is sufficiently large to ensure that $\varphi(k) \leq (1-\kappa)r - \kappa\max_{\beta\in\Omega_\beta}\|\beta-\beta^*\|_2$. Then for any initialization $\widetilde{\theta}^0 = (\widetilde{\mu}^0, \widetilde{\beta}^0) \in \mathbb{B}_2\big(r; \mu^*\big) \times \Omega_\beta$, the sequence $\{\widetilde{\theta}^t\}_{t=0}^{\infty}$ generated by $\overline{M}^k$ satisfies the bound*

$$\|\widetilde{\theta}^t - \theta^*\|_\star \leq \kappa^t\|\widetilde{\theta}^0 - \theta^*\|_\star + \frac{1}{1-\kappa}\varphi(k) \tag{32}$$

for all iterations $t = 1, 2, \ldots$.

From definition (31) it follows that as long as the signal-to-noise ratio η is larger than a universal constant, the convergence rate $\kappa(\eta) < 1$. The bound (32) then ensures a type of contraction and the pre-condition $\varphi(k) \leq (1-\kappa)r - \kappa\max_{\beta\in\Omega_\beta}\|\beta-\beta^*\|_2$ can be satisfied by choosing the truncation parameter k large enough. If we use a finite truncation parameter k, then the contraction occurs up to the error floor given by $\varphi(k)$, which reflects the bias introduced by truncating the likelihood to a window of size k. At the population level (in which the effective sample size is infinite), we could take the limit $k \to \infty$ so as to eliminate this bias. However, this is no longer possible in the finite sample setting, in which we must necessarily have $k \ll n$. While large k give a better truncation approximation, it allows for fewer samples which are "sufficiently independent" from each other within the sequence. We can see in the proof of Corollary 2 that $k \gtrsim \log n$ is a good choice to obtain an adequate trade-off.

Corollary 2 (Sample Baum-Welch iterates) *For a given tolerance $\delta \in (0,1)$, suppose that the sample size is lower bounded as $n \geq \frac{C_1}{\|\mu^*\|_2^2\sigma^2}(\eta^2+1)^3 d\log^8(\frac{d}{\delta})$. Then under the conditions of Corollary 1 and $\eta^2 \geq C\log\frac{1}{1-b^2}$, with probability at least $1-\delta$, we have*

$$\|\widetilde{\theta}^t - \theta^*\|_\star \leq \kappa^t\|\widehat{\theta}^0 - \theta^*\|_\star + \frac{C}{\sigma}\frac{\left(\frac{\|\mu^*\|_2^2}{\sigma^2}+1\right)^{3/2}\sqrt{\frac{d\log^8(n/\delta)}{n}}}{1-\kappa}. \tag{33}$$

Remarks: As a consequence of the bound (33), if we are given a sample size $n \gtrsim d\log^8 d$, then taking $T \approx \log n$ iterations is guaranteed to return an estimate $(\widehat{\mu}^T, \widehat{\beta}^T)$ with error of the order $\sqrt{\frac{d\log^8(n)}{n}}$.

In order to interpret this guarantee, note that in the case of symmetric Gaussian output HMMs as in Section 4, standard techniques can be used to show that the minimax rate of estimating μ^* in Euclidean norm scales as $\sqrt{\frac{d}{n}}$. If we could compute the MLE in polynomial time, then its error would also exhibit this scaling. The significance of Corollary 2 is that it shows that the Baum-Welch update achieves this minimax risk of estimation up to logarithmic factors.

Moreover, it should be noted that the initialization radius given here is essentially optimal up to constants. Because of the symmetric nature of the population log-likelihood, the all zeroes vector is a stationary point. Consequently, the maximum Euclidean radius of any basin of attraction for one of the observation parameters—that is, either μ^* or $-\mu^*$—can at most be $r = \|\mu^*\|_2$. Note that our initialization radius only differs from this maximal radius by a small constant factor.

4.3 Simulations

In this section, we provide the results of simulations that confirm the accuracy of our theoretical predictions for two-state Gaussian output HMMs. In all cases, we update the estimates for the mean vector $\widehat{\mu}^{t+1}$ and transition probability $\hat{\zeta}^{t+1}$ according to equation (30); for convenience, we update ζ as opposed to β. The true parameters are denoted by μ^* and ζ^*.

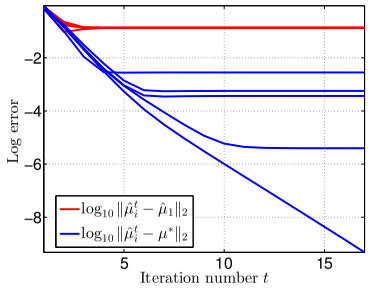


Figure 3: Plot of the convergence of the optimization error $\log\|\widehat{\mu}_i^t - \widehat{\mu}_1\|_2$, plotted in blue, and the statistical error $\log\|\widehat{\mu}_i^t - \mu^*\|_2$, plotted in red, for 5 different initializations. The parameter settings were $d = 10$, $n = 1000$, $\rho_{\mathrm{mix}} = 0.6$ and SNR $\frac{\|\mu^*\|_2}{\sigma} = 1.5$. See the main text for further details.

In all simulations, we fix the mixing parameter to $\rho_{\mathrm{mix}} = 0.6$, generate initial vectors $\widehat{\mu}^0$ randomly in a ball of radius $r := \frac{\|\mu^*\|_2}{4}$ around the true parameter μ^*, and set $\widehat{\zeta}^0 = \frac{1}{2}$. Finally, the estimation error of the mean vector μ is computed as $\log_{10}\|\hat{\mu} - \mu^*\|_2$. Since the transition parameter estimation errors behave similarly to the observation parameter in simulations, we omit the corresponding figures here.

Figure 3 depicts the convergence behavior of the Baum-Welch updates, as assessed in terms of both the optimization and the statistical error. Here we run the Baum-Welch algorithm for a fixed sample sequence X_1^n drawn from a model with SNR $\eta^2 = 1.5$ and $\zeta = 0.2$, using different random initializations in the ball around μ^* with radius $\frac{\|\mu^*\|_2}{4}$. We denote the final estimate of the $i-$th trial by $\widehat{\mu}_i$. The curves in blue depict the *optimization error*—that is, the differences between the Baum-Welch iterates $\widehat{\mu}_i^t$ using the i-th initialization, and $\widehat{\mu}_1$. On the other hand, the red lines represent the *statistical error*—that is, the distance of the iterates from the true parameter μ^*.

For both family of curves, we observe linear convergence in the first few iterations until an error floor is reached. The convergence of the statistical error aligns with the theoretical prediction in upper bound (33) of Corollary 2. The (minimax-optimal) error floor in the curve corresponds to the residual error and the e_n–region in Figure 1. In addition, the blue optimization error curves show that for different initializations, the Baum-Welch algorithm converges to *different stationary points* $\widehat{\mu}_i$; however, all of these points have roughly the same distance from μ^*. This phenomenon highlights the importance of the change of perspective

in our analysis—that is, focusing on the true parameter as opposed to the MLE. Given the presence of all these local optima in a small neighborhood of μ^*, the basin of attraction of the MLE must necessarily be much smaller than the initialization radius guaranteed by our theory.

Figure 4 shows how the convergence rate of the Baum-Welch algorithm depends on the underlying SNR parameter η^2; this behavior confirms the predictions given in Corollary 2. Lines of the same color represent different random draws of parameters given a fix SNR. Clearly, the convergence is linear for high SNR, and the rate decreases with decreasing SNR.

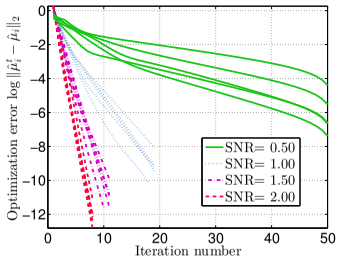


Figure 4: Plot of convergence behavior for different SNR, where for each curve, different parameters were chosen. The parameter settings are $d = 10$, $n = 1000$ and $\rho_{\mathrm{mix}} = 0.6$.

5. Proofs

In this section, we collect the proofs of our main results. In all cases, we provide the main bodies of the proofs here, deferring the more technical details to the appendices.

5.1 Proof of Theorem 1

Throughout this proof, we make use of the shorthand $\widetilde{\rho}_{\mathrm{mix}} = 1 - \epsilon_{\mathrm{mix}}\overline{\pi}_{\min}$. Also we denote the separate components of the population EM operators by $\overline{M}(\theta) =: (\overline{M}^{\mu}(\theta), \overline{M}^{\beta}(\theta))^T$ and their truncated equivalents by $\overline{M}^k(\theta) =: (\overline{M}^{\mu,k}(\theta), \overline{M}^{\beta,k}(\theta))^T$. We begin by proving the bound given in part (a). Since $\overline{Q} = \lim_{n\to\infty}\mathbb{E}[Q_n]$, we have

$$\|\overline{Q} - \overline{Q}^k\|_\infty = \|\lim_{n\to\infty}\mathbb{E}[Q_n] - \overline{Q}^k\|_\infty \leq \frac{Cs^4}{\epsilon_{\mathrm{mix}}^9\overline{\pi}_{\min}^2}\widetilde{\rho}_{\mathrm{mix}}^k,$$

where we have exchanged the supremum and the limit before applying the bound (17). The same holds for the separate functions $\overline{Q}_1, \overline{Q}_2$.

Using this bound and the fact that for $\overline{Q}_1$ we have $\overline{Q}_1(\overline{M}^{\mu}(\theta) \mid \theta) \geq \overline{Q}_1(\overline{M}^{\mu,k}(\theta) \mid \theta)$, we find that

$$\overline{Q}_1(\overline{M}^{\mu}(\theta) \mid \theta) \geq \overline{Q}_1^k(\overline{M}^{\mu,k}(\theta) \mid \theta) - \frac{Cs^4}{\epsilon_{\mathrm{mix}}^9\overline{\pi}_{\min}^2}\widetilde{\rho}_{\mathrm{mix}}^k.$$

Since $\overline{M}^{\mu,k}(\theta)$ is optimal, the first-order conditions for optimality imply that

$$\langle \bar{Q}_1^k(\overline{M}^{\mu,k}(\theta) \mid \theta),\, \theta - \overline{M}^{\mu,k}(\theta)\rangle \leq 0 \qquad \text{for all } \theta \in \Omega.$$

Combining this fact with strong concavity of $\bar{Q}^k(\cdot|\theta)$ for all θ, we obtain

$$\begin{aligned}
\frac{Cs^4}{\epsilon_{\min}^9 \bar{\pi}_{\min}^2}\widetilde{\rho}_{\min}^k &\geq \bar{Q}_1^k(\overline{M}^{\mu,k}(\theta) \mid \theta) - \bar{Q}_1(\overline{M}^{\mu}(\theta) \mid \theta) \\
&\geq \bar{Q}_1^k(\overline{M}^{\mu,k}(\theta) \mid \theta) - \bar{Q}_1^k(\overline{M}^{\mu}(\theta) \mid \theta) - \frac{Cs^4}{\epsilon_{\min}^9 \bar{\pi}_{\min}^2}\widetilde{\rho}_{\min}^k \\
&\geq \frac{\lambda_\mu}{2}\|\overline{M}^{\mu}(\theta) - \overline{M}^{\mu,k}(\theta)\|_2^2 - \frac{Cs^4}{\epsilon_{\min}^9 \bar{\pi}_{\min}^2}\widetilde{\rho}_{\min}^k
\end{aligned}$$

and therefore $\|\overline{M}^{\mu}(\theta) - \overline{M}^{\mu,k}(\theta)\|_2^2 \leq 4\frac{Cs^4}{\lambda\epsilon_{\min}^9 \bar{\pi}_{\min}^2}\widetilde{\rho}_{\min}^k$. In particular, setting $\theta = \theta^*$ and identifiability, i.e. $\overline{M}^{\mu}(\theta^*) = \theta^*$, yields

$$\|\overline{M}^{\mu,k}(\theta^*) - \theta^*\|_2^2 \leq 4\frac{Cs^4}{\lambda_\mu \epsilon_{\min}^9 \bar{\pi}_{\min}^2}\widetilde{\rho}_{\min}^k,$$

and the equivalent bound can be obtained for $\overline{M}^{\beta,k}(\cdot)$ which yields the claim.

We now turn to the proof of part (b). Let us suppose that the recursive bound (26) holds, and use it to complete the proof of this claim. We first show that if $\widetilde{\mu}^t \in \mathbb{B}_2(r;\mu^*)$, then we must have $\widetilde{\mu}^{t+1} \in \mathbb{B}_2(r;\mu^*)$ as well. Indeed, if $\widetilde{\mu}^t \in \mathbb{B}_2(r;\mu^*)$, then we have by triangle inequality and contraction in (26)

$$\begin{aligned}
\|\overline{M}^{\mu,k}(\widetilde{\theta}^t) - \mu^*\|_2 &\leq \|\overline{M}^{\mu,k}(\widetilde{\theta}^t) - \overline{M}^{\mu,k}(\theta^*)\|_2 + \|\overline{M}^{\mu,k}(\theta^*) - \mu^*\|_2 \\
&\leq \kappa[\|\widetilde{\mu}^t - \mu^*\|_2 + \|\widetilde{\beta}^t - \beta^*\|_2] + \varphi(k) \\
&\leq \kappa(r + \max_{\beta\in\Omega_\beta}\|\beta - \beta^*\|_2) + \varphi(k) \leq r,
\end{aligned}$$

where the final step uses the assumed bound on φ. For the joint parameter update we in turn have

$$\begin{aligned}
\|\overline{M}^{k}(\widetilde{\theta}^t) - \theta^*\|_\star &\leq \|\overline{M}^{k}(\widetilde{\theta}^t) - \overline{M}^{k}(\theta^*)\|_\star + \|\overline{M}^{k}(\theta^*) - \theta^*\|_\star \\
&\leq \kappa\|\widetilde{\theta}^t - \theta^*\|_\star + \varphi(k). \qquad (34)
\end{aligned}$$

By repeatedly applying inequality (34) and summing the geometric series, the claimed bound (25) follows.

It remains to prove the bound (26). Since the vector $\overline{M}^k(\theta^*)$ maximizes the function $\theta \mapsto \bar{Q}_1^k(\theta \mid \theta^*)$, we have the first-order optimality condition

$$\langle \nabla\bar{Q}_1^k(\overline{M}^{\mu,k}(\theta^*) \mid \theta^*),\, \overline{M}^{\mu,k}(\theta) - \overline{M}^{\mu,k}(\theta^*)\rangle \leq 0, \qquad \text{valid for any } \theta.$$

Similarly, we have $\langle \nabla\bar{Q}_1^k(\overline{M}^{\mu,k}(\theta) \mid \theta),\, \overline{M}^{\mu,k}(\theta^*) - \overline{M}^{\mu,k}(\theta)\rangle \leq 0$, and adding together these two inequalities yields

$$0 \leq \langle \nabla\bar{Q}_1^k(\overline{M}^{\mu,k}(\theta^*) \mid \theta^*) - \nabla\bar{Q}_1^k(\overline{M}^{\mu,k}(\theta) \mid \theta),\, \overline{M}^{\mu,k}(\theta^*) - \overline{M}^{\mu,k}(\theta)\rangle$$

On the other hand, by the λ-strong concavity condition, we have

$$\lambda_\mu\|\overline{M}^{\mu,k}(\theta) - \overline{M}^{\mu,k}(\theta^*)\|_2^2 \leq \langle \nabla\bar{Q}_1^k(\overline{M}^{\mu,k}(\theta) \mid \theta^*) - \nabla\bar{Q}_1^k(\overline{M}^{\mu,k}(\theta^*) \mid \theta^*),\, \overline{M}^{\mu,k}(\theta^*) - \overline{M}^{\mu,k}(\theta)\rangle$$

Combining these two inequalities with the (L_μ, L_β)-FOS condition yields

$$\begin{aligned}
\lambda_\mu\|\overline{M}^{\mu,k}(\theta) - \overline{M}^{\mu,k}(\theta^*)\|_2^2 &\leq \langle \nabla\bar{Q}_1^k(\overline{M}^{\mu,k}(\theta) \mid \theta^*) - \nabla\bar{Q}_1^k(\overline{M}^{\mu,k}(\theta) \mid \theta),\, \overline{M}^{\mu,k}(\theta^*) - \overline{M}^{\mu,k}(\theta)\rangle \\
&\leq \big[L_{\mu_1}\|\mu - \mu^*\|_2 + L_{\mu_2}\|\beta - \beta^*\|_2\big]\|\overline{M}^{\mu,k}(\theta) - \overline{M}^{\mu,k}(\theta^*)\|_2,
\end{aligned}$$

and similarly we obtain $\lambda_\beta\|\overline{M}^{\beta,k}(\theta) - \overline{M}^{\beta,k}(\theta^*)\|_2 \leq \big[L_{\beta_1}\|\mu - \mu^*\|_2 + L_{\beta_2}\|\beta - \beta^*\|_2\big]$. Adding both inequalities yields the claim (26).

5.2 Proof of Theorem 2

By the triangle inequality and inequality (34), we have with probability at least $1 - 2\delta$ that for any iteration

$$\begin{aligned}
\|\widehat{\theta}^{t+1} - \theta^*\|_\star &\leq \|M_n(\widehat{\theta}^t) - M_n^k(\widehat{\theta}^t)\|_\star + \|M_n^k(\widehat{\theta}^t) - \overline{M}^k(\widehat{\theta}^t)\|_\star + \|\overline{M}^k(\widehat{\theta}^t) - \theta^*\|_\star \\
&\leq \varphi_n(\delta,k) + \epsilon_n(\delta,k) + \kappa\|\widehat{\theta}^t - \theta^*\|_\star + \varphi(k).
\end{aligned}$$

In order to see that the iterates do not leave $\mathbb{B}_2\big(r;\mu^*\big)$, observe that

$$\begin{aligned}
\|\widehat{\mu}^{t+1} - \mu^*\|_2 &\leq \|M_n^\mu(\widehat{\theta}^t) - M_n^{\mu,k}(\widehat{\theta}^t)\|_2 + \|M_n^{\mu,k}(\widehat{\theta}^t) - \overline{M}^{\mu,k}(\widehat{\theta}^t)\|_2 + \|\overline{M}^{\mu,k}(\widehat{\theta}^t) - \mu^*\|_2 \\
&\leq \varphi_n(\delta,k) + \epsilon_n^\mu(\delta,k) + \kappa(\|\widehat{\mu}^t - \mu^*\|_2 + \max_{\beta\in\Omega_\beta}\|\beta - \beta^*\|_2) + \varphi(k). \qquad (35)
\end{aligned}$$

Consequently, as long as $\|\widehat{\mu}^t - \mu^*\|_2 \leq r$, we also have $\|\widehat{\mu}^{t+1} - \mu^*\|_2 \leq r$ whenever

$$\varphi_n(\delta,k) + \varphi(k) + \epsilon_n^\mu(\delta,k) \leq (1-\kappa)\,r - \kappa\max_{\beta\in\Omega_\beta}\|\beta - \beta^*\|_2.$$

Combining inequality (35) with the equivalent bound for β, we obtain

$$\|\widehat{\theta}^t - \theta^*\|_\star \leq \kappa\|\widehat{\theta}^{t-1} - \theta^*\|_\star + \varphi_n(\delta,k) + \epsilon_n(\delta,k) + \varphi(k)$$

Summing the geometric series yields the bound (28b).

5.3 Proof of Corollary 1

The boundedness condition (Assumption (16)) is easy to check since for $X \sim N(\mu^*, \sigma^2)$, the quantity $\sup_{\mu\in\mathbb{B}_2(r;\mu^*)} \mathbb{E}\big[\max\{\|X - \mu\|_2, \|X + \mu\|_2\}\big]$ is finite for any choice of radius $r < \infty$.

By Theorem 1, the k-truncated population EM iterates satisfy the bound

$$\|\widetilde{\theta}^t - \theta^*\|_\star \leq \kappa^t\|\widetilde{\theta}^0 - \theta^*\|_\star + \frac{1}{1-\kappa}\varphi(k), \qquad (36)$$

if the strong concavity (20) and FOS conditions (21), (22) hold with suitable parameters.

In the remainder of proof—and the bulk of the technical work— we show that:

- strong concavity holds with $\lambda_\mu = 1$ and $\lambda_\beta \geq \frac{2}{3}(1 - b^2)$;

- the FOS conditions hold with

$$L_{\mu,1} = c\ (\eta^2+1)\varphi_2(\epsilon_{\mathrm{mix}})\eta^2 \mathrm{e}^{-c\eta^2}, \quad \text{and} \quad L_{\mu,2} = c\sqrt{\|\mu^*\|_2^2+\sigma^2}\varphi_2(\epsilon_{\mathrm{mix}})\eta^2\mathrm{e}^{-c\eta^2}$$
$$L_{\beta,1} = c\frac{1-b}{1+b}\varphi_2(\epsilon_{\mathrm{mix}})\eta^2\mathrm{e}^{-c\eta^2} \quad \text{and} \quad L_{\beta,2} = c\sqrt{\|\mu^*\|_2^2+\sigma^2}\varphi_2(\epsilon_{\mathrm{mix}})\eta^2\mathrm{e}^{-c\eta^2},$$

where $\varphi_2(\epsilon_{\mathrm{mix}}) := \left(\frac{1}{\log(1/(1-\epsilon_{\mathrm{mix}}))} + \frac{1}{\epsilon_{\mathrm{mix}}}\right)$. Substuting these choices into the bound (36) and performing some algebra yields the claim.

5.3.1 Establishing strong concavity

We first show concavity of $\overline{Q}_1^k(\cdot \mid \theta')$ and $\overline{Q}_2^k(\cdot \mid \theta')$ separately. For strong concavity of $\overline{Q}_1^k(\cdot \mid \theta')$, observe that

$$\overline{Q}_1^k(\mu \mid \theta') = -\frac{1}{2}\mathbb{E}\left[p(z_0 = 1 \mid X_{-k}^k; \theta')\|X_0 - \mu\|_2^2 + (1 - p(z_0 = 1|X_{-k}^k; \theta'))\|X_0 + \mu\|_2^2 + c\right],$$

where c is a quantity independent of μ. By inspection, this function is strongly concave in μ with parameter $\lambda_\mu = 1$.

On the other hand, we have

$$\overline{Q}_2^k(\beta \mid \theta') = \mathbb{E}_{X_{-k}^k|\theta^*} \sum_{z_0,z_1} p(z_0, z_1 \mid X_{-k}^k; \theta') \log\left(\frac{\mathrm{e}^{\beta z_0 z_1}}{\mathrm{e}^{\beta} + \mathrm{e}^{-\beta}}\right).$$

This function has second derivative $\frac{\partial^2}{\partial\beta^2}\overline{Q}_2^k(\beta \mid \theta') = -4\frac{\mathrm{e}^{-2\beta}}{(\mathrm{e}^{-2\beta}+1)^2}$. As a function of $\beta \in \Omega_\beta$, this second derivative is maximized at $\beta = \frac{1}{2}\log\left(\frac{1+b}{1-b}\right)$. Consequently, the function $\overline{Q}_2^k(\cdot \mid \theta')$ is strongly concave with parameter $\lambda_\beta \geq \frac{2}{3}(1-b^2)$.

5.3.2 Separate FOS conditions

We now turn to proving that the FOS conditions in equations (21) and (22) hold. A key ingredient in our proof is the fact that the conditional density $p(z_{-k}^k \mid x_{-k}^k; \mu, \beta)$ belongs to the exponential family with parameters $\beta \in \mathbb{R}$, and $\gamma_i := \frac{\langle \mu, x_i\rangle}{\sigma^2} \in \mathbb{R}$ for $i = -k, \ldots, k$ which define the vector $\gamma = (\gamma_{-k}, \ldots, \gamma_k)$ (see Wainwright and Jordan (2008) for more details on exponential families.) In particular, we have

$$\underbrace{p(z_{-k}^k \mid x_{-k}^k, \mu, \beta)}_{:=p(z_{-k}^k;\gamma,\beta)} = \exp\left\{\sum_{\ell=-k}^{k}\gamma_\ell z_\ell + \beta\sum_{\ell=-k}^{k-1} z_\ell z_{\ell+1} - \Phi(\gamma,\beta)\right\}, \tag{37}$$

where the function h absorbs various coupling terms. Note that this exponential family is a specific case of the following exponential family distribution

$$\underbrace{\tilde{p}(z_{-k}^k \mid x_{-k}^k, \mu, \beta)}_{:=\tilde{p}(z_{-k}^k;\gamma,\beta)} = \exp\left\{\sum_{\ell=-k}^{k}\gamma_\ell z_\ell + \sum_{\ell=-k}^{k-1}\beta_\ell z_\ell z_{\ell+1} - \Phi(\gamma,\beta)\right\}. \tag{38}$$

The distribution in (37) corresponds to (38) with $\beta_\ell = \beta$ for all ℓ and the so-called partition function Φ is given by

$$\Phi(\gamma,\beta) = \log\sum_z \exp\left\{\sum_{\ell=-k}^{k}\gamma_\ell z_\ell + \sum_{\ell=-k}^{k-1}\beta_\ell z_\ell z_{\ell+1}\right\}.$$

The reason to view our distribution as a special case of the more general one in (38) becomes clear when we consider the equivalence of expectations and the derivatives of the cumulant function

$$\left.\frac{\partial\Phi}{\partial\gamma_\ell}\right|_{\theta'} = \mathbb{E}_{Z_{-k}^k|x_{-k}^k,\theta'}Z_\ell \quad \text{and} \quad \left.\frac{\partial\Phi}{\partial\beta_0}\right|_{\theta'} = \mathbb{E}_{Z_{-k}^k|x_{-k}^k,\theta'}Z_0Z_1, \tag{39}$$

where we recall that $\mathbb{E}_{Z_{-k}^k|x_{-k}^k,\theta'}$ is the expectation with respect to the distribution $\tilde{p}(Z_{-k}^k \mid x_{-k}^k; \mu', \beta')$ with $\beta_\ell = \beta'$. Note that in the following any value θ' for $\tilde{p}$ is taken to be on the manifold on which $\beta_\ell = \beta'$ for some β' since this is the manifold the algorithm works on. Also, as before, $\mathbb{E}$ denotes the expectation over the joint distribution of all samples X_ℓ drawn according to $p(\cdot;\theta^*)$, in this case X_{-k}^k.

Similarly to equations (39), the covariances of the sufficient statistics correspond to the second derivatives of the cumulant function

$$\left.\frac{\partial^2\Phi}{\partial\beta_\ell\partial\beta_0}\right|_{\theta} = \mathrm{cov}(Z_0Z_1, Z_\ell Z_{\ell+1} \mid X_{-k}^k, \theta) \tag{40a}$$
$$\left.\frac{\partial^2\Phi}{\partial\gamma_\ell\partial\gamma_0}\right|_{\theta} = \mathrm{cov}(Z_0, Z_\ell \mid X_{-k}^k, \theta) \tag{40b}$$
$$\left.\frac{\partial^2\Phi}{\partial\beta_\ell\partial\gamma_0}\right|_{\theta} = \mathrm{cov}(Z_0, Z_\ell Z_{\ell+1} \mid X_{-k}^k, \theta). \tag{40c}$$

In the following, we adopt the shorthand

$$\begin{aligned}\mathrm{cov}(Z_\ell, Z_{\ell+1} \mid \gamma', \beta') &= \mathrm{cov}(Z_\ell, Z_{\ell+1} \mid X_{-k}^k, \theta') \\ &= \mathbb{E}_{Z_\ell^{\ell+1}|X_{-k}^k,\theta'}(Z_\ell - \mathbb{E}_{Z_\ell^{\ell+1}|X_{-k}^k,\theta'}Z_\ell)(Z_{\ell+1} - \mathbb{E}_{Z_\ell^{\ell+1}|X_{-k}^k,\theta'}Z_{\ell+1})\end{aligned}$$

where the dependence on β is occasionally omitted so as to simplify notation.

5.3.3 Proof of inequality (21a)

By an application of the mean value theorem, we have

$$\|\nabla_\mu\overline{Q}_1^k(\mu \mid \mu', \beta') - \nabla_\mu\overline{Q}_1^k(\mu \mid \mu^*, \beta')\| \leq \underbrace{\left\|\mathbb{E}\sum_{\ell=-k}^{k}\left.\frac{\partial^2\Phi}{\partial\gamma_\ell\partial\gamma_0}\right|_{\theta=\widetilde{\theta}}(\gamma'_\ell - \gamma^*_\ell)X_0\right\|}_{T_1}$$

where $\widetilde{\theta} = \theta' + t(\theta^* - \theta')$ for some $t \in (0,1)$. Since second derivatives yield covariances (see equation (40)), we can write

$$T_1 = \left\|\sum_{\ell=-k}^{k}\mathbb{E}X_0\mathbb{E}\left[\mathrm{cov}(Z_0, Z_\ell \mid \widetilde{\gamma})\frac{\langle\mu' - \mu^*, X_\ell\rangle}{\sigma^2}\middle| X_0\right]\right\|_2,$$

so that it suffices to control the expected conditional covariance. By the Cauchy-Schwarz inequality and the fact that $\operatorname{cov}(X,Y) \leq \sqrt{\operatorname{var} X}\sqrt{\operatorname{var} Y}$ and $\operatorname{var}(Z_0 \mid X) \leq 1$, we obtain the following bound on the expected conditional covariance by using Lemma 4 (see Appendix B)

$$\begin{aligned}\mathbb{E}\left[\left|\operatorname{cov}(Z_0, Z_\ell \mid \widetilde{\gamma}) \mid X_0\right|\right] &\leq \sqrt{\mathbb{E}\left[\operatorname{var}(Z_0 \mid \widetilde{\gamma}) \mid X_0\right]}\sqrt{\mathbb{E}\left[\operatorname{var}(Z_\ell \mid \widetilde{\gamma}) \mid X_0\right]} \\ &\leq \sqrt{\operatorname{var}(Z_0 \mid \widetilde{\gamma}_0)}. \end{aligned} \tag{41a}$$

Furthermore, by Lemma 5 and 6 (see Appendix B), we have

$$|\operatorname{cov}(Z_0, Z_\ell \mid \widetilde{\gamma})| \leq 2\rho_{\mathrm{mix}}^{\ell}, \quad \text{and} \quad \left\|\mathbb{E}(\operatorname{var}(Z_0|\widetilde{\gamma}_0))^{1/2} X_0 X_0^T\right\|_{op} \leq C\mathrm{e}^{-c\eta^2}. \tag{41b}$$

From the definition of the operator norm, we have

$$\begin{aligned}\|\mathbb{E}\operatorname{cov}(Z_0, Z_\ell \mid \widetilde{\gamma}) X_0 X_\ell^T\|_{op} &= \sup_{\substack{\|u\|_2=1 \\ \|v\|_2=1}} \mathbb{E}\operatorname{cov}(Z_0, Z_\ell \mid \widetilde{\gamma})\langle X_0,\, v\rangle\,\langle X_\ell,\, u\rangle \\ &\leq \sup_{\|v\|_2=1} \mathbb{E}|\operatorname{cov}(Z_0, Z_\ell \mid \widetilde{\gamma})|\langle X_0,\, v\rangle^2 \\ &+ \sup_{\|u\|_2=1} \mathbb{E}|\operatorname{cov}(Z_0, Z_\ell \mid \widetilde{\gamma})|\langle X_\ell,\, u\rangle^2 \\ &= \|\mathbb{E} X_0 X_0^T \mathbb{E}\big[|\operatorname{cov}(Z_0, Z_\ell \mid \widetilde{\gamma}) \mid X_0\big]\|_{op} \\ &+ \|\mathbb{E} X_\ell X_\ell^T \mathbb{E}\big[\operatorname{cov}(Z_0, Z_\ell \mid \widetilde{\gamma}) \mid X_\ell\big]\|_{op} \\ &\overset{(i)}{\leq} 2\min\{\rho_{\mathrm{mix}}^{|\ell|}\|\mathbb{E} X_0 X_0^T\|_{op}, \|\mathbb{E}\operatorname{var}(Z_0 \mid \widetilde{\gamma}_0)^{1/2} X_0 X_0^T\|_{op}\} \\ &\overset{(ii)}{\leq} 2\min\{(\|\mu^*\|_2^2 + \sigma^2)\rho_{\mathrm{mix}}^{|\ell|}, C'\mathrm{e}^{-c\eta^2}\}, \end{aligned} \tag{42}$$

where inequality (i) makes use of inequalities (41a) and (41b), and step (ii) makes use of the second inequality in line (41b).

By inequality (42), we find that

$$\begin{aligned} T_1 &\leq \frac{\|\mu' - \mu^*\|_2}{\sigma^2} \sum_{\ell=-k}^{k} \|\mathbb{E}\operatorname{cov}(Z_0, Z_\ell \mid \widetilde{\gamma}) X_0 X_\ell^T\|_{op} \\ &\leq 2\frac{\|\mu' - \mu^*\|_2}{\sigma^2} \sum_{\ell=-k}^{k} \min\{(\|\mu^*\|_2^2 + \sigma^2)\rho_{\mathrm{mix}}^{|\ell|}, C\mathrm{e}^{-c\eta^2}\} \\ &\leq 4(\eta^2+1)\big(mC\mathrm{e}^{-c\eta^2} + \frac{\rho_{\mathrm{mix}}^m}{1-\rho_{\mathrm{mix}}}\big)\|\mu' - \mu^*\|_2. \end{aligned}$$

where $m = \frac{c\eta^2}{\log(1/\rho_{\mathrm{mix}})}$ is the smallest integer such that $\rho_{\mathrm{mix}}^m \leq C\mathrm{e}^{-c\eta^2}$ The last inequality follows from the proof of Corollary 1 in the paper Balakrishnan et al. (2014) if $\eta^2 > C$ for some universal constant C. We have thus shown that

$$\|\nabla_\mu \overline{Q}_1^k(\mu \mid \mu', \beta') - \nabla_\mu \overline{Q}_1^k(\mu \mid \mu^*, \beta')\| \leq L_{\mu,1}\|\mu' - \mu^*\|_2,$$

where $L_{\mu,1} = c\ \varphi_1(\eta)\varphi_2(\epsilon_{\mathrm{mix}})\eta^2(\eta^2+1)\mathrm{e}^{-c\eta^2}$ as claimed.

5.3.4 Proof of inequality (21b)

The same argument via the mean value theorem guarantees that

$$\|\frac{\partial}{\partial\beta}\overline{Q}_2^k(\beta \mid \mu', \beta') - \frac{\partial}{\partial\beta}\overline{Q}_2^k(\beta \mid \mu', \beta^*)\| \leq \left\|\mathbb{E}\sum_{\ell=-k}^{k} \frac{\partial^2\Phi}{\partial\beta_\ell\partial\gamma_0}\Big|_{\theta=\widetilde{\theta}}(\beta' - \beta^*)X_0\right\|_2.$$

In order to bound this quantity, we again use the equivalence (40) and bound the expected conditional covariance. Furthermore, Lemma 5 and 6 yield

$$\operatorname{cov}(Z_0, Z_\ell Z_{\ell+1} \mid \widetilde{\gamma}) \overset{(i)}{\leq} 2\rho_{\mathrm{mix}}^{\ell} \quad \text{and} \quad \left\|\mathbb{E}\operatorname{var}(Z_0 \mid \widetilde{\gamma}_0) X_0 X_0^T\right\|_{op} \overset{(ii)}{\leq} c\mathrm{e}^{-c\eta^2}. \tag{43}$$

Here inequality (ii) follows by combining inequality (54c) from Lemma 5 with the fact that $\operatorname{var}(Z_0 \mid \widetilde{\gamma}_0) \leq 1$.

$$\begin{aligned}\|\mathbb{E} X_0 \operatorname{cov}(Z_0, Z_\ell Z_{\ell+1} \mid \widetilde{\gamma})\|_2 &= \sup_{\|u\|_2=1} \mathbb{E}\langle X_0, u\rangle \operatorname{cov}(Z_0, Z_\ell Z_{\ell+1} \mid \widetilde{\gamma}) \\ &\leq \sup_{\|u\|_2=1} \mathbb{E}|\langle X_0, u\rangle|\mathbb{E}\big[|\operatorname{cov}(Z_0, Z_\ell Z_{\ell+1} \mid \widetilde{\gamma})| \mid X_0\big] \\ &\overset{(iii)}{\leq} \sup_{\|u\|_2=1} \mathbb{E}|\langle X_0, u\rangle| \min\{\rho_{\mathrm{mix}}^{|\ell|}, (\operatorname{var}(Z_0 \mid \widetilde{\gamma}_0))^{1/2}\} \\ &\overset{(iv)}{\leq} \min\{\sup_{\|u\|_2=1} \sqrt{\mathbb{E}\langle X_0, u\rangle^2}\rho_{\mathrm{mix}}^{|\ell|}, \sup_{\|u\|_2=1} \sqrt{\mathbb{E}\langle X_0, u\rangle^2 \operatorname{var}(Z_0 \mid \widetilde{\gamma}_0))}\} \\ &\overset{(v)}{\leq} \min\{\rho_{\mathrm{mix}}^{|\ell|}\sqrt{\|\mathbb{E} X_0 X_0^T\|_{op}}, \sqrt{\|\mathbb{E}\operatorname{var}(Z_0 \mid \widetilde{\gamma}_0) X_0 X_0^T\|_{op}}\} \\ &\overset{(vi)}{\leq} \min\{\rho_{\mathrm{mix}}^{|\ell|}\sqrt{\|\mu^*\|_2^2 + \sigma^2}, C\mathrm{e}^{-c\eta^2}\} \end{aligned}$$

where step (iii) uses inequality (43); step (iv) follows from the Cauchy-Schwarz inequality; step (v) follows from the definition of the operator norm; and step (vi) uses inequality (43) again.

Putting together the pieces, we find that

$$\begin{aligned}\left\|\mathbb{E}\sum_{\ell=-k}^{k} \frac{\partial^2\Phi}{\partial\beta_\ell\partial\gamma_0} X_0\right\|_2 |\beta' - \beta^*| &\leq \sum_{\ell=-k}^{k} \|\mathbb{E} X_0 \mathbb{E}[\operatorname{cov}(Z_0, Z_\ell Z_{\ell+1} \mid \widetilde{\gamma}) \mid X_0]\|_2 |\beta' - \beta^*| \\ &\leq 4\sqrt{\|\mu^*\|_2^2 + \sigma^2}\left(c\,m\,\mathrm{e}^{-c\eta^2} + \frac{\rho_{\mathrm{mix}}^m}{1-\rho_{\mathrm{mix}}}\right)|\beta' - \beta^*|. \end{aligned}$$

again with $m = \frac{c\eta^2}{\log(1/\rho_{\mathrm{mix}})}$, we find that inequality (21b) holds with $L_{\mu,2} = c\varphi_2(\epsilon_{\mathrm{mix}})\sqrt{\|\mu^*\|_2 + \sigma^2}\eta^2\mathrm{e}^{-c\eta^2}$, as claimed.

5.3.5 Proof of inequality (22a)

By the same argument via the mean value theorem, we find that

$$\|\frac{\partial}{\partial\beta}\overline{Q}_2^k(\beta \mid \beta', \mu') - \frac{\partial}{\partial\beta}\overline{Q}_2^k(\beta \mid \beta', \mu^*)\| \leq \left|\mathbb{E}\sum_{\ell=-k}^{k} \frac{\partial^2\Phi}{\partial\gamma_\ell\partial\beta_0}\Big|_{\theta=\widetilde{\theta}} \frac{\langle\mu' - \mu^*, X_\ell\rangle}{\sigma^2}\right|.$$

Equation (40) guarantees that $\frac{\partial^2 \Phi}{\partial \gamma_\ell \partial \beta_0} = \operatorname{cov}(Z_0 Z_1, Z_\ell \mid \gamma)$. Therefore, by similar arguments as in the proof of inequalities (21), we have

$$\begin{aligned} T &:= \Big| \sum_{\ell=-k}^{k} \mathbb{E}\langle \mu' - \mu^*, X_\ell \rangle \mathbb{E}[\operatorname{cov}(Z_0 Z_1, Z_\ell \mid \widetilde{\gamma}_\ell, \beta') | X_\ell] \Big| \\ &\leq \Big| \sum_{\ell=-k}^{k} \mathbb{E}|\langle \mu' - \mu^*, X_\ell \rangle| \min\{\rho_{\mathrm{mix}}^{|\ell|}, (\operatorname{var}(Z_\ell \mid \widetilde{\gamma}_\ell, \beta'))^{1/2}\} \Big| \\ &\leq \Big| \sum_{\ell=-k}^{k} \min\big\{\rho_{\mathrm{mix}}^{|\ell|}, \sqrt{\mathbb{E}\operatorname{var}(Z_\ell \mid \widetilde{\gamma}_\ell, \beta')}\big\} \sqrt{\mathbb{E}\langle \mu' - \mu^*, X_\ell \rangle^2} \Big| \\ &\leq \sqrt{\|\mu^*\|_2^2 + \sigma^2} \big(mc\, \mathrm{e}^{-c\eta^2} + 2 \sum_{\ell=m+1}^{k} \rho_{\mathrm{mix}}^{\ell} \big). \end{aligned}$$

where we have used inequality (54b) from Lemma 6. Finally, again noting that $m = \frac{c\eta^2}{\log(1/\rho_{\mathrm{mix}})}$ yields that the FOS condition holds with $L_{\beta,2} = c\sqrt{\|\mu^*\|_2 + \sigma^2}\varphi_2(\epsilon_{\mathrm{mix}})\eta^2 \mathrm{e}^{-c\eta^2}$, as claimed.

5.3.6 Proof of inequality (22b)

By the same mean value argument, we find that

$$\|\frac{\partial}{\partial \beta} \overline{Q}_2^k(\beta \mid \beta', \mu') - \frac{\partial}{\partial \beta} \overline{Q}_2^k(\beta \mid \beta^*, \mu')\| \leq \Big| \mathbb{E} \sum_{\ell=-k}^{k} \frac{\partial^2 \Phi}{\partial \beta_\ell \partial \beta_0} \Big|_{\theta = \widetilde{\theta}} (\beta' - \beta^*) \Big|.$$

By the exponential family view in equality (40) it suffices to control the expected conditional covariance. Lemma 5 and 6 guarantee that

$$|\operatorname{cov}(Z_0 Z_1, Z_\ell Z_{\ell+1} \mid X_{-k}^k, \widetilde{\gamma})| \leq \rho_{\mathrm{mix}}^{|\ell|}, \quad \text{and} \quad \mathbb{E}\operatorname{var}(Z_0 Z_1 \mid \widetilde{\gamma}_0^1, \widetilde{\beta}) \leq c\, \frac{1+b}{1-b}\, \mathrm{e}^{-c\eta^2}. \tag{44}$$

Furthermore, the Cauchy-Schwarz inequality combined with the bound (53a) from Lemma 4 yields

$$\begin{aligned} \mathbb{E}\big|\operatorname{cov}(Z_0 Z_1, Z_\ell Z_{\ell+1} \mid \widetilde{\gamma})\big| &\leq \sqrt{\mathbb{E}\operatorname{var}(Z_0 Z_1 \mid \widetilde{\gamma}, \widetilde{\beta})} \sqrt{\mathbb{E}\operatorname{var}(Z_\ell Z_{\ell+1} \mid \widetilde{\gamma}, \widetilde{\beta})} \\ &\leq \sqrt{\mathbb{E}\operatorname{var}(Z_0 Z_1 \mid \widetilde{\gamma}_0^1, \widetilde{\beta})} \sqrt{\mathbb{E}\operatorname{var}(Z_\ell Z_{\ell+1} \mid \widetilde{\gamma}_\ell^{\ell+1}, \widetilde{\beta})} \\ &\leq \mathbb{E}\operatorname{var}(Z_0 Z_1 \mid \widetilde{\gamma}_0^1, \widetilde{\beta}). \end{aligned} \tag{45}$$

Combining the bounds (44) and (45) yields

$$\begin{aligned} \Big| \sum_{\ell=-k}^{k} \mathbb{E} \frac{\partial^2 \Phi}{\partial \beta_\ell \partial \beta_0} (\beta' - \beta^*) \Big| &\leq \sum_{\ell=-k}^{k} \big| \mathbb{E}\operatorname{cov}(Z_0 Z_1, Z_\ell Z_{\ell+1} \mid \widetilde{\gamma}_{-k}^k, \widetilde{\beta}) \big|\, |\beta' - \beta^*| \\ &\leq \sum_{\ell=-k}^{k} \min\Big\{ \rho_{\mathrm{mix}}^{|\ell|}, \mathbb{E}\operatorname{var}(Z_0 Z_1 \mid \widetilde{\gamma}_0^1, \widetilde{\beta}) \Big\} |\beta' - \beta^*| \\ &\leq 2 \left(c \frac{1+b}{1-b}\, m \mathrm{e}^{-c\eta^2} + \sum_{l=m+1}^{k} \rho_{\mathrm{mix}}^{\ell} \right) |\beta' - \beta^*| \\ &\leq 2c \frac{1+b}{1-b} \varphi_2(\epsilon_{\mathrm{mix}}) \eta^2 \mathrm{e}^{-c\eta^2} |\beta' - \beta^*| \end{aligned}$$

where the final inequality follows by setting $m = \frac{c\eta^2}{\log(1/\rho_{\mathrm{mix}})}$. Therefore, the FOS condition holds with $L_{\beta,1} = c\frac{1-b}{1+b}\varphi_2(\epsilon_{\mathrm{mix}})\eta^2 \mathrm{e}^{-c\eta^2}$, as claimed.

5.4 Proof of Corollary 2

In order to prove this corollary, it is again convenient to separate the updates on the mean vectors μ from those applied to the transition parameter β. Recall the definitions of φ, φ_n and ϵ_n from equations (24) and (27a) respectively, as well as $\widetilde{\rho}_{\mathrm{mix}} = 1 - \epsilon_{\mathrm{mix}} \overline{\pi}_{\min}$.

Using Theorem 2 we readily have that given any initialization $\widehat{\theta}^0 \in \Omega$, with probability at least $1 - 2\delta$, we are guaranteed that

$$\|\widehat{\theta}^T - \theta^*\|_\star \leq \kappa^T \|\widehat{\theta}^0 - \theta^*\|_\star + \frac{\varphi_n(\delta, k) + \epsilon_n(\delta, k) + \varphi(k)}{1 - \kappa}. \tag{46}$$

In order to leverage the bound (46), we need to find appropriate upper bounds on the quantities $\varphi_n(\delta, k)$, $\epsilon_n(\delta, k)$.

Lemma 1 *Suppose that the truncation level satisifes the lower bound*

$$k \geq \log\left(\frac{C_\epsilon n}{\delta}\right) \left(\log \frac{1}{\widetilde{\rho}_{\mathrm{mix}}}\right)^{-1} \qquad \text{where } C_\epsilon := \frac{C}{\epsilon_{\mathrm{mix}}^3 \overline{\pi}_{\min}^3}. \tag{47a}$$

Then, when the number of observations n satifies the lower bound in the assumptions of the corollary and the radius is chosen to be $r = \frac{\|\mu^*\|_2}{4}$, *we have*

$$\epsilon_n^{\mu}(\delta, k) \leq C_0 \frac{1}{\sigma} \big(\frac{\|\mu^*\|_2^2}{\sigma^2} + 1 \big)^{3/2} \log(k^2/\delta) \sqrt{\frac{k^3 d \log n}{n}}, \quad \text{and} \tag{47b}$$

$$\epsilon_n^{\beta}(\delta, k) \leq C_0\, \frac{1}{\sigma} \sqrt{\frac{\|\mu^*\|_2^2}{\sigma^2} + 1} \sqrt{\frac{k^3 \log(k^2/\delta)}{n}}. \tag{47c}$$

Lemma 2 *Suppose that* $\frac{1}{2}\frac{\log\left(C_\epsilon n/\delta\right)}{\log(1/\widetilde{\rho}_{\mathrm{mix}})} \leq k \leq C\frac{\log\left(C_\epsilon n/\delta\right)}{\log(1/\widetilde{\rho}_{\mathrm{mix}})}$ *with* $C > 1$. *Then by choosing* $r = \frac{\|\mu^*\|_2}{4}$ *and* C_1 *large enough, we have*

$$\varphi_n(\delta, k) \leq C_1 \left\{ \sqrt{\frac{d \log^2(C_\epsilon n/\delta)}{\sigma n}} + \sqrt{\frac{\|\mu^*\|_2}{\sigma} \frac{\log^2(C_\epsilon n/\delta)}{n}} + \frac{\|\mu^*\|_2}{\sigma} \sqrt{\frac{\delta}{n}} \right\}. \tag{48}$$

See Appendices C.1 and C.2, respectively, for the proofs of these two lemmas.

Note that the set for which k simultaneously satisfies the conditions in Lemma 1 and 2 is nonempty. Furtheremore, the choice of k is made purely for analysis purposes – it does not have any consequence on the Using these two lemmas, we can now complete the proof of the corollary. From the definition (31) of κ, under the stated lower bound on η^2, we can ensure that $\kappa \leq \frac{1}{2}$. Under this condition, inequality (28a) with $r = \|\mu^*\|_2/4$ reduces to showing that

$$\varphi_n(\delta, k) + \epsilon_n^{\mu}(\delta, k) + \varphi(k) \leq \frac{\|\mu^*\|_2}{8}. \tag{49}$$

Now any choice of k satisfying both conditions in Lemmas 1 and 2 guarantees that

$$\varphi_n(\delta, k) + \epsilon_n^{\mu}(\delta, k) + \epsilon_n^{\beta}(\delta, k) + \varphi(k) \leq \frac{C}{\sigma}\Big(\frac{\|\mu^*\|_2^2}{\sigma^2} + 1\Big)^{3/2}\sqrt{\frac{d\log^8(n/\delta)}{n}}. \tag{50}$$

Furthermore, as long as $n \geq \frac{C_1}{\|\mu^*\|_2^2\sigma^2}(\eta^2+1)^3 d\log^8(d/\delta)$ for a sufficiently large C_1, we are guaranteed that the bound (49) holds. Substituting the bound (50) into inequality (46) completes the proof of the corollary.

6. Discussion

In this paper, we provided general global convergence guarantees for the Baum-Welch algorithm as well as specific results for a hidden Markov mixture of two isotropic Gaussians. In contrast to the classical perspective of focusing on the MLE, we focused on bounding the distance between the Baum-Welch iterates and the true parameter. Under suitable regularity conditions, our theory guarantees that the iterates converge to an e_n-ball of the true parameter, where e_n represents a form of statistical error. It is important to note that our theory does not guarantee convergence to the MLE itself, but rather to a ball that contains the true parameter, and asymptotically the MLE as well. When applied to the Gaussian mixture HMM, we proved that the Baum-Welch algorithm achieves estimation error that is minimax optimal up to logarithmic factors. To the best of our knowledge, these are the first rigorous guarantees for the Baum-Welch algorithm that allow for a large initialization radius.

Acknowledgments

This work was partially supported by NSF grant CIF-31712-23800, ONR-MURI grant DOD-002888, and AFOSR grant FA9550-14-1-0016 to MJW.

Appendix A. Proof of Proposition 1

In order to show that the limit $\lim_{n\to\infty}\mathbb{E}Q_n(\theta \mid \theta')$ exists, it suffices to show that the sequence of functions $\{\mathbb{E}Q_1, \mathbb{E}Q_2, \ldots, \mathbb{E}Q_n\}$ is Cauchy in the sup-norm (as defined previously in equation (15)). In particular, it suffices to show that for every $\epsilon > 0$ there is a positive integer $N(\epsilon)$ such that for $m, n \geq N(\epsilon)$,

$$\|\mathbb{E}Q_m - \mathbb{E}Q_n\|_\infty \leq \epsilon.$$

In order to do so, we make use of the previously stated bound (17) relating $\mathbb{E}Q_n$ to $\bar{Q}^k$. Taking this bound as given for the moment, an application of the triangle inequality yields

$$\|\mathbb{E}Q_m - \mathbb{E}Q_n\|_\infty \leq \|\mathbb{E}Q_m - \bar{Q}^k\|_\infty + \|\mathbb{E}Q_n - \bar{Q}^k\|_\infty \;\leq\; \epsilon,$$

the final inequality follows as long as we choose $N(\epsilon)$ and k large enough (roughly proportional to $\log(1/\epsilon)$).

It remains to prove the claim (17). In order to do so, we require an auxiliary lemma:

Lemma 3 (Approximation by truncation) *For a Markov chain satisfying the mixing condition* (3)*, we have*

$$\sup_{\theta'\in\Omega}\sup_{x}\sum_{z_i}|p(z_i \mid x_1^n;\theta') - p(z_i \mid x_{i-k}^{i+k};\theta')| \leq \frac{Cs^2}{\epsilon_{\text{mix}}^8\overline{\pi}_{\min}}\big(1-\epsilon_{\text{mix}}\overline{\pi}_{\min}\big)^{\min\{i,n-i,k\}} \tag{51}$$

for all $i \in [0, n]$*, where* $\overline{\pi}_{\min} = \min_{j\in[s],\beta\in\Omega_\beta}\overline{\pi}(j;\beta)$.

See Appendix D.2 for the proof of this lemma.

Using Lemma 3, let us now prove the claim (17). Introducing the shorthand notation

$$h(X_i, z_i, \theta, \theta') := \log p(X_i \mid z_i;\theta) + \sum_{z_{i-1}} p(z_i \mid z_{i-1};\theta')\log p(z_i|z_{i-1},\theta),$$

we can verify by applying Lemma 3 that

$$\begin{aligned}
&\|\mathbb{E}Q_n - \bar{Q}^k\|_\infty \qquad\qquad (52)\\
&= \Big|\sup_{\theta,\theta'}\frac{1}{n}\sum_{i=1}^n\sum_{z_i}\mathbb{E}(p(z_i \mid X_1^n,\theta') - p(z_i \mid X_{i-k}^{i+k},\theta'))h(X_i,z_i,\theta,\theta')\Big|\\
&+ \Big|\frac{1}{n}\sup_{\theta,\theta'}\mathbb{E}\sum_{z_0}p(z_0 \mid X_1^n,\theta')\log p(z_0;\theta)\Big|\\
&\leq \sup_{\theta,\theta'}\frac{1}{n}\sum_{i=1}^n\sum_{z_i}\sup_x\big|p(z_i \mid x_1^n,\theta') - p(z_i \mid x_{i-k}^{i+k},\theta')\big|\,\mathbb{E}\,|h(X_i,z_i,\theta,\theta')| + \frac{1}{n}\log\overline{\pi}_{\min}^{-1}\\
&\leq \frac{Cs^3}{\epsilon_{\text{mix}}^8\overline{\pi}_{\min}n}\Big(2\sum_{i=1}^k(1-\epsilon_{\text{mix}}\overline{\pi}_{\min})^i + (n-2k)(1-\epsilon_{\text{mix}}\overline{\pi}_{\min})^k\Big)\Big[\max_{z_i\in[s]}\mathbb{E}|h(X_i,z_i,\theta,\theta')|\Big] + \frac{1}{n}\log\overline{\pi}_{\min}^{-1}\\
&\leq \frac{Cs^3}{\epsilon_{\text{mix}}^8\overline{\pi}_{\min}}\Big(\frac{2}{n\epsilon_{\text{mix}}\overline{\pi}_{\min}} + \frac{n-2k}{n}(1-\epsilon_{\text{mix}}\overline{\pi}_{\min})^k\Big)\Big[\max_{z_i\in[s]}\mathbb{E}|h(X_i,z_i,\theta,\theta')|\Big] + \frac{1}{n}\log\overline{\pi}_{\min}^{-1}\\
&\leq \frac{Cs^4}{\epsilon_{\text{mix}}^9\overline{\pi}_{\min}^2}\big(1-\epsilon_{\text{mix}}\overline{\pi}_{\min}\big)^k + \frac{1}{n}\left(\log\overline{\pi}_{\min}^{-1} + \frac{Cs^4}{\epsilon_{\text{mix}}^{10}\overline{\pi}_{\min}^3}\right),
\end{aligned}$$

using the crude bound

$$\max_{z_i \in [s]} \mathbb{E}|h(X_i, z_i, \theta, \theta')| \leq \mathbb{E} \max_{z_i \in [s]} \big| \log p(X_i \mid z_i, \theta) \big| + s \log(\overline{\pi}_{\min} \epsilon_{\mathrm{mix}})^{-1} \leq \frac{Cs}{\overline{\pi}_{\min} \epsilon_{\mathrm{mix}}}.$$

which uses condition (16) and where C denotes generic constants which are potentially different each time they appear.

Appendix B. Technical details for Corollary 1

In this section, we collect some auxiliary bounds on conditional covariances in hidden Markov models. These results are used in the proof of Corollary 1.

Lemma 4 *For any HMM with observed-hidden states (X_i, Z_i), we have*

$$\mathbb{E}\Big[\operatorname{var}(Z_0 Z_1 \mid X_{-k}^k)\Big] \leq \mathbb{E} \operatorname{var}(Z_0 Z_1 \mid X_0^1) \tag{53a}$$

$$\mathbb{E}\Big[\operatorname{var}(Z_0 \mid X_{-k}^k) \mid X_0\Big] \leq \operatorname{var}(Z_0 \mid X_0) \tag{53b}$$

where we have omitted the dependence on the parameters.

Proof We use the law of total variance, which guarantees that $\operatorname{var} Z = \mathbb{E}\big[\operatorname{var}(Z \mid X)\big] + \operatorname{var}\mathbb{E}[Z \mid X]$. Using this decomposition, we have

$$\mathbb{E}[\operatorname{var}(Z_0 \mid X_0^1) \mid X_0] \leq \operatorname{var}(Z_0 \mid X_0)$$
$$\mathbb{E}[\operatorname{var}(Z_0 Z_1 \mid X_0^2) \mid X_0^1] \leq \operatorname{var}(Z_0 Z_1 \mid X_0^1).$$

The result then follows by induction. ∎

We now show that the expected conditional variance of the hidden state (or pairs thereof) conditioned on the corresponding observation (pairs of observations) decays exponentially with the SNR.

Lemma 5 *For a 2-state Markov chain with true parameter θ^*, we have for $\mu \in \mathbb{B}_2\big(\frac{\|\mu^*\|_2}{4}; \mu^*\big)$ and $\beta \in \Omega_\beta$*

$$\Big\| \mathbb{E} X_0 X_0^T (\operatorname{var}(Z_0 \mid \gamma_0, \beta))^{1/2} \Big\|_{op} \leq c_0 \, \mathrm{e}^{-c\eta^2} \tag{54a}$$

$$\mathbb{E} \operatorname{var}(Z_\ell \mid \gamma_\ell, \beta) \leq c_0 \, \mathrm{e}^{-c\eta^2} \tag{54b}$$

$$\mathbb{E} \operatorname{var}(Z_0 Z_1 \mid \gamma_0^1, \beta) \leq c_0 \frac{1+b}{1-b} \, \mathrm{e}^{-c\eta^2}. \tag{54c}$$

Proof By definition of the Gaussian HMM example, we have $\operatorname{var}(Z_i \mid \gamma_i) = \frac{4}{(\mathrm{e}^{\gamma_i} + \mathrm{e}^{-\gamma_i})^2}$. Moreover, following the proof of Corollary 1 in the paper Balakrishnan et al. (2014), we are guaranteed that $\mathbb{E} \operatorname{var}(Z_i \mid \gamma_i) \leq 8\mathrm{e}^{-\frac{\eta^2}{32}}$ and $\|\mathbb{E} X_i X_i^T (\operatorname{var}(Z_i|\gamma_i))^{1/2}\|_{op} \leq c_0 \mathrm{e}^{-\frac{\eta^2}{32}}$, from which inequalities (54a) and (54b) follow.

We now prove inequality (54c) for $\beta \in \Omega_\beta$ and $\mu \in \mathbb{B}_2\big(\frac{\|\mu^*\|_2}{4}; \mu^*\big)$. Note that

$$\begin{aligned}
\frac{1}{4} \operatorname{var}(Z_0 Z_1 \mid \gamma_0^1, \beta) &= \frac{\mathrm{e}^{2\gamma_1} + \mathrm{e}^{-2\gamma_1} + \mathrm{e}^{2\gamma_0} + \mathrm{e}^{-2\gamma_0}}{\big[\mathrm{e}^{\beta}(\mathrm{e}^{\gamma_0+\gamma_1} + \mathrm{e}^{-(\gamma_0+\gamma_1)}) + \mathrm{e}^{-\beta}(\mathrm{e}^{\gamma_0-\gamma_1} + \mathrm{e}^{-(\gamma_0-\gamma_1)})\big]^2} \\
&\leq \mathrm{e}^{2|\beta|} \frac{\mathrm{e}^{2\gamma_1} + \mathrm{e}^{-2\gamma_1} + \mathrm{e}^{2\gamma_0} + \mathrm{e}^{-2\gamma_0}}{(\mathrm{e}^{\gamma_0+\gamma_1} + \mathrm{e}^{-(\gamma_0+\gamma_1)} + \mathrm{e}^{\gamma_0-\gamma_1} + \mathrm{e}^{-(\gamma_0-\gamma_1)})^2} \\
&\leq \left(\frac{1+b}{1-b}\right) \left[\frac{\mathrm{e}^{|\gamma_0|}}{\mathrm{e}^{2\gamma_0} + \mathrm{e}^{-2\gamma_0}} + \frac{\mathrm{e}^{|\gamma_1|}}{\mathrm{e}^{2\gamma_1} + \mathrm{e}^{-2\gamma_1}}\right]
\end{aligned}$$

where γ are now random variables and we used

$$\begin{aligned}
&(\mathrm{e}^{\gamma_0+\gamma_1} + \mathrm{e}^{-(\gamma_0+\gamma_1)} + \mathrm{e}^{\gamma_0-\gamma_1} + \mathrm{e}^{-(\gamma_0-\gamma_1)})^2 \\
&\geq \mathrm{e}^{-|\gamma_0|}(\mathrm{e}^{-\gamma_0} + \mathrm{e}^{\gamma_0})(\mathrm{e}^{2\gamma_1} + \mathrm{e}^{-2\gamma_1}) + \mathrm{e}^{-|\gamma_1|}(\mathrm{e}^{-\gamma_1} + \mathrm{e}^{\gamma_1})(\mathrm{e}^{2\gamma_0} + \mathrm{e}^{-2\gamma_0}) \\
&\geq (\mathrm{e}^{-|\gamma_0|} + \mathrm{e}^{-|\gamma_1|})(\mathrm{e}^{2\gamma_0} + \mathrm{e}^{-2\gamma_0})(\mathrm{e}^{2\gamma_1} + \mathrm{e}^{-2\gamma_1}).
\end{aligned}$$

It directly follows that

$$\begin{aligned}
\frac{1}{4} \mathbb{E} \operatorname{var}(Z_0 Z_1 \mid \gamma_0^1, \beta) &\leq 2 \left(\frac{1+b}{1-b}\right) \mathbb{E}\left[\frac{1}{\mathrm{e}^{\gamma_0} + \mathrm{e}^{-3\gamma_0}} \mathbf{1}_{\gamma_0 \geq 0} + \frac{1}{\mathrm{e}^{3\gamma_0} + \mathrm{e}^{-\gamma_0}} \mathbf{1}_{\gamma_0 \leq 0}\right] \\
&\leq 2 \left(\frac{1+b}{1-b}\right) \left(\mathbb{E}[\mathrm{e}^{-\gamma_0} \mathbf{1}_{\gamma_0 \geq 0}] + \mathbb{E}[\mathrm{e}^{\gamma_0} \mathbf{1}_{\gamma_0 \leq 0}]\right) \\
&\leq 4 \left(\frac{1+b}{1-b}\right) \mathbb{E}[\mathrm{e}^{-\gamma_0} \mathbf{1}_{\gamma_0 \geq 0}]
\end{aligned}$$

where the last inequality follows from symmetry of the random variables X_i. One can then bound

$$\mathbb{E}[\mathrm{e}^{-\gamma_0} \mathbf{1}_{\gamma_0 \geq 0}] = \mathbb{E}\, \mathrm{e}^{-\frac{\|\mu\|_2 V_1}{\sigma^2}} \mathbf{1}_{V_1 \geq 0} \leq 2\mathrm{e}^{-\frac{\eta^2}{32}}$$

by employing a similar procedure as in the proof of Corollary 1 in Balakrishnan et al. (2014). Inequality (54c) then follows. ∎

The last lemma provides rigorous confirmation of the intuition that the covariance between any pair of hidden states should decay exponentially in their separation ℓ:

Lemma 6 *For a 2-state Markov chain with mixing coefficient ϵ_{mix} and uniform stationary distribution, we have*

$$\max\Big\{ \operatorname{cov}(Z_0, Z_\ell \mid \gamma),\ \operatorname{cov}(Z_0 Z_1, Z_\ell Z_{\ell+1} \mid \gamma),\ \operatorname{cov}(Z_0, Z_\ell Z_{\ell+1} \mid \gamma) \Big\} \leq 2\rho_{\mathrm{mix}}^\ell \tag{55}$$

with $\rho_{\mathrm{mix}} = 1 - \epsilon_{\mathrm{mix}}$ for all $\theta \in \Omega$.

Lemma 6 is a mixing result and its proof is found in Section D.3.

Appendix C. Technical details for Corollary 2

In this section we prove Lemmas 1 and 2. In order to do so, we leverage the independent blocks approach used in the analysis of dependent data (see, for instance, the papers Yu (1994); Nobel and Dembo (1993)). For future reference, we state here an auxiliary lemma that plays an important role in both proofs.

Let $\{X_i\}_{i=-\infty}^{\infty}$ be a sequence sampled from a Markov chain with mixing rate $\rho_{\mathrm{mix}} = 1 - \epsilon_{\mathrm{mix}}$, $\overline{\pi}_{\min}$ be the minimum entry of the stationary distribution and $\widetilde{\rho}_{\mathrm{mix}} = 1 - \epsilon_{\mathrm{mix}}\overline{\pi}_{\min}$. Given some functions $f_1 : \mathbb{R}^{2k} \to \mathbb{R}^d$ and $f_2 : \mathbb{R} \to \mathbb{R}^d$ in some function class $\mathcal{F}_1, \mathcal{F}_2$ respectively, our goal is to control the difference between the functions

$$g_1(X) := \frac{1}{n}\sum_{i=1}^{n} f_1(X_{i-k}^{i+k}), \qquad g_2(X) := \frac{1}{n}\sum_{i=1}^{n} f_2(X_i) \tag{56a}$$

and their expectation. Defining $m_1 := \lfloor n/4k \rfloor$ and $m_2 := \lfloor n/k \rfloor$, we say that f_1 respectively f_2 is (δ, k)-concentrated if

$$\mathbb{P}\Big[\sup_{f\in\mathcal{F}_1} \|\frac{1}{m_1}\sum_{i=1}^{m_1} f_1(\widetilde{X}_{i;2k}) - \mathbb{E}f_1(\widetilde{X}_{1;2k})\|_2 \geq \epsilon\Big] \leq \frac{\delta}{8k}, \tag{56b}$$
$$\mathbb{P}\Big[\sup_{f\in\mathcal{F}_2} \|\frac{1}{m_2}\sum_{i=1}^{m_2} f_2(\widetilde{X}_i) - \mathbb{E}f_2(\widetilde{X}_1)\|_2 \geq \epsilon\Big] \leq \frac{\delta}{2k}$$

where $\{\widetilde{X}_{i;2k}\}_{i\in\mathbb{N}}$ are a collection of i.i.d. sequences of length $2k$ drawn from the same Markov chain and $\{\widetilde{X}_i\}_{i\in\mathbb{N}}$ a collection of i.i.d. variables drawn from the same stationary distribution. In our notation, $\{\widetilde{X}_{i;2k}\}_{i\in\mathbb{N}}$ under $\mathbb{P}$ are identically distributed as $\{X_{i;2k}\}_{i\in\mathbb{N}}$ under $\mathbb{P}_0$.

Lemma 7 *Consider functions f_1, f_2 that are (δ, k)-concentrated* (56b) *for a truncation parameter $k \geq \log\left(\frac{Cn}{\overline{\pi}_{\min}^3 \epsilon_{\mathrm{mix}}^3 \delta}\right)(\log\frac{1}{\widetilde{\rho}_{\mathrm{mix}}})^{-1}$. Then the averaged functions g_1, g_2 from equation* (56a) *satisfy the bounds*

$$\mathbb{P}\Big[\sup_{g\in\mathcal{F}_1} \|g_1(X) - \mathbb{E}g_1(X)\|_2 \geq \epsilon\Big] \leq \delta \quad \text{and} \quad \mathbb{P}\Big[\sup_{g\in\mathcal{F}_2} \|g_2(X) - \mathbb{E}g_2(X)\|_2 \geq \epsilon\Big] \leq \delta. \tag{57}$$

Proof We prove the lemma for functions of the type (f_1, g_1); the proof for the case (f_2, g_2) is very similar. In order to simplify notation, we assume throughout the proof that the effective sample size n is a multiple of $4k$, so that the block size $m = \frac{n}{4k}$ is integral. By definition (56a), the function g is a function of the sequences $\{X_{1-k}^{1+k}, X_{2-k}^{2+k}, \ldots, X_{n-k}^{n+k}\}$. We begin by dividing these sequences into blocks. Let us define the subsets of indices

$$H_i^j = \{4k(i-1) + k + j \mid 4k(i-1) + 3k + j\}, \quad \text{and}$$
$$R_i^j = \{4k(i-1) - k + j \mid 4k(i-1) + k - 1 + j\}.$$

With this notation, we have the decomposition

$$g(X) = \frac{1}{2}\left(\frac{1}{2k}\sum_{j=1}^{2k}\underbrace{\frac{1}{m}\sum_{i=1}^{m} f(X_{H_i^j})}_{g^{H^j}(X)} + \frac{1}{2k}\sum_{j=1}^{2k}\underbrace{\frac{1}{m}\sum_{i=1}^{m} f(X_{R_i^j})}_{g^{R^j}(X)}\right),$$

from which we find that

$$\begin{aligned}\mathbb{P}\big[\sup_{g\in\mathcal{F}} \|g(X) - \mathbb{E}g(X)\|_2 \leq \epsilon\big] &\geq \mathbb{P}\big(\bigcap_{j=1}^{2k}\{\sup_{g\in\mathcal{F}} \|g^{H^j}(X) - \mathbb{E}g(X)\|_2 \leq \epsilon\} \\ &\qquad \cap \{\sup_{g\in\mathcal{F}} \|g^{R^j}(X) - \mathbb{E}g(X)\|_2 \leq \epsilon\}\big) \\ &\overset{\text{(i)}}{\geq} 1 - 4k\,\mathbb{P}(\sup_{g\in\mathcal{F}} \|g^{H^1}(X) - \mathbb{E}g(X)\|_2 \geq \epsilon),\end{aligned}$$

where (i) follows using stationarity of the underlying sequence combined with the union bound.

In order to bound the probability $\mathbb{P}\big[\|g^{H^1}(X) - \mathbb{E}g(X)\|_2 \geq \epsilon\big]$, it is convenient to relate it to the probability of the same event under the product measure $\mathbb{P}_0$ on the blocks $\{H_1^1, \ldots, H_m^1\}$. In particular, we have $\mathbb{P}(\|g^{H^1}(X) - \mathbb{E}g(X)\|_2 \geq \epsilon) \leq T_1 + T_2$, where

$$\begin{aligned}T_1 &:= \mathbb{P}_0(\|g^{H^1}(X) - \mathbb{E}g(X)\|_2 \geq \epsilon), \quad \text{and} \\ T_2 &:= |\mathbb{P}(\|g^{H^1}(X) - \mathbb{E}g(X)\|_2 \geq \epsilon) - \mathbb{P}_0(\|g^{H^1}(X) - \mathbb{E}g(X)\|_2 \geq \epsilon)|.\end{aligned}$$

By our assumed concentration (56b), we have $T_1 \leq \frac{\delta}{8k}$, and so it remains to show that $T_2 \leq \frac{\delta}{8k}$.

Now following standard arguments (e.g., see the papers Nobel and Dembo (1993); Yu (1994)), we first define

$$\beta(k) = \sup_{A\in\sigma(\mathcal{S}_{-\infty}^0, \mathcal{S}_k^\infty)} |\mathbb{P}(A) - \mathbb{P}_{-\infty}^0 \times \mathbb{P}_1^\infty(A)|, \tag{58}$$

where $\mathcal{S}_{-\infty}^0$ and $\mathcal{S}_k^\infty$ are the σ-algebras generated by the random vector $X_{-\infty}^0$ and X_k^∞ respectively, and $\mathbb{P}_{-\infty}^0 \times \mathbb{P}_1^\infty$ is the product measure under which the sequences $X_{-\infty}^0$ and X_1^∞ are independent. Define $\mathcal{S}_i$ to be the σ-algebra generated by $X_{H_i^j}$ for $i = \{1, \ldots, m\}$; it then follows by induction that $\sup_{A\in\sigma(\mathcal{S}_1,\ldots,\mathcal{S}_m)} |\mathbb{P}(A) - \mathbb{P}_0(A)| \leq m\beta(k)$. An identical relation holds over the blocks R_i^j.

For our two-state HMM, Lemma 12 implies that

$$\begin{aligned}\beta(k) = |p(x) - p(x_k^\infty)p(x_{-\infty}^0)| &\leq |p(x_{-\infty}^0 \mid x_k^n) - p(x_{-\infty}^0)| \\ &\leq |p(z_0 \mid x_k^n) - p(z_0)| \\ &\overset{\text{(i)}}{\leq} \frac{3}{\overline{\pi}_{\min}^3 \epsilon_{\mathrm{mix}}^3}\rho_{\mathrm{mix}}^k = \frac{3}{\overline{\pi}_{\min}^3 \epsilon_{\mathrm{mix}}^3}\mathrm{e}^{-k\log(1/\rho_{\mathrm{mix}})},\end{aligned} \tag{59}$$

where step (i) follows from inequality (73). From our assumed lower bound on k, we conclude that $m\beta(k) \leq \frac{\delta}{8k}$, which completes the proof. ■

In the following sections we apply it in order to prove the bounds on the approximation and sample error of the M-operators.

C.1 Proof of Lemma 1

We prove each of the two inequalities in equations (47b) and (47c) in turn by using suitable choices of the function f in Lemma 7. Throughout, note that our function class is parameterized and $f \in \mathcal{F}$ is equivalent to $\theta \in \Omega = \mathbb{B}_2(r; \mu^*) \times \Omega_\beta$.

Proof of inequality (47b)**:** We use the notation from the proof of Lemma 7 and furthermore define the weights $w_\theta(X_{i-k}^{i+k-1}) = p(Z_i = 1 \mid X_{i-k}^{i+k-1}, \theta)$, as well as the function $f_0(X_{i-k}^{i+k-1}, \theta') = (2w_{\theta'}(X_{i-k}^{i+k-1}) - 1)X_i$. It is then possible to write the EM operator explicitly as the average

$$M_n^{\mu,k}(\theta') = \arg\max_{\mu \in \widetilde{\Omega}} \frac{1}{n}\Big[\sum_{i=1}^n \mathbb{E}_{Z_i | X_{i-k}^{i+k}, \theta'} \log p(X_i \mid Z_i, \mu)\Big] = \frac{1}{n}\sum_{i=1}^n f_0(X_{i-k}^{i+k-1}, \theta').$$

We are now ready to apply Lemma 7 with the choices $f_1 = f_0$, $g_1(X) = M_n^{\mu,k}(\theta)$. According to Lemma 7, given that the lower bound on the truncation parameter k holds, we now need to show that f_0 is (δ, k)-concentrated, that means finding ϵ_n^μ such that

$$\mathbb{P}_0\left[\sup_{\theta\in\Omega} \big\|\frac{1}{m}\sum_{i=1}^m f_0(\widetilde{X}_{i;2k}, \theta') - \mathbb{E} f_0(\widetilde{X}_{i;2k}, \theta')\big\|_2 \geq \epsilon_n^\mu\right] \leq \frac{\delta}{8k},$$

where $\mathbb{P}_0$ denotes the product measure over the independent blocks and $m := m_1 = \lfloor n/4k \rfloor$.

Let X_i be the middle element of the (i.i.d. drawn) sequence $\widetilde{X}_{i;2k}$ and Z_i, V_i the corresponding latent and noise variable. We can then write $X_i = Z_i + V_i$ where V_i are zero-mean Gaussian random variables with covariance matrix $\sigma^2 I$.

With a minor abuse of notation, let us use $X_{i,\ell}$ to denote ℓ^{th} element in the block $\widetilde{X}_{i;2k} = (X_{i,1}, \ldots, X_{i,2k})^T$, and write $\widetilde{X} = \{\widetilde{X}_{i;2k}\}_{i=1}^n$. In view of Lemma 7, our objective is to find the smallest scalar ϵ_n^μ such that

$$\mathbb{P}\big[\sup_{\theta\in\Omega} \|\frac{1}{m}\sum_{i=1}^m \underbrace{(2w_\theta(\widetilde{X}_{i;2k}) - 1)X_{i,k} - \mathbb{E}(2w_\theta(\widetilde{X}_{i;2k}) - 1)X_{i,k}}_{f_\theta(\widetilde{X}_{i;2k})} \|_2 \geq \epsilon_n^\mu\big] \leq \frac{\delta}{8k} \qquad (60)$$

For each unit norm vector $u \in \mathbb{R}^d$, define the random variable

$$\tilde{V}_m(\widetilde{X}; u) = \sup_{\theta\in\Omega} \frac{1}{m}\sum_{i=1}^m (2w_\theta(\widetilde{X}_{i;2k}) - 1)\langle X_{i,k}, u\rangle - \mathbb{E}(2w_\theta(\widetilde{X}_{i;2k}) - 1)\langle X_{i,k}, u\rangle.$$

Let $\{u^{(1)}, \ldots, u^{(T)}\}$ denote a 1/2-cover of the unit sphere in $\mathbb{R}^d$; by standard arguments, we can find such a set with cardinality $\log T \leq d \log 5$. Using this covering, we have

$$\sup_{\theta\in\Omega} \|\frac{1}{m}\sum_{i=1}^m f_\theta(\widetilde{X}_{i;2k})\|_2 = \sup_{\|u\|_2 \leq 1} \tilde{V}_m(\widetilde{X}; u) \leq 2 \max_{j\in[T]} \tilde{V}_m(\widetilde{X}; u^{(j)}),$$

where the inequality follows by a discretization argument. Consequently, we have

$$\mathbb{P}\big[\sup_{\theta\in\Omega} \|\frac{1}{m}\sum_{i=1}^m f_\theta(\widetilde{X}_{i;2k})\|_2 \geq \epsilon_n^\mu\big] \leq \mathbb{P}\big[\max_{j\in[T]} \tilde{V}_m(\widetilde{X}; u^{(j)}) \geq \frac{\epsilon_n^\mu}{2}\big] \leq T \max_{j\in[T]} \mathbb{P}\big[\tilde{V}_m(\widetilde{X}; u^{(j)}) \geq \frac{\epsilon_n^\mu}{2}\big].$$

The remainder of our analysis focuses on bounding the tail probability for a fixed unit vector u, in particular ensuring an exponent small enough to cancel the $T \leq e^{d \log 5}$ pre-factor. By Lemma 2.3.7 of van der Vaart and Wellner (1996), for any $t > 0$, we have

$$\mathbb{P}_X\big[\tilde{V}_m(\widetilde{X}; u) \geq t\big] \leq c\mathbb{P}_{X,\epsilon}\big[V_m(\widetilde{X}; u) \geq \frac{t}{4}\big],$$

where $V_m(\widetilde{X}; u) = \sup_{\theta\in\Omega} \big|\frac{1}{m}\sum_{i=1}^m \epsilon_i(2w_\theta(\widetilde{X}_{i;2k}) - 1)\langle X_{i,k}, u\rangle\big|$, and $\{\epsilon_i\}_{i=1}^m$ is a sequence of i.i.d. Rademacher variables.

We now require a sequence of technical lemmas; see Section C.3 for their proofs. Our first lemma shows that the variable $V_m(\widetilde{X}; u)$, viewed as a function of the Rademacher sequence, is concentrated:

Lemma 8 *For any fixed $(\widetilde{X}, u)$, we have*

$$\mathbb{P}_\epsilon\big[V_m(\widetilde{X}; u) \geq \mathbb{E}_\epsilon V_m(\widetilde{X}; u) + t\big] \leq 2e^{-\frac{t^2}{16L_m^2(\widetilde{X}; u)}}, \qquad (61)$$

where $L_m(\widetilde{X}; u) = \frac{1}{m}\sqrt{\sum_{i=1}^m \langle X_{i,k}, u\rangle^2}$.

Our next lemma bounds the expectation with respect to the Rademacher random vector:

Lemma 9 *There exists a universal constant c such that for each fixed $(\widetilde{X}; u)$, we have*

$$\mathbb{E}_\epsilon V_m(\widetilde{X}; u) \leq \underbrace{c\frac{\|\mu^*\|_2}{\sigma^2}\sqrt{\log m}\,\Big[\sum_{\ell=1}^{2k} \mathbb{E}_{\tilde{\epsilon}} \|\frac{1}{m}\sum_{i=1}^m \tilde{\epsilon}_{i,\ell} X_{i,\ell}\langle X_{i,k}, u\rangle\|_2\Big]}_{M_m(\widetilde{X}; u)} + \underbrace{\mathbb{E}_g\big|\frac{1}{m}\sum_{i=1}^m g_{i,2k+1}\langle X_{i,k}, u\rangle\big|}_{N_m(\widetilde{X}; u)} \qquad (62)$$

where $\epsilon, \tilde{\epsilon} \in \mathbb{R}^m$ are random vectors with i.i.d. Rademacher components, and g is a random vector with i.i.d. $\mathcal{N}(0,1)$ components.

We now bound the three quantities $L_m(\widetilde{X}; u)$, $M_m(\widetilde{X}; u)$, and $N_m(\widetilde{X}; u)$ appearing in the previous two lemmas. In particular, let us introduce the quantities $L' = cL\|\mu^*\|_2\big(\frac{\|\mu^*\|_2^2}{\sigma^2} + 1\big)$, $L'' = L\sqrt{\|\mu^*\|_2^2 + \sigma^2}$ and $L = \frac{\sqrt{8}}{1-\rho_{\mathrm{mix}}}$.

Lemma 10 *Define the event*

$$\mathcal{E} = \Bigg\{ L_m(\widetilde{X}; u) \leq \tilde{c}\sqrt{\frac{2(\|\mu^*\|_2^2 + \sigma^2)\log\frac{1}{\delta}}{m}}, \quad M_m(\widetilde{X}; u) \leq L'k\sqrt{\frac{d\log m \log\frac{k}{\delta}}{m}} \\ \text{and } N_m(\widetilde{X}; u) \leq cL''\sqrt{\frac{d\log\frac{1}{\delta}}{m}} \Bigg\}.$$

Then we have $\mathbb{P}[\mathcal{E}] \geq 1 - \mathrm{e}^{-c' \, d \log \frac{1}{\delta}}$ *for* $m > d$ *and a universal constant* $c' > 0$ *which increases with the constants* c *in* L', N_m *and* $\tilde{c}$ *in* L_m.

In conjunction, Lemmas 8 and 9 imply that conditionally on the event $\mathcal{E}$, we have

$$\mathbb{E}_\epsilon\big[V_m(\widetilde{X}; u)\big] \leq c\sqrt{\|\mu^*\|_2^2 + \sigma^2}\big(\frac{\|\mu^*\|_2^2}{\sigma^2} + 1\big)k\sqrt{\frac{d \log m \log \frac{k}{\delta}}{m}}.$$

Note that by assumption on n we also have $m \geq d$ so that we can combine this bound with Lemma 10 which yields

$$\begin{aligned} T\,\mathbb{P}_X\big[\tilde{V}_m(\widetilde{X}; u) \geq t\big] &\leq T\,\mathbb{P}_{X,\epsilon}\big[V_m(\widetilde{X}; u) \geq \frac{t}{4} \mid \mathcal{E}\big] + T\,\mathbb{P}\big[\mathcal{E}^c\big] \\ &\leq 2\mathrm{e}^{4d - \left(\frac{c}{\tilde{c}}\right)^2 k^2 d \log m \log \frac{k}{\delta}} + \delta \mathrm{e}^{4d - c' d} \\ &\leq \delta, \end{aligned}$$

where the second inequality follows by setting $t/4 = c\|\mu^*\|_2\big(\frac{\|\mu^*\|_2^2}{\sigma^2} + 1\big)k\log\big(\frac{k}{\delta}\big)\sqrt{\frac{d\log m}{m}}$ and the final inequality holds for c', c and $\tilde{c}$ big enough. After rescaling δ by $8k$ and setting $m = \frac{n}{4k}$, the result follows after an application of Lemma 7.

Proof of inequality (47c)**:** In order to bound $|M_n^{\beta,k}(\theta) - \overline{M}^{\beta,k}(\theta)|$, we need a few extra steps. First, let us define new weights

$$v_\theta(X_{i-k}^{i+k-1}) = p(Z_0 = Z_1 = 1 \mid X_{i-k}^{i+k-1}, \theta) + p(Z_0 = Z_1 = -1 \mid X_{i-k}^{i+k-1}, \theta),$$

and also write the update in the form

$$\begin{aligned} M_n^{\beta,k}(\theta) &= \arg\max_{\zeta \in \Omega_\zeta} \Big\{ \mathbb{E}_{Z_1 | X_{i-k}^{i+k}, \theta} \log p(Z_1 \mid \zeta) + \sum_{i=2}^n \mathbb{E}_{Z_{i-1}^i | X_{i-k}^{i+k}, \theta} \log p(Z_i \mid Z_{i-1}, \zeta) \Big\} \\ &= \arg\max_{\zeta \in \Omega_\zeta} \Big\{ \frac{1}{2} + \sum_{i=2}^n \mathbb{E}_{Z_{i-1}^i | X_{i-k}^{i+k}, \theta} \log p(Z_i \mid Z_{i-1}, \zeta) \Big\} \\ &= \Pi_{\Omega_\zeta}\Big(\frac{1}{n}\sum_{i=2}^n v_\theta(X_{i-k}^{i+k-1})\Big), \end{aligned}$$

where we have reparameterized the transition probabilities with ζ via the equivalences $\beta = h(\zeta) := \frac{1}{2}\log\left(\frac{\zeta}{1-\zeta}\right)$. Note that the original EM operator is obtained via the transformation $M_n^{\beta,k}(\theta') = h(M_n^{\beta,k}(\theta'))$ and we have $\overline{M}^{\beta,k}(\theta) = \Pi_{\Omega_\zeta}\mathbb{E}v_\theta(X_{i-k}^{i+k-1})$ by definition.

Given this set-up, we can now pursue an argument similar to that of inequality (47b). The new weights remain Lipschitz with the same constant—that is, we have the bound $|v_\theta(\widetilde{X}_{i;2k}) - v_{\theta'}(\widetilde{X}_{i;2k})| \leq L\|\tilde{\theta}_i - \tilde{\theta}_i'\|_2$. As a consequence, we can write

$$\mathbb{P}\big[\sup_{\theta \in \Omega} \big| \frac{1}{m}\sum_{i=1}^m v_\theta(\widetilde{X}_{i;2k}) - \mathbb{E}v_\theta(\widetilde{X}_{i;2k})\big| \geq \epsilon_n^\beta\big] \leq \frac{\delta}{8k},$$

with ϵ_n^β defined as in the lemma statement. In this case, it is not necessary to perform the covering step, nor to introduce extra Rademacher variables after the Gaussian comparison step; consequently, the two constants ϵ_n^β and ϵ_n^μ differ by a factor of $\sqrt{d \log n}$ modulo constants.

Applying Lemma 7 then yields a tail bound for the quantity $\big|\frac{1}{n}\sum_{i=1}^n v_\theta(\widetilde{X}_{i;2k}) - \mathbb{E}v_\theta(\widetilde{X}_{i;2k})\big|$ with probability δ. Since projection onto a convex set only decreases the distance, we find that

$$\mathbb{P}\left[|M_n^{\beta,k}(\theta) - \overline{M}^{\beta,k}(\theta)| \geq C\frac{\sqrt{\|\mu^*\|_2^2 + \sigma^2}}{\sigma^2}\sqrt{\frac{k^3 \log(k^2/\delta)}{n}}\right] \leq \delta.$$

In order to prove the result, the last step needed is the fact that

$$\frac{1}{2}\Big|\log\frac{x}{1-x} - \log\frac{y}{1-y}\Big| \leq \frac{1}{\tilde{x}(1-\tilde{x})}|x-y| \leq \frac{2}{1-b^2}|x-y| =: L|x-y|$$

for $x, y, \tilde{x} \in \Omega_\zeta$. Since $M_n^{\beta,k}(\theta) \in \Omega_\zeta$ we finally arrive at

$$\mathbb{P}\Big[|M_n^{\beta,k}(\theta) - \overline{M}^{\beta,k}(\theta)| \geq C(1-b^2)\frac{\sqrt{\|\mu^*\|_2^2 + \sigma^2}}{\sigma^2}\sqrt{\frac{k^3 \log\left(\frac{k^2}{\delta}\right)}{n}}\Big] \leq \delta$$

and the proof is complete.

C.2 Proof of Lemma 2

We need to show that

$$\mathbb{P}\Big[\sup_{\theta \in \Omega} \|M_n(\theta) - M_n^k(\theta)\|_\star^2 \geq c_1\varphi_n^2(\delta, k)\Big] \leq \delta$$

with

$$\varphi_n^2(\delta, k) = \frac{Cs^4}{(1-b^2)\epsilon_{\mathrm{mix}}^{10}\bar{\pi}_{\min}^3}\Big[\frac{1}{\sigma}\frac{d\log^2(C_\epsilon n/\delta)}{n} + \frac{\|\mu^*\|_2}{\sigma}\frac{\log^2(C_\epsilon n/\delta)}{n} + \frac{\|\mu^*\|_2^2}{\sigma^2}\frac{\delta}{n}\Big].$$

We first claim that

$$\sup_{\theta \in \Omega}\|M_n(\theta) - M_n^k(\theta)\|_\star^2 \leq \frac{8\|Q_n - Q_n^k\|_\infty}{\lambda}, \quad \text{where} \quad \lambda \geq \frac{2}{3}(1-b^2). \tag{63}$$

In Section 5.3.1. we showed that population operators are strongly concave with parameter at least λ. We make the added observation that using our parameterization, the sample Q functions $Q_n^k(\cdot \mid \theta'), Q_n(\cdot \mid \theta')$ are also strongly concave. This is because the concavity results for the population operators did not use any property of the covariates in the HMM, in particular not the expectation operator, and the single term $\frac{1}{n}\mathbb{E}\sum_{z_0} p(z_0 \mid X_1^n, \beta')\log p(z_0; \beta) = \frac{1}{n}\log\frac{1}{2}$ is constant for all $\beta \in \Omega_\beta$. From this λ-strong concavity, the bound (63) follows immediately using the same argumentation as in the proof of Theorem 1.

Given the bound (63), the remainder of the proof focuses on bounding the difference $\|Q_n - Q_n^k\|_\infty$. Recalling the shorthand notation

$$h(X_i, z_i, \theta, \theta') = \log p(X_i|z_i, \theta) + \sum_{z_{i-1}} p(z_i|z_{i-1}, \theta') \log p(z_i|z_{i-1}, \theta),$$

we use a similar argumentation as in the Proof of Proposition 1 equation (52) to obtain

$$\begin{aligned}
\|Q_n - Q_n^k\|_\infty &= \Big| \sup_{\theta,\theta' \in \Omega} \frac{1}{n} \sum_{i=1}^{n} \sum_{z_i} (p(z_i|X_1^n, \theta') - p(z_i|X_{i-k}^{i+k}, \theta')) h(X_i, z_i, \theta, \theta') \Big| \qquad (64) \\
&+ \Big| \sup_{\theta,\theta' \in \Omega} \frac{1}{n} \sum_{z_0} p(z_0 \mid x_1^n, \theta') \log p(z_0 \mid \theta) \Big| \\
&\leq \frac{2Cs^3}{\epsilon_{\mathrm{mix}}^8 \overline{\pi}_{\min}} \frac{1}{n} \big[\sum_{i=1}^{k} \widetilde{\rho}_{\mathrm{mix}}^i \max_{z_i \in [s]} \big| h(X_i, z_i, \theta, \theta') \big| + \log \overline{\pi}_{\min}^{-1} \big] \\
&+ \frac{Cs^3 \widetilde{\rho}_{\mathrm{mix}}^k}{\epsilon_{\mathrm{mix}}^8 \overline{\pi}_{\min}} \frac{1}{n-2k} \sum_{i=k}^{n-k} \max_{z_i \in [s]} \big| h(X_i, z_i, \theta, \theta') \big| \\
&\leq \underbrace{\frac{2Cs^3}{\epsilon_{\mathrm{mix}}^8 \overline{\pi}_{\min} n} \big[\max_{z_i \in [s], X_1^k} \big| \log p(X_i \mid z_i, \theta) \big| \sum_{i=1}^{k} \widetilde{\rho}_{\mathrm{mix}}^i + \frac{s \log(\overline{\pi}_{\min} \epsilon_{\mathrm{mix}})^{-1}}{\overline{\pi}_{\min} \epsilon_{\mathrm{mix}}} + \log \overline{\pi}_{\min}^{-1} \big]}_{S_1} \\
&+ \frac{Cs^3 \widetilde{\rho}_{\mathrm{mix}}^k}{\epsilon_{\mathrm{mix}}^8 \overline{\pi}_{\min}} \big[\mathbb{E} \max_{z_i \in [s]} \big| \log p(X_i \mid z_i, \theta') \big| + e_{n-2k}(X) + s \log(\overline{\pi}_{\min} \epsilon_{\mathrm{mix}})^{-1} \big]
\end{aligned}$$

where we use $\max_{z_i \in [s]} \big| h(X_i, z_i, \theta, \theta') \big| \leq \max_{z_i \in [s]} \big| \log p(X_i|z_i, \theta) \big| + s \log(\overline{\pi}_{\min} \epsilon_{\mathrm{mix}})^{-1}$, and

$$e_n(X) := \Big| \frac{1}{n} \sum_{i=1}^{n} \max_{z_i \in [s]} \Big| \log p(X_i \mid z_i, \theta) \Big| - \mathbb{E} \max_{z_i \in [s]} \Big| \log p(X_i \mid z_i, \theta) \Big| \Big|.$$

By assumption, we have that $\mathbb{E} \max_{z_i \in [s]} |\log p(X_i \mid z_i, \theta)|$ is bounded by an appropriately large universal constant. We therefore have with probability one that

$$S_1 \leq \frac{Cs^4}{\epsilon_{\mathrm{mix}}^9 \overline{\pi}_{\min}^2} \frac{k}{n} \log(\epsilon_{\mathrm{mix}} \overline{\pi}_{\min})^{-1}.$$

Putting these together, we find that

$$\sup_{\theta \in \Omega} \|M_n(\theta) - M_n^k(\theta)\|_\star^2 \leq \frac{Cs^4}{\lambda \epsilon_{\mathrm{mix}}^9 \overline{\pi}_{\min}^2} \big[\frac{k}{n} \log(\epsilon_{\mathrm{mix}} \overline{\pi}_{\min})^{-1} + \widetilde{\rho}_{\mathrm{mix}}^k e_{n-2k}(X) \big].$$

Suppose that we can show that

$$\mathbb{P}\Big(e_n(X) \geq c_0 \Big(\frac{1}{\sigma} \sqrt{\frac{d \log^2(C_\epsilon n/\delta)}{n}} + \frac{\|\mu^*\|_2}{\sigma} \sqrt{\frac{\log^2(C_\epsilon n/\delta)}{n}} + \frac{\|\mu^*\|_2^2}{\sigma^2} \Big) \Big) \leq \delta, \qquad (65)$$

where c_0 is a universal constant and $C_\epsilon = \frac{C}{\epsilon_{\mathrm{mix}}^3 \overline{\pi}_{\min}^3}$. By assumption we have $\frac{1}{2} \frac{\log\left(Cn/\delta\right)}{\log(1/\widetilde{\rho}_{\mathrm{mix}})} \leq k \leq C \frac{\log\left(Cn/\delta\right)}{\log(1/\widetilde{\rho}_{\mathrm{mix}})}$ so that we obtain

$$\sup_{\theta \in \Omega} \|M_n(\theta) - M_n^k(\theta)\|_\star^2 \leq \varphi_n^2(\delta, k)$$

with probability at least $1 - \frac{\delta}{3}$ for an appropriate choice of C.

We now move on to prove the bound (65). Observe that we have

$$\begin{aligned}
e_n(X) &= \frac{1}{2n\sigma^2} \sum_{i=1}^{n} \big[\max\{\|X_i + \mu\|_2^2, \|X_i - \mu\|_2^2\} - \mathbb{E} \max\{\|X_i + \mu\|_2^2, \|X_i - \mu\|_2^2\} \big] \\
&= \frac{1}{2n\sigma^2} \sum_{i=1}^{n} \left(\|X_i\|_2^2 - \mathbb{E}\|X_i\|_2^2 \right) + \frac{1}{n\sigma^2} \sum_{i=1}^{n} \left(|X_i^T \mu| - \mathbb{E}|X_i^T \mu| \right).
\end{aligned}$$

Note that we are again dealing with a dependent sequence so that we cannot use usual Hoeffding type bounds. For some $\tilde{k}$ to be chosen later on, and $m = n/\tilde{k}$ using the proof idea of Lemma 7 with $f_2(X_i) = |X_i^T \mu|$ and $f_2(X_i) = \|X_i\|_2^2$, we can write

$$\begin{aligned}
\mathbb{P}(e_n(X) \geq \frac{t}{2\sigma^2}) \leq \tilde{k} \Big(& \underbrace{\mathbb{P}_0 \Big(|\frac{1}{m} \sum_{i=1}^{m} \|X_i\|_2^2 - \mathbb{E}\|X_i\|_2^2| \geq \frac{t}{2} \Big)}_{T_1} \\
& + \underbrace{\mathbb{P}_0 \Big(|\frac{1}{m} \sum_{i=1}^{m} |X_i^T \mu| - \mathbb{E}|X_i^T \mu|| \geq \frac{t}{4} \Big)}_{T_2} + m\beta(\tilde{k}) \Big),
\end{aligned}$$

where $\beta(\tilde{k})$ was previously defined in equation (58). We claim that the choices

$$t := c_1 \Big(\sigma \sqrt{\frac{d \log(\tilde{k}/\delta)}{m}} + \sigma \|\mu^*\|_2 \sqrt{\frac{\log(\tilde{k}/\delta)}{m}} + \|\mu^*\|_2^2 \Big), \quad \text{and} \quad \tilde{k} := \frac{C_2 \log(\frac{3n}{\epsilon_{\mathrm{mix}}^3 \overline{\pi}_{\min}^3 \delta})}{\log 1/\widetilde{\rho}_{\mathrm{mix}}},$$

suffice to ensure that $\mathbb{P}(e_n(X) \geq t/(2\sigma^2)) \leq \delta$. Notice that the bound (59) implies that

$$m\beta(\tilde{k}) \leq \frac{cm\rho_{\mathrm{mix}}^{\tilde{k}}}{\epsilon_{\mathrm{mix}}^3 \overline{\pi}_{\min}^3} \leq \frac{\delta}{3\tilde{k}}.$$

In the sequel we develop bounds on T_1 and T_2. For T_1, observe that since $X_i \sim Z_i \mu^* + \epsilon_i$ where ϵ_i is a Gaussian vector with covariance $\sigma^2 I$ and Z_i independent under $\mathbb{P}_0$, standard χ^2 tail bounds imply that

$$\mathbb{P}_0 \Big[|\frac{1}{m} \sum_{i=1}^{m} \|X_i\|_2^2 - \mathbb{E}\|X_i\|_2^2| \geq \frac{t}{2} \Big] \leq \frac{\delta}{3\tilde{k}}.$$

Finally, we turn our attention to the term T_2. Observe that,

$$X_i^T\mu \sim \frac{1}{2}\mathcal{N}(\mu^T\mu^*, \sigma^2\|\mu\|_2^2) + \frac{1}{2}\mathcal{N}(-\mu^T\mu^*, \sigma^2\|\mu\|_2^2),$$

so that $|\ X_i^T\mu \sim |\mathcal{N}(\mu^T\mu^*, \sigma^2\|\mu\|_2^2)|$. Denote $U_i =|\ X_i^T\mu\ |$. Letting ϵ denote a Rademacher random variable, observe that

$$\mathbb{E}\exp(tU_i) \overset{\text{(i)}}{\leq} \mathbb{E}\exp(2t\epsilon U_i) \overset{\text{(ii)}}{\leq} \exp\left(2t^2\sigma^2\|\mu\|_2^2 + 2t\mu^T\mu^*\right),$$

where (i) follows using symmetrization, and (ii) follows since the random variable ϵU_i is a Gaussian mixture. Observe that

$$\mathbb{E}U_i \overset{\text{(iii)}}{\leq} |\mu^T\mu^*| + \sigma\|\mu\|_2 \overset{\text{(iv)}}{\leq} \underbrace{2(\sigma + \|\mu^*\|_2)\|\mu^*\|_2}_{M},$$

where we have used for (iii) that U_i is a folded normal, and for (iv) that $\|\mu - \mu^*\|_2 \leq \frac{\|\mu^*\|_2}{4}$. Setting $D := 4\sigma\|\mu^*\|_2\sqrt{\frac{\log(6\tilde{k}/\delta)}{m}}$ observe that $\frac{t}{4} \geq 2M + D$ for big enough c_1. Thus, applying the Chernoff bound yields

$$\begin{aligned} T_2 \leq \mathbb{P}_0\Big[|\frac{1}{m}\sum_{i=1}^m U_i - \mathbb{E}U_i| \geq 2M + D\Big] &\leq \mathbb{P}_0\Big(\ |\ \frac{1}{m}\sum_{i=1}^m U_i\ |\geq M + D\Big) \\ &\leq 2\inf_{t\geq 0}\Big\{\mathbb{E}\exp\Big(\frac{t}{m}\sum_{i=1}^m U_i - Mt - Dt\Big)\Big\}, \\ &\leq 2\exp\Big(-\frac{mD^2}{8\sigma^2\|\mu\|_2^2}\Big) \leq \frac{\delta}{3\tilde{k}}. \end{aligned}$$

By combining the bounds on T_1 and T_2, some algebra shows that our choices of $t, \tilde{k}$ yield the claimed bound—namely, that $\mathbb{P}\big[e_n(X) \geq t/(2\sigma^2)\big] \leq \delta$.

C.3 Proofs of technical lemmas

In this section, we collect the proofs of various technical lemmas cited in the previous sections.

C.3.1 Proof of Lemma 8

We use the following concentration theorem (e.g., Ledoux (1997)): suppose that the function $f : \mathbb{R}^n \to \mathbb{R}$ is coordinate-wise convex and L-Lipschitz with respect to the Euclidean norm. Then for any i.i.d. sequence of variables $\{X_i\}_{i=1}^n$ taking values in the interval $[a, b]$, we have

$$\mathbb{P}\big[f(X) \geq \mathbb{E}f(X) + \delta\big] \leq \mathrm{e}^{-\frac{\delta^2}{4L^2(b-a)^2}} \tag{66}$$

We consider the process without absolute values (which introduces the factor of two in the lemma) and see that $\epsilon := (\epsilon_1, \ldots, \epsilon_n)$ is a random vector with bounded entries and that the supremum over affine functions is convex.

It remains to show that the function $\epsilon \mapsto V_m(\widetilde{X}, u)$ is Lipschitz with $L_m(\widetilde{X}; u)$ as follows

$$\begin{aligned} &\big|\sup_\theta \frac{1}{m}\sum_{i=1}^m \epsilon_i f_\theta(\widetilde{X}_{i;2k}) - \sup_\theta \frac{1}{m}\sum_{i=1}^m \epsilon_i' f_\theta(\widetilde{X}_{i;2k})\big| \\ &\leq \frac{1}{m}|\sum_{i=1}^m (\epsilon_i - \epsilon_i') f_\theta(\widetilde{X}_{i;2k})| \\ &\leq \frac{1}{m}\sqrt{\sum_{i=1}^m (2w_{\tilde{\theta}}(\widetilde{X}_{i;2k}) - 1)^2\langle X_{i,k}, u\rangle^2}\|\epsilon - \epsilon'\|_2 \\ &\leq L_m(\widetilde{X}; u)\|\epsilon - \epsilon'\|_2 \end{aligned}$$

where $\tilde{\theta} = \arg\max_{\theta\in\Omega}\sum_i \epsilon_i f_\theta(\widetilde{X}_{i;2k})$ in the last line and we use that $|2w_\theta(\widetilde{X}_{i;2k}) - 1| \leq 1$.

C.3.2 Proof of Lemma 9

The proof consists of three steps. First, we observe that the Rademacher complexity is upper bounded by the Gaussian complexity. Then we use Gaussian comparison inequalities to reduce the process to a simpler one, followed by a final step to convert it back to a Rademacher process.

Relating the Gaussian and Rademacher complexity: Let $g_i \sim \mathcal{N}(0, 1)$. It is easy to see that using Jensen's inequality and the fact that $\epsilon_i|g_i| \overset{d}{=} g_i$

$$\begin{aligned} \mathbb{E}_\epsilon \sup_\theta \frac{1}{m}\sum_{i=1}^m \epsilon_i f_\theta(\widetilde{X}_{i;2k}) &= \sqrt{\frac{2}{\pi}}\mathbb{E}_\epsilon \sup_\theta \frac{1}{m}\sum_{i=1}^m \epsilon_i \mathbb{E}_g[|g_i|] f_\theta(\widetilde{X}_{i;2k}) \\ &\leq \sqrt{\frac{2}{\pi}}\mathbb{E}_g \sup_\theta \frac{1}{m}\sum_{i=1}^m g_i f_\theta(\widetilde{X}_{i;2k}). \end{aligned}$$

Lipschitz continuity: For $\theta = (\mu, \beta)$ define the corresponding effective parameter that is obtained by treating the observed variables X as fixed

$$\tilde{\theta}_i := (\gamma_i, \beta) = (\frac{\langle \mu, X_{i,1}\rangle}{\sigma^2}, \ldots, \frac{\langle \mu, X_{i,2k}\rangle}{\sigma^2}, \beta). \tag{67}$$

Now we can use results in the proof of Corollary 1 to see that $\tilde{\theta}_i \mapsto F(\tilde{\theta}_i; \widetilde{X}_{i;2k}) := f_\theta(\widetilde{X}_{i;2k})$ is Lipschitz in the Euclidean norm, i.e. there exists an L, only dependent on ρ_{mix} such that

$$|F(\tilde{\theta}_i; \widetilde{X}_{i;2k}) - F(\tilde{\theta}_i'; \widetilde{X}_{i;2k})| \leq L\|\tilde{\theta}_i - \tilde{\theta}_i'\|_2|\langle X_{i,k}, u\rangle| \tag{68}$$

For this we directly use results (exponential family representation) that were used to show Corollary 1. We overload notation and write $X_\ell := X_{1,\ell}$ and analyze Lipschitz continuity for the first block. First note that $F(\tilde{\theta}_i, X_{1;2k}) = (2\mathbb{E}_{Z_k|X_1^{2k},\theta}Z_k - 1)X_{i,k}$. By Taylor's theorem, we then have

$$\begin{aligned} |F(\tilde{\theta}_i; \widetilde{X}_{i;2k}) - F(\tilde{\theta}_i'; \widetilde{X}_{i;2k})| &= |\langle X_{i,k}, u\rangle||\mathbb{E}_{Z_k|\widetilde{X}_{i;2k},\theta}Z_k - \mathbb{E}_{Z_k|\widetilde{X}_{i;2k},\theta'}Z_k| \\ &\leq |\langle X_{i,k}, u\rangle||\mathbb{E}_{Z_k|\widetilde{X}_{i;2k},(\mu,\beta)}Z_k - \mathbb{E}_{Z_k|\widetilde{X}_{i;2k},(\mu',\beta)}Z_k| \\ &\quad + |\langle X_{i,k}, u\rangle||\mathbb{E}_{Z_k|\widetilde{X}_{i;2k},(\mu',\beta)}Z_k - \mathbb{E}_{Z_k|\widetilde{X}_{i;2k},(\mu',\beta')}Z_k| \end{aligned}$$

Let us examine each of the summands separately. By the Cauchy-Schwartz inequality and Lemma 6, we have

$$\begin{aligned}|\mathbb{E}_{Z_k|\widetilde{X}_{i;2k},(\mu,\beta)}Z_k - \mathbb{E}_{Z_k|\widetilde{X}_{i;2k},(\mu',\beta)}Z_k| &= \frac{1}{\sigma}|\sum_{\ell=1}^{2k}\frac{\partial^2\Phi}{\partial\gamma_\ell\partial\gamma_0}\Big|_{\theta=\tilde{\theta}}(\gamma_\ell-\gamma'_\ell)| \\ &= |\sum_{\ell=1}^{2k}\mathrm{cov}(Z_0, Z_\ell \mid \widetilde{X}_{i;2k},\tilde{\theta})(\langle\mu, X_\ell\rangle - \langle\mu', X_\ell\rangle)| \\ &\leq \sqrt{(\sum_{\ell=1}^{2k}4\rho_{\mathrm{mix}}^{2\ell})\sum_{\ell=1}^{2k}(\gamma_\ell-\gamma'_\ell)^2},\end{aligned}$$

as well as

$$\begin{aligned}|\mathbb{E}_{Z_k|\widetilde{X}_{i;2k},(\mu',\beta)}Z_k - \mathbb{E}_{Z_k|\widetilde{X}_{i;2k},(\mu',\beta')}Z_k| &= |\sum_{\ell=1}^{2k}\frac{\partial^2\Phi}{\partial\beta_\ell\partial\gamma_0}\Big|_{\theta=\tilde{\theta}}(\beta-\beta')| \\ &= |\sum_{\ell=1}^{2k}\mathrm{cov}(Z_0, Z_\ell Z_{\ell+1} \mid \widetilde{X}_{i;2k},\tilde{\theta})(\beta-\beta')| \\ &\leq \frac{2}{1-\rho_{\mathrm{mix}}}|\beta-\beta'|.\end{aligned}$$

Combining these two bounds yields

$$\begin{aligned}|F(\tilde{\theta}_i;\widetilde{X}_{i;2k}) - F(\tilde{\theta}'_i;\widetilde{X}_{i;2k})|^2 &\leq \langle X_{i,k},u\rangle^2 L\big(\sum_{\ell=1}^{2k}(\gamma_\ell-\gamma'_\ell)^2 + (\beta-\beta')^2\big) \\ &= \langle X_{i,k},u\rangle^2 L^2\|\tilde{\theta}_i - \tilde{\theta}'_i\|_2^2\end{aligned}$$

with $L^2 = \frac{8}{(1-\rho_{\mathrm{mix}})^2}$.

Applying the Sudakov-Fernique Gaussian comparison: Let us introduce the shorthands $X_\theta = \frac{1}{m}\sum_i g_i f_\theta(\widetilde{X}_{i;2k})$, and

$$Y_\theta = \frac{1}{m}L\sum_i\big(\sum_{\ell=1}^{2k}g_{i\ell}\frac{\langle\mu, X_{i,\ell}\rangle}{\sigma^2} + g_{i,2k+1}\beta\big)\langle X_{i,k},u\rangle.$$

By construction, the random variable $X_\theta - X'_\theta$ is a zero-mean Gaussian variable with variance

$$\begin{aligned}\mathbb{E}_g(X_\theta - X_{\theta'})^2 &= \sum_i(F(\tilde{\theta};\widetilde{X}_{i;2k}) - F(\tilde{\theta}';\widetilde{X}_{i;2k}))^2 \\ &\leq L^2\sum_i\langle X_{i,k},u\rangle^2\big(\sum_{\ell=1}^{2k}(\gamma_{i,\ell}-\gamma'_{i,\ell})^2 + (\beta-\beta')^2\big) \\ &= \mathbb{E}_g(Y_\theta - Y_{\theta'})^2 \qquad (69)\end{aligned}$$

By the Sudakov-Fernique comparison Ledoux and Talagrand (2013),we are then guaranteed that $\mathbb{E}\sup_\theta X_\theta \leq \mathbb{E}\sup_\theta Y_\theta$. Therefore, it is sufficient to bound

$$\mathbb{E}_g\sup_{\theta\in\Omega}Y_\theta = \underbrace{\mathbb{E}_g\sup_\theta\frac{L}{\sigma^2 n}\sum_{i=1}^m\sum_{\ell=1}^{2k}g_{i\ell}\langle\mu, X_{i,\ell}\rangle\langle X_{i,k},u\rangle}_{T_1} + \underbrace{\mathbb{E}_g\sup_\theta\frac{L}{n}\sum_{i=1}^m g_{i,2k+1}\beta\langle X_{i,k},u\rangle}_{T_2}.$$

Converting back to a Rademacher process: We now convert the term T_1 back to a Rademacher process, which allows us to use sub-exponential tail bounds. We do so by re-introducing additional Rademacher variables, and then removing the term $\max_i|g_i|$ via the Ledoux-Talagrand contraction theorem (Ledoux and Talagrand, 2013). Given a Rademacher variable ϵ_{il} independent of g, note the distributional equivalence $\epsilon_{il}g_{il}\stackrel{d}{=}g_{i\ell}$. Then consider the function $\phi_{i\ell}(g_{i\ell}) := g_{i\ell}h_{i\ell}$ with $h_{i\ell} := \langle\mu, X_{i,\ell}\rangle\langle X_{i,k},u\rangle$ for which it is easy to see that

$$|\phi_{i\ell}(g_{i\ell},h_{i\ell}) - \phi_{i\ell}(g_{i\ell},h'_{i\ell})| \leq |g_{i\ell}||h_{i\ell}-h'_{i\ell}| \qquad (70)$$

Applying Theorem 4.12. in Ledoux and Talgrand Ledoux and Talagrand (2013) yields

$$\mathbb{E}\sup_\theta\frac{1}{m}\sum_{i=1}^m\sum_{\ell=1}^{2k}\epsilon_{i\ell}g_{i\ell}\langle\mu, X_{i\ell}\rangle\langle X_{i,k},u\rangle \leq \mathbb{E}_g\|g\|_\infty\mathbb{E}_\epsilon\sup_\theta\frac{1}{m}\sum_{i=1}^m\sum_{\ell=1}^{2k}\epsilon_{i\ell}\langle\mu, X_{i,\ell}\rangle\langle X_{i,k},u\rangle.$$

Putting together the pieces yields the claim (62).

C.3.3 Proof of Lemma 10

We prove that the probability of each of the events corresponding to the inequalities is smaller than $\frac{1}{3}\mathrm{e}^{-\tilde{c}d\log(\frac{k}{\delta})}$.

Bounding L_m: We start by bounding $L_m(\widetilde{X};u)$. Note that

$$\frac{1}{m}\sum_{i=1}^m\langle X_{i,k},u\rangle^2 \leq \|\mu^*\|_2^2 + \frac{1}{m}\sum_{i=1}^m\langle n_{i,k},u\rangle^2 + \frac{1}{m}\sum_{i=1}^m\langle\mu^*,u\rangle\langle n_{i,k},u\rangle$$

where the sum $\sum_{i=1}^m\langle n_{i,k},u\rangle^2$ is sub-exponential random variable with parameters $(2\sqrt{m}\sigma^2,4)$ so that

$$\mathbb{P}(\frac{1}{m}\sum_{i=1}^m\langle n_{i,k},u\rangle^2 - \sigma^2 \geq \tilde{c}^2\sigma^2\log(1/\delta)) \leq \mathrm{e}^{-c'm\log\frac{1}{\delta}} \leq \mathrm{e}^{-c'd\log\frac{1}{\delta}}$$

where the last inequality follows since $m \geq d$ by assumption. Since $\langle n_{i,k},u\rangle$ can be readily bounded by a sub-Gaussian tailbound it then follows directly that $L_m^2(\widetilde{X};u) \leq \tilde{c}^2\frac{2(\|\mu^*\|_2^2+\sigma^2)\log\frac{1}{\delta}}{m}$ with probability at least $1-\frac{1}{3}\mathrm{e}^{-c'd\log(\frac{k}{\delta})}$ for c' large enough.

Bounding N_m: In order to bound $N_m(\widetilde{X};u)$, we first introduce an extra Rademacher random variable into its definition; doing so does not change its value (now defined by an expectation over both g and the Rademacher variables). We now require a result for a product of the form ϵgh where g,h are independent Gaussian random variables.

Lemma 11 *Let (ϵ, g, h) be independent random variables, with ϵ Rademacher, $g \sim \mathcal{N}(0, \sigma_g^2)$, and $h \sim \mathcal{N}(0, \sigma_h^2)$. Then the random variable $Z = \epsilon gh$ is a zero-mean sub-exponential random variable with parameters $(\frac{\sigma_g^2\sigma_h^2}{2}, \frac{1}{4})$.*

Proof Note that $g' = \alpha h$ with $\alpha = \frac{\sigma_g}{\sigma_h}$ is identically distributed as g. Therefore, we have

$$gh = \frac{1}{\alpha} g g' = \frac{1}{4\alpha}[(g - g')^2 + (g + g')^2]$$

The random variables $g - g'$ and $g + g'$ are independent and therefore $(g - g')^2, (g + g')^2$ are sub-exponential with parameters $\nu^2 = 4\sigma_g^4$, $b = \frac{1}{4}$. This directly yields

$$\mathbb{E}\mathrm{e}^{\lambda\epsilon[(g+g')^2 - (g-g')^2]} \leq \mathrm{e}^{4\lambda^2\sigma_g^4}$$

for $|\lambda| \leq \frac{1}{b}$. Therefore $\mathbb{E}\mathrm{e}^{\lambda\epsilon gh} \leq \mathrm{e}^{\frac{\lambda^2\sigma_g^2\sigma_h^2}{4}}$, which shows that ϵgh is sub-exponential with parameters $(\frac{\sigma_g^2\sigma_h^2}{2}, \frac{1}{4})$. ■

Returning to the random variable $N_m(\widetilde{X}; u)$, each term $\epsilon_i g_{i,2k+1}\langle X_{i,k}, u\rangle$ is a sub-exponential random variable with mean zero and parameter $\nu^2 = \|\mu^*\|_2^2 + \frac{\sigma^2}{2}$. Consequently, there are universal constants such that $N_m(\widetilde{X}; u) \leq cL\nu\sqrt{\frac{d\log\frac{k}{\delta}}{m}}$ with probability at least $1 - \frac{1}{3}\mathrm{e}^{-c' d\log(\frac{k}{\delta})}$.

Bounding M_m: Our next claim is that with probability at least $1 - \frac{1}{3}\mathrm{e}^{-c' d\log(\frac{k}{\delta})}$, we have

$$\mathbb{E}_\epsilon\|\frac{1}{m}\sum_{i=1}^m \epsilon_{i\ell} X_{i,\ell}\langle X_{i,k}, u\rangle\|_2 \leq (\|\mu^*\|_2^2 + \sigma^2)\sqrt{\frac{d\log\frac{k}{\delta}}{m}}, \tag{71}$$

which then implies that $M_m(\widetilde{X}; u) \leq c\|\mu^*\|_2\big(\frac{\|\mu^*\|_2}{\sigma^2} + 1\big)k\sqrt{\frac{d\log m\log\frac{k}{\delta}}{m}}$. In order to establish this claim, we first observe that by Lemma 11, the random variable $\epsilon_{i,\ell}\langle X_{i,\ell}, u\rangle\langle X_{i,k}, u\rangle$ is zero mean, sub-exponential with parameter at most $\nu^2 = (\|\mu^*\|_2^2 + \sigma^2)^2$. The bound then follows by the same argument used to bound the quantity N_m.

Appendix D. Mixing related results

In the following we use the shorthand notation $\pi_k^\theta := p(z_k \mid x_0^k, \theta)$ which we refer to the filtering distribution which is tied to some distribution μ on z_0.

Introducing the shorthand notation $p_\mu(x_k) := \sum_{z_k}\sum_{z_{k-1}} p(x_k \mid z_k)p(z_k \mid z_{k-1})\mu(z_{k-1})$, we define the filter operator

$$F_i\nu(z_i) := \frac{\sum_{z_{i-1}} p(x_i \mid z_i)p(z_i \mid z_{i-1})\nu(z_{i-1})}{\sum_{z_i}\sum_{z_{i-1}} p(x_i \mid z_i)p(z_i \mid z_{i-1})\nu(z_{i-1})} = \sum_{z_{i-1}} \frac{p(x_i \mid z_i)p(z_i \mid z_{i-1})}{p_\nu(x_i)}\nu(z_{i-1}).$$

where the observations x are fix. Using this notation, the filtering distribution can then be rewritten in the more compact form $\pi_k^\theta = p(z_k \mid x_0^k, \theta) = F_k \ldots F_1\mu$. Similarly, we define

$$K_{j|i}(z_j \mid z_{j-1}) := \frac{p(z_j \mid z_{j-1})p(x_j \mid z_j)p(x_{j+1}^i \mid z_j)}{\sum_{z_j} p(z_j \mid z_{j-1})p(x_j \mid z_j)p(x_{j+1}^i|z_j)}, \quad \text{and} \quad \nu_{\ell|i} := \frac{p(x_{\ell+1}^i \mid z_\ell)\nu(z_\ell)}{\sum_{z_\ell} p(x_{\ell+1}^i \mid z_\ell)\nu(z_\ell)}$$

Note that $\epsilon_{\mathrm{mix}}C_0 \leq p(x_{\ell+1}^i \mid z_\ell) \leq \epsilon_{\mathrm{mix}}^{-1}C_0$ where

$$C_0 = \sum_{z_i \ldots z_{\ell+1}} p(x_i \mid z_i)p(z_i \mid z_{i-1}) \ldots p(x_{\ell+1} \mid z_{\ell+1})\overline{\pi}(z_{\ell+1})$$

and therefore by definition of ϵ_{mix} (3)

$$\sup_x \frac{\sup_z p(x_{\ell+1}^i \mid z_\ell)}{\inf_z p(x_{\ell+1}^i \mid z_\ell)} \leq \epsilon_{\mathrm{mix}}^{-2}. \tag{72}$$

With these definitions, it can be verified (e.g., see Chapter 5 of van Handel (2008)) that $F_i \ldots F_{\ell+1}\nu = \nu_{\ell+1|i}^T K_{\ell+1|i} \ldots K_{i|i}$, where $\nu^T K := \int \nu(x')K(x|x')\mathrm{d}x'$. In the discrete setting, this relation can be written as the row vector ν being right multiplied by the kernel matrix K.

D.1 Consequences of mixing

In this technical appendix we derive several useful consequences of the geometric mixing condition on the stochastic process Z_i.

Lemma 12 *For any geometrically ρ_{mix}-mixing and time reversible Markov chain $\{Z_i\}$ with s states, there is a universal constant c such that*

$$\sup_{z_0} \big|p(z_0 \mid x_k^n) - p(z_0)\big| < \frac{c(s+1)}{\overline{\pi}_{\min}^3 \epsilon_{\mathrm{mix}}^3}\rho_{\mathrm{mix}}^k. \tag{73}$$

Proof We first prove the following relation

$$\sup_x |p(z_i \mid x_{i+k}) - p(z_i)| \leq c_0 \frac{\rho_{\mathrm{mix}}^k}{\overline{\pi}_{\min}}. \tag{74}$$

Using time reversibility and the definition of mixing (4) we obtain

$$\begin{aligned}
\max_x (p(z_0 \mid x_k) - \overline{\pi}(z_0)) &= \sum_{z_k} (p(z_0 \mid z_k) - \overline{\pi}(z_0))p(z_k \mid x_k) \\
&\leq \max_{z_k} |(p(z_0 \mid z_k) - \overline{\pi}(z_0))| \sum_{z_k} p(z_k \mid x_k) \\
&\leq \max_{z_k} \Big|\frac{p(z_k \mid z_0)\overline{\pi}(z_0)}{\overline{\pi}(z_k)} - \frac{\overline{\pi}(z_0)\overline{\pi}(z_k)}{\overline{\pi}(z_k)}\Big| \\
&\leq \frac{\overline{\pi}(z_0)}{\overline{\pi}(z_k)} \max_{z_k} |p(z_k \mid z_0) - \overline{\pi}(z_0)| \leq \frac{c_0\rho_{\mathrm{mix}}^k}{\overline{\pi}_{\min}}
\end{aligned}$$

where $p(z_k \mid z_0) = P(Z_k = z_k \mid Z_0 = z_0)$ and $p(z_0 \mid z_k) = P(Z_0 = z_0 \mid Z_k = z_k)$.

Using this result we can now prove inequality (73). By definition, we have

$$p(z_0) = \frac{p(x_{k+1}^n \mid x_k)p(x_k)p(z_0)}{p(x_{k+1}^n, x_k)}, \quad \text{and} \quad p(z_0 \mid x_k, x_{k+1}^n) = \frac{p(x_{k+1}^n \mid x_k, z_0)p(x_k \mid z_0)p(z_0)}{p(x_{k+1}^n, x_k)}$$

and therefore

$$|p(z_0) - p(z_0 \mid x_k^n)| \leq \frac{p(x_k)p(z_0)}{p(x_k^n)}|p(x_{k+1}^n \mid x_k) - p(x_{k+1}^n \mid x_k, z_0)| + \frac{p(x_{k+1}^n \mid x_k, z_0)p(z_0)}{p(x_{k+1}^n \mid x_k)}|p(x_k) - p(x_k \mid z_0)| \quad (75)$$

In the following we bound each of the two differences. Note that

$$\begin{aligned}|p(x_{k+1}^n \mid x_k, z_0) - p(x_{k+1}^n \mid x_k)| &= \sum_{z_k}\sum_{z_{k+1}} p(x_{k+1}^n \mid z_{k+1})p(z_{k+1} \mid z_k)|p(z_k \mid x_k, z_0) - p(z_k \mid x_k)| \\ &\leq \sup_{z_k, x_k} |p(z_k \mid x_k, z_0) - p(z_k \mid x_k)| \sum_{z_k} p(x_{k+1}^n \mid z_k) \quad (76)\end{aligned}$$

The last term $\sum_{z_k} p(x_{k+1}^n \mid z_k)$ is bounded by s for s-state models. Using the bound (74), we obtain

$$|p(x_k \mid z_0) - p(x_k)| = \frac{|p(z_0 \mid x_k) - \overline{\pi}(z_0)|p(x_k)}{\overline{\pi}(z_0)} \leq \frac{p(x_k)}{\overline{\pi}_{\min}^2}\rho_{\mathrm{mix}}^k \quad (77)$$

which yields

$$\begin{aligned}|p(z_k \mid x_k, z_0) - p(z_k \mid x_k)| &= p(x_k \mid z_k)\left|\frac{p(z_k \mid z_0)}{p(x_k \mid z_0)} - \frac{\overline{\pi}(z_k)}{p(x_k)}\right| \\ &\leq \frac{p(x_k \mid z_k)}{p(x_k \mid z_0)}\left(|p(z_k \mid z_0) - \overline{\pi}(z_k)| + \frac{\overline{\pi}(z_k)}{p(x_k)}|p(x_k \mid z_0) - p(x_k)|\right) \\ &\leq \frac{p(x_k \mid z_k)}{p(x_k \mid z_0)}\left(\rho_{\mathrm{mix}}^k + \frac{1}{\overline{\pi}_{\min}^2}\rho_{\mathrm{mix}}^k\right) \\ &\leq \frac{2\rho_{\mathrm{mix}}^k}{p(z_k \mid z_0)\overline{\pi}_{\min}^2} \leq \frac{2}{\overline{\pi}_{\min}^3\epsilon_{\mathrm{mix}}}\rho_{\mathrm{mix}}^k. \quad (78)\end{aligned}$$

The last statement is true because one can check that for all $t \in \mathbb{N}$ we have

$$\min_{z_k, z_0} p(z_k \mid z_0) = \min_{ij}(A^t)_{ij} \geq \min_{ij}(A)_{ij} \geq \epsilon_{\mathrm{mix}}\overline{\pi}_{\min}$$

for any stochastic matrix A which satisfies the mixing condition (3).

Substituting (76) with (78) and (77) into (75), we obtain

$$\begin{aligned}|p(z_0) - p(z_0 \mid x_k^n)| &\leq \frac{\sum_{z_k} p(x_{k+1}^n \mid z_k)p(z_0)}{\sum_{z_k} p(x_{k+1}^n \mid z_k)p(z_k \mid x_k)}\frac{2\rho_{\mathrm{mix}}^k}{\overline{\pi}_{\min}^3\epsilon_{\mathrm{mix}}} + \frac{p(x_{k+1}^n \mid x_k, z_0)p(z_0)}{p(x_{k+1}^n \mid x_k)}\frac{\rho_{\mathrm{mix}}^k}{\overline{\pi}_{\min}} \\ &\leq \left(\frac{2s}{\overline{\pi}_{\min}^3\epsilon_{\mathrm{mix}}^3} + \frac{1}{\epsilon_{\mathrm{mix}}^2\overline{\pi}_{\min}}\right)\rho_{\mathrm{mix}}^k \leq \frac{2s+1}{\overline{\pi}_{\min}^3\epsilon_{\mathrm{mix}}^3}\rho_{\mathrm{mix}}^k\end{aligned}$$

where we use (72) to see that

$$\frac{\sum_{z_k} p(x_{k+1}^n \mid z_k)p(z_k \mid x_k, z_0)}{\sum_{z_k} p(x_{k+1}^n \mid z_k)p(z_k \mid x_k)} \leq \frac{\max_{z_k} p(x_{k+1}^n \mid z_k)}{\min_{z_k} p(x_{k+1}^n \mid z_k)} \leq \epsilon_{\mathrm{mix}}^{-2}$$

and similarly for the first term. ■

Lemma 13 (Filter stability) *For any mixing Markov chain which fulfills condition* (3)*, the following holds*

$$\|F_i \dots F_1(\nu - \nu')\|_\infty \leq \epsilon_{\mathrm{mix}}^{-2}\widetilde{\rho}_{\mathrm{mix}}^i\|\nu - \nu'\|_1$$

where $\widetilde{\rho}_{\mathrm{mix}} = 1 - \epsilon_{\mathrm{mix}}\overline{\pi}_{\min}$*. In particular we have*

$$\sup_{z_i} |p(z_i \mid x_1^i) - p(z_i \mid x_{-n}^i)| \leq 2\epsilon_{\mathrm{mix}}^{-2}\widetilde{\rho}_{\mathrm{mix}}^i. \quad (79)$$

Proof Given the mixing assumption (3) we can show that $K_{j|i}(x|y) \geq \epsilon p_{j|i}(x)$ with $\epsilon = \epsilon_{\mathrm{mix}}\overline{\pi}_{\min}$ for some probability distribution $p_{j|i}(\cdot)$. This is because we can lower bound

$$\begin{aligned}K_{j|i}(z_j \mid z_{j-1}) &= \frac{p(z_j \mid z_{j-1})p(x_j \mid z_j)p(x_{j+1}^i \mid z_j)}{\sum_{z_j} p(z_j \mid z_{j-1})p(x_j \mid z_j)p(x_{j+1}^i \mid z_j)} \\ &\geq \underbrace{\frac{\epsilon_{\mathrm{mix}}\overline{\pi}(z_j)p(x_j \mid z_j)p(x_{j+1}^i \mid z_j)}{\sum_{z_j}\frac{\overline{\pi}(z_j)}{\overline{\pi}_{\min}}p(x_j \mid z_j)p(x_{j+1}^i \mid z_j)}}_{=:\epsilon p_{j|i}(z_j)}\end{aligned}$$

with $\epsilon = \epsilon_{\mathrm{mix}}\overline{\pi}_{\min}$. This allows us to define the stochastic matrix

$$Q_{j|i} = \frac{1}{1-\epsilon}(K_{j|i} - \epsilon P_{j|i}) \text{ or } K_{j|i} = \epsilon P_{j|i} + (1-\epsilon)Q_{j|i}.$$

where $(P_{j|i})_{k\ell} = p_{j|i}(\ell)$ and for any two probability distributions ν_1, ν_2 we have $(\nu_1 - \nu_2)^T P_{j|i} = 0$. Using $\widetilde{\rho}_{\mathrm{mix}} = 1 - \epsilon$ we then obtain by induction, Hoelder's inequality and

inequality (72)

$$\begin{aligned}
&\|(\nu_{1|i} - \nu'_{1|i})^T K_{1|i} \dots K_{i|i}\|_\infty \\
&\leq \prod_{j=1}^{i} (1-\epsilon) \|(\nu_{1|i} - \nu'_{1|i})^T \otimes_{j=1}^{i} Q_{j|i}\|_2 \\
&\leq \widetilde{\rho}^i_{\text{mix}} \|\nu_{1|i} - \nu'_{1|i}\|_2 \prod_{j=1}^{i} \|Q^T_{j|i}\|_{op} \leq \widetilde{\rho}^i_{\text{mix}} \|\nu_{1|i} - \nu'_{1|i}\|_2 \\
&\leq \widetilde{\rho}^i_{\text{mix}} \left\| \frac{p(x^i_2 \mid \cdot)\nu(\cdot)}{\sum_{z_1} p(x^i_2 \mid z_1)\nu(z_1)} - \frac{p(x^i_2 \mid \cdot)\nu'(\cdot)}{\sum_{z_1} p(x^i_2 \mid z_1)\nu'(z_1)} \right\|_2 \\
&\leq \widetilde{\rho}^i_{\text{mix}} \Big[\Big\| \frac{p(x^i_2 \mid \cdot)}{\sum_{z_1} p(x^i_2 \mid z_1)\nu(z_1)} (\nu(\cdot) - \nu'(\cdot)) \Big\|_2 \\
&+ \Big| \sup_{z_1} p(x^i_2 \mid z_1) \Big(\frac{1}{\sum_{z_1} p(x^i_2 \mid z_1)\nu(z_1)} - \frac{1}{\sum_{z_1} p(x^i_2 \mid z_1)\nu'(z_1)} \Big) \Big| \|\nu'(\cdot)\|_1 \Big] \\
&\leq \widetilde{\rho}^i_{\text{mix}} \Big(\frac{\sup_{z_1} p(x^i_2 \mid z_1)}{\inf_{z_1} p(x^i_2 \mid z_1)} \Big)^2 \|\nu - \nu'\|_1 \leq \epsilon^{-2}_{\text{mix}} \widetilde{\rho}^i_{\text{mix}} \|\nu - \nu'\|_1,
\end{aligned}$$

since $Q_{j|i}$ are stochastic matrices and $\|\nu\|_2 \leq \|\nu\|_1 \leq 1$ for probability vectors. The second statement is readily derived by substituting $\nu(z_1) = p(z_1)$ and $\nu'(z_1) = p(z_1 \mid x^1_{-n})$. ■

D.2 Proof of Lemma 3

Recall the shorthand $\widetilde{\rho}_{\text{mix}} = 1 - \epsilon_{\text{mix}} \overline{\pi}_{\min}$. First observe that

$$\begin{aligned}
\sup_{z_i} |p(z_i \mid x^n_1) - p(z_i \mid x^{i+k}_{i-k})| &\leq |p(z_i|x^n_{i+1})p(z_i|x^i_1) - p(z_i|x^{i+k}_{i+1})p(z_i|x^i_{i-k+1})| \frac{p(x^n_{i+1})}{p(x^n_{i+1}|x^i_1)p(z_i)} \\
&\quad + |A-1| \frac{p(x^{i+k}_{i+1})}{p(x^{i+k}_{i+1}|x^i_{i-k+1})} \frac{1}{p(z_i)}
\end{aligned}$$

where $A = \frac{p(x^n_{i+1})}{p(x^n_{i+1}|x^i_1)} \left(\frac{p(x^{i+k}_{i+1})}{p(x^{i+k}_{i+1}|x^i_{i-k+1})} \right)^{-1}$. We bound the two terms in the sum separately.

From Lemma 13 we directly obtain the following upper bounds

$$\begin{aligned}
\sup_{z,x} |p(z_i \mid x^i_1) - p(z_i \mid x^i_{i-k+1})| &\leq \epsilon^{-2}_{\text{mix}} \widetilde{\rho}^{\min\{i,k\}}_{\text{mix}} \\
\sup_{z,x} |p(z_i \mid x^n_{i+1}) - p(z_i \mid x^{i+k}_{i+1})| &\leq \epsilon^{-2}_{\text{mix}} \widetilde{\rho}^{\max\{n-i,k\}}_{\text{mix}}
\end{aligned}$$

where the latter follows because of reversibility assumption (2) of the Markov chain. Inequality (72) can also be used to show that $\frac{p(x^n_{i+1})}{p(x^n_{i+1}|x^i_1)}, \frac{p(x^{i+k}_{i+1})}{p(x^{i+k}_{i+1}|x^i_{i-k+1})} \leq \epsilon^{-2}_{\text{mix}}$. A proof for a similar statement is given after inequality (80). The first term of the sum is therefore bounded

above by $2 \frac{\widetilde{\rho}^{\min\{i,n-i,k\}}_{\text{mix}}}{\pi_{\min} \epsilon^4_{\text{mix}}}$.

For the second term, we mainly need to bound $|A-1|$. In order to simplify the notation in the proof, we divide the sequence of values all observed variables in the window $i-k, i+k$ around index i, i.e. $x^{\max\{i+k,n\}}_{\min\{i-k,1\}}$, into four disjoint chunks and call them a, b, c, d in chronological order, explicitly defined as

$$a := x^{\max\{i-k,1\}}_{\min\{i-k,1\}} \quad b := x^{i}_{\max\{i-k,1\}+1} \quad c := x^{\min\{i+k,n\}}_{i+1} \quad d := x^{\max\{i+k,n\}}_{\min\{i+k,n\}+1}.$$

Note that the definition depends on whether $i-k > 1$ or $i+k < n$. Depending on the combination of $i+k < n$ and $i-k > 1$ being true or false,

$$A = \begin{cases} \frac{p(d|c)p(a|b)}{p(d|a,b,c)p(a|b,c)} & \text{if } i-k > 1,\ i+k < n \\ \frac{p(a|b)p(d|a,b,c)}{p(a|b,c,d)p(d|c)} & \text{if } i-k > 1,\ i+k > n \end{cases}.$$

For the other two possible cases, A is an inverse of the above. We demonstrate the main argument by looking into these two cases in more detail. Observe that the following inequality holds for the first case

$$|A-1| \leq \frac{|p(d \mid c) - p(d \mid a,b,c)|}{p(d \mid a,b,c)} \frac{p(a \mid b)}{p(a \mid b,c)} + \frac{|p(a \mid b) - p(a \mid b,c)|}{p(a \mid b,c)}$$

holds for all x. For the second case there is only an additional conditioning on d for the second term on the right hand side. In the inverse case that $i-k < 1, i+k > n$ we have

$$|A-1| \leq \frac{|p(d \mid c) - p(d \mid a,b,c)|}{p(d \mid c)} \frac{p(a \mid b,c)}{p(a \mid b)} + \frac{|p(a \mid b) - p(a \mid b,c)|}{p(a \mid b)}$$

and equivalently with an additional conditioning on d for $i-k < 1, i+k < n$. It is thus sufficient to consider $\sup_x |p(d \mid c) - p(d \mid a,b,c)|$ and $\sup_x |p(a \mid b) - p(a \mid b,c,d)|$. We see later that this is also the critical quantity to bound for the inverses.

First note that

$$\max\Big\{ \frac{p(a \mid b)}{p(a \mid b,c)}, \frac{p(a \mid b,c)}{p(a \mid b)} \Big\} \leq \epsilon^{-2}_{\text{mix}}. \tag{80}$$

For the first term we see that for all x we have

$$\begin{aligned}
\frac{p(a \mid b)}{p(a \mid b,c)} = \frac{p(x^\beta_\alpha \mid x^i_{\beta+1})}{p(x^\beta_\alpha \mid x^i_{\beta+1}, x^\gamma_{i+1})} &= \frac{\sum_{z_{\beta+1}} p(x^\beta_\alpha \mid z_{\beta+1}) p(z_{\beta+1} \mid x^i_{\beta+1})}{\sum_{z_{\beta+1}} p(x^\beta_\alpha \mid z_{\beta+1}) p(z_{\beta+1} \mid x^\gamma_{\beta+1})} \\
&\leq \frac{\sup_z p(x^\beta_\alpha \mid z_{\beta+1})}{\inf_z p(x^\beta_\alpha \mid z_{\beta+1})} \leq \epsilon^{-2}_{\text{mix}}
\end{aligned}$$

where the second inequality holds because of conditional independence of $x^i_{\beta+1}$ and x^β_α given $z_{\beta+1}$ in an HMM, and the last line holds because of inequality (72) and the fact that the Markov chain is invertible. Observe that the same arguments goes through for the inverse as well so that inequality (80) holds.

Let us now look at the rest of the terms involving differences. For the sake of simplification, let us introduce the shorthand notation

$$\alpha := \min\{i-k,1\}, \quad \beta := \max\{i-k,1\}, \quad \gamma := \min\{i+k,n\}, \quad \text{and} \quad \delta := \max\{i+k,n\}.$$

Then we can write $x_\alpha^\beta = a; x_{\beta+1}^i = b; x_{i+1}^\gamma = c; x_{\gamma+1}^\delta = d$ and bound $|p(d \mid c) - p(d \mid a,b,c)|$. Using Lemma 13 and inequality (72), we have the bound

$$\begin{aligned}
\frac{p(d \mid c) - p(d \mid a,b,c)}{p(d \mid a,b,c)} &= \frac{|p(x_{\gamma+1}^\delta | x_{i+1}^\gamma) - p(x_{\gamma+1}^\delta | x_\alpha^i, x_{i+1}^\gamma)|}{p(x_{\gamma+1}^\delta | x_\alpha^i, x_{i+1}^\gamma)} \\
&\leq \frac{\sum_{z_\gamma} p(x_{\gamma+1}^\delta \mid z_\gamma)|p(z_\gamma|x_{i+1}^\gamma) - p(z_\gamma|x_\alpha^i, x_{i+1}^\gamma)|}{\sum_{z_\gamma} p(x_{\gamma+1}^\delta|z_\gamma)p(z_\gamma|x_\alpha^\gamma)} \\
&\leq \frac{\sup_z p(x_{\gamma+1}^\delta|z_\gamma)}{\inf_z p(x_{\gamma+1}^\delta|z_\gamma)} \sum_{z_\gamma} |p(z_\gamma|x_{i+1}^\gamma) - p(z_\gamma|x_\alpha^\gamma)| \\
&\leq Cs\epsilon_{\text{mix}}^{-4}\tilde{\rho}_{\text{mix}}^{\gamma-i} = Cs\epsilon_{\text{mix}}^{-4}\tilde{\rho}_{\text{mix}}^{\min\{n-i,k\}}.
\end{aligned}$$

The same argument applies if the denominator is $p(d \mid c)$. Analogously, we have that

$$\frac{|p(a \mid b) - p(a \mid b,c)|}{p(a \mid b,c)} \leq Cs\epsilon_{\text{mix}}^{-4}\tilde{\rho}_{\text{mix}}^{i-\beta+1} = Cs\epsilon_{\text{mix}}^{-4}\tilde{\rho}_{\text{mix}}^{\min\{k,i\}}$$

and the same holds for the case when the denominator is $p(a \mid b)$ by inequality (80).

Note that the additional conditioning on d, does not change the result. Also, considering the inverses we see that the inequalities still hold.

Putting everything together now yields

$$|A - 1| \leq C's\epsilon_{\text{mix}}^{-6}\tilde{\rho}_{\text{mix}}^{\min\{n-i,i,k\}},$$

where C' is a generic constant and thus

$$\sup_{z_i} |p(z_i \mid x_1^n) - p(z_i \mid x_{i-k}^{i+k})| \leq C\frac{s\tilde{\rho}_{\text{mix}}^{\min\{i,n-i,k\}}}{\bar{\pi}_{\min}\epsilon_{\text{mix}}^8}.$$

D.3 Proof of Lemma 6

The latter inequality is valid in our particular case because

$$\begin{aligned}
|\operatorname{cov}(z_0, z_\ell \mid x_0, \ldots, x_k)| &= |\sum_{z_0,z_\ell} z_0 z_\ell p(z_\ell \mid z_0, x)p(z_0|x) - \sum_{z_0} z_0 p(z_0|x) \sum_{z_\ell} z_\ell p(z_\ell|x)| \\
&= |\sum_{z_0,z_\ell} z_0 z_\ell p(z_0|x)(p(z_\ell|z_0,x) - p(z_\ell \mid x))| \\
&\leq \sup_{z_\ell,z_0} |p(z_\ell|z_0,x) - p(z_\ell \mid x)| \sum_{z_0}\sum_{z_\ell} |z_0 z_\ell| p(z_0 \mid x)
\end{aligned}$$

Let us now show that $\sup_{z_\ell,z_0} \left|p(z_\ell \mid z_0,x) - p(z_\ell \mid x)\right| \leq \rho_{\text{mix}}^\ell$. Introducing the shorthand $\Delta(\ell) = p(z_\ell = 1 \mid z_0 = 1, x) - p(z_\ell = 1 \mid z_0 = -1, x)$, we first claim that

$$|\Delta(1)| \leq \rho_{\text{mix}} \tag{81}$$

To establish this fact, note that

$$\begin{aligned}
\Delta(1) &= \left|\frac{p(x \mid z_\ell = 1)}{p(x \mid z_{\ell-1} = 1)} p(z_\ell = 1 \mid z_{\ell-1} = 1) - \frac{p(x \mid z_\ell = 1)}{p(x \mid z_{\ell-1} = -1)} p(z_\ell = 1 \mid z_{\ell-1} = -1)\right| \\
&= \frac{ap}{ap + b(1-p)} - \frac{a(1-p)}{a(1-p) + bp} \\
&= \frac{ab}{(ap + b(1-p))(a(1-p) + bp)}(2p - 1)
\end{aligned}$$

where we write $a = p(x \mid z_\ell = 1)$ and $b = p(x \mid z_\ell = -1)$. The denominator is minimized at $p = 1$ so that inequality (81) is shown. The same argument shows that $|\Delta(-1)| \leq \rho_{\text{mix}}$.

Induction step: Assume that $\Delta(\ell - 1) \leq \rho_{\text{mix}}^{\ell-1}$. It then follows that

$$\begin{aligned}
&|p(z_\ell = 1 \mid z_0 = 1, x) - p(z_\ell = 1 \mid z_0 = -1, x)| \\
&= |\sum_{z_{\ell-1}} p(z_\ell = 1 \mid z_{\ell-1}, x)p(z_{\ell-1} \mid z_0 = 1, x) - p(z_\ell = 1 \mid z_{\ell-1}, x)p(z_{\ell-1}|z_0 = -1, x)| \\
&= \Delta(1)\Delta(\ell - 1) \leq \rho_{\text{mix}}^\ell
\end{aligned}$$

Since

$$p(z_\ell = 1 \mid z_0 = -1, x) - p(z_\ell = 1 \mid z_0 = 1, x) = -p(z_\ell = -1 \mid z_0 = -1, x) + p(z_\ell = -1 \mid z_0 = 1, x)$$

we use the shorthand $s = p(z_0 = 1 \mid x)$ to obtain

$$\begin{aligned}
&\sup_{z_\ell,z_0} |\, p(z_\ell \mid z_0, x) - p(z_\ell \mid x)| \\
&= \sup_{b_\ell,b_0} p(z_\ell = b_\ell \mid z_0 = b_0, x) - [(p(z_\ell = b_\ell \mid z_0 = 1, x)s + p(z_\ell = b_\ell \mid z_0 = -1, x)(1 - s)] \\
&\leq (1 - s)|\Delta(\ell)| \leq \rho_{\text{mix}}
\end{aligned}$$

which proves the bound for $\operatorname{cov}(Z_0, Z_1 \mid \gamma)$.

For the two state mixing we define $\widetilde{\Delta}(\ell) = p(z_\ell = 1 \mid z_1 z_0 = 1, x) - p(z_\ell = 1 \mid z_1 z_0 = -1, x)$ and can readily see that $|\widetilde{\Delta}(1)| \leq \rho_{\text{mix}}$ and

$$\begin{aligned}
&|p(z_{\ell+1}z_{\ell+2} = 1 \mid z_\ell z_{\ell-1} = 1, x) - p(z_{\ell+1}z_{\ell+2} = 1 \mid z_\ell z_{\ell-1} = -1, x)| \\
&= [p(z_{\ell+2} = 1 \mid z_{\ell+1} = 1, x) - p(z_{\ell+2} = -1 \mid z_{\ell+1} = -1, x)]\tilde{\Delta}(2)
\end{aligned}$$

Using equation (81), we obtain

$$|\widetilde{\Delta}(2)| = |p(z_1 = 1 \mid z_0 = 1, x) - p(z_1 = -1 \mid z_0 = -1, x)|\widetilde{\Delta}(1) \leq \rho_{\text{mix}} \tag{82}$$

from which it directly follows that

$$|p(z_{\ell+1}z_{\ell+2} = 1 \mid z_\ell z_{\ell-1} = 1, x) - p(z_{\ell+1}z_{\ell+2} = 1 \mid z_\ell z_{\ell-1} = -1, x)| \leq \rho_{\text{mix}}$$

The rest follows the same arguments as above and the bound for $\operatorname{cov}(Z_0 Z_1, Z_\ell Z_{\ell+1} \mid \gamma)$ in inequality (55) is shown.

Finally, the bound for $\operatorname{cov}(Z_0, Z_\ell Z_{\ell+1} \mid \gamma)$ in inequality (55) follows in a straightforward way using the relation (82) and induction with equation (81), as above.

References

S. Balakrishnan, M. J. Wainwright, and B. Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. *arXiv preprint arXiv:1408.2156*, 2014.

L.E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The Annals of Mathematical Statistics*, pages 1554–1563, 1966.

L.E. Baum, T. Petrie, G. Soules, and N. Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, pages 164–171, 1970.

M. Belkin and K. Sinha. Toward learning Gaussian mixtures with arbitrary separation. In *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 407–419, 2010.

P.J. Bickel, Y. Ritov, and T. Rydén. Asymptotic normality of the maximum-likelihood estimator for general Hidden Markov Models. *The Annals of Statistics*, 26(4):1614–1635, 08 1998. doi: 10.1214/aos/1024691255.

O. Cappé, E. Moulines, and T. Rydén. Hidden Markov Models, 2004.

A.T. Chaganty and P. Liang. Spectral experts for estimating mixtures of linear regressions. *arXiv preprint arXiv:1306.3729*, 2013.

S. Dasgupta. Learning mixtures of Gaussians. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 634–644, 1999.

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. B*, pages 1–38, 1977.

R. Durbin. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998. ISBN 9780521620413. URL `https://books.google.co.in/books?id=R5P2GlJvigQC`.

R.J. Elliott, L. Aggoun, and J.B. Moore. *Hidden Markov Models: Estimation and Control*. Applications of Mathematics. Springer, 1995. ISBN 9780387943640. URL `https://books.google.co.in/books?id=aR6-ASc_efQC`.

D. Hsu and S.M. Kakade. Learning mixtures of spherical Gaussians: Moment methods and spectral decompositions. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, pages 11–20, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1859-4. doi: 10.1145/2422436.2422439. URL `http://doi.acm.org/10.1145/2422436.2422439`.

D. Hsu, S. Kakade, and T. Zhang. A spectral algorithm for learning Hidden Markov Models. *J. Comput. Syst. Sci.*, 78(5):1460–1480, 2012.

C.J. Kim and C.R. Nelson. *State-space Models with Regime Switching: Classical and Gibbs-sampling Approaches with Applications*. MIT Press, 1999. ISBN 9780262112383. URL `https://books.google.co.in/books?id=eQFsQgAACAAJ`.

L. A. Kontorovich, B. Nadler, and R. Weiss. On learning parametric-output HMMs. In *Proc. 30th International Conference Machine Learning*, pages 702–710, June 2013.

F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Info. Theory*, 47(2):498–519, February 2001.

Michel Ledoux. On Talagrand's deviation inequalities for product measures. *ESAIM: Probability and statistics*, 1:63–87, 1997.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*, volume 23. Springer Science & Business Media, 2013.

A. Moitra and G. Valiant. Settling the polynomial learnability of mixtures of Gaussians. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 93–102, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-4244-7. doi: 10.1109/FOCS.2010.15. URL `http://dx.doi.org/10.1109/FOCS.2010.15`.

E. Mossel and S. Roch. Learning nonsingular phylogenies and Hidden Markov Models. *The Annals of Applied Probability*, 16(2):583–614, 05 2006.

A. Nobel and A. Dembo. A note on uniform laws of averages for dependent processes. *Statistics & Probability Letters*, 17(3):169–172, 1993.

L.R. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Pearson Education Signal Processing Series. Pearson Education, 1993. ISBN 9788129701381. URL `https://books.google.co.in/books?id=hoVLAAAACAAJ`.

S. M. Siddiqi, B. Boots, and G. J. Gordon. Reduced-rank Hidden Markov Models. In *Proc. 13th International Conference on Artificial Intelligence and Statistics*, 2010.

S. Terwijn. On the learnability of Hidden Markov Models. In *Proceedings of the 6th International Colloquium on Grammatical Inference: Algorithms and Applications*, ICGI '02, pages 261–268, London, UK, UK, 2002. Springer-Verlag.

Aad W. van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996.

R. van Handel. Hidden Markov Models. *Unpublished lecture notes*, 2008.

S. Vempala and G. Wang. A spectral algorithm for learning mixture models. *J. Comput. Syst. Sci.*, 68(4):841–860, June 2004. ISSN 0022-0000. doi: 10.1016/j.jcss.2003.11.008. URL `http://dx.doi.org/10.1016/j.jcss.2003.11.008`.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1—305, December 2008.

Zhaoran Wang, Quanquan Gu, Yang Ning, and Han Liu. High dimensional expectation-maximization algorithm: Statistical optimization and asymptotic normality. *arXiv preprint arXiv:1412.8729*, 2014.

J. C. Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, pages 95–103, 1983.

Xinyang Yi and Constantine Caramanis. Regularized EM algorithms: A unified framework and provable statistical guarantees. *arXiv preprint arXiv:1511.08551*, 2015.

B. Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, pages 94–116, 1994.

Journal of Machine Learning Research 18 (2017) 1-45 Submitted 7/16; Revised 10/17; Published 11/17

Online but Accurate Inference for Latent Variable Models with Local Gibbs Sampling

Christophe Dupuy CHIRSTOPHE.IH.DUPUY@GMAIL.COM
Technicolor
INRIA
Departement d'Informatique de l'ENS, Ecole normale suprieure, CNRS, PSL Research University
Paris, France.

Francis Bach FRANCIS.BACH@ENS.FR
INRIA
Departement d'Informatique de l'ENS, Ecole normale suprieure, CNRS, PSL Research University
Paris, France.

Editor: Francçis Caron

Abstract

We study parameter inference in large-scale latent variable models. We first propose a unified treatment of online inference for latent variable models from a non-canonical exponential family, and draw explicit links between several previously proposed frequentist or Bayesian methods. We then propose a novel inference method for the frequentist estimation of parameters, that adapts MCMC methods to online inference of latent variable models with the proper use of local Gibbs sampling. Then, for latent Dirichlet allocation,we provide an extensive set of experiments and comparisons with existing work, where our new approach outperforms all previously proposed methods. In particular, using Gibbs sampling for latent variable inference is superior to variational inference in terms of test log-likelihoods. Moreover, Bayesian inference through variational methods perform poorly, sometimes leading to worse fits with latent variables of higher dimensionality.

Keywords: Latent Variables Models, Online Learning, Gibbs Sampling, Topic Modelling, Latent Dirichlet Allocation

1. Introduction

Probabilistic graphical models provide general modelling tools for complex data, where it is natural to include assumptions on the data generating process by adding latent variables in the model. Such latent variable models are adapted to a wide variety of unsupervised learning tasks (Koller and Friedman, 2009; Murphy, 2012). In this paper, we focus on parameter inference in such latent variable models where the main operation needed for the standard expectation-maximization (EM) algorithm is intractable, namely dealing with conditional distributions over latent variables given the observed variables; latent Dirichlet allocation (LDA) (Blei et al., 2003) is our motivating example, but many hierarchical models exhibit this behavior, e.g., ICA with heavy-tailed priors. For such models, there exist two main classes of methods to deal efficiently with intractable exact inference in large-scale situations: sampling methods or variational methods.

Sampling methods can handle arbitrary distributions and lead to simple inference algorithms while converging to exact inference. However it may be slow to converge and non scalable to big datasets in practice. In particular, although efficient implementations have been developed, for example for LDA (Zhao et al., 2014; Yan et al., 2009), MCMC methods may not deal efficiently yet with continuous streams of data for our general class of models.

On the other hand, *variational inference* builds an approximate model for the posterior distribution over latent variables—called variational—and infer parameters of the true model through this approximation. The fitting of this variational distribution is formulated as an optimization problem where efficient (deterministic) iterative techniques such as gradient or coordinate ascent methods apply. This approach leads to scalable inference schemes (Hoffman et al., 2013), but due to approximations, there always remains a gap between the variational posterior and the true posterior distribution, inherent to algorithm design, and that will not vanish when the number of samples and the number of iterations increase.

Beyond the choice of approximate inference techniques for latent variables, parameter inference may be treated either from the *frequentist* point of view, e.g., using maximum likelihood inference, or a *Bayesian* point of view, where the posterior distribution of the parameter given the observed data is approximated. With massive numbers of observations, this posterior distribution is typically peaked around the maximum likelihood estimate, and the two inference frameworks should not differ much (Van der Vaart, 2000).

In this paper, we focus on methods that make a single pass over the data to estimate parameters. We make the following contributions:

1. We review and compare existing methods for online inference for latent variable models from a non-canonical exponential family in Section 2, and draw explicit links between several previously proposed frequentist or Bayesian methods. Given the large number of existing methods, our unifying framework allows to understand differences and similarities between all of them.

2. We propose in Section 3 a novel inference method for the frequentist estimation of parameters, that adapts MCMC methods to online inference of latent variable models with the proper use of "local" Gibbs sampling. In our online scheme, we apply Gibbs sampling to the current observation, which is "local", as opposed to "global" batch schemes where Gibbs sampling is applied to the entire dataset.

3. After formulating LDA as a non-canonical exponential family in Section 4, we provide an extensive set of experiments in Section 6, where our new approach outperforms all previously proposed methods. In particular, using Gibbs sampling for latent variable inference is superior to variational inference in terms of test log-likelihoods. Moreover, Bayesian inference through variational methods perform poorly, sometimes leading to worse fits with latent variables of higher dimensionality.

2. Online EM

We consider an *exponential family* model on random variables (X, h) with parameter $\eta \in \mathcal{E} \subseteq \mathbb{R}^d$ and with density (Lehmann and Casella, 1998):

$$p(X, h|\eta) = a(X, h) \exp\big[\langle \phi(\eta), S(X, h)\rangle - \psi(\eta)\big]. \tag{1}$$

We assume that h is hidden and X is observed. The vector $\phi(\eta) \in \mathbb{R}^d$ represents the natural parameter, $S(X, h) \in \mathbb{R}^d$ is the vector of sufficient statistics, $\psi(\eta)$ is the log-normalizer, and $a(X, h)$

©2017 Christophe Dupuy and Francis Bach.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-374.html.

is the underlying base measure. We consider a *non-canonical* family as in many models (such as LDA), the natural parameter $\phi(\eta)$ does not coincide with the model parameter η, that is, $\phi(\eta) \not\equiv \eta$; we however assume that ϕ is injective.

We consider N *i.i.d.* observations $(X_i)_{i=1,\dots,N}$ from a distribution $t(X)$, which may be of the form $P(X|\eta^*) = \int_h p(X,h|\eta^*)\mathrm{d}h$ for our model above and a certain $\eta^* \in \mathcal{E}$ (well-specified model) or not (misspecified model). Our goal is to obtain a predictive density $r(X)$ built from the data and using the model defined in (1), with the maximal expected log-likelihood $\mathbb{E}_{t(X)} \log r(X)$.

2.1 Maximum Likelihood Estimation

In the frequentist perpective, the predictive distribution $r(X)$ is of the form $p(X|\hat{\eta})$, for a well-defined estimator $\hat{\eta} \in \mathcal{E}$. The most common method is the EM algorithm (Dempster et al., 1977), which is an algorithm that aims at maximizing the likelihood of the observed data, that is,

$$\max_{\eta \in \mathcal{E}} \sum_{i=1}^{N} \log p(X_i|\eta). \tag{2}$$

More precisely, the EM algorithm is an iterative process to find the maximum likelihood (ML) estimate given observations $(X_i)_{i=1,\dots,N}$ associated to hidden variables $(h_i)_{i=1,\dots,N}$. It may be seen as the iterative construction of lower bounds of the log-likelihood function (Bishop, 2006). In the exponential family setting (1), we have, by Jensen's inequality, given the model defined by $\eta' \in \mathcal{E}$ from the previous iteration, and for any parameter $\eta \in \mathcal{E}$:

$$\begin{aligned}
\log p(X_i|\eta) &= \log \int p(X_i,h_i|\eta) dh_i \\
&\geq \int p(h_i|X_i,\eta') \log \frac{p(X_i,h_i|\eta)}{p(h_i|X_i,\eta')} dh_i \\
&= \int p(h_i|X_i,\eta') \left(\langle \phi(\eta), S(X_i,h_i) \rangle - \psi(\eta) \right) dh_i - C_i(\eta') \\
&= \langle \phi(\eta), \mathbb{E}_{p(h_i|X_i,\eta')} \left[S(X_i,h_i) \right] \rangle - \psi(\eta) - C_i(\eta'),
\end{aligned}$$

for a certain constant $C_i(\eta')$, with equality if $\eta' = \eta$. Thus, EM-type algorithms build locally tight lower bounds of the log-likelihood in (2), which are equal to

$$\langle \phi(\eta), \textstyle\sum_{i=1}^{N} s_i \rangle - N\psi(\eta) + \text{cst},$$

for appropriate values of $s_i \in \mathbb{R}^d$ obtained by computing conditional expectations with the distribution of h_i given X_i for the current model defined by η' (E-step), i.e., $s_i = \mathbb{E}_{p(h_i|X_i,\eta')} \left[S(X_i,h_i) \right]$. Then this function of η is maximized to obtain the next iterate (M-step). In standard EM applications, these two steps are assumed tractable. In Section 3, we will only assume that the M-step is tractable while the E-step is intractable.

Standard EM will consider $s_i = \mathbb{E}_{p(h_i|X_i,\eta')} \left[S(X_i,h) \right]$ for the previous value of the parameter η for all i, and hence, at every iteration, all observations X_i, $i = 1, \dots, N$ are considered for latent variable inference, leading to a slow "batch" algorithm for large N.

Incremental EM (Neal and Hinton, 1998) will only update a single element s_i coming from a single observation X_i and update the corresponding part of the sum $\sum_{j=1}^{N} s_j$ without changing

other elements. In the extreme case where a single pass over the data is made, then the M-step at iteration i maximizes

$$\langle \phi(\eta), \sum_{j=1}^{i} \mathbb{E}_{p(h_j|X_j,\eta_{j-1})} \left[S(X_j,h_j) \right] \rangle - i\psi(\eta),$$

with respect to η. In the next section, we provide a (known) other interpretation of this algorithm.

2.2 Stochastic Approximation

Given our frequentist objective $\mathbb{E}_{t(X)} \log p(X|\eta)$ to maximize defined as an expectation, we may consider two forms of stochastic approximation (Kushner and Yin, 2003), where observations X_i sampled from $t(X)$ are processed only once. The first one is stochastic gradient ascent, of the form

$$\eta_i = \eta_{i-1} + \rho_i \frac{\partial \log p(X_i|\eta)}{\partial \eta},$$

or appropriately renormalized version thereof, i.e., $\eta_i = \eta_{i-1} + \rho_i H^{-1} \frac{\partial \log p(X_i|\eta)}{\partial \eta}$, with several possibilities for the $d \times d$ matrix H, such as the negative Hessian of the partial or the full log-likelihood, or the negative covariance matrix of gradients, which can be seen as versions of natural gradient—see Titterington (1984); Delyon et al. (1999); Cappé and Moulines (2009). This either leads to slow convergence (without H) or expensive iterations (with H), with the added difficulty of choosing a proper scale and decay for the step-size ρ_i.

A key insight of Delyon et al. (1999); Cappé and Moulines (2009) is to use a different formulation of stochastic approximation, *not explicitly based on stochastic gradient ascent*. Indeed, they consider the stationary equation $\mathbb{E}_{t(X)} \left[\frac{\partial \log p(X|\eta)}{\partial \eta} \right] = 0$ and expand it using the exponential family model (1) as follows:

$$\begin{aligned}
\frac{\partial \log p(X|\eta)}{\partial \eta} &= \frac{\partial \log \int p(X,h|\eta) dh}{\partial \eta} \\
&= \phi'(\eta) \mathbb{E}_{p(h|X,\eta)} \left[S(X,h) \right] - \psi'(\eta).
\end{aligned}$$

Given standard properties of the exponential family, namely

$$\psi'(\eta) = \phi'(\eta) \mathbb{E}_{p(h,X|\eta)} \left[S(X,h) \right] \text{[1]},$$

and assuming invertibility of $\phi'(\eta)$, this leads to the following stationary equation:

$$\mathbb{E}_{t(X)} \left[\mathbb{E}_{p(h|X,\eta)} \left[S(X,h) \right] \right] = \mathbb{E}_{p(h,X|\eta)} \left[S(X,h) \right].$$

1. Proof: Given (1), $\int_{X,h} p(X,h|\eta) \mathrm{d}(X,h) = 1 \Rightarrow \psi(\eta) = \log \left[\int_{X,h} a(X,h) e^{\langle \phi(\eta), S(X,h) \rangle} \mathrm{d}(X,h) \right]$. We then derive this identity with respect to η, which gives:

$$\begin{aligned}
\psi'(\eta) &= \frac{\int_{X,h} \phi'(\eta) S(X,h) a(X,h) e^{\langle \phi(\eta), S(X,h) \rangle} \mathrm{d}(X,h)}{\int_{X,h} a(X,h) e^{\langle \phi(\eta), S(X,h) \rangle} \mathrm{d}(X,h)} \\
&= \frac{\phi'(\eta) \int_{X,h} S(X,h) a(X,h) e^{\langle \phi(\eta), S(X,h) \rangle} \mathrm{d}(X,h)}{e^{\psi(\eta)}} \\
&= \phi'(\eta) \int_{X,h} S(X,h) p(X,h|\eta) \mathrm{d}(X,h) \\
&= \phi'(\eta) \mathbb{E}_{p(h,X|\eta)} \left[S(X,h) \right].
\end{aligned}$$

This stationary equation states that at optimality the sufficient statitics have the same expectation for the full model $p(h, X|\eta)$ and the joint "model/data" distribution $t(X)p(h|X, \eta)$.

Another important insight of Delyon et al. (1999); Cappé and Moulines (2009) is to consider the change of variable $s(\eta) = \mathbb{E}_{p(h,X|\eta)}\left[S(X, h)\right]$ on sufficient statistics, which is equivalent to

$$\eta = \eta^*(s) \in \arg\max \langle \phi(\eta), s\rangle - \psi(\eta),$$

(which is the usual M-step update). See Cappé and Moulines (2009) for detailed assumptions allowing this inversion. We may then rewrite the equation above as

$$\mathbb{E}_{t(X)}\big(\mathbb{E}_{p(h|X,\eta^*(s))}\left[S(X, h)\right]\big) = s.$$

This is a non-linear equation in $s \in \mathbb{R}^d$, with an expectation with respect to $t(X)$ which is only accessed through i.i.d. samples X_i, and thus a good candidate for the Robbins-Monro algorithm to solve stationary equations (and not to minimize functions) (Kushner and Yin, 2003), which takes the simple form:

$$s_i = s_{i-1} - \rho_i\big(s_{i-1} - \mathbb{E}_{p(h_i|X_i,\eta^*(s_{i-1}))}\left[S(X_i, h_i)\right]\big),$$

with a step-size ρ_i. It may be rewritten as

$$\left\{ \begin{array}{rl} s_i &= (1-\rho_i)s_{i-1} + \rho_i \mathbb{E}_{p(h_i|X_i,\eta_{i-1})}[S(X_i, h_i)] \\ \eta_i &= \eta^*(s_i), \end{array} \right. \tag{3}$$

which has a particularly simple interpretation: instead of computing the expectation for all observations as in full EM, this stochastic version keeps tracks of old sufficient statistics through the variable s_{i-1} which is updated towards the current value $\mathbb{E}_{p(h_i|X_i,\eta_{i-1})}[S(X_i, h_i)]$. The parameter η is then updated to the value $\eta^*(s_i)$. Cappé and Moulines (2009) show that this update is asymptotically equivalent to the natural gradient update with three main improvements: (a) no matrix inversion is needed, (b) the algorithm may be accelerated through Polyak-Ruppert averaging (Polyak and Juditsky, 1992), i.e., using the average $\bar{\eta}_N$ of all η_i instead of the last iterate η_N, and (c) the step-size is particularly simple to set, as we are taking *convex* combinations of sufficient statistics, and hence only the decay rate of ρ_i has to be chosen, i.e., of the form $\rho_i = i^{-\kappa}$, for $\kappa \in (0, 1]$, without any multiplicative constant.

2.2.1 INCREMENTAL VIEW

For the specific stepsize $\rho_i = 1/i$, the online EM algorithm (3) corresponds exactly to the incremental EM presented above (Neal and Hinton, 1998), as then

$$s_i = \frac{1}{i}\sum_{j=1}^{i} \mathbb{E}_{p(h_j|X_j,\eta_{j-1})}[S(X_j, h_j)].$$

See Mairal (2014) for a detailed convergence analysis of incremental algorithms, in particular showing that step-sizes larger than $1/i$ are preferable (we observe this in practice in Section 6).

2.2.2 MONTE CARLO METHODS

There exist alternative methods to the EM algorithm based on Monte Carlo sampling to compute the maximum likelihood. For instance, the Monte Carlo EM method (MCEM) (Wei and Tanner, 1990) is a general Bayesian approach (i.e., η is a random variable) to approximate the maximizer of the posterior distribution $p(\eta|X, h)$ with Monte Carlo sampling. More precisely, in the MCEM method, similarly to EM, a surrogate function of the log-likelihood is used, given by:

$$Q(\eta, \eta_t) = \int_h \log[p(\eta|X, h)]p(h|X, \eta_t)\mathrm{d}h.$$

The function Q is approximated by sampling the latent variables h from the current conditional $p(h|X, \eta_t)$:

$$\hat{Q}(\eta, \eta_t) = \sum_{i=1}^{P} \log p(\eta|X, h^i),$$

where $(h^i)_{i=1,\dots,P}$ are the samples drawn from the conditional $p(h|X, \eta_t)$. The approximation $\hat{Q}$ is then maximize with respect to η. Note that this method is a batch method, namely, samples are drawn over all the dataset.

Other sequential Monte Carlo methods (SMC) use importance sampling to estimate the conditional distributions $p(h|X)$ using an auxiliary density function. This auxiliary distribution is then reweighted at each iteration. In our case where the conditional distribution $p(h|X)$ is intractable to compute, these methods fail to accurately approximate this distribution (Cappé et al., 2005; Kantas et al., 2015) and our Gibbs sampling scheme is more relevant. The SMC methods are also more adapted to models with dependency between observations.

The two Monte Carlo methods mentioned above also consist in sufficient statistics updates for the class of models considered here.

3. Online EM with Intractable Models

The online EM updates in (3) lead to a scalable algorithm for optimization when the local E-step is tractable. However, in many latent variable models—e.g., LDA, hierarchical Dirichlet processes (Teh et al., 2006), or ICA (Hyvärinen et al., 2004)—it is intractable to compute the conditional expectation $\mathbb{E}_{p(h|X,\eta)}[S(X, h)]$.

Following Rohde and Cappé (2011), we propose to leverage the scalability of online EM updates (3) and locally approximate the conditional distribution $p(h|X, \eta)$ in the case this distribution is intractable to compute. We will however consider different approximate methods, namely Gibbs sampling or variational inference. Our method is thus restricted to models where the hidden variable h may naturally be splitted in two or more groups of simple random variables. Our algorithm is described in Algorithm 1 and may be instantiated with two approximate inference schemes which we now describe.

Algorithm 1 Gibbs / Variational online EM

Input: η_0, s_0, $\kappa \in (0, 1]$.
for $i = 1, \dots, N$ **do**
- Collect observation X_i,
- Estimate $p(h_i|X_i, \eta_{i-1})$ with sampling (`G-OEM`) or variational inference (`V-OEM`),
- Apply (3) to sufficient statistics s_i and parameter η_i with $\rho_i = 1/i^\kappa$,

end for
Output: $\bar{\eta}_N = \frac{1}{N}\sum_{i=1}^N \eta_i$ or η_N.

3.1 Variational Inference: `V-OEM`

While variational inference had been considered before for online estimation of latent variable models, in particular for LDA for incremental EM (Sato et al., 2010), using it for online EM (which is empirically faster) had not been proposed and allows to use bigger step-sizes (e.g., $\kappa = 1/2$). These methods are based on maximizing the negative variational "free-energy"

$$\mathbb{E}_{q(h|\eta)}\left[\log \frac{p(X, h|\eta)}{q(h|\eta)}\right], \tag{4}$$

with respect to $q(h|\eta)$ having a certain factorized form adapted to the model at hand, so that efficient coordinate ascent may be used. See, e.g., Hoffman et al. (2013). We now denote online EM with variational approximation of the conditional distribution $p(h|X, \eta)$ as `V-OEM`.

3.2 Sampling Methods: `G-OEM`

MCMC methods to approximate the conditional distribution of latent variables with online EM have been considered by Rohde and Cappé (2011), who apply locally the Metropolis-Hasting (M-H) algorithm (Metropolis et al., 1953; Hastings, 1970), and show results on simple synthetic datasets. While Gibbs sampling is widely used for many models such as LDA due to its simplicity and lack of external parameters, M-H requires a proper proposal distribution with frequent acceptance and fast mixing, which may be hard to find in high dimensions. We provide a different simpler local scheme based on Gibbs sampling (thus adapted to a wide variety of models), and propose a thorough favorable comparison on synthetic and real datasets with existing methods.

The Gibbs sampler is used to estimate posterior distributions by alternatively sampling parts of the variables given the other ones (see Casella and George, 1992, for details), and is standard and easy to use in many common latent variable models. In the following, the online EM method with Gibbs estimation of the conditional distribution $p(h|X, \eta)$ is denoted `G-OEM`.

As mentioned above, the online EM updates correspond to a stochastic approximation algorithm and thus are robust to random noise in the local E-step. As a result, our sampling method is particularly adapted as it is a random estimate of the E-step—see a theoretical analysis by Rohde and Cappé (2011), and thus we only need to compute a few Gibbs samples for the estimation of $p(h|X_i, \eta_{i-1})$. A key contribution of our paper is to reuse sampling techniques that have proved competitive in the batch set-up and to compare them to existing variational approaches.

3.3 "Boosted" Inference

As the variational and MCMC estimations of $p(h|X_i, \eta_{i-1})$ are done with iterative methods, we can boost the inference of Algorithm 1 by applying the update in the parameter η in (3) after each iteration of the estimation of $p(h|X_i, \eta_{i-1})$. In the context of LDA, this was proposed by Sato et al. (2010) for incremental EM and we extend it to all versions of online EM. With this boost, we expect that the global parameters η converge faster, as they are updated more often. In the following, we denote by `G-OEM++` (resp. `V-OEM++`) the method `G-OEM` (resp. `V-OEM`) augmented with this boost.

3.4 Variational Bayesian Estimation

In the Bayesian perspective where η is seen as a random variable, we either consider a distribution based on model averaging, e.g., $r(X) = \int p(X|\eta)q(\eta)d\eta$ where $q(\eta) \propto \prod_{i=1}^N p(X_i|\eta)p(\eta)$ is the posterior distribution, or

$$r(X) = p(X|\bar{\eta}),$$

where $\bar{\eta}$ is the summary (e.g., the mean) of the posterior distribution $q(\eta)$, or of an approximation, which is usually done in practice (see, e.g., Hoffman and Blei, 2015) and is asymptotically equivalent when N tends to infinity.

The main problem is that, *even when the conditional distribution of latent variables is tractable*, it is intractable to manipulate the *joint* posterior distribution over the latent variables $h_1, \dots, h_N$, and the parameter η. Variational inference techniques consider an approximation where hidden variables are independent of the parameter η, i.e., such that

$$p(\eta, h_1, \dots, h_N|X_1, \dots, X_N) \approx q(\eta)\prod_{i=1}^N q(h_i),$$

which corresponds to the maximization of the following lower bound—called Evidence Lower BOund (ELBO)—on the log-likelihood $\log p(X_1, \dots, X_n)$ (Bishop, 2006):

$$\int q(\eta)\prod_{i=1}^N q(h_i)\log\frac{p(\eta)\prod_{i=1}^n p(X_i, h_i|\eta)}{q(\eta)\prod_{i=1}^N q(h_i)}d\eta dh_1\cdots dh_N.$$

The key insight from Hoffman et al. (2010); Broderick et al. (2013) is to consider the variational distribution $q(\eta)$ as the global parameter, and the cost function above as a sum of *local* functions that depend on the data X_i and the variational distribution $q(h_i)$. Once the local variational distribution $q(h_i)$ is maximized out, the sum structure may be leveraged in similar ways than for frequentist estimation, either by direct (natural) stochastic gradient (Hoffman et al., 2010) or incremental techniques that accumulate sufficient statistics (Broderick et al., 2013). A nice feature of these techniques is that they extend directly to models with intractable latent variable inference, by making additional assumptions on $q(h_i)$ (see for example the LDA situation in Section 4).

In terms of actual updates, they are similar to online EM in Section 3.1, with a few changes, but which turn out to lead to significant differences in practice. The similarity comes from the expansion of the ELBO as

$$\mathbb{E}_{q(\eta)}\Big[\sum_{i=1}^N \mathbb{E}_{q(h_i)}\log\frac{p(X_i, h_i|\eta)}{q(h_i)}\Big] + \mathbb{E}_{q(\eta)}\left[\log\frac{p(\eta)}{q(\eta)}\right].$$

The left hand side has the same structure than the variational EM update in (4), thus leading to similar updates, while the right hand side corresponds to the "Bayesian layer", and the maximization with respect to $q(\eta)$ is similar to the M-step of EM (where η is seen as a parameter).

Like online EM techniques presented in Section 3, approximate inference for latent variable is used, but, when using Bayesian stochastic variational inference techniques, there are two additional sources of inefficiencies: (a) extra assumptions regarding the independence of η and $h_1, \dots, h_N$, and (b) the lack of explicit formulation as the minimization of an expectation, which prevents the simple use of the most efficient stochastic approximation techniques (together with their guarantees). While (b) can simply slow down the algorithm, (a) may lead to results which are far away from exact inference, even for large numbers of samples (see examples in Section 6).

Beyond variational inference, Gibbs sampling has been recently considered by Gao et al. (2016): their method consists in sampling hidden variables for the current document given current parameters, but (a) only some of the new parameters are updated by incrementally aggregating the samples of the current document with current parameters, and (b) the method is slower than `G-OEM` (see Section 6).

4. Application to LDA

LDA (Blei et al., 2003) is a probabilistic model that infers hidden topics given a text corpus where each document of the corpus can be represented as topic probabilities. In particular, the assumption behind LDA is that each document is generated from a mixture of topics and the model infers the hidden topics and the topic proportions of each document. In practice, inference is done using Bayesian variational EM (Blei et al., 2003), Gibbs sampling (Griffiths and Steyvers, 2004; Wallach, 2006) or stochastic variational inference (Hoffman et al., 2010; Broderick et al., 2013; Sato et al., 2010).

4.0.1 HIERARCHICAL PROBABILISTIC MODEL.

Let $\mathcal{C} = \{X_1, \dots, X_D\}$ be a corpus of D documents, V the number of words in our vocabulary and K the number of latent topics in the corpus. Each topic β^k corresponds to a discrete distribution on the V words (that is an element of the simplex in V dimensions). A hidden discrete distribution θ_i over the K topics (that is an element of the simplex in K dimensions) is attached to each document X_i. LDA is a generative model applied to a corpus of text documents which assumes that each word of the ith document X_i is generated as follows:

- Choose $\theta_i \sim \text{Dirichlet}(\alpha)$,
- For each word $x_n \in X_i = \left(x_1, \dots, x_{N_{X_i}}\right)$:
  - Choose a topic $z_n \sim \text{Multinomial}(\theta_i)$,
  - Choose a word $x_n \sim \text{Multinomial}(\beta^{z_n})$.

In our settings, an observation is a document $X_i = (x_1, \dots, x_{N_{X_i}})$ where for all $1 \leq n \leq N_{X_i}$, $x_n \in \{0,1\}^V$ and $\sum_{v=1}^V x_{nv} = 1$. Each observation X_i is associated with the hidden variables h_i, with $h_i \equiv (Z_i = (z_1, \dots, z_{N_{X_i}}), \theta_i)$. The vector θ_i represents the topic proportions of document X_i and Z_i is the vector of topic assignments of each word of X_i. The variable h_i is local, i.e., attached

to one observation X_i. The parameters of the model are global, represented by $\eta \equiv (\beta, \alpha)$, where β represents the topic matrix and α represents the Dirichlet prior on topic proportions.

We derive the LDA model in Section 4.1 to find ϕ, S, ψ and a such that the joint probability $p(Z, \theta | X, \alpha, \beta)$ is in a non-canonical exponential family (1).

We may then readily apply all algorithms from Section 3 by estimating the conditional expectation $\mathbb{E}_{Z,\theta|X,\alpha,\beta}[S(X, Z, \theta)]$ with either variational inference (`V-OEM`) or Gibbs sampling (`G-OEM`). See Sections 4.2 and 4.3 for online EM derivations. Note that the key difficulty of LDA is the presence of two interacting hidden variables Z and θ.

4.1 LDA and Exponential Families

An observation X is a document of length N_X, where $X = (x_1, \dots, x_{N_X})$, each word is represented by $x_n \in \{0,1\}^V$ with $\sum_{v=1}^V x_{nv} = 1$. Our corpus $\mathcal{C}$ is a set of D observations $\mathcal{C} = (X^1, \dots, X^D)$. For each document X^i a hidden variable θ^i is associated, corresponding to the topic distribution of document X^i. For each word x_n of document X^i a hidden variable $z_n \in \{0,1\}^K$ is attached, corresponding to the topic assignment of word x_n. We want to find ϕ, S, ψ and a such that, the joint probability is in the exponential family (1):

$$p(X, Z, \theta | \beta, \alpha) = a(X, Z, \theta) \exp\left[\langle \phi(\beta, \alpha), S(X, Z, \theta) \rangle - \psi(\beta, \alpha)\right],$$

given an observation X and hidden variables Z and θ. For the LDA model, we have:

$$p(X, Z, \theta | \beta, \alpha) = \prod_{n=1}^{N_X} p(x_n | z_n, \beta) p(z_n | \theta) p(\theta | \alpha) = \prod_{n=1}^{N_X} \prod_{k=1}^{K} \prod_{v=1}^{V} \left[(\beta_v^k)^{x_{nv}} \theta_k\right]^{z_{nk}} p(\theta | \alpha),$$

which we can expand as:

$$p(X, Z, \theta | \beta, \alpha) = \exp\left[\sum_{n=1}^{N_X} \sum_{k=1}^{K} z_{nk} \log \theta_k\right] \times \exp\left[\sum_{n=1}^{N_X} \sum_{k=1}^{K} \sum_{v=1}^{V} x_{nv} z_{nk} \log \beta_v^k\right] \times \exp\left[\sum_{k=1}^{K} (\alpha_k - 1) \log \theta_k + B(\alpha)\right],$$

with $B(\alpha) = \log\left[\Gamma\left(\sum_{i=1}^K \alpha_i\right)\right] - \sum_{i=1}^K \log[\Gamma(\alpha_i)]$, where Γ is the gamma function. We deduce the non-canonical exponential family setting ϕ, S, ψ a:

$$S(X, Z, \theta) = \begin{pmatrix} S_{kv}^1 \equiv \left[\sum_{n=1}^{N_X} z_{nk} x_{nv}\right]_{kv} \\ S_k^2 \equiv [\log \theta_k]_k \end{pmatrix}, \tag{5}$$

$$\phi(\beta, \alpha) = \begin{pmatrix} \phi_{kv}^1 \equiv \left[\log \beta_v^k\right]_{kv} \\ \phi_k^2 \equiv [\alpha_k]_k \end{pmatrix}, \tag{6}$$

with $S^1, \phi^1 \in \mathbb{R}^{K\times V}$ and $S^2, \phi^2 \in \mathbb{R}^K$,

$$\psi(\beta,\alpha) = \sum_{i=1}^{K} \log[\Gamma(\alpha_i)] - \log\left[\Gamma\left(\sum_{i=1}^{K}\alpha_i\right)\right], \tag{7}$$

and

$$a(X,Z,\theta) = \exp\left[\sum_{k=1}^{K}\left(\sum_{n=1}^{N_X} z_{nk} - 1\right)\log\theta_k\right].$$

The one-to one mapping between the sufficient statistics $s = \binom{s^1}{s^2}$ and (β,α) is defined by:

$$(\beta,\alpha)^*[s] = \begin{cases} \arg\max_{\beta\geq 0,\alpha\geq 0} & \langle\phi(\beta,\alpha), s\rangle - \psi(\beta,\alpha) \\ \text{s.t.} & \beta^\top \mathbf{1} = \mathbf{1}, \end{cases}$$

where $\mathbf{1}$ denotes the vector whose all entries equal 1. The objective $\langle\phi(\beta,\alpha), s\rangle - \psi(\beta,\alpha)$ is concave in β from the concavity of log and concave in any α_k for $\alpha \geq 0$ as the function $B(\alpha)$ is concave as the negative log-partition of the Dirichlet distribution. We use the Lagrangian method for β:

$$L(\beta,\lambda) = \sum_{k=1}^{K}\sum_{v=1}^{V} s^1_{kv}\log\beta_v^k + \lambda^\top(\beta^\top\mathbf{1} - \mathbf{1}),$$

with $\lambda \in \mathbb{R}^K$. The derivative of L is set to zero when:

$$\forall(k,v), \frac{s^1_{kv}}{\beta_v^k} + \lambda_k = 0 \Rightarrow \lambda_k = -\sum_{v=1}^{V} s^1_{kv},$$

as $\sum_{v=1}^{V}\beta_v^k = 1$. We then have $(\beta^*(s))_{kv} = \frac{s^1_{kv}}{\sum_j s^1_{kj}}$. This mapping satisfies the constraint $\beta \geq 0$ because for any observation X and hidden variable Z, we have $S^1(X,Z)_{kv} \geq 0$. This comes from (5) and the fact that $\forall(n,k,v), (x_{nv}, z_{nk}) \in \{0,1\}^2$. We find the condition on α by setting the derivatives to 0, which gives $\forall k \in [\![1,K]\!]$:

$$s_k^2 - \Psi([\alpha^*(s)]_k) + \Psi\left(\sum_{i=1}^{K}[\alpha^*(s)]_i\right) = 0,$$

where $\Psi : x \mapsto \frac{\partial}{\partial x}[\log\Gamma](x)$ is the digamma function. Finally, $(\alpha^*(s), \beta^*(s))$ satisfies $\forall(k,v)$:

$$\begin{cases} (\beta^*(s))_{kv} & \equiv & \left[\frac{s^1_{kv}}{\sum_j s^1_{kj}}\right]_{kv} \\ \Psi([\alpha^*(s)]_k) - \Psi\left(\sum_{i=1}^{K}[\alpha^*(s)]_i\right) & = & s_k^2. \end{cases} \tag{8}$$

The parameter α^* is usually estimated with gradient ascent (Blei et al., 2003; Hoffman et al., 2010). We can also estimate α with the fixed point iteration (Minka, 2000) which consists in repeating the following update until convergence:

$$\alpha_k^{new} = \Psi^{-1}\left(\Psi\left(\sum_{i=1}^{K}\alpha_i^{old}\right) + s_k^2\right).$$

We use the fixed point iteration to estimate α^* as it is more stable in practice. We study different updates for α in Appendix C.

We can now apply Algorithm 1 to LDA. The only missing step is the estimation of the conditional expectation $\mathbb{E}_{Z,\theta|X,\alpha_t,\beta_t}[S(X,Z,\theta)]$, with $X = (x_1,\dots,x_{N_X})$ and $Z = (z_1,\dots,z_{N_X})$. We explain how to approximate this expectation with variational inference and Gibbs sampling.

4.2 Variational Online EM Applied to LDA (V−OEM)

In this section we explain how to approximate $\mathbb{E}_{Z,\theta|X,\alpha_t,\beta_t}[S(X,Z,\theta)]$ with variational inference, in the frequentist setting. See Hoffman et al. (2013) for detailed derivations of variational inference for LDA in the Bayesian setting (from which the updates in the frequentist setting may be easily obtained). The idea behind variational inference is to maximize the Evidence Lower BOund (ELBO), a lower bound on the probability of the observations:

$$p(X) \geq \text{ELBO}(X,p,q),$$

where q represents the variational model. In the case of LDA, the variational model is often set with a Dirichlet(γ) prior on θ and a multinomial prior on Z (Hoffman et al., 2013):

$$q(Z,\theta) = q(\theta|\gamma)\prod_{n=1}^{N_X} q(z_n|\zeta_n). \tag{9}$$

We then maximize the ELBO with respect to γ and ζ, which is equivalent to minimizing the Kullback-Leibler (KL) divergence between the variational posterior and the true posterior:

$$\max_{\gamma,\zeta}\ \text{ELBO}(X,p,q) \Leftrightarrow \min_{\gamma,\zeta}\ \text{KL}[p(Z,\theta|X)||q(\theta,Z)]. \tag{10}$$

We solve this problem with block coordinate descent, which leads to iteratively updating γ and ζ as follows:

$$\zeta_{nk} \propto \prod_{v=1}^{V}\left(\beta_v^k\right)^{x_{nv}}\exp\left[\Psi(\gamma_k)\right], \tag{11}$$

$$\gamma_k = \alpha_k + \textstyle\sum_{n=1}^{N_X}\zeta_{nk}. \tag{12}$$

We then approximate $\mathbb{E}_{Z,\theta|X,\alpha_t,\beta_t}[S(X,Z,\theta)]$ with the variational posterior. Given (5) and (9), we have:

$$\mathbb{E}_{p(Z,\theta|X)}[S(X,Z,\theta)] \approx \mathbb{E}_{q(Z,\theta)}[S(X,Z,\theta)] = \begin{bmatrix} \left(\sum_{n=1}^{N_{X_{t+1}}}\zeta_{nk}x_{nv}\right)_{kv} \\ \left(\Psi(\gamma_k) - \Psi\left(\sum_{j=1}^{K}\gamma_j\right)\right)_k \end{bmatrix}. \tag{13}$$

The variational approximation of $\mathbb{E}_{p(Z,\theta|X)}[S(X,Z,\theta)]$ is then done in two steps:

1. Iteratively update ζ with (11) and γ with (12),
2. $\mathbb{E}_{p(Z,\theta|X)}[S(X,Z,\theta)] \leftarrow \mathbb{E}_{q(Z,\theta|\gamma,\zeta)}[S(X,Z,\theta)]$ with equation (13).

As γ and ζ are set to minimize the distance between the variational posterior and the true posterior (10) we expect that this approximation is close to the true expectation. However, as the variational model is a simplified version of the true model, there always remains a gap between the true posterior and the variational posterior.

4.3 Gibbs Online EM Applied to LDA (G-OEM)

In this section we explain how to approximate $\mathbb{E}_{Z,\theta|X,\alpha_t,\beta_t}[S(X,Z,\theta)]$ with Gibbs sampling.

4.3.1 EXPECTATION OF S^1.

Given (5), we have $\forall k \in [\![1,K]\!]$, $\forall v \in [\![1,V]\!]$:

$$\mathbb{E}_{Z,\theta|X,\alpha,\beta}\left[(S^1(X,Z))_{kv}\right] = \mathbb{E}_{Z,\theta|X,\alpha,\beta}\left[\sum_{n=1}^{N_X} z_{nk}x_{nv}\right]$$
$$= \sum_{n=1}^{N_X}\int_{Z,\theta} z_{nk}x_{nv}p(z_n,\theta|X,\beta,\alpha)\mathrm{d}\theta\mathrm{d}z = \sum_{n=1}^{N_X} x_{nv}p(z_{nk}=1|X,\beta,\alpha).$$

We see that we only need the probability of z, and can thus use collapsed Gibbs sampling (Griffiths and Steyvers, 2004). We have, following Bayes rule:

$$p(z_{nk}=1|z_{-n},X,\beta,\alpha) \propto p(x_n|z_{nk}=1,\beta)p(z_{nk}=1|z_{-n},\alpha),$$

where z_{-n} is the topic assignments except index n. In the LDA model, each word x_n is drawn from a multinomial with parameter β^{z_n}, which gives:

$$p(x_n|z_{nk}=1,\beta) = \sum_{v=1}^{V} x_{nv}\beta_v^k.$$

In the following, we use the notation $\beta_{x_n}^k \equiv \sum_{v=1}^V x_{nv}\beta_v^k$ for the sake of simplicty. We then use the fact that the topic proportions θ has a Dirichlet(α) prior, which implies that $Z|\alpha$ follows a Dirichlet-multinomial distribution (or multivariate Pólya distribution). As a result, the conditional distribution is:

$$p(z_{nk}=1|z_{-n},\alpha) = \frac{N_{-n,k}+\alpha_k}{(N_X-1)+\sum_j \alpha_j},$$

with $N_{-n,k}$ the number of words assigned to topic k in the current document, except index n. Finally, we have the following relation (Griffiths and Steyvers, 2004):

$$p(z_{nk}=1|z_{-n},X,\beta,\alpha) \propto \beta_{x_n}^k \times \frac{N_{-n,k}+\alpha_k}{(N_X-1)+\sum_j \alpha_j}. \tag{14}$$

We estimate $p(z_{nk}=1|X,\beta,\alpha)$ with Gibbs sampling by iteratively sampling topic assignments z_n for each word, as detailed in Algorithm 2. We average over the last quarter of samples to reduce noise in the final output. We then incorporate the output in Algorithm 1.

4.3.2 EXPECTATION OF S^2.

Given (5), we also have $\forall k \in [\![1,K]\!]$, $\forall v \in [\![1,V]\!]$:

$$\mathbb{E}_{Z,\theta|X,\alpha,\beta}[(S^2(X,Z))_k] = \mathbb{E}_{Z,\theta|X,\alpha,\beta}[\log\theta_k].$$

On the one hand, we have:

$$p(Z,\theta|X,\beta,\alpha) = p(Z|\theta,X,\beta,\alpha)p(\theta|X,\beta,\alpha) = C(\alpha)\prod_{k=1}^K \theta_k^{\left(\sum_{n=1}^{N_X} z_{nk}\right)+\alpha_k-1},$$

with $C(\alpha) = \frac{\Gamma\left(\sum_{i=1}^K \alpha_i\right)}{\prod_{i=1}^K \Gamma(\alpha_i)}$. On the other hand:

$$p(Z,\theta|X,\beta,\alpha) \propto p(\theta|Z,\alpha)p(Z|X,\beta).$$

We deduce from the two identities:

$$p(\theta|Z,\alpha) \propto \prod_{k=1}^K \theta_k^{\left(\sum_{n=1}^{N_X} z_{nk}\right)+\alpha_k-1} \Rightarrow \theta|Z,\alpha \sim \text{Dirichlet}\left(\alpha + \textstyle\sum_{n=1}^{N_X} z_n\right).$$

Finally, the expectation is:

$$\begin{aligned}\mathbb{E}_{Z,\theta|X,\alpha,\beta}[(S^2(X,Z))_k] =& \mathbb{E}_{Z,\theta|X,\alpha,\beta}[\log\theta_k]\\ =& \mathbb{E}_{Z|X,\beta,\alpha}\left[\mathbb{E}_{\theta|Z,\alpha}\left[\log\theta_k\right]\right]\\ =& \mathbb{E}_{Z|X,\beta,\alpha}\left[\Psi\left([\alpha(s)]_k + \textstyle\sum_{n=1}^{N_X} z_{nk}\right)\right] - \Psi\left(\sum_{i=1}^K [\alpha(s)]_i + N_X\right),\end{aligned}$$

as the distribution of $\theta|Z,\alpha$ is Dirichlet $\left(\alpha + \sum\limits_{n=1}^{N_X} z_n\right)$. We use the values of z sampled with Algorithm 2 to estimate this expectation. More precisely, keeping notations of Algorithm 2:

$$\mathbb{E}_{Z|X,\beta,\alpha}\left[\Psi\left([\alpha(s)]_k + \textstyle\sum_{n=1}^{N_X} z_{nk}\right)\right] \approx \frac{1}{P}\sum_{t=1}^P \Psi\left([\alpha(s)]_k + \sum_{n=1}^{N_X} z_{nk}^t\right).$$

4.4 Bayesian Approach

In a Bayesian setting, we consider β as a random variable, with $\beta \sim \text{Dirichlet}(b\mathbf{1})$, with $b \in \mathbb{R}$ and $\mathbf{1} \in \mathbb{R}^V$ denotes the vector whose all entries equal 1. The variational distribution of the global parameter β is then set to $q(\beta^k|\lambda^k) = \text{Dirichlet}(\lambda^k)$, with $\lambda^k \in \mathbb{R}^V$ $\forall k = 1,\dots,K$. The main difference with the frequentist methods above (G-OEM and V-OEM) is to optimize the ELBO with respect to the variational parameters $(\lambda^k)_k$. In practice, it is equivalent to replace β_v^k by $\exp\left[\mathbb{E}_q[\log\beta_v^k]\right]$ in all the updates above (i.e., in Equation (11) for V-OEM and in Equation (14) for G-OEM). The variational parmater $\lambda^k \in \mathbb{R}^V$ is updated with stochastic gradient on the ELBO, which gives, at iteration t:

$$\lambda^k(t+1) = \rho_t\lambda^k(t) + (1-\rho_t)\hat{\lambda}^k, \tag{15}$$

with $\hat{\lambda}^k \in \mathbb{R}^V$, $\hat{\lambda}_v^k = b + D\mathbb{E}_q\left[S_{kv}^1\right]$, where D is the total number of documents in the dataset and b is the prior on β^k (Hoffman et al., 2013).

Algorithm 2 Gibbs sampling scheme to approximate $p(z_{nk} = 1|X, \beta, \alpha)$

Input: β, α, X.

Initialization: $Z_n^0 \sim \text{Mult}([\bar{\beta}_{x_n}^k]_{k=1,\dots,K})$, with $\bar{\beta}_{x_n}^k = \frac{\beta_{x_n}^k}{\sum_j \beta_{x_n}^j}$ $\forall n \in [\![1, N_X]\!]$.

for $t = 1, 2, \dots, P$ **do**

Compute random permutation σ^t on $[\![1, N_X]\!]$,

for $n = 0, 1, \dots, N_X$ **do**

- Set $Z_{-n}^t = \left\{ (z_{\sigma^t(i)}^t)_{1 \leq i < n}, (z_{\sigma^t(i)}^{t-1})_{n < i \leq N_X} \right\}$,
- Compute $\forall k$, $p(z_{\sigma^t(n)k} = 1|Z_{-n}^t, X, \beta, \alpha)$ with Equation (14),
- Sample $z_{\sigma^t(n)}^t \sim \text{Mult}\left[p(z_{\sigma^t(n)}|Z_{-n}^t, X, \beta, \alpha)\right]$,

end for

end for

for $n = 0, 1, \dots, N_X$ **do**

Set $Z_{-n}^t = \left\{ (z_i^t)_{1 \leq i < n}, (z_i^t)_{n < i \leq N_X} \right\}$ for $t \geq \frac{3}{4}P$,

$p(z_{nk} = 1|X, \beta, \alpha) \leftarrow \frac{4}{P} \sum_{t=\frac{3}{4}P}^{P} p(z_{nk} = 1|Z_{-n}^t, X, \beta, \alpha)$

end for

Output: $\forall k, \forall n$: $(z_n^t)_{t=1,\dots,P}$, $p(z_{nk} = 1|X, \beta, \alpha)$.

5. Application to Hierarchical Dirichlet Process (HDP) (Teh et al., 2006)

The HDP model is a generative process to model documents from an infinite set of topics β_k, $k = 1, 2, 3, \dots$. Each topic is a discrete distribution of size V, the size of the vocabulary. Each topic is associated to a weight $\pi_k \in [0, 1]$, representing the importance of the topic in the corpus. For each document d, the (infinite) topic proportions ν_d are drawn from $\nu_d \sim \text{Dirichlet}(b\pi)$. We then generate words with a similar scheme to LDA scheme. More formally a corpus is generated as follows:

1. Draw an infinite number of topics $\beta_k \sim \text{Dirichlet}(\eta)$, for $k \in \{1, 2, 3, \dots\}$;
2. Draw corpus breaking proportions $\bar{\pi}_k \sim \text{Beta}(1, \alpha)$, for $k = 1, 2, 3, \dots$; with $\pi_k = \sigma_k(\bar{\pi})$;
3. For each document d:
   (a) Draw document-level topic proportions: $\nu_d \sim \text{Dirichlet}(b\pi)$;
   (b) For each word n in d:
      i. Draw topic assignment $z_{dn} \sim \text{Multinomial}(\nu_d)$;
      ii. Draw word $w_n \sim \text{Multinomial}(\beta_{z_{dn}})$.

In practice, we set the initial number of topics to $T = 2$. We then increase the number of topics used in the corpus using Gibbs sampling and $p(z_{dn} > T|X, \eta) \propto b(1 - \sum_{i=1}^{T} \pi_i)$. See Section 5.2 for details.

5.1 HDP and Exponential Families

We consider an exponential family model on random variables (X, h) with parameter $\eta \in \mathcal{E} \subseteq \mathbb{R}^d$ and with density:

$$p(X, h|\eta) = a(X, h) \exp\left[\langle \phi(\eta), S(X, h) \rangle - \Psi(\eta)\right].$$

In the case of HDP, an observation X is a document of length N_X, where $X = (x_1, \dots, x_{N_X})$, $x_n \in \{0, 1\}^V$ and $\sum_{v=1}^{V} x_{nv} = 1$. In the frequentist approach, the parameters of the model are global, represented by $\eta \equiv (\beta, \pi)$, where β represents the corpus topics, π represents the corpus breaking proportions. Our corpus $\mathcal{C}$ is a set of D observations $\mathcal{C} = (X^1, \dots, X^D)$. For each document X^d, the associated hidden variables are $\nu_d \in [0, 1]^K$ corresponding to document-level topic proportions. For each word x_n of document X^d, a hidden variable $z_n \in \{0, 1\}^T$ is attached, corresponding to the topic assignment of word x_n.

We want to find ϕ, S, ψ and a such that, the joint probability is in the exponential family:

$$p(X, Z, \nu|\beta, \pi) = a(X, Z, \nu) \exp\left[\langle \phi(\beta, \pi), S(X, Z, \nu) \rangle - \psi(\beta, \pi)\right],$$

given an observation X and hidden variables Z and ν. For the HDP model, we have:

$$\begin{aligned}
p(X, Z, \nu|\beta, \pi) =& p(\nu|\pi) \prod_{n=1}^{N_X} p(x_n|z_n, \beta) p(z_n|\nu) \\
=& W(\pi) \prod_{k \in \mathbb{N}^*} (\nu_k)^{b\pi_k - 1} \prod_{n=1}^{N_X} \prod_k (\nu_k)^{z_{nk}} \prod_v (\beta_{k,v})^{x_{nv} z_{nk}} \\
=& \exp\left[-\psi(\pi)\right] \exp\left[\sum_k \log \nu_k \left(\sum_{n=1}^{N_X} z_{nk} - 1 \right)\right] \\
& \times \exp\left[\sum_k (b\pi_k) \log \nu_k\right] \\
& \times \exp\left[\sum_{k,v} \log \beta_{k,v} \sum_{n=1}^{N_X} x_{nv} z_{nk}\right],
\end{aligned}$$

with $\psi(\pi) = \sum_k \log \Gamma(b\pi_k) - \log \Gamma(b)$.We deduce the exponential family setting ϕ, S, a:

$$S(X, Z, \nu) = \begin{pmatrix} S_k^1 \equiv [\log \nu_k]_k \\ S_{kv}^2 \equiv \left[\sum_{n=1}^{N_X} z_{nk} x_{nv} \right]_{kv} \end{pmatrix}, \tag{16}$$

$$\phi(\beta, \pi) = \begin{pmatrix} \phi_k^1 \equiv [b\pi_k]_k \\ \phi_{kv}^2 \equiv [\log \beta_{k,v}]_{kv} \end{pmatrix}, \tag{17}$$

with

$$a(X, Z, \nu) = \exp\left[\sum_k \log \nu_k \left(\sum_{n=1}^{N_X} z_{nk} - 1 \right)\right].$$

The one-to one mapping between the sufficient statistics $s = (s^1, s^2)^\top$ and (β, π) is defined by:

$$(\beta, \pi)^*[s] = \begin{cases} \arg\max_{\beta \geq 0, 1 \geq \pi \geq 0} & \langle \phi(\beta, \pi), s \rangle - \psi(\beta, \pi) \\ \text{s.t.} & \beta^\top \mathbf{1} = \mathbf{1}, \end{cases}$$

where $\mathbf{1}$ denotes the vector whose all entries equal 1.

With the same computation than LDA, $\beta^*(s)_{kv} \equiv \left[\frac{s^2_{kv}}{\sum_j s^2_{kj}}\right]$. We find $\pi^*(s)$ by solving:

$$\pi^*(s) = \arg\max_{1 \geq \pi \geq 0} \sum_{k=1}^{K} \left(b\pi_k s^1_k - \log\Gamma(b\pi_k)\right) + \log\Gamma(b\sum_k \pi_k),$$

which gives:

$$\Psi(b\pi^*(s)_k) - \Psi\left(b\sum_i \pi^*(s)_i\right) = s^1_k.$$

where $\Psi : x \mapsto \frac{\partial}{\partial x}\left[\log\Gamma\right](x)$ is the digamma function. We estimate $(b\pi)^*$ with the fixed point iteration which consists in repeating the following update until convergence:

$$(b\pi)^{new}_k = \Psi^{-1}\left(\Psi\left(\sum_i (b\pi_i)^{old}\right) + s^1_k\right).$$

Finally, $(\beta, \pi)^*[s]$ satisfies $\forall (k, v)$:

$$\boxed{\begin{cases} (\beta^*(s))_{kv} & = & \left[\frac{s^2_{kv}}{\sum_j s^2_{kj}}\right] \\ \Psi(b\pi^*(s)_k) - \Psi\left(b\sum_i \pi^*(s)_i\right) & = & s^1_k. \end{cases}}$$

5.2 Inference with Online EM

In this section, we explain how to approximate $\mathbb{E}_{Z,\nu|X,\eta}[S(X, Z, \nu)]$ with Gibbs sampling from a frequentist and a Bayesian perspective. In particular, as the total number of topics is infinite, we need to keep track of the previously used topics and iteratively extend the number of topics considered.

5.2.1 GIBBS ONLINE EM (G-OEM)

In our frequentist G-OEM approach, η is a parameter. The Gibbs sampling scheme to approximate $\mathbb{E}_{Z,\nu|X,\eta}[S(X, Z, \nu)]$ is different from LDA and a probability of adding a new topic to the current list is computed at each iteration, as explained below.

5.2.2 EXPECTATION OF S^1.

We have:

$$\begin{aligned} \mathbb{E}_{Z,\nu|X,\eta}[S^1(X, Z, \nu)]_k &= \mathbb{E}_{Z,\nu|X,\eta}\left[\log\nu_k\right] \\ &= \mathbb{E}_{Z|X,\eta}\left[\Psi\left(b\pi_k + \sum_{n=1}^{N_X} z_{nk}\right)\right] - \Psi\left(b\sum_i \pi_i + N_X\right), \end{aligned}$$

and we use the values of z sampled with Gibbs sampling to compute:

$$\mathbb{E}_{Z|X,\eta}\left[\Psi\left(b\pi_k + \sum_{n=1}^{N_X} z_{nk}\right)\right] \approx \frac{1}{P}\sum_{t=1}^{P} \Psi\left(b\pi_k + \sum_{n=1}^{N_X} z^t_{nk}\right)$$

5.2.3 EXPECTATION OF S^2.

We have:

$$\mathbb{E}_{Z,\nu|X,\eta}[S^2(X, \nu)]_{kv} = \mathbb{E}_{Z,\nu|X,\eta}\left[\sum_{n=1}^{N_X} z_{nk}x_{nv}\right] = \sum_{n=1}^{N_X} x_{nv}p(z_{nk} = 1|X, \eta)$$

5.2.4 SAMPLING $z|X, \eta$.

If T is the current number of topics, we have:

$$\begin{aligned} \forall k \in \{1, \ldots, T\}, \;\; p(z_{nk} = 1|z^{-n}, X, \eta) &\propto (N^{-n}_k + b\pi_k) \times p(x_n|z_{ni} = 1, c_{ik} = 1, \eta) \\ &\propto (N^{-n}_k + b\pi_k) \times \beta_{k,x_n}, \end{aligned}$$

and the probability of sampling a new topic is given by:

$$p(z_n > T|z^{-n}, X, \eta) \propto b\left(1 - \sum_{t=1}^{T} \pi_k\right)/V.$$

When a new topic is generated, we initialize the probability π_{T+1} with $\bar{\pi}_{T+1} \sim \text{Beta}(1, \alpha)$ and $\pi_{T+1} = \bar{\pi}_{T+1}\prod_{t=1}^{T}(1 - \bar{\pi}_t)$.

5.2.5 BAYESIAN APPROACH: VARGIBBS (WANG AND BLEI, 2012)

In a Bayesian settings where $\beta^k \sim \text{Dirichlet}(\eta)$; $q(\beta^k|\lambda) = \text{Dirichlet}(\lambda^k)$ and $\pi_k \sim \text{Beta}(1, a)$; $q(\pi_k|a_k, b_k) = \text{Beta}(a_k, b_k)$, the sampling scheme is different as we also sample π and an auxiliary variable s_{dk} corresponding to the number of "tables" serving "dish" k in "restaurant" d (in the fomulation of HDP as a Chinese restaurant process; see Wang and Blei (2012) for details).
Sampling z:

$$p(z_{nk} = 1|z^{-n}, \lambda, \pi) \propto (N^{-n}_{dk} + b\pi_k)\frac{N^{-n}_{kx_n} + \lambda_{kx_n}}{N^{-n}_k + \sum_v \lambda_{kv}}.$$

Sampling s:

$$p(s_{dk}|N_{dk}, b\pi_k) = \frac{\Gamma(b\pi_k)}{\Gamma(b\pi_k + N_{dk})}S(N_{dk}, s_{dk})\,(b\pi_k)^{s_{dk}},$$

with $S(n, m)$ are unsigned Stirling number of the first kind.
Sampling π:

$$p(\bar{\pi}_k) \propto \bar{\pi}_k^{a_k - 1 + \sum_{d \in \mathbb{S}} s_{dk}}\,(1 - \bar{\pi}_k)^{b_k - \alpha + \sum_{d \in \mathbb{S}} \sum_{j=k+1}^{\infty} s_{dj}}$$

We then set:

$$\begin{cases} \hat{\lambda}_{kv} = \eta + D \sum_{n=1}^{N_X} z_{nk} x_{nv} \\ \hat{a}_k = 1 + D s_{dk} \\ \hat{b}_k = \alpha + D \sum_{j=k+1}^{\infty} s_{dj} \end{cases} \tag{18}$$

and $\lambda^{t+1} = (1-\rho_t)\lambda^t + \rho_t \hat{\lambda}$; $a^{t+1} = (1-\rho_t)a^t + \rho_t \hat{a}$; $b^{t+1} = (1-\rho_t)b^t + \rho_t \hat{b}$.

In practice, for each document we sample the hidden variables z for each word and compute the topic counts N_{dk} for topic k in document d, then we sample the variable s. Finally, we perform the online EM algorithm by making the approximation $\mathbb{E}_{p(h|X,\eta)}[S(X,h)] \approx \mathbb{E}_{q(h)}[S(X,h)]$, which corresponds to equation (18). Note that in this Bayesian approach, the parameters (λ, a, b) represent the distribution parameters of the random variables β and π.

6. Evaluation

We evaluate our method by computing the likelihood on held-out documents, that is $p(X|\beta,\alpha)$ for any test document X. For LDA, the likelihood is intractable to compute. We approximate $p(X|\beta,\alpha)$ with the "left-to-right" evaluation algorithm (Wallach et al., 2009) applied to each test document. This algorithm is a mix of particle filtering and Gibbs sampling. On any experiments, this leads essentially to the same log-likelihood than Gibbs sampling with sufficiently enough samples—e.g., 200. In the following, we present results in terms of log-perplexity, defined as the opposite of the log-likelihood $-\log p(X|\eta)$. The lower the log-perplexity, the better the corresponding model. In our experiments, we compute the average test log-perplexity on N_t documents. We compare eight different methods:

- `G-OEM` (our main algorithm): Gibbs online EM. Online EM algorithm with Gibbs estimation of the conditional distribution $p(h|X,\eta)$ (Algorithm 2). Frequentist approach and step-size $\rho_i = 1/\sqrt{i}$;
- `V-OEM++`: variational online EM (also a new algorithm). Online EM algorithm with variational estimation of the conditional distribution $p(h|X,\eta)$, augmented with inference boosting from Section 3.3. Frequentist approach and step-size $\rho_i = 1/\sqrt{i}$;
- `OLDA`: online LDA (Hoffman et al., 2010). Bayesian approach which maximizes the ELBO from Section 3.4, with natural stochastic gradient ascent and a step-size $\rho_i = 1/\sqrt{i}$;
- `VarGibbs`: Sparse stochastic inference for LDA (Mimno et al., 2012). This method also maximizes the ELBO but estimates the variational expectations $q(Z,\theta)$ with Gibbs sampling instead of iterative maximization of variational parameters—see Section 4.2;
- `SVB`: streaming variational Bayes (Broderick et al., 2013). A variational Bayesian equivalent of `V-OEM` with step-size $\rho_i = 1/i$;

- `SPLDA`: single pass LDA (Sato et al., 2010). The difference with `V-OEM++` is that $\rho_i = 1/i$ and the updates in α done with a Gamma prior (see Appendix C);
- `SGS`: streaming Gibbs sampling (Gao et al., 2016). This method is related to `G-OEM` with $\rho_i = 1/i$. In this method, α is not optimized and set to a constant C_α. For comparison purposes, for each dataset, we set C_α to be the averaged final parameter $\hat{\alpha}$ obtained with `G-OEM` on the same dataset: $C_\alpha = \frac{1}{K}\sum_k \hat{\alpha}_k$. For each observation, only the last Gibbs sample is considered, leading to extra noise in the output;
- `LDS`: Stochastic gradient Riemannian Langevin dynamics sampler (Patterson and Teh, 2013). The authors use the Langevin Monte Carlo methods on probability simplex and apply their online algorithm to LDA. For this method and only this method, we set to $P = 200$ the number of internal updates.

For existing variational methods—`OLDA`, `SVB`, `SPLDA`—β is a random variable with prior $q(\beta)$. We estimate the likelihood $p(X|\hat{\beta},\alpha)$ with the "left-to-right" algorithm by setting $\hat{\beta} = \mathbb{E}_q[\beta]$ for Bayesian methods. For simplicity, we only present our results obtained with `G-OEM` and `V-OEM++`. Indeed, the inference boost presented in Section 4 is only beneficial for `V-OEM`. A detailed analysis is presented in Appendix A.1.

6.1 Explicit Links for LDA

In this section, we propose to make the links between the methods listed above explicit, using the framework described in Section 4 for the particular LDA model. We present in Table 1 a summary of the compared method.

6.1.1 CATEGORY

In the frequentist approach, β is a parameter and is updated with Equation (8), as the "M-step" in online EM.

In a Bayesian setting, β is a random variable with prior $\beta \sim \text{Dirichlet}(b\mathbf{1})$, with $b \in \mathbb{R}$ and $\mathbf{1} \in \mathbb{R}^V$ denotes the vector whose all entries equal 1. The variational distribution of the global parameter β is then set to $q(\beta^k|\lambda^k) = \text{Dirichlet}(\lambda^k)$, with $\lambda^k \in \mathbb{R}^V \ \forall k = 1,\dots,K$. The variational parameter λ^k is updated by maximizing the ELBO with stochastic gradient ascent (Equation (15)).

	CATEGORY	$\mathbb{E}_{Z\|X,\eta}[S(X,Z)]$	STEP-SIZE ρ_t	UPDATE FOR α
G-OEM	FREQUENTIST	GIBBS SAMPLING	FREE	FIXED POINT
V-OEM	FREQUENTIST	VARIATIONAL	FREE	FIXED POINT
OLDA	BAYESIAN	VARIATIONAL	FREE	GRADIENT ASCENT
VARGIBBS	BAYESIAN	GIBBS SAMPLING	FREE	α FIXED
SVB	BAYESIAN	VARIATIONAL	FIXED: $1/t$	GRADIENT ASCENT
SPLDA	FREQUENTIST	VARIATIONAL	FIXED: $1/t$	GAMMA PRIOR
SGS	FREQUENTIST	GIBBS SAMPLING	FIXED: $1/t$	α FIXED

Table 1: Comparison of existing methods for LDA.

6.1.2 ESTIMATION OF $\mathbb{E}_{Z|X,\eta}[S(X,Z)]$

For LDA, the expectation $\mathbb{E}_{Z|X,\eta}[S(X,Z)]$ can either be estimated with Gibbs sampling—using Equation (14)—or with variational approximation—using Equation (10).

6.1.3 STEP-SIZE

Some of the methods listed above (`SVB`, `SPLDA` and `SGS`) are incremental, which means the sufficient statistics are incrementally aggregated $s_t = s_{t-1} + \mathbb{E}_{Z_t|X_t,\eta}[S(X_t,Z_t)]$. For LDA, it exactly corresponds to a step-size $\rho_t = 1/t$ in the online EM setting, even though the link is not explicit in the corresponding papers.

For the other listed methods, the step-size exponent κ is chosen arbitrarily in $[0.5, 1)$, with $\rho_t = 1/t^\kappa$. However, results are mostly presented with $\kappa = 1/2$ and $\rho_t = 1/\sqrt{t}$.

6.2 General Settings

6.2.1 INITIALIZATION

We initialize randomly $\eta \equiv (\beta, \alpha)$. For a given experiment, we initialize all the methods with the same values of (β, α) for fair comparison, except `SPLDA` that has its own initilization scheme—see Sato et al. (2010) for more details.

6.2.2 MINIBATCH

We consider minibatches of size 100 documents for each update in order to reduce noise (Liang and Klein, 2009). In the case of online EM in Equation (3), we estimate an expectation for each observation of the minibatch. We update the new sufficient statistics s towards the average of the expectations over the minibatch. We do the same averaging for all the presented methods.

6.2.3 NUMBER OF LOCAL UPDATES

For all the presented methods, we set the number of passes through each minibatch to $P = 20$. For `G-OEM`, this means that we perform 20 Gibbs sampling for each word of the minibatch. All other methods access each document 20 times (e.g., 20 iterations of variational inference on each document). For `G-OEM`, inference with larger values for P (e.g., $P = 50$ or $P = 100$) leads to very similar results.

6.2.4 DATASETS

We apply the methods on six differents datasets, summarized in Table 2 ($\overline{N_X}$ is the average length of documents). Following Blei et al. (2003), the synthetic dataset has been generated from 10 topics and the length of each document drawn from a Poisson(60). The 10 topics are inferred with online LDA (Hoffman et al., 2010) from 50,000 reviews of the IMDB dataset with a vocabulary size of 10,000. We only consider the entries of the 1,000 most frequent words of this dataset that we normalize to satisfy the constraint $\sum_v \beta_v^k = 1$.

The words in the datasets IMDB, Wikipedia, New York Times, Pubmed and Amazon movies are filtered by removing the stop-words and we select the most frequent words of the datasets. For the synthetic dataset, IMDB, Pubmed and Amazon movies, the size of the test sets is $N_t = 5{,}000$ documents. For Wikipedia and New York Times, the test sets contain $N_t = 2{,}000$ documents.

DATASET	#DOCUMENTS	$\overline{N_X}$	#WORDS
SYNTHETIC	1,000,000	60	1,000
WIKIPEDIA[1]	1,010,000	162.3	7702
IMDB[2]	614,589	82.2	10,000
AMAZON MOVIES[3]	338,565	75.4	10,000
NEW YORK TIMES[4]	299,877	287.4	44,228
PUBMED[4]	2,100,000	82.0	113,568

Table 2: Datasets.

We run the methods on 11 differents train/test splits of each dataset. For all the presented results, we plot the median from the 11 experiments as a line—solid or dashed. For the sake of readability, we only present the same plots with error bars between the third and the seventh decile in Appendix D and Appendix E.

6.2.5 COMPUTATION TIME

For each presented method and dataset, the computational time is reported in Table 3. Although all methods have the same running-time complexities, coded in Python, sampling methods (`G-OEM`, `VarGibbs` and `SGS`) need an actual loop over all documents while variational methods (`OLDA`, `SVB`, `SPLDA` and `V-OEM++`) may use vector operations, and may thus be up to twice faster. This could be mitigated by using efficient implementations of Gibbs sampling on minibatches (Yan et al., 2009; Zhao et al., 2014; Gao et al., 2016). Note also that to attain a given log-likelihood, our method `G-OEM` is significantly faster and often attains log-likelihoods not attainable by other methods (e.g., for the dataset New York Times).

6.2.6 STEP-SIZE

In the following, we compare the results of our methods `G-OEM` and `V-OEM++` with $\kappa = 1/2$, i.e., the step-size $\rho_t = 1/\sqrt{t}$, without averaging. Detailed analysis of different settings of our method can be found in Appendix A. In particular, we compare different step-sizes and the effect of averaging over all iterates. We also compare the performance of `OLDA` with different step-sizes in Appendix A.2 and observe that results are very similar for all the step-sizes that we try. Note that for incremental methods (`SVB`, `SPLDA`, `SGS`), the step-size is fixed to $\rho_t = 1/t$. For `LDS`, we run the method with parameters as close as possible to our method for fair comparison.

6.3 Results on LDA

Results obtained with the presented methods applied to LDA on different datasets for different values of the number K of topics are presented in Figure 1. Performance through iterations (i.e., as the number of documents increases) is presented in Figure 4. We first observe that for all experiments, our new method `G-OEM` performs better—often significantly—than all existing methods. In particular, it is highly robust to diversity of datasets.

1. Code available from Hoffman et al. (2010)
2. Dataset described in Diao et al. (2014)
3. Data from Leskovec and Krevl (2014)
4. UCI dataset (Lichman, 2013)

	IMDB	WIKIPEDIA	NYT	PUBMED
G-OEM	13H	55H	30H	58H
V-OEM++	9H	37H	20H	54H
OLDA	7H	33H	8H	30H
VARGIBBS	12H	50H	28H	54H
SVB	7H	34H	9H	30H
SPLDA	9H	37H	20H	54H
SGS	11H	48H	27H	50H
LDS	7H	17H	12H	40H

Table 3: Average computational time (in hours) for each method—$K = 128$.

6.3.1 INFLUENCE OF THE NUMBER OF TOPICS K

As shown in Figure 1, for synthetic data in plot (a), although the true number of topics is $K^* = 10$, SPLDA, OLDA, VarGibbs and SGS perform slightly better with $K = 20$, while G-OEM has the better fit for the correct value of K; moreover, SVB has very similar performances for any value of K, which highlights the fact that this method does not capture more information with a higher value of K. LDS performs very poorly on this dataset—for any value of K the log-perplexity is around 400—and is not displayed in Figure 1 (a) for clarity.

On non-synthetic datasets in plots (b)-(f), while the log-perplexity of frequentist methods—G-OEM, V-OEM++ and SPLDA—decreases with K, the log-perplexity of variational Bayesian methods—OLDA and SVB—does not decrease significantly with K. As explained below, our interpretation is that the actual maximization of the ELBO does not lead to an improvement in log-likelihood. The hybrid Bayesian method VarGibbs—which uses Gibbs sampling for local updates (θ, z) and variational updates for global parameters (β, α)—performs much better than the variational Bayesian methods. Our interpretation is that the objective function maximized with VarGibbs is a much better approximation of the log-likelihood than the ELBO.

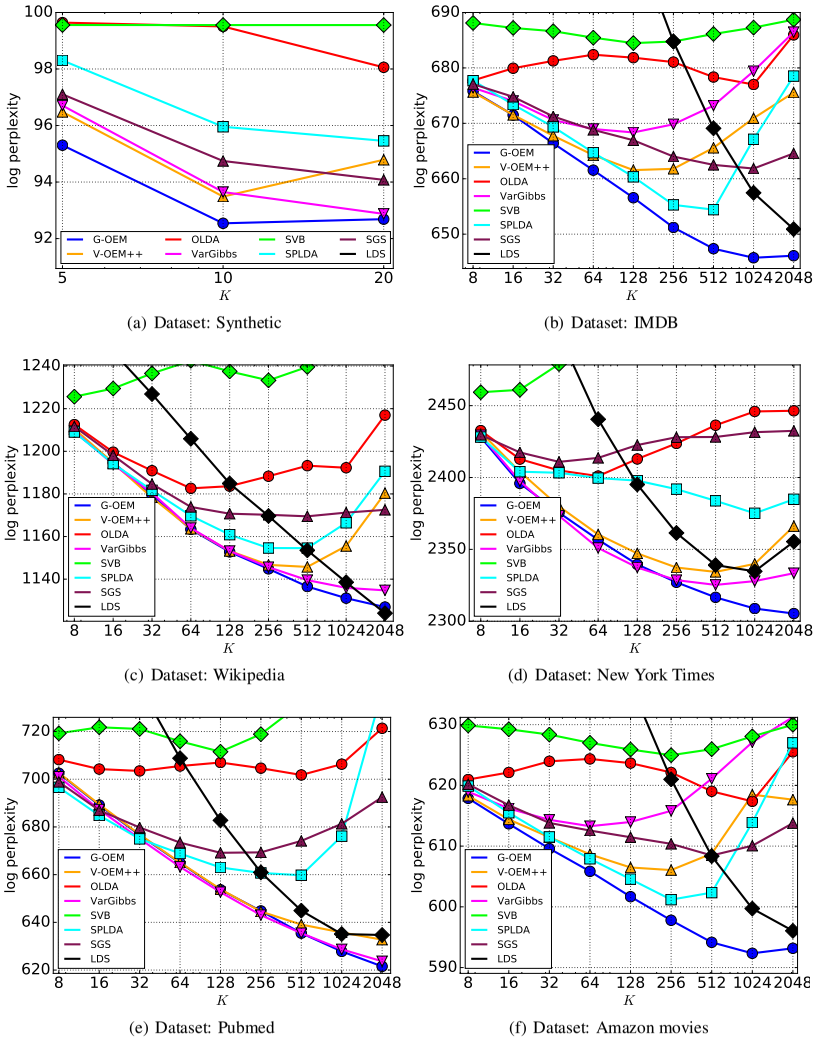
In terms of robustness, G-OEM and LDS are the only methods that do not display overfitting on any dataset. However, LDS is only competitive for the highest values of K—$K \geq 500$.

6.3.2 PERFORMANCE THROUGH ITERATIONS

As shown in Figure 4, for synthetic data in plot (a), after only few dozens of iterations—few thousands of documents seen—G-OEM, V-OEM++ and VarGibbs outperform the other presented methods. Variational Bayesian methods again do converge but to a worse parameter value. On real datasets in plots (b)-(f), G-OEM and VarGibbs are significantly faster; we can indeed still observe that after around 100 iterations—10,000 documents seen—G-OEM and VarGibbs perform better than other methods on all the datasets except Pubmed, where the performances of G-OEM, V-OEM++, VarGibbs and SPLDA are similar. Note that

6.3.3 VARIATIONAL VS. SAMPLING

Our method G-OEM directly optimizes the likelihood with a consistent approximation, and performs better than its variational counterparts SPLDA and V-OEM++ in all experiments. The hybrid method VarGibbs is less robust than G-OEM as it performs either similarly to G-OEM—for the datasets



(a) Dataset: Synthetic

(b) Dataset: IMDB

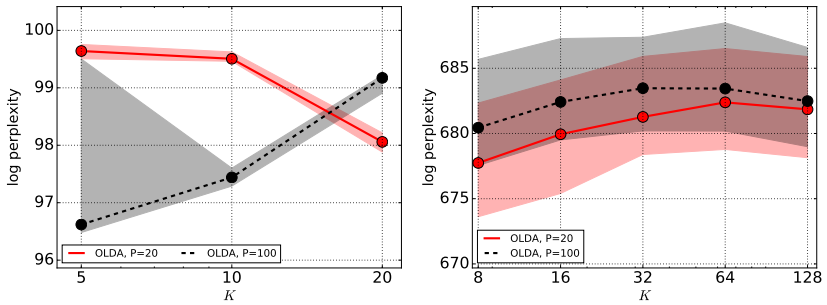
(c) Dataset: Wikipedia

(d) Dataset: New York Times

(e) Dataset: Pubmed

(f) Dataset: Amazon movies

Figure 1: Perplexity on different test sets as a function of K, the number of topics inferred. Best seen in color.



(a) Synthetic dataset. (b) IMDB.

Figure 2: OLDA. Perplexity on different test sets as a function of K for OLDA with $P = 10$ (red) and $P = 100$ (black) internal updates.

Wikipedia, New York Times and Pubmed—or worse than G-OEM and its variational counterparts SPLDA and V-OEM++—for the datasets IMDB and Amazon.

6.3.4 FREQUENTIST VS. BAYESIAN

In all our experiments we observe that frequentist methods—G-OEM, V-OEM++ and SPLDA—outperform variational Bayesian methods—OLDA and SVB. As described in Section 3.4, variational Bayesian methods maximize the ELBO, which makes additional strong independence assumptions and here leads to poor results. For example, as the number K of topics increases, the log-likelihood goes down for some datasets. In order to investigate if this is an issue of slow convergence, we show on Figure 2 (dotted black line) that running $P = 100$ internal updates in OLDA to get a finer estimate of the ELBO for each document may deteriorate the performance. Moreover, Figure 3 presents the evolution of the ELBO, which does always increase when K increases, showing that the online methods do optimize correctly the ELBO (while not improving the true log-likelihood). See Appendix B for additional results on the convergence of the ELBO. The results are mitigated for the hybrid Bayesian method VarGibbs. The performance of this method is either similar to G-OEM and V-OEM++ or significantly worse than both G-OEM and V-OEM++.

6.3.5 SMALL STEP-SIZES VS. LARGE STEP-SIZES

SPLDA is also a variational method which is equivalent to V-OEM++, but with a step-size $1/t$, which is often slower than bigger step-sizes (Mairal, 2014), which we do observe—see Appendix A.2 for a further analysis on the effect of the choice of step-sizes as $1/i^\kappa$ on G-OEM. Note that we run all the methods on a fixed (finite) number of observations. If we were to extend to infinite datasets, the difference between the step-sizes should be the speed of convergence. However, even if the number of observations is large, the gap between the step-sizes is still significant to justify the use of $1/\sqrt{t}$ for the step-size. Indeed, when considering large datasets, the contribution of each

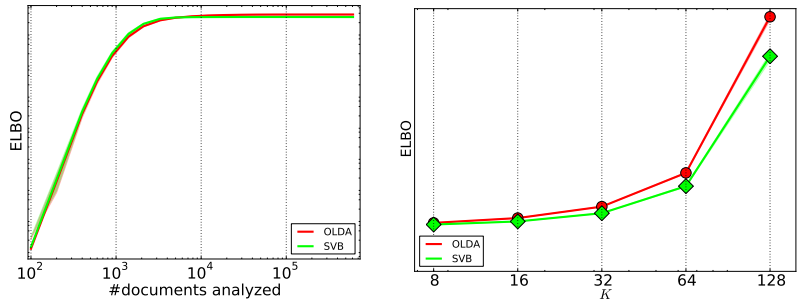


Figure 3: Dataset: IMDB. Evidence Lower BOund (ELBO) computed on test sets (20 internal iterations and 1 pass over the dataset). *Left*: ELBO through iterations with $K = 128$. *Right*: ELBO as a function of the number of topics K.

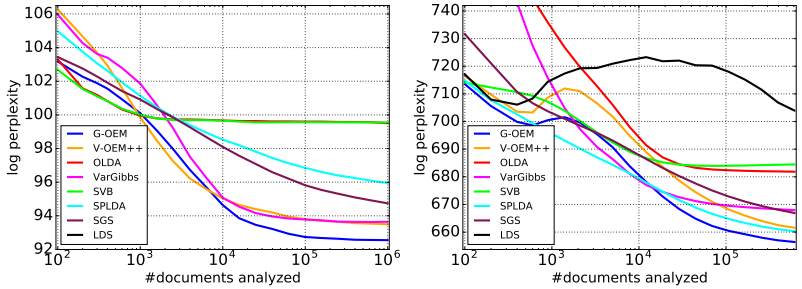
iteration at the end of the pass over the data is squeezed by the step-size in $1/t$. When the number of observations is large enough to prevent the use of batch algorithms but still insufficient for an online algorithm to converge in one pass, a possible solution could be to consider constant step-sizes in order to converge even faster to a local maxima. As proposed, we do not have any guarantee for our methods to converge with constant step-sizes, but previous works have shown the benefits of using constant step-sizes under certain assumptions (e.g., Bach and Moulines (2013))

6.4 Empirical Analysis

In this section we provide a qualitative empirical analysis on the topics extracted with the different methods. We note this is clearly a subjective analysis but it stresses the benefits of a "better" inference mechanism in terms of log-likelihood (Chang et al., 2009). Examples of eight topics extracted with G-OEM and OLDA on the IMDB dataset of movie reviews are presented in Table 4 page 30.

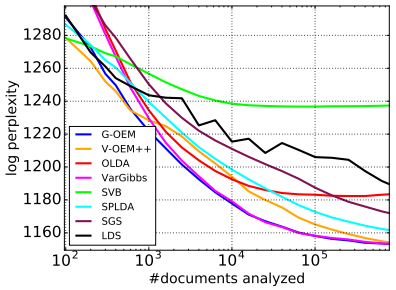
We first compute the KL divergence between the $K = 128$ topics extracted with G-OEM and the $K = 128$ topics extracted with OLDA. We run the Hungarian algorithm on the resulting distance matrix to assign each topic extracted with G-OEM to a single topic of OLDA. We choose manually eight topics extracted with G-OEM that are representative of the usual behavior, and display the eight corresponding topics of OLDA assigned with the above method.

We observe that the topics extracted with G-OEM are more consistent than topics extracted with OLDA: topics of G-OEM precisely describe only one aspect of the reviews while the topics of OLDA tend to mix several aspects in each topic. For instance, the words of topic 1 extracted with G-OEM are related to *horror* movies. The words of the corresponding topic extracted with OLDA mix *horror* movies—e.g., *horror, scary*—and *ghost* movies—e.g., *ghost, haunt*. In this OLDA topic 1, we can also observe less relevant words, like *effective, mysterious*, which are not directly linked with *horror* and *ghost* vocabularies. We can make the same remarks with topic 2 and topic 3, respectively related to *comedy* movies and *romantic comedy* movies. In topic 2 extracted with G-OEM, the least related

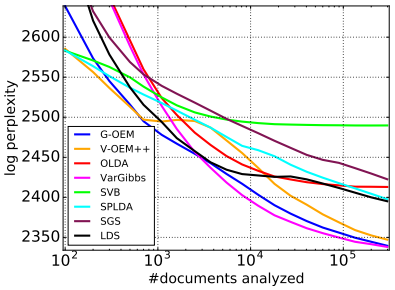


(a) Synthetic, $K = 10$

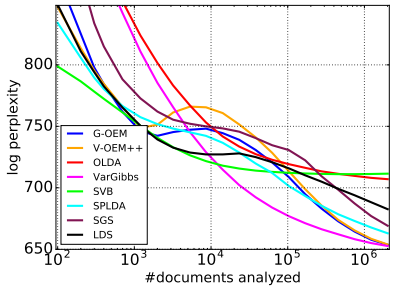
(b) IMDB, $K = 128$



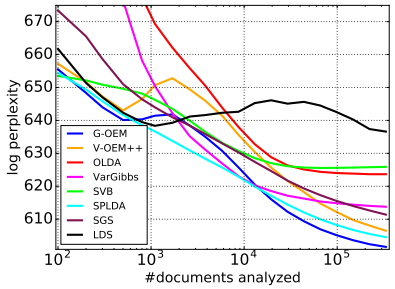
(c) Wikipedia, $K = 128$



(d) New York Times, $K = 128$



(e) Pubmed, $K = 128$



(f) Amazon movies, $K = 128$

Figure 4: Perplexity through iterations on different test sets with the presented methods. Best seen in color.

words to *comedy* are names of characters/actors—i.e., *steve* and *seth*—while the words not related to *comedy* in topic 25 of OLDA are more general, belonging to a different lexical field—e.g., *sport, site, progress, brave, definition*. In topic 3 of G-OEM, all the presented words are related to *romantic comedy* while in topic 3 of OLDA, the words *old, hard* and *review* are not related to this genre.

We also observe that G-OEM extracts strongly "qualitative" topics—topic 4 and topic 5—which is not done with OLDA. Indeed, it is difficult to group the top words of topic 4 or topic 5 of OLDA in the same lexical field. Except *dialogue* and *suppose*, all the top words of topic 4 of G-OEM are negative words. These two words may appear in a lot of negative sentences, leading to a high weight in this topic. In topic 5 of G-OEM, the words *absolutely* and *visual* are non strictly positive words while the thirteen other words in this topic convey a positive opinion. The word *absolutely* is an adverb much more employed in positive sentences than negative or neutral sentences, which can explain its high weight in topic 5.

The topic 6 of both G-OEM and OLDA can be considered as a "junk" topic, as for both method, most of its top words are contractions of modal verbs or frequent words—e.g., *didn't, isn't, wait, bad*. The contractions are not filtered when removing the stop words as they are not included in the list of words removed[1].

For both G-OEM and OLDA, the top words of topic 7 are general words about movies. These words are usually employed to describe a movie as a whole—e.g., *narrative, filmmaker*.

Finally, the top words of topic 8 of G-OEM are related to the situation of the scenes. We could not find such topic in the other presented methods and we can see that the top words of topic 8 of OLDA—supposedly close to topic 8 of G-OEM—are related to *family* movies. Each word of topic 8 of G-OEM—except *group* and *beautiful*—are related to a *spatial location*, and may help answer the question "*where does the scene take place?*".

6.5 Results on HDP

For the HDP model, we compare our G-OEM method to the Bayesian VarGibbs (Wang and Blei, 2012) method. We set the initial number of topics to $T = 2$. We present in Figure 5 results obtained with G-OEM and VarGibbs applied to both LDA and HDP. Results with error bars are presented in Appendix F. For both LDA and HDP, G-OEM outperforms the Bayesian method VarGibbs.

7. Conclusion

We have developed an online inference scheme to handle intractable conditional distributions of latent variables, with a proper use of local Gibbs sampling within online EM, that leads to significant improvements over variational methods and Bayesian estimation procedures. Note that all methods for the same problem are similar (in fact a few characters away from each other); ours is based on a proper stochastic approximation maximum likelihood framework and is empirically the most robust. It would be interesting to explore distributed large-scale settings (Broderick et al., 2013; Yan et al., 2009; Gao et al., 2016) and potentially larger (e.g., constant) step-sizes that have proved efficient in supervised learning (Bach and Moulines, 2013).

1. See NLTK toolbox (Bird et al., 2009) for the exhaustive list of stop words.

(a) IMDB. (b) Wikipedia

Figure 5: Perplexity through iterations on different test sets with `G-OEM` and `VarGibbs` applied to both LDA and HDP. Best seen in color.

Acknowledgments

We would like to thank David Blei, Olivier Cappé, Nicolas Flammarion and John Canny for helpful discussions related to this work.

Table 4: Comparison of topics extracted on IMDB dataset, $K = 128$—15 top words of eight topics extracted with `G-OEM` and `OLDA`.

	`G-OEM`						
#	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
1	VIOLENCE	COMEDY	ROMANTIC	BAD	BRILLIANT	DIDN'T	NARRATIVE	TOWN
2	VIOLENT	FUNNY	COMEDY	WORST	PERFECT	I'VE	ULTIMATELY	LOCAL
3	DISTURBING	LAUGH	LOVE	WASTE	BEAUTIFUL	WASN'T	CINEMATIC	MOUNTAIN
4	BRUTAL	JOKE	ROMANCE	BORING	MASTERPIECE	ISN'T	APPROACH	VILLAGE
5	MURDER	HILARIOUS	FUNNY	AWFUL	AMAZING	WE'RE	PROTAGONIST	LOCATION
6	GRAPHIC	COMIC	CHARMING	POOR	SUPERB	I'LL	SEEMINGLY	ROAD
7	KILLER	COMEDIC	CHEMISTRY	DIALOGUE	STUNNING	COULDN'T	NATURE	JOURNEY
8	TORTURE	STEVE	SWEET	WORSE	WONDERFUL	WOULDN'T	TONE	GROUP
9	VICTIM	AMUSING	ENJOY	DULL	ABSOLUTELY	PRETTY	FILMMAKER	TRAVEL
10	RAPE	FUN	HEART	FAIL	BEST	BAD	WHOSE	COUNTRY
11	KILL	GAG	NICE	MESS	INCREDIBLE	HAVEN'T	CRAFT	LANDSCAPE
12	HORROR	SETH	CHARM	RIDICULOUS	BRILLIANTLY	GUESS	CONTEMPORARY	LAND
13	BLOODY	FUNNIEST	GREAT	SUPPOSE	BEAUTIFULLY	AREN'T	MANNER	BEAUTIFUL
14	REVENGE	CAMEO	FUN	TERRIBLE	VISUAL	ENJOY	SERVE	AREA
15	BLOOD	SITUATION	WONDERFUL	UNFORTUNATELY	PERFECTLY	REVIEW	MATERIAL	TRIP
	`OLDA`							
#	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
1	HORROR	HILARIOUS	LOVE	BAD	GREAT	WRONG	VISUAL	YOUNG
2	NIGHT	ROMANCE	ENJOY	ACTION	BEST	DIDN'T	FOCUS	FAMILY
3	DEAD	CLEVER	PRETTY	INTERESTING	STAR	I'VE	REALITY	CHILD
4	TWIST	SMART	OLD	ORIGINAL	LONG	WAIT	DIFFICULT	FATHER
5	SCARY	INTRIGUING	FUNNY	FAR	JOHN	CATCH	FILMMAKER	SON
6	EFFECTIVE	COMEDIC	COMEDY	SPECIAL	EXCELLENT	EXACTLY	IMAGE	AGE
7	MYSTERIOUS	FUNNIEST	FUN	FIGHT	CLASSIC	WASN'T	NARRATIVE	WHOSE
8	BLOODY	PROGRESS	HARD	HERO	BEAUTIFUL	HUGE	INTELLIGENT	TALE
9	GHOST	SPORT	PERFECT	ENTIRE	DRAMA	I'LL	ACCEPT	DISCOVER
10	HAUNT	SITE	LAUGH	HALF	WONDERFUL	CHOICE	IMPRESSION	EASILY
11	FEAR	BRAVE	ENTERTAINING	SAVE	MICHAEL	ETC	EXTREME	DREAM
12	EVIL	DREADFUL	WORTH	DIALOGUE	HEART	SERIOUSLY	CENTRAL	INTRODUCE
13	NIGHTMARE	SHOULDER	NICE	FULL	FORGET	NOTICE	MAINTAIN	MARRY
14	GORY	GIMMICK	FAVORITE	VIOLENCE	ROBERT	RIDICULOUS	DENY	RAISE
15	MASK	DEFINITION	REVIEW	EXAMPLE	EARLY	ANSWER	NAIL	RULE

Appendix A. Gibbs/Variational Online EM Analysis

In this section we evaluate the proposed methods G−OEM and V−OEM with different settings in terms of step-sizes, averaging outputs and boosting internal updates.

A.1 Effect of Inference Boosting on G−OEM and V−OEM

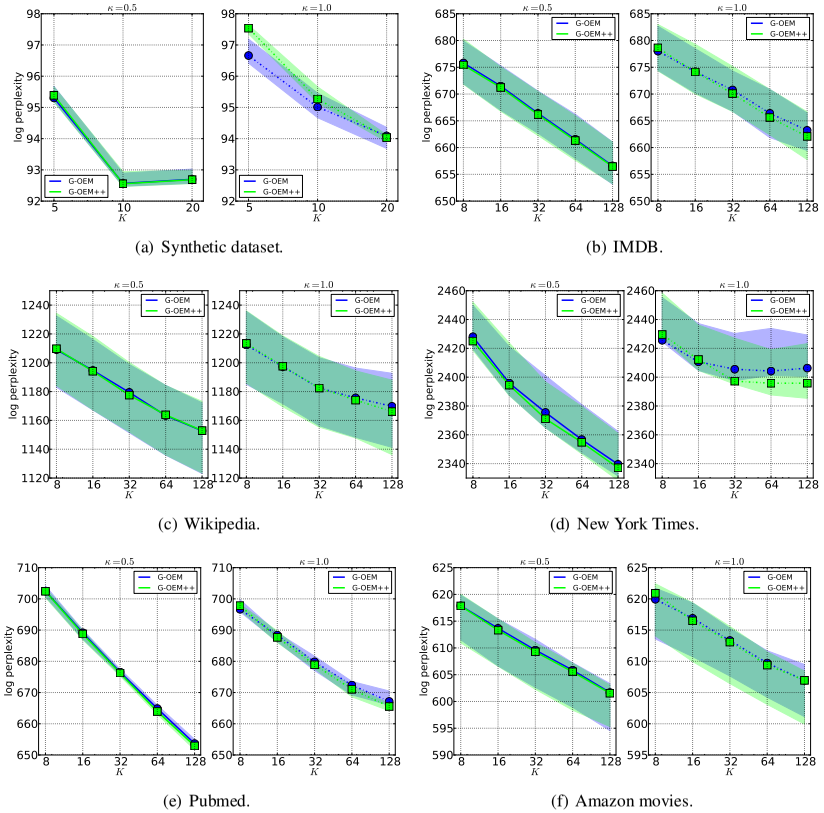
The effect of the inference boost as described in Section 3.3 on G−OEM and V−OEM with synthetic and IMDB datasets is presented in Figure 6 and in Figure 7. It leads to a minor improvement for G−OEM++ and a significant one for V−OEM++.

A.2 Step-sizes and Averaging

We apply G−OEM with different stepsizes $\rho_i = \frac{1}{i^\kappa}$. Note that because we average sufficient statistics, there is no needed proportionality constants. We first compare the performance of the last iterate η_N (without averaging) and the average of the iterates $\bar{\eta}_N = \frac{1}{N}\sum_{i=0}^{N} \eta_i$ (with averaging) for different values of κ.

Results are presented in Figure 8 on the synthetic data and in Figure 9 on the IMDB dataset. For $\kappa \in \left[0, \frac{1}{2}\right[$, averaging improves the performance while for $\kappa \in \left]\frac{1}{2}, 1\right]$, averaging deteriorates the performance. For $\kappa = \frac{1}{2}$, averaging is only slightly beneficial on IMDB dataset. For constant stepsizes $\kappa = 0$ the averaging improves significantly the performance, as the iterates do not converge and tend to oscillate around a local optimum (Bach and Moulines, 2013). We can expect the same effect for $\kappa \in \left[0, \frac{1}{2}\right[$ as the function $n \mapsto \frac{1}{n^\kappa}$ deacreases slowly for such values of κ. For $\kappa \in \left]\frac{1}{2}, 1\right]$, the stochastic gradient ascent scheme is guaranteed to converge to a local optimum (Bottou, 1998). The averaging then deteriorates the performance as it incorporates the first iterates, which gets the last iterate away from local optimum. However, the stepsize $1/i$ ($\kappa = 1$) is not competitive. The performance with $\kappa = 0.75$ is only slightly better on IMDB dataset. The setting $\kappa = \frac{1}{2}$ represents a good balance between first and last iterates. For this step-size, performances with or without averaging are similar but results without averaging seem to be more stable, hence our choice for all our other simulations.

We also apply OLDA with different step-sizes $\rho_t = \tau/t^\kappa$ for different values of τ, κ. Results are presented in Figure 10 without error bars and in Figure 11 with error bars. For OLDA, results are very similar for any step-size.

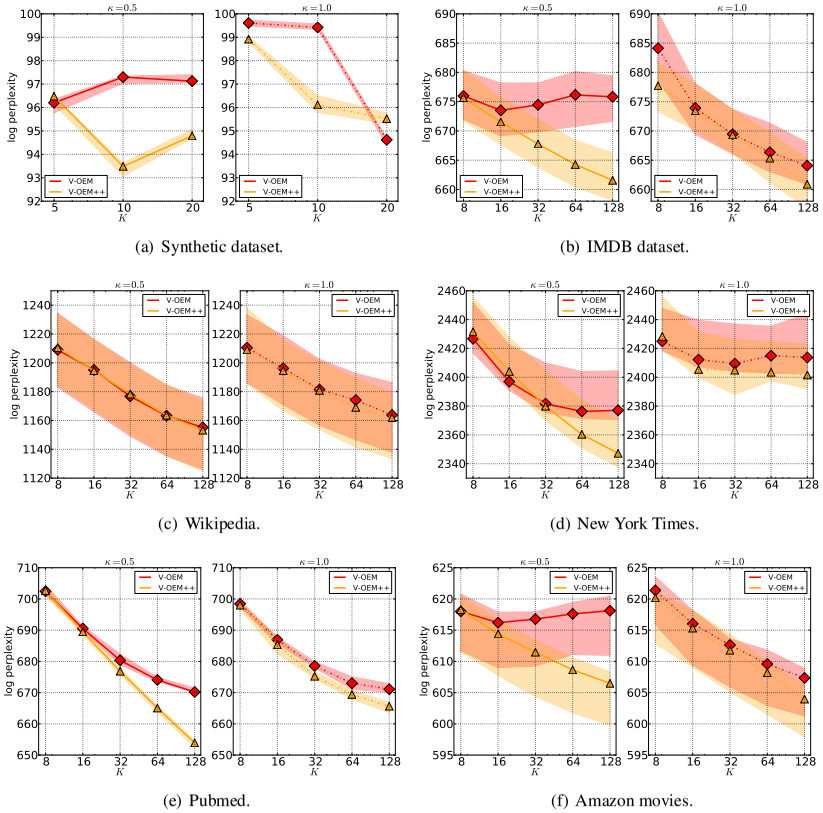


(a) Synthetic dataset. (b) IMDB.

(c) Wikipedia. (d) New York Times.

(e) Pubmed. (f) Amazon movies.

Figure 6: G−OEM. Perplexity on different test sets as a function of the number of topics K for regular EM and boosted EM (++). We observe that for almost all datasets, there is no significant improvement when boosting the inference. Our interpretation is that each Gibbs sample is noisy and does not provide a stable boost. Best seen in color.



(a) Synthetic dataset.

(b) IMDB dataset.

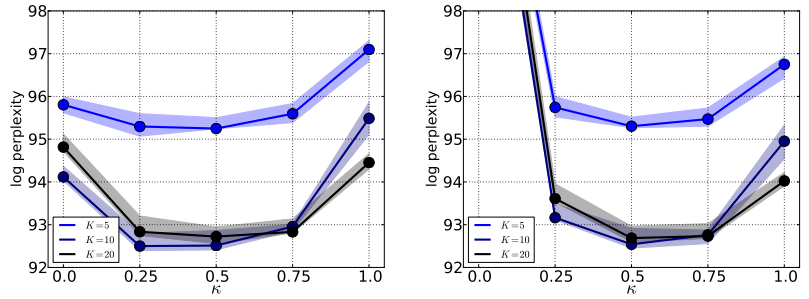
(c) Wikipedia.

(d) New York Times.

(e) Pubmed.

(f) Amazon movies.

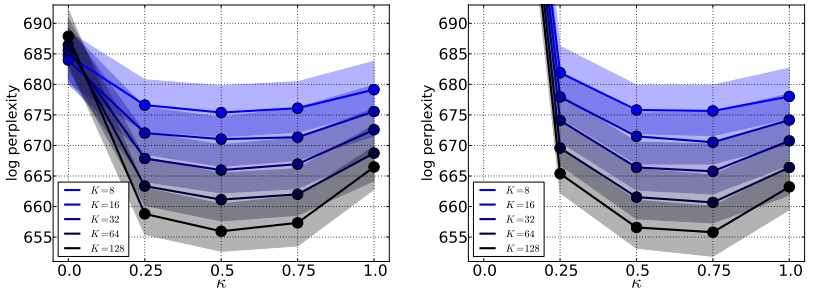
Figure 7: V−OEM. Perplexity on different test sets as a function of the number of topics K for regular EM and boosted EM (++). We observe that boosting inference improves significantly the results on all the datasets excepted on Wikipedia where V−OEM and V−OEM++ have similar performances. The variational estimation of the posterior is finer and finer through iterations. When updating the parameters at each iteration of the posterior estimation, the inference is indeed boosted. Best seen in color.



(a) *With* averaging

(b) *Without* averaging

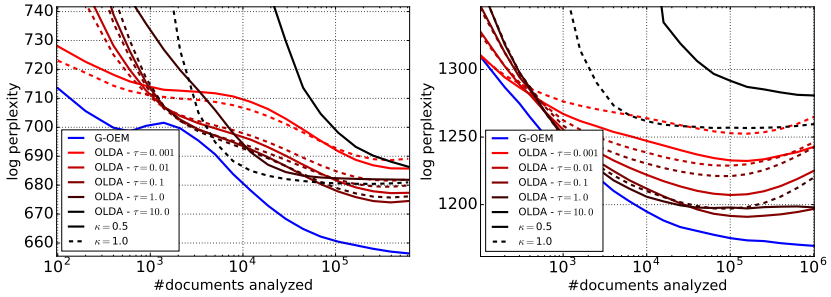
Figure 8: Dataset: synthetic. Perplexity on different test sets as a function of the exponent κ—the corresponding stepsize is $\rho_i = \frac{1}{i^\kappa}$—for G−OEM with averaging (left) and without averaging (right). The number of topics inferred K goes from 5 (the lightest) to 20 (the darkest). Best seen in color.



(a) *With* averaging

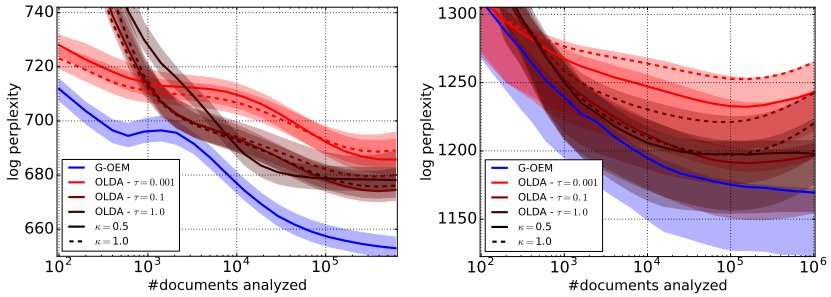
(b) *Without* averaging

Figure 9: Dataset: IMDB. Perplexity on different test sets as a function of the exponent κ—the corresponding stepsize is $\rho_i = \frac{1}{i^\kappa}$—for G−OEM with averaging (left) and without averaging (right). The number of topics inferred K goes from 8 (the lightest) to 128 (the darkest). Best seen in color.



(a) IMDB. (b) Wikipedia.

Figure 10: Evolution of perplexity on different test sets as a function of the number of documents analyzed. For OLDA, we compare the performance with different step-sizes $\rho_t = \tau/t^\kappa$ for different values of τ, κ. Solid line: $\kappa = 1/2$; Dashed line: $\kappa = 1$.

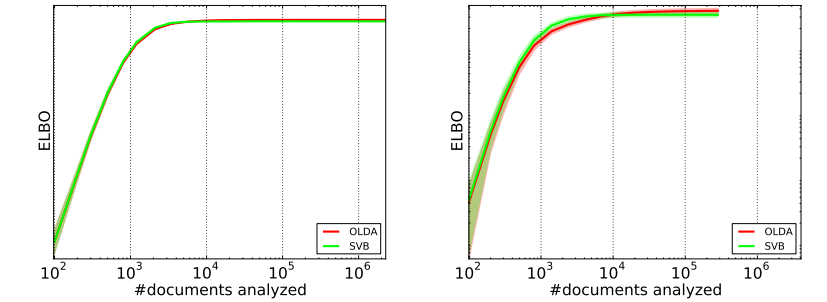


(a) IMDB. (b) Wikipedia.

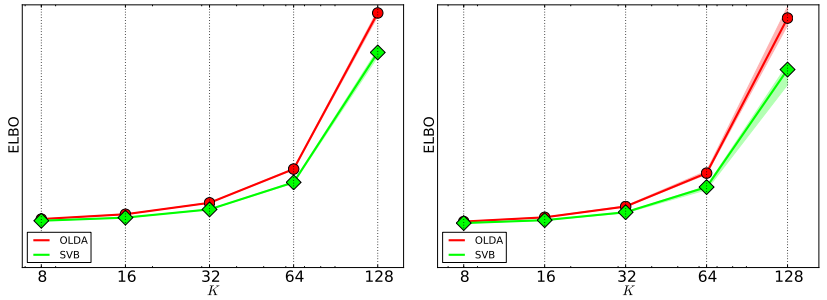
Figure 11: Evolution of perplexity on different test sets as a function of the number of documents analyzed, with error bars. For OLDA, we compare the performance with different step-sizes $\rho_t = \tau/t^\kappa$ for different values of τ, κ. Solid line: $\kappa = 1/2$; Dashed line: $\kappa = 1$.

Appendix B. Evolution of the ELBO

Figure 12 presents the evolution of the ELBO for online LDA (OLDA) and SVB on different test sets. We compute the ELBO on test documents as described by Hoffman et al. (2010). This plot helps us to observe that even if the ELBO reaches a local maximum (i.e., it stabilizes), the quality of the model in terms of perplexity is not controllable. We can also see in Figure 12 that the ELBO is much better optimized with $K = 128$ than with other values of K for both SVB and OLDA, that is, as expected, latent variables of higher dimensionality lead to better fits for the cost function which is optimized. However, for several datasets the performance in terms of perplexity is better with low values of K ($K = 8$ or $K = 16$) than with high dimensional variables ($K = 64$ or $K = 128$).



(a) IMDB, $K = 128$ (b) Wikipedia, $K = 128$



(c) IMDB (d) Wikipedia

Figure 12: Evidence Lower BOund (ELBO) computed on different test sets. Top: ELBO through iterations, with 4 passes over each dataset and 200 internal iterations. Bottom: ELBO as a function of the number of topics K, with 20 internal iterations and 1 pass over each dataset. Best seen in color.

In order to check if more internal iterations could help variational Bayesian methods, we present in Table 5 the values of perplexity reached by OLDA when running 4 passes over each dataset with $P = 200$ internal iterations 1 pass over each dataset with $P = 20$ internal iterations. We observe that the ELBO converges quickly to a local optimum and doing ten times more internal iterations does not change significantly the final performance.

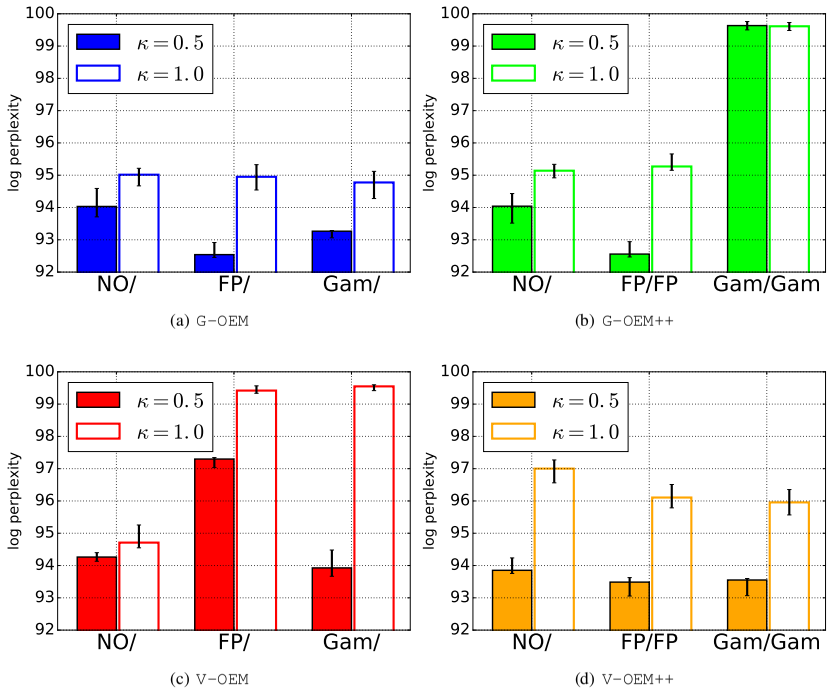
	$P = 200$, 4 passes	$P = 20$, 1 pass
OLDA	682.6±3.7	681.9±3.9
SVB	683.8±3.8	684.5±3.8

Table 5: Comparison of log-perplexity levels reached with OLDA and SVB on IMDB dataset.

Appendix C. Updates in α

In this section we compare the different types of updates for α. Figure 13 presents results obtained on synthetic dataset for fixed point iteration algorithm (Minka, 2000) and by putting a gamma prior on α (Sato et al., 2010). We observe that the fixed point method leads to better performance for G-OEM and G-OEM++. For V-OEM, the gamma updates better perform for $\kappa = \frac{1}{2}$. The performances of the gamma updates and the fixed point method are very similar for V-OEM++. Note that the algorithm V-OEM++ with $\kappa = 1$ and gamma updates on α is exactly equivalent to SPLDA (Sato et al., 2010). The performance of this method can be improved by setting $\kappa = \frac{1}{2}$ with any update on α.

We also observe that fixing α to α_{true} that generated the data does not necessarily lead to better performance.



(a) G-OEM (b) G-OEM++

(c) V-OEM (d) V-OEM++

Figure 13: Dataset: Synthetic, $K = 10$. Perplexity on different test sets for different types of updates for α; for boosted methods, we use the same inference for α for local and global updates. NO: α is fixed and set to α_{true} that generated the data; FP: fixed point iteration; Gam: gamma prior on α (Sato et al., 2010). Best seen in color.

Appendix D. Performance with Different K, with Error Bars

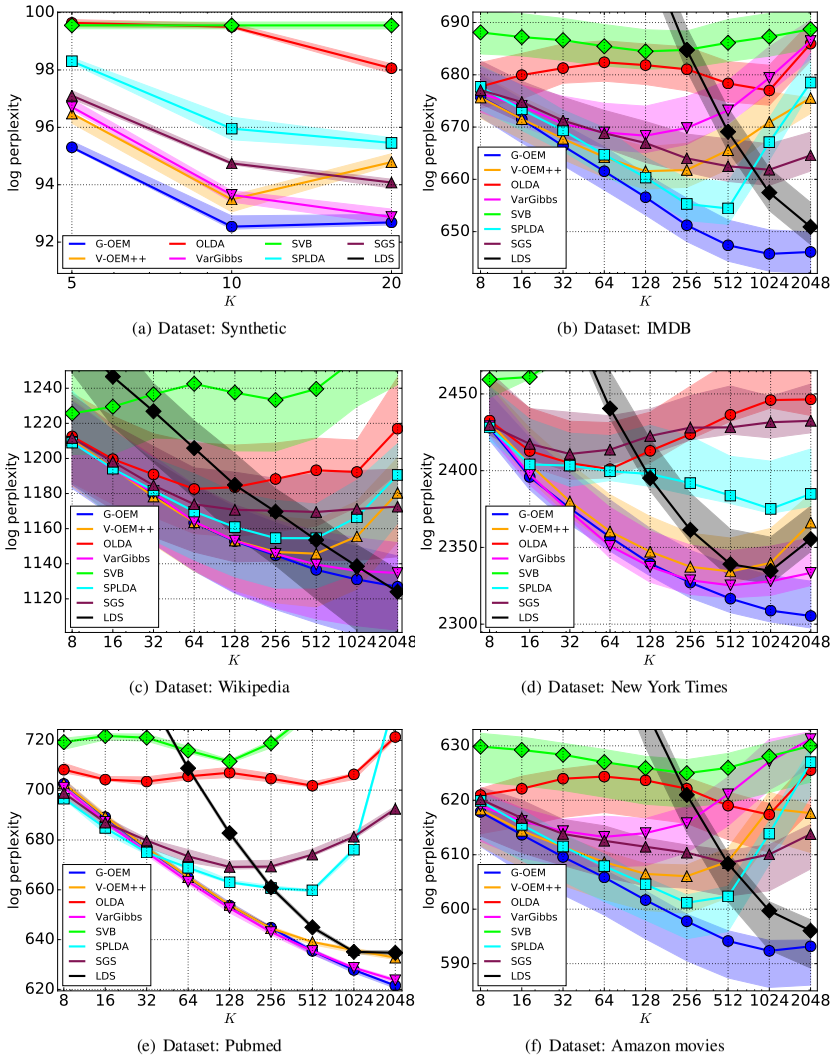
The performance of the presented methods for different values of K on the different datasets is presented in Figure 14. We plot the median from the 11 experiments as a line—solid or dashed—and a shaded region between the third and the seventh decile.

Appendix E. Performance Through Iterations, with Error Bars

The performance through iterations of the presented methods on the different datasets is presented in Figure 15. We plot the median from the 11 experiments as a line—solid or dashed—and a shaded region between the third and the seventh decile.

Appendix F. Results on HDP, with Error Bars

The performance through iterations of the `G-OEM` and `VarGibss` applied to both LDA and HDP is presented in Figure 15. We plot the median from the 11 experiments as a line—solid or dashed—and a shaded region between the third and the seventh decile.



(a) Dataset: Synthetic

(b) Dataset: IMDB

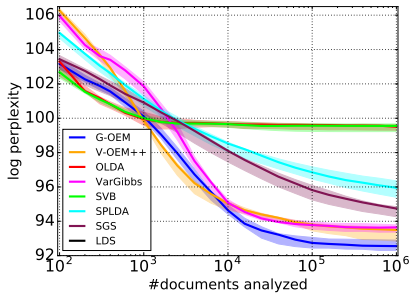
(c) Dataset: Wikipedia

(d) Dataset: New York Times

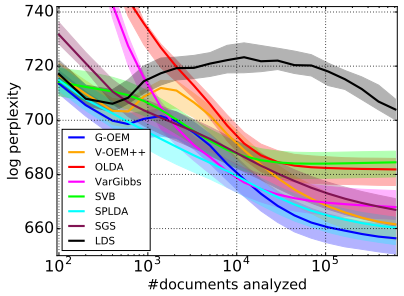
(e) Dataset: Pubmed

(f) Dataset: Amazon movies

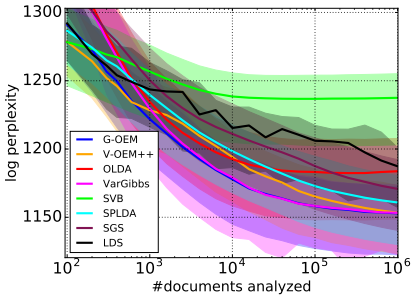
Figure 14: Perplexity on different test sets as a function of K, the number of topics inferred. Same as Figure 1, but with error bars. Best seen in colors.



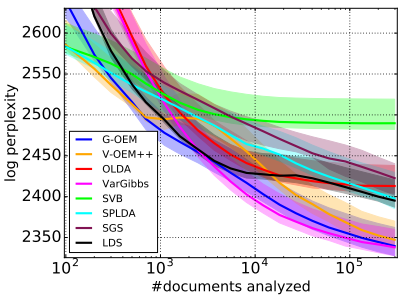
(a) Synthetic, $K = 10$



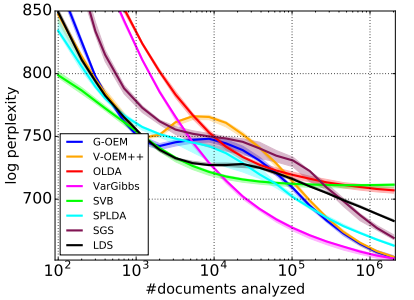
(b) IMDB, $K = 128$



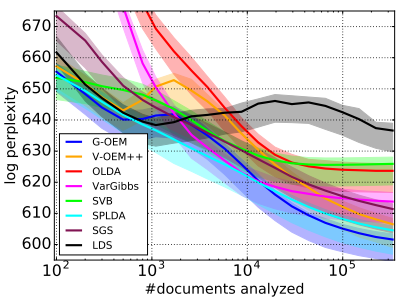
(c) Wikipedia, $K = 128$



(d) New York Times, $K = 128$

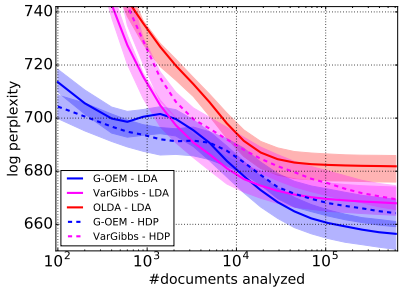


(e) Pubmed, $K = 128$

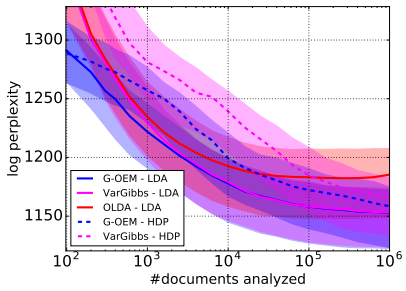


(f) Amazon movies, $K = 128$

Figure 15: Perplexity through iterations on different test sets with the presented methods. Same as Figure 4, but with error bars. Best seen in colors.



(a) IMDB.



(b) Wikipedia

Figure 16: Perplexity through iterations on different test sets with `G-OEM` and `VarGibbs` applied to both LDA and HDP. Best seen in color.

References

F. Bach and E. Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate $O(1/n)$. In *Adv. NIPS*, 2013.

S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. O'Reilly Media, Inc., 2009.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17: 9, 1998.

T. Broderick, N. Boyd, A. Wibisono, A. Wilson, and M. Jordan. Streaming variational Bayes. 2013.

O. Cappé and E. Moulines. Online EM algorithm for latent data models. *Journal of the Royal Statistical Society*, 71(3):593–613, 2009.

O. Cappé, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer, 2005.

G. Casella and E. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

J. Chang, S. Gerrish, C. Wang, J. Boyd-Graber, and D. Blei. Reading tea leaves: How humans interpret topic models. 2009.

B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1):94–128, 1999.

A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 39(1):1–38, 1977.

Q. Diao, M. Qiu, C.-Y. Wu, A. J. Smola, J. Jiang, and C. Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proc. ACM SIGKDD*, 2014.

Y. Gao, J. Chen, and J. Zhu. Streaming Gibbs sampling for LDA model. *arXiv preprint arXiv:1601.01142*, 2016.

T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235, 2004.

K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

M. Hoffman and D. Blei. Structured stochastic variational inference. In *Proc. AISTATS*, 2015.

M. Hoffman, D. Blei, and F. Bach. Online learning for latent Dirichlet allocation. 2010.

M. Hoffman, D. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

N. Kantas, A. Doucet, S. Singh, J. Maciejowski, and N. Chopin. On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3):328–351, 2015.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

H. Kushner and G. Yin. *Stochastic approximation and recursive algorithms and applications*. Springer-Verlag, second edition, 2003.

E. Lehmann and G. Casella. *Theory of point estimation*, volume 31. Springer Science & Business Media, 1998.

J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

P. Liang and D. Klein. Online EM for unsupervised models. In *Proc. NAACL HLT*, 2009.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *arXiv preprint arXiv:1402.4419*, 2014.

N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

D. Mimno, M. Hoffman, and D. Blei. Sparse stochastic inference for latent Dirichlet allocation. *Proc. ICML*, 2012.

T. Minka. Estimating a Dirichlet distribution. Technical report, 2000.

K. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.

R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

S. Patterson and Y. Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Adv. NIPS*, 2013.

B. Polyak and A. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

D. Rohde and O. Cappé. Online maximum-likelihood estimation for latent factor models. In *Proc. IEEE SSP Workshop*, 2011.

I. Sato, K. Kurihara, and H. Nakagawa. Deterministic single-pass algorithm for LDA. In *Adv. NIPS*, 2010.

Y. Teh, M. Jordan, M. Beal, and D. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

M. Titterington. Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(2):257–267, 1984.

A. Van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000.

H. Wallach. Topic modeling: beyond bag-of-words. 2006.

H. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proc. ICML*, 2009.

C. Wang and D. Blei. Truncation-free online variational inference for Bayesian nonparametric models. In *Adv. NIPS*, 2012.

G. Wei and M. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.

F. Yan, N. Xu, and Y. Qi. Parallel inference for latent Dirichlet allocation on graphics processing units. In *Adv. NIPS*, 2009.

H. Zhao, B. Jiang, and J. Canny. SAME but different: Fast and high-quality Gibbs parameter estimation. *arXiv preprint arXiv:1409.5402*, 2014.

Journal of Machine Learning Research 18 (2017) 1-45 Submitted 10/16; Revised 5/17; Published 12/17

Poisson Random Fields for Dynamic Feature Models

Valerio Perrone[*] V.PERRONE@WARWICK.AC.UK
Paul A. Jenkins[*†] P.JENKINS@WARWICK.AC.UK
Dario Spanò[*] D.SPANO@WARWICK.AC.UK
[*]*Department of Statistics*
[†]*Department of Computer Science*
University of Warwick
Coventry, CV4 7AL, UK

Yee Whye Teh Y.W.TEH@STATS.OX.AC.UK
Department of Statistics
University of Oxford
Oxford, OX1 3LB, UK

Editor: Lawrence Carin

Abstract

We present the *Wright-Fisher Indian buffet process* (WF-IBP), a probabilistic model for time-dependent data assumed to have been generated by an unknown number of latent features. This model is suitable as a prior in Bayesian nonparametric feature allocation models in which the features underlying the observed data exhibit a dependency structure over time. More specifically, we establish a new framework for generating dependent Indian buffet processes, where the Poisson random field model from population genetics is used as a way of constructing dependent beta processes. Inference in the model is complex, and we describe a sophisticated Markov Chain Monte Carlo algorithm for exact posterior simulation. We apply our construction to develop a nonparametric focused topic model for collections of time-stamped text documents and test it on the full corpus of NIPS papers published from 1987 to 2015.

Keywords: Bayesian nonparametrics, Indian buffet process, topic model, Markov chain Monte Carlo, Poisson random field

1. Introduction

The Indian buffet process (IBP) (Griffiths and Ghahramani, 2011) is a distribution for sampling binary matrices with any finite number of rows and an unbounded number of columns, such that rows are exchangeable while columns are independent. It is used as a prior in Bayesian nonparametric models where rows represent objects and columns represent an unbounded array of features. In many settings the prevalence of features exhibits some sort of dependency structure over time and modeling data via a set of independent IBPs may not be appropriate. There has been previous work dedicated to extending the IBP to dependent settings (e.g., Williamson et al., 2010a; Zhou et al., 2011; Miller et al., 2012; Gershman et al., 2015). In this paper we present a novel approach that achieves this by means of a particular time-evolving beta process, which has a number of desirable properties and is better-suited for a different range of applications.

©2017 Valerio Perrone, Paul A. Jenkins, Dario Spano and Yee Whye Teh.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-541.html.

For each discrete time $t_0, \ldots, t_T$ at which the data is observed, denote by Z_t the feature allocation matrix whose entries are binary random variables such that $Z_{ikt} = 1$ if object i possesses feature k at time t and 0 otherwise. Denote by $X_k(t)$ the probability that $Z_{ikt} = 1$, namely the probability that feature k is active at time t, and by $X(t)$ the collection of these probabilities at time t. The idea is to define a prior over the stochastic process $\{X(t)\}_{t\geq 0}$ which governs its evolution in continuous time. In particular, for each feature k, $X_k(t)$ evolves independently, while features are born and die over time. This is a desirable property in several applications such as in topic modeling, where at some point in time a new topic may be discovered (birth) or forsaken (death). Our model benefits from these properties while retaining a very simple prior where sample paths are continuous and Markovian. Finally, we show that our construction defines a time-dependent beta process from which the two-parameter generalization of the IBP is marginally recovered for every fixed time t (Ghahramani et al., 2007).

The stochastic process we use is a modification of the so-called Poisson Random Field (PRF), a model widely used in population genetics (e.g., Sawyer and Hartl, 1992; Hartl et al., 1994; Bustamante et al., 2001, 2003; Williamson et al., 2005; Boyko et al., 2008; Gutenkunst et al., 2009; Amei and Sawyer, 2010, 2012). In this setting, new features can arise over time and each of them evolves via an independent Wright-Fisher (W-F) diffusion. The PRF model describes the evolution of feature probabilities within the interval $[0, 1]$, allows for flexible boundary behaviours and gives access to several off-the-shelf results from population genetics about quantities of interest, such as the expected lifetime of features or the expected time feature probabilities spend in a given subset of $[0, 1]$ (see Ewens, 2004).

We apply the WF-IBP to a topic modeling setting, where a set of time-stamped documents is described using a collection of latent topics whose probabilities evolve over time. The WF-IBP prior allows us to incorporate time dependency into the focused topic model construction described in Williamson et al. (2010b), where the IBP is used as a prior on the topic allocation matrix determining which topics underlie each observed document. As opposed to several existing approaches to topic modeling, which require specifying the total number of topics in the corpus in advance (see for instance Blei et al., 2003), adopting a nonparametric approach saves expensive model selection procedures (such as the one described in Griffiths and Steyvers, 2004). This is also reasonable in view of the fact that the total number of topics in a corpus is expected to grow as new documents accrue. Most existing nonparametric approaches to topic modeling are not designed to capture the evolution of the popularity of topics over time and may thus not be suitable for corpora that span large time periods. On the other hand, existing nonparametric and time-dependent topic models are mostly based on the Hierarchical Dirichlet Process (HDP) (Teh et al., 2006), which implicitly assumes a coupling between the probability of topics and the proportion of words that topics explain within each document. This assumption is undesirable since rare topics may account for a large proportion of words in the few documents in which they appear. Our construction inherits from the static model presented in Williamson et al. (2010b) the advantage of eliminating this coupling. Moreover, it keeps inference straightforward while using an unbounded number of topics and flexibly capturing the evolution of their popularity continuously over time.

Section 2 introduces the beta process construction of the IBP and the PRF, providing the background for Section 3 where we modify the PRF to construct the WF-IBP, our

novel time-varying feature allocation model. Section 4 describes a fixed-K approximation and an intuitive inference scheme that is built upon in Section 5 to develop an exact MCMC algorithm for posterior inference with the WF-IBP. Section 6 combines the model with a linear-Gaussian likelihood and evaluates it on a range of synthetic data sets. Finally, Section 7 illustrates the application of the WF-IBP to topic modeling and presents results obtained on both synthetic data and on the real-world data set consisting of the full text of papers from the NIPS conferences between the years 1987 and 2015.

2. Background

Before introducing our time-varying feature allocation model we first review its two building blocks, namely the IBP and the PRF. We show how the beta process connects these two models, and we adjust the PRF to develop a time-dependent extension of the two-parameter IBP in the next section.

2.1 The beta and Indian buffet processes

A completely random measure B over a measurable space (Ω,Σ) is a random measure that assigns independent masses to disjoint subsets of Ω. Any positive completely random measure is uniquely characterized by a certain Levy measure on $\Omega \times \mathbb{R}^+$ (see Kingman, 1967). Denote by c a positive function over Ω (concentration function) and by B_0 a fixed measure on Ω (base measure), and assume the base measure B_0 is continuous. A beta process B $\sim BP(c, B_0)$ on Ω is a completely random measure uniquely characterized by the Levy measure

$$\nu(d\omega, dx) = B_0(d\omega)c(\omega)x^{-1}(1-x)^{c(\omega)-1}dx,$$

where $x \in [0,1]$ and $\omega \in \Omega$. This completely random measure can be represented via a Poisson process. In order to draw $B \sim BP(c, B_0)$, draw a set of points $(\omega_i, x_i) \in \Omega \times [0,1]$ from a Poisson point process with base measure $\nu(d\omega, dx)$ and let

$$B = \sum_{k=1}^{\infty} x_k \delta_{\omega_k},$$

where $\{\omega_i\}$ are the atoms of the measure B and $\{x_i\}$ their respective weights. Given a realization B of a beta process $BP(c, \alpha B_0)$ with $\alpha > 0$, one can obtain a draw from the IBP (Thibaux and Jordan, 2007). This is done by sampling each row z_i, $i = 1, \dots, N$, of an allocation matrix Z from a Bernoulli process $z_i \mid B \sim BeP(B)$ defined as

$$z_i = \sum_{k=1}^{\infty} p_{ik}\delta_{\omega_k}, \quad p_{ik} \overset{iid}{\sim} \text{Bernoulli}(x_k).$$

Notice that as the location of the atoms is not in fact relevant for constructing the matrix Z, the IBP depends only on the so-called mass parameter $B_0(\Omega) = \alpha$ and on $c(\omega)$. In the standard IBP, $c(\omega)$ is set to be identically 1, so that the Levy measure of the corresponding one-parameter beta process is

$$\nu(d\omega, dx) = B_0(d\omega)x^{-1}dx. \tag{1}$$

If $c(\omega)$ is equal to a constant β we have the two-parameter generalization of the IBP (Thibaux and Jordan, 2007).

2.2 The Wright-Fisher model

The starting point of the PRF is the Wright-Fisher (W-F) model from population genetics (Ewens, 2004), which we briefly summarize here. Consider a finite population of organisms of size G such that $i \in \{0, 1, \dots, G\}$ individuals have the mutant version of a gene at generation k, while the rest has the non-mutant variant. Assume that each individual produces an infinite number of gametes such that the gametes yielded by a non-mutant become mutant with probability μ_G and, conversely, those yielded by a mutant become non-mutant with probability β_G. Finally, assume that the next generation of G individuals is formed by simple random sampling from this infinite pool of gametes. The evolution of the number $Y^G(k)$ of mutant genes at time k is described by a Markov chain on the discrete space $\{0, \dots, G\}$. The transition probability p_{ij} of switching from i mutants (at time k) to j mutants (at time $k+1$) is given by the following binomial sampling formula:

$$p_{ij} = \binom{G}{j} (\Psi_i)^j (1-\Psi_i)^{G-j},$$
$$\Psi_i = \frac{i(1-\beta_G) + (G-i)\mu_G}{G}.$$

Assume the initial state is $Y^G(0) = y_0$ and denote the resulting Markov chain by $Y^G = (Y^G(k))_{k=1,2,\dots} \sim \text{W-F}^G(\mu_G, \beta_G)$. Notice that, if $\mu_G = 0$ and/or $\beta_G = 0$, the states 0 and/or G are absorbing states that respectively correspond to the extinction and fixation of the mutation.

A continuous-time diffusion limit of the W-F model can be obtained, by rescaling time as $t = k/G$, and taking $G \to \infty$. The Markov chain $Y^G(\lfloor Gt \rfloor)/G$ converges to a diffusion process on $[0,1]$ (see Ethier and Kurtz, 1986; Sawyer and Hartl, 1992) which obeys the one-dimensional stochastic differential equation

$$dX(t) = \gamma(X(t))dt + \sigma(X(t))dB(t),$$

where

$$\gamma(x) = \frac{1}{2}[\mu(1-x) - \beta x], \tag{2}$$
$$\sigma(x) = \sqrt{x(1-x)}, \tag{3}$$

with some initial state $X(0) = x_0$, over the time interval $t \in [0, T]$, with rescaled parameters $\mu = \lim_{G\to\infty} 2G\mu_G$, $\beta = \lim_{G\to\infty} 2G\beta_G$, and with $B(t)$ denoting a standard Brownian motion. The terms $\gamma(x)$ and $\sigma(x)$ are respectively referred to as the drift term and the diffusion term. Denote the diffusion process as $X \sim \text{W-F}(\mu, \beta)$.

When $x(t) \to 0$ (respectively $x(t) \to 1$), then the diffusion term tends to 0 while the drift term tends to $\frac{\mu}{2}$ (respectively $-\frac{\beta}{2}$), preventing absorption at 0 or 1 provided that $\mu > 0$ (respectively $\beta > 0$). Otherwise, 0 is an absorbing extinction state (respectively, 1 is an absorbing fixation state). Moreover, if both $\mu, \beta > 0$ then the diffusion is ergodic and has a stationary distribution that is a Beta(μ, β).

As there exists no closed form expression for its transition function, simulating from the W-F diffusion requires non-trivial computational techniques. The method we used is outlined in Dangerfield et al. (2012), a stochastic Taylor scheme tailored to the W-F diffusion. A novel alternative approach that allows for exact simulation has very recently been proposed by Jenkins and Spanò (2017). Finally, note that simulating from the W-F diffusion implies doing so for a given diffusion time unit Δt. While in population genetics diffusion time is related to the population size, in different applications it can be used to regulate the degree of sample path variance, which is linear in time and equal to $\mathrm{Var}(x(1-x)\Delta t)$.

2.3 The Poisson random field

The W-F model describes the evolution of a gene at one particular site. The PRF generalizes it to modeling an infinite collection of sites, each of which evolves independently according to the W-F model. As before, we start with a model with a population of finite size G, before taking the diffusion limit as $G \to \infty$. For a site i at which some individuals carry the mutant gene, denote by $X_i(k)$ the fraction of mutants in generation k. Each site evolves independently according to the W-F$^G(0,0)$ model. Further suppose that at each generation k a number of mutations $M \sim \mathrm{Poisson}(\nu_G)$ arise in new sites with indices $j_1, j_2, \ldots j_M$. ν_G is referred to as the immigration parameter of the PRF. Assume that each of the new mutations occurs at a new site in a single individual, with initial frequency $X_{j_m}(k) = 1/G$. Subsequently, each new process $X_{j_m}(k+1), X_{j_m}(k+2), \ldots$ evolves independently according to the W-F$^G(0,0)$ model as well (Figure 1). As with pre-existing mutant sites, each process eventually hits one of the boundaries $\{0,1\}$ and stays there (we say that the mutation is extinct/has been fixed).

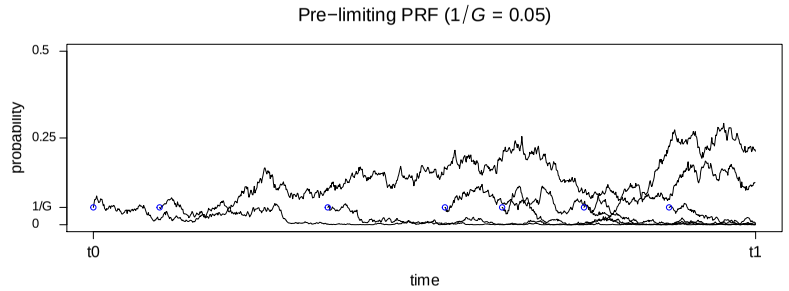


Figure 1: Evolution of mutant sites over time in the pre-limiting PRF model. The blue circles indicate mutations arising at a new site.

Consider the limit $G \to \infty$, so that after the same rescaling $t = k/G$ of time as in Section 2.2 each site evolves as an independent W-F diffusion $X_i \sim \text{W-F}(0,0)$. We also assume that $\nu_G \to \alpha$ as $G \to \infty$. This means that in the diffusion time scale the immigration rate is $G\nu_G \to \infty$, which suggests that the number of sites with mutant genes should explode. However, the initial frequency of each diffusion is $1/G \to 0$ as $G \to \infty$, and moreover 0 is an absorbing state. It can be shown (Sawyer and Hartl, 1992; Amei and Sawyer, 2010) that only

$O(G^{-1})$ of the newborn processes are not almost immediately absorbed. Therefore, there is a balance between the infinite number of newborn mutations and the infinite number of them going extinct in the first few generations, in such a way that the net immigration rate is $O(G\nu_G \times G^{-1}) = O(\alpha)$, and hence the limiting stationary measure is nontrivial. Provided that we remove from the model all sites whose frequency hits either the boundary 1 or 0, Sawyer and Hartl (1992) prove that the limiting distribution of the fractions of mutants in the interval $[0,1]$ is a Poisson random field with mean density

$$\alpha x^{-1} dx. \tag{4}$$

Interestingly, the rate measure of the PRF coincides with the distribution of weights in the one-parameter beta process given in Equation (1). This means that at equilibrium the number of sites whose frequencies $X_i(t)$ are in any given interval $(a,b]$ is Poisson distributed with rate $\alpha \int_a^b x^{-1} dx$, and these are independent for nonoverlapping intervals. Integrating (4) over $[0,1]$ shows that the number of mutations in the population that has not been fixed or gone extinct is infinite. However, most mutations are present in a very small proportion of the population.

3. Time-Varying Feature Allocation Model

The derivation of the PRF in the previous section shows that, as long as sites reaching frequency 1 or 0 are removed from the model, the equilibrium distribution of the PRF is related to the one-parameter beta process. In this section we generalize the PRF so that it is better adapted to applications in feature allocation modeling. Specifically, we identify mutant sites with features, and identify the proportion of the population having the mutant gene with the probability of the feature occurring in a data observation. The PRF can be then used in a time-varying feature allocation model whereby features arise at some unknown time point, change their probability smoothly according to a W-F diffusion process and eventually die when their probability reaches zero.

3.1 The WF-IBP

Recall from the previous section that mutant sites whose frequency hits 1 are removed from the PRF model. This means that features with high probability of occurrence can be removed from the model instantaneously, which does not make modeling sense. Instead, one expects a feature probability to change smoothly and to be removed from the model only once its probability of occurrence is *small*. A simple solution to this conundrum is to prevent 1 from being an absorbing state by using instead a W-F$(0,\beta)$ diffusion with $\beta > 0$. This is a departure from Sawyer and Hartl (1992), due to the differing modeling requirements of genetics versus feature allocation modeling. At the same time, both models let features disappear once their probability gets to 0, which is suitable from a feature allocation perspective and, as we now see, allows for a nontrival equilibrium mean density.

We shall denote the modified stochastic process as PRF(α,β). The following theorem derives the equilibrium mean density of PRF(α,β), with proof given in Appendix A:

Theorem 1 *The equilibrium mean density of the* PRF(α,β) *is*

$$l(x) = \alpha x^{-1}(1-x)^{\beta-1} dx. \tag{5}$$

In other words, the mean density of the PRF(α, β) is the Lévy measure of the two-parameter beta process, with the immigration rate α identified with the mass parameter, and β identified with the concentration parameter of the beta process. When $\beta = 1$ the one-parameter beta process is recovered. We assume that the initial distribution of PRF(α, β) is its equilibrium distribution, that is, a Poisson random field with mean density (5), so that the marginal distribution of the PRF at any point in time is the same.

We will now make the connection more precise by specifying how a PRF can be used in a time-varying feature allocation model. Denote by $X_k(t)$ the probability of feature k being active at time t and define our PRF as the stochastic process $X := \{X_k(t)\}$. Assume that at a finite number of time points $t = t_0, \ldots, t_T$ there are N_t objects whose observable properties depend on a potentially infinite number of latent features. Let D_{it} be the observation associated with object $i = 1, \ldots, N_t$ at time $t = t_0, \ldots, t_T$. Consider a set of random feature allocation matrices Z_t such that entry Z_{ikt} is equal to 1 if object i at time t possesses feature k, and 0 otherwise. Let $Z := \{Z_{ikt}\}$. Finally, let ρ_k be some latent parameters of feature k and $\rho = \{\rho_k\}$ be the set of all feature parameters. Our complete WF-IBP model is given as follows.

$$\begin{aligned} X &\sim \text{PRF}(\alpha, \beta), \\ Z_{ikt} \mid X &\overset{ind}{\sim} \text{Bernoulli}(X_k(t)), \\ \rho_k &\overset{iid}{\sim} H, \\ D_{it} \mid \rho, Z_{it} &\overset{ind}{\sim} F(\{\rho_k : Z_{ikt} = 1\}), \end{aligned} \tag{6}$$

where $i = 1, \ldots, N_t$, $t = t_0, \ldots, t_T$ and $k = 1, 2, \ldots, H$ is the prior distribution for feature parameters, and where $F(\rho)$ is the observation model for an object with a set of features with parameters ρ.

Since the feature probabilities X have marginal density (5), at each time t the feature allocation matrix Z_t has marginal distribution given by the two-parameter Indian buffet process (Thibaux and Jordan, 2007). Further, since X varies over time, the complete model is a time-varying Indian buffet process feature allocation model. The corresponding De Finetti measure would then be a time-varying beta process. More precisely, this is the measure-valued stochastic process $G = \{G(t)\}$ where

$$G(t) = \sum_{k=1}^{\infty} X_k(t) \delta_{\rho_k},$$

which has marginal distribution given by a beta process with parameters α, β and base distribution H. We denote the distribution of G as WFBP(α, β, H). We can also express the feature allocations using random measures as well. In particular, let

$$B_{it} = \sum_{k=1}^{\infty} Z_{ikt} \delta_{\rho_k}$$

be a Bernoulli process BeP$(G(t))$ with mean measure given by the beta process $G(t)$ at time t. An equivalent way to express our model (6) using the introduced random measures

is then

$$\begin{aligned} G &\sim \text{WFBP}(\alpha, \beta, H), \\ B_{it} \mid G &\sim \text{BeP}(G(t)), \\ D_{it} \mid B_{it} &\sim F(B_{it}), \end{aligned}$$

where WFBP denotes our time-varying beta process, and we have used $F(B)$ to denote the same observation model as before, but with B being a random measure with an atom for each feature, and whose location is the corresponding feature parameter. In the following, we use the notation introduced in (6) instead of in terms of beta and Bernoulli processes for simplicity.

As in the two-parameter IBP, α and β separately control the distribution of the number of features per object and the total number of features. In addition, the discrete time points $t_0, \ldots, t_T$ at which the observations are given influence the number of time units for which the W-F diffusions should be simulated. This introduces a parameter that regulates the strength of the time-dependency or accounts for gaps of varying sizes between successive observations. The time parameter can also be chosen according to the expected life time of features (see Chapter 15 of Karlin and Taylor, 1981). A detailed description of how to simulate from the model is given in Appendix B.

3.2 Related models

The WF-IBP fits a line of research that aims at introducing dependency structures into the IBP. A number of these extensions are designed to drop the exchangeability assumption from the IBP by coupling the rows and columns of the feature allocation matrix, and are thus orthogonal to our work (see Zhou et al., 2011; Miller et al., 2012; Gershman et al., 2015). What we are rather interested in achieving is partial exchangeabilty, whereby objects can be permuted independently at each time point without changing the probability of the process. This is necessary for time-dependent topic models where each time has a different set of documents and there is no correspondence between documents at different times.

One example that is more closely related to our model is the dependent Indian buffet process (dIBP) (Williamson et al., 2010a), which can also be applied to the partially exchangeable case. The dIBP uses a hierarchical Gaussian process to introduce couplings between features and items in such a way that, for an appropriate choice of the kernel, items can be permuted independently at each time. Although both the dIBP and the WF-IBP try to achieve the same goal, their methodologies are substantially different: while the former introduces dependencies at the feature-matrix level, the latter does so at the beta-process level. The construction of a dependent beta process represents a significantly different research direction, and is indeed anticipated as future work in Williamson et al. (2010a). First, an important consequence is that the WF-IBP prior describes the evolution of feature probabilities explicitly, whereas the dIBP fixes them and effectively describes a time-evolving Bernoulli process. The WF-IBP is then preferable in all settings where one is interested in a direct interpretation of the evolution of features. Second, the dIBP suffers from a less flexible boundary behaviour as it does not allow features to be born and die over time. Third, the dIBP is based on the stick-breaking construction of the IBP, which is only available for the one-parameter IBP; instead, our approach extends the two-parameter

IBP and models the dimensionality of the feature allocation matrix and its sparsity independently.

4. Fixed-K approximation

In order to give more intuition on the exact inference with the WF-IBP that is developed in Section 5, we first describe a finite-dimensional approximation where the number of features is a finite number K, and show that this marginally converges to the WF-IBP as $K \to \infty$.

Assume that the random feature allocation matrix Z_t has a fixed number of features, say K. First, let $\alpha, \beta > 0$ and consider the beta-binomial model

$$Z_{ikt} \mid \{X_k(t) = x_k(t)\} \overset{iid}{\sim} \text{Bernoulli}(x_k(t)),$$
$$X_k(t) \overset{iid}{\sim} \text{Beta}\left(\frac{\alpha\beta}{K}, \beta\right),$$

$\forall k = 1, \dots, K, \forall i = 1, \dots, N_t$. This coincides with the pre-limiting model of the two-parameter IBP presented in Ghahramani et al. (2007). Then, for each feature, think of the $\text{Beta}\left(\frac{\alpha\beta}{K}, \beta\right)$ distribution as the stationary distribution of a W-F diffusion with parameters $\frac{\alpha\beta}{K} > 0$ and $\beta > 0$. This suggests making the model time-dependent by letting each feature evolve, starting at stationarity, as an independent W-F diffusion with these parameters. Generate for all times $t = t_0, \dots, t_T$ the binary variables z_{ikt} as

$$Z_{ikt} \mid \{X_k(t) = x_k(t)\} \overset{iid}{\sim} \text{Bernoulli}(x_k(t)),$$
$$X_k \sim \text{WF}\left(\frac{\alpha\beta}{K}, \beta\right),$$

$\forall k = 1, \dots, K, \forall i = 1, \dots, N_t$. In this way, the closer two time points, the stronger the dependency between the probabilities of a given feature (Figure 2). Moreover, as we assume the W-F diffusion to start at stationarity, this construction coincides marginally with the beta-binomial model. The parameters of the W-F diffusion are positive, so that neither fixation nor absorption ever occurs and the number K of features remains constant.

4.1 Fixed-K MCMC inference

Given a set of observations D, a natural inference problem would be to recover the latent feature allocation matrices $Z = \{Z_t\}_{t=t_0}^{t_T}$ responsible for generating the observed data, the underlying feature probabilities X and their parameters ρ. Inference is straightforward; we propose the following updates.

- $Z \mid X, D, \rho$ via Gibbs sampling.
- $\rho \mid D, Z$ according to the likelihood model.
- $X \mid Z$ via Particle Gibbs.

Consider first the Gibbs sampling step to perform posterior inference over the matrices Z. Denote by $Z_{-(ik)t}$ all the components of the matrix Z_t excluding Z_{ikt}, and by Z_{i-kt} all

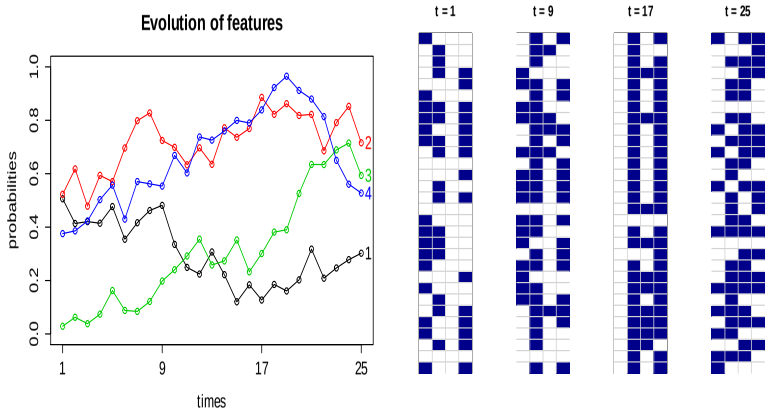


Figure 2: Left: underlying feature probabilities over time. Right: corresponding feature allocation matrices. Rows represent objects and columns features, which can be active (blue) or inactive (white).

the components in row i excluding k. We can easily derive for all t the distribution of a given component Z_{ikt} conditioning on the state of all other components $Z_{-(ik)t}$, on data D_{it}, on the parameters ρ and on the prior probability $X_k(t)$ of feature k. The full conditional probability of the entry Z_{ikt} being active is equal to

$$P(Z_{ikt} = 1 \mid Z_{-(ik)t}, X_k(t), D_{it}, \rho) \propto x_k(t) P(D_{it} \mid Z_{i-kt}, Z_{ikt} = 1, \rho). \tag{7}$$

By the same token, the full conditional probability of the entry Z_{ikt} being inactive is

$$P(Z_{ikt} = 0 \mid Z_{-(ik)t}, X_k(t), D_{it}, \rho) \propto (1 - x_k(t)) P(D_{it} \mid Z_{i-kt}, Z_{ikt} = 0, \rho). \tag{8}$$

As the matrices Z are conditionally independent given the feature probabilities X, equations (7) and (8) can be used to sample the matrices Z independently given the respective feature probabilities at each time. Note that the likelihood $P(D_{it} \mid Z_t, \rho)$ needs to be specified according to the problem at hand. A typical choice, detailed in Section 6, is the linear-Gaussian likelihood model, whose parameters can easily be integrated out (Griffiths and Ghahramani, 2011). The update $\rho \mid D, Z$ over the feature parameters is also specific to the likelihood model and, as we will illustrate, can easily be derived in conjugate models such as the linear-Gaussian one.

Consider now the Particle Gibbs (PG) step (Andrieu et al., 2010) to perform Bayesian inference on the feature trajectories continuously over the interval $[t_0, t_T]$. As the prior probability of each feature is a Beta $\left(\frac{\alpha\beta}{K}, \beta\right)$ and the column-wise sums of Z_{t_0} are realizations from binomial distributions, by conjugacy we have

$$X_k(t_0) \mid \{Z_{t_0} = z_{t_0}\} \sim \text{Beta}\left(\frac{\alpha\beta}{K} + n_{kt_0}, \beta + N_{t_0} - n_{kt_0}\right),$$

where $k = 1, \ldots, K$, and $n_{kt} := \sum_{i=1}^{N_t} z_{ikt}$ denotes the number of objects in matrix Z_t possessing feature k. This posterior distribution can be therefore used to draw the whole set of features at time t_0 and the trajectories in the interval $[t_0, t_T]$ can be obtained via PG.

More precisely, start with an initial reference trajectory $x^{r,k}_{t_0:t_T} := (x^{r,k}_{t_0}, \ldots, x^{r,k}_{t_T})$ for $k = 1, \ldots, K$ and, independently for each feature, iterate the following procedure. Draw a given number of particles from the posterior beta distribution at time t_0 and propagate them forward to time t_1 according to $\mathrm{WF}(\frac{\alpha\beta}{K}, \beta)$. At time t_1, assign each of these particles and the reference feature a weight given by the binomial likelihood of seeing that feature active in $n(t)$ objects out of $N(t)$ in $Z(t)$, i.e., $x_k(t)^{n_{kt}}(1 - x_k(t))^{N_t - n_{kt}}$. Sample the weighted particles with replacement and propagate the off-springs forward. This corresponds to using a bootstrap filter with multinomial resampling, but other choices to improve on the performance of the sampler can be made (see Andrieu et al., 2010). Repeat this procedure up to time t_T and sample only one particle at that time. Reject all the others and keep the trajectory that led to the sampled particle as the reference trajectory for the next iteration. Notice that the reference feature is kept intact throughout each iteration of the algorithm.

This procedure is illustrated more precisely by Algorithm 1, which needs to be iterated independently for each feature to provide posterior samples from their trajectories. To simplify the notation, we drop the index k and write $x^i_t | x^i_{t_0} \sim \mathrm{WF}(\frac{\alpha\beta}{K}, \beta)$ for $t \in [t_0, t_1]$ to denote the following: simulate from a W-F diffusion with initial value $x(t_0)$ and set $X(t_1) = x(t_1)$, the value of the diffusion at time t_1.

4.2 Approximation for large K

As already noted, the marginal distribution with the fixed-K approximation corresponds to the beta-binomial model, which is the pre-limiting model of the two-parameter IBP. As a consequence, at any fixed time t and as $K \to \infty$, the fixed-K approximation converges to the two-parameter generalization of the IBP, which in turns coincides with the marginal distribution of the WF-IBP. An aspect of interest is then whether the whole dynamics of the fixed-K approximation can be used as a finite approximation of the infinite model in such a way that, the larger K, the better the approximation. Two caveats need to be noted. First, only in the infinite model can features be born. For large K, however, the number of particles in the fixed-K approximation whose mass is close to zero becomes so large that, with a sufficient amount of time, some of them gain enough mass to become 'visible'. The behavior of these particles resembles the behavior of the newborn features of the infinite model. Second, as the fixed-K approximation has an upwards drift equal to $\alpha\beta/K$, only the infinite model allows for features to be absorbed at 0. This discrepancy is however overcome by the fact that, when K goes to infinity, 0 behaves like an absorbing boundary, in that features get trapped at probabilities close to 0. For these reasons, a comparison of the two models requires relabeling the particles in the finite model in such a way that, whenever a particle goes below a certain threshold $\epsilon \approx 0$, it is considered to have gone extinct, while if its probability is below ϵ and later exceeds ϵ the particle is labeled as newborn. We choose this threshold to be $\epsilon = 1/K$, as then $\lim_{K\to\infty} \epsilon = 0$.

Taking these caveats into account, we performed an empirical comparison of the fixed-K approximation with the infinite model. Consider the joint distribution at two given time points $t_0 = 0$ and $t_1 = 1$ of the feature probabilities, first in the fixed-K and then in the

Algorithm 1: Particle Gibbs

Input: Reference trajectory $x^r_{t_0:t_T}$; M.
Set $x^M_{t_0} = x^r_{t_0}$;
Draw $x^i_{t_0} \sim \mathrm{Beta}(\frac{\alpha\beta}{K} + n_{t_0},\ \beta + N_{t_0} - n_{t_0})$ for $i = 1, \ldots, M-1$;
Simulate $x^i_t | x^i_{t_0} \sim \mathrm{WF}(\frac{\alpha\beta}{K}, \beta)$ for $t \in [t_0, t_1]$ for $i = 1, \ldots, M-1$;
Set $x^M_{t_1} = x^r_{t_1}$;
Compute $w^i_{t_1} = (x^i_{t_1})^{n_{t_1}}(1 - x^i_{t_1})^{N_{t_1} - n_{t_1}}$ for $i = 1, \ldots, M$;
Sample $\bar{x}^i_{t_1}$ with $P(\bar{x}^i_{t_1} = x^i_{t_1}) \propto w^i_{t_1}$ for $i = 1, \ldots, M-1$;
Set $\bar{x}^M_{t_1} = x^r_{t_1}$;
Simulate $x^i_t | x^i_{t_1} \sim \mathrm{WF}(\frac{\alpha\beta}{K}, \beta)$ for $t \in [t_1, t_2]$ for $i = 1, \ldots, M-1$;
Set $j \leftarrow 2$;
while $t_j < t_T$ **do**
  Set $x^M_{t_j} = x^r_{t_j}$;
  Compute $w^i_{t_j} = (x^i_{t_j})^{n_{t_j}}(1 - x^i_{t_j})^{N_{t_j} - n_{t_j}} w^i_{t_{j-1}}$ for $i = 1, \ldots, M$;
  Sample $\bar{x}^i_{t_j}$ with $P(\bar{x}^i_{t_j} = x^i_{t_j}) \propto w^i_{t_j}$ for $i = 1, \ldots, M-1$;
  Set $\bar{x}^M_{t_j} = x^r_{t_j}$;
  Simulate $x^i_t | \bar{x}^i_{t_j} \sim \mathrm{WF}(\frac{\alpha\beta}{K}, \beta)$ for $t \in [t_j, t_{j+1}]$ for $i = 1, \ldots, M-1$;
  Set $j \leftarrow j + 1$;
end
Compute $w^i_{t_T} = (x^i_{t_T})^{n_{t_T}}(1 - x^i_{t_T})^{N_{t_T} - n_{t_T}} w^i_{t_{T-1}}$ for $i = 1, \ldots, M$;
Sample r_{new} with $P(r_{new} = i) \propto w^i_{t_T}$, where $i = 1, \ldots, M$;
Output: New reference trajectory $x^{r_{new}}_{t_0:t_T}$.

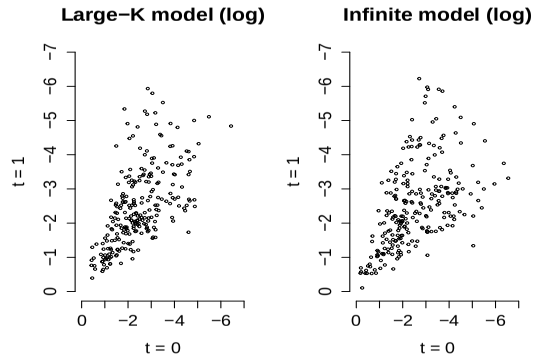


Figure 3: Scatterplot of the log feature probabilities greater than $1/K$ at time $t_0 = 0$ and at time $t_1 = 1$ to compare the fixed-K ($K = 1000$) and infinite model.

infinite model. Separately for each model, we took 1000 samples of the feature probabilities at time 0 and at time 1, excluding the ones below $1/K$. Figure 3 shows the logarithm of these values for the two models, suggesting a remarkable similarity between the two underlying joint distributions. The validity of this comparison is supported by the maximum mean discrepancy (mmd) test (Gretton et al., 2006), which does not reject the null hypothesis of the two joint distributions being the same. Although this suggests a strong similarity between the dynamics of the fixed-K approximation and the infinite model, we leave a proof of the convergence of these joint distributions as $K \to \infty$ for future work.

5. Exact MCMC inference

Building on the fixed-K approximation, we now develop a sophisticated MCMC algorithm for exact inference with the WF-IBP. The first point to notice is that, while in the fixed-K approximation the total number of features is constant over time and equal to K, in the WF-IBP model this is not a finite number. In order to use the WF-IBP for inference it is necessary to augment the state space with the features that are not seen in the feature allocation matrices, but simulating the dynamics of the PRF would require generating an infinite number of features, which is clearly unfeasible. One way to deal with this could be to resort to some sort of truncation, considering only features whose probability is greater than a given threshold and are likely to be seen in the data. We rather choose this truncation level adaptively by introducing a collection of slice variables $\{S_t\}_{t=1}^{T_t}$ and adopting conditional slice sampling (Walker, 2007; Teh et al., 2007). This scheme, which is detailed in Section 5.1, has the advantage of making inference tractable without introducing approximations.

Partition the set of features into two subsets, one containing the features that have been seen at least once among times $t = t_0, \dots, t_T$, and the other containing the features that have never been seen for all $t = t_0, \dots, t_T$, so that $X = X_{\text{seen}} \cup X_{\text{unseen}}$. Since the unseen

features cannot be identified individually based on the matrices Z, and as all features are conditionally independent given Z, we consider seen and unseen features separately. As for the seen features, we use the same Particle Gibbs scheme as in the fixed-K approximation. As for the unseen features, we simulate them via a thinning scheme. The exact MCMC inference scheme can be then summarized by the following updates.

- $Z \mid X, S, D, \rho$ via Gibbs sampling.
- $S \mid Z, X$ via slice sampling.
- $\rho \mid D, Z$ according to the likelihood model.
- $X_{\text{seen}} \mid Z$ via Particle Gibbs.
- $X_{\text{unseen}} \mid S$ via thinning.

We present each of these steps in Sections 5.1 and 5.2, which build on the simpler inference scheme developed in Section 4.

5.1 Gibbs and slice sampling

The first step is to augment the parameter space with a set of slice variables. Given the feature allocation matrices Z_t at times $t = t_0, \dots, t_T$, draw a slice variable $S_t \sim \text{Uniform}[0, x^{min}(Z_t)]$ for each time t, where $x^{min}(Z_t)$ is the minimum among the probabilities of the features seen in the feature matrix Z_t. In this way, when conditioning on the value s_t of the slice variable, we have a truncation level s_t and only need to sample the finite number of features whose probability is above this threshold (Teh et al., 2007). In other words, for all $t = t_0, \dots, t_T$, we only need to update the columns of Z_t whose corresponding feature probability $x_k(t)$ is greater than or equal to the slice variable s_t (note that these include both seen and currently unseen features). Observe that we defined a different slice variable for each time point, while an alternative choice could have been drawing a single slice from a uniform between 0 and the minimum feature probability across all times. Having multiple slice variables makes it possible to simulate fewer feature trajectories while keeping inference exact, reducing the computational cost of simulating features with small probabilities of being active. Although this comes at the cost of a larger number of parameters, in experiments we find that having multiple slices does not compromise mixing nor predictive performance.

Accounting for the slice variables, the full conditional probability of the entry Z_{ikt} being active is directly proportional to

$$P(Z_{ikt} = 1 \mid Z_{-(ik)t}, X_k(t), D_{it}, \rho) P(S_t \mid Z_{ikt} = 1, Z_{-(ik)t}) \propto$$
$$x_k(t) P(D_{it} \mid Z_{i-kt}, Z_{ikt} = 1, \rho) \frac{1[0 \le s_t \le x^{min}(Z_{-(ik)t}, Z_{ikt} = 1)]}{x^{min}(Z_{-(ik)t}, Z_{ikt} = 1)}, \tag{9}$$

while the full conditional probability of the entry Z_{ikt} being inactive is directly proportional to

$$P(Z_{ikt} = 0 \mid Z_{-(ik)t}, X_k(t), D_{it}, \rho) P(S_t \mid Z_{ikt} = 0, Z_{-(ik)t}) \propto$$
$$(1 - x_k(t)) P(D_{it} \mid Z_{i-kt}, Z_{ikt} = 0, \rho) \frac{1[0 \le s_t \le x^{min}(Z_{-(ik)t}, Z_{ikt} = 0)]}{x^{min}(Z_{-(ik)t}, Z_{ikt} = 0)}. \tag{10}$$

The term $P(S_t \mid Z_t)$ is not constant as updating Z_{ikt} for a currently unseen feature may modify the value of the minimum probability of the active features.

5.2 Particle Gibbs and thinning

Assume that the feature allocation matrices Z_t are given at the time points $t = t_0, \dots, t_T$ and we are interested in inferring the probabilities X of the underlying features. This section gives the details of inference for each of the following subpartitions of seen and unseen features: features seen for the first time at a given time t_j (for $j = 0, \dots, T$), unseen features alive at time t_0 and unseen features born between any two consecutive times t_j and t_{j+1} (for $j = 0, \dots, T-1$).

5.2.1 Seen features

As already mentioned, we can apply PG to sample from the posterior trajectories of the seen features. In particular, for features that are seen at time t_0 we can simply apply Algorithm 1 as in the fixed-K approximation by replacing the term $\frac{\alpha\beta}{K}$ with 0. This is possible as observing a feature allocation matrix Z_t updates the prior probability of features as in the posterior beta process (Thibaux and Jordan, 2007), meaning that we can draw each seen feature k from a Beta$(n_{kt},\, \beta + N_t - n_{kt})$ (recall that n_{kt} is the number of objects in which feature k is active at time t).

More generally, consider features that are seen for the first time at a given time t_j. As they cannot be identified individually based on any feature matrix Z_{t_k} for $k < j$, these features need to be drawn from the posterior beta process at time t_j and propagated both forward and backwards. Note that simulating from the W-F diffusion backwards in time is not a problem as each W-F$(0, \beta)$ diffusion is time-reversible with respect to the speed density of the PRF (Griffiths, 2003). The additional backward propagation requires adjusting Algorithm 1, already modified by replacing $\frac{\alpha\beta}{K}$ with 0, by further replacing the steps before the while loop with Algorithm 2, where for simplicity we describe the particular case of features seen for the first time at time t_1. This description can be easily generalized to features that are seen for the first time at a generic time point $t \in \{t_0, \dots, t_T\}$.

5.2.2 Unseen features

We now describe a thinning scheme to simulate the unseen features alive at time t_0. Denote the slice variable values at each time by $s_{t_0}, \dots, s_{t_T}$ and note that sampling the set of unseen features from the truncated posterior beta process at time t means drawing samples from a Poisson process on $[s_t, 1)$ with rate measure $x^{-1}(1-x)^{\beta+N_t-1}dx$ (Thibaux and Jordan, 2007), which yields only a finite number of features whose probability is larger than s_t. First, draw the unseen features from the truncated posterior beta process at time t_0. Then, propagate them forward to time t_1 according to the W-F diffusion and accept them with probability $(1-x(t_1))^{N(t_1)}$, namely the binomial likelihood of not seeing them in any object at time t_1. Finally, iterate this propagation and rejection steps up to time t_T. The details of this thinning scheme are given in Algorithm 3.

Notice that simulating the trajectories of the unseen features born between time t_0 and t_1 is equivalent to Algorithm 3 from time t_1 onwards. The only difference is that these features, drawn at time t_1, need to be simulated backwards to time t_0 as well, hence the

Algorithm 2: PG: features seen for the first time at time t_1.

Input: Reference trajectory $x^r_{t_0:t_T}$; M.
Set $x^M_{t_1} = x^r_{t_1}$;
Draw $x^i_{t_1} \sim \text{Beta}(n_{t_1},\, \beta + N_{t_1} - n_{t_1})$ for $i = 1, \dots, M-1$;
Set $x^M_{t_0} = x^r_{t_0}$;
Simulate $x^i_t | x^i_{t_1} \sim \text{WF}(0, \beta)$ backwards for $t \in [t_1, t_0]$ for $i = 1, \dots, M-1$;
Set $x^M_{t_2} = x^r_{t_2}$;
Simulate $x^i_t | x^i_{t_1} \sim \text{WF}(0, \beta)$ for $t \in [t_1, t_2]$ for $i = 1, \dots, M-1$;
Compute $w^i_{t_0} = (1 - x^i_{t_0})^{N_{t_0}}$ for $i = 1, \dots, M$;
Compute $w^i_{t_2} = (x^i_{t_2})^{n_{t_2}} (1 - x^i_{t_2})^{N_{t_2} - n_{t_2}} w^i_{t_0}$ for $i = 1, \dots, M$;
Draw $\bar{x}^i_{t_2}$ with $P(\bar{x}^i_{t_2} = x^i_{t_2}) \propto w^i_{t_2}$ for $i = 1, \dots, M-1$;
Set $\bar{x}^M_{t_2} = x^r_{t_2}$;
Simulate $x^i_t | x^i_{t_2} \sim \text{WF}(0, \beta)$ for $t \in [t_2, t_3]$ for $i = 1, \dots, M-1$;
Set $j \leftarrow 3$;

Algorithm 3: Thinning: unseen features alive at time t_0

Draw from a Poisson process on $[s_{t_0}, 1)$ with rate measure $\alpha x^{-1}(1-x)^{\beta+N_{t_0}-1}dx$ and denote by $\{x^i_{t_0}\}_{i \in A}$ the resulting candidate particles;
Set $j \leftarrow 1$;
while $t_j < t_T$ **do**
  Simulate $x^i_l | x^i_{t_j} \sim \text{WF}(0, \beta)$ for $t \in [t_j, t_{j+1}]$ for all $i \in A$;
  Accept $x^i_{t_{j+1}}$ with probability $(1 - x^i_{t_{j+1}})^{N_{t_{j+1}}}$ for all $i \in A$;
  Remove from A the indices of the rejected particles;
  Set $j \leftarrow j + 1$;
end
Output: Trajectories $\{x^i_{t_0:t_T}\}_{i \in A}$ of the unseen features alive at time t_0 from the truncated PRF(α, β).

Algorithm 4: Thinning: unseen features born between time t_0 and t_1

Draw from a Poisson process on $[s_{t_1}, 1)$ with rate measure $\alpha x^{-1}(1-x)^{\beta+N_{t_1}-1}dx$ and denote by $\{x^i_{t_1}\}_{i\in A}$ the resulting candidate particles;
Simulate $x^i_t | x^i_{t_0} \sim \mathrm{WF}(0,\beta)$ for $t \in [t_0, t_1]$ for all $i \in A$;
for *all* $i \in A$ **do**
  if $x^i_{t_0} > s_{t_0}$ **then**
    Reject $x^i_{t_0}$;
    Set $A \leftarrow A \setminus \{i\}$;
  else
    Accept $x^i_{t_0}$ with probability $(1 - x^i_{t_0})^{N_{t_0}}$;
  end
end
Set $j \leftarrow 1$;
while $t_j < t_T$ **do**
  Simulate $x^i_t | x^i_{t_j} \sim \mathrm{WF}(0,\beta)$ for $t \in [t_j, t_{j+1}]$ for all $i \in A$;
  Accept $x^i_{t_{j+1}}$ with probability $(1 - x^i_{t_{j+1}})^{N_{t_{j+1}}}$ for all $i \in A$;
  Remove from A the indices of the rejected particles;
  Set $j \leftarrow j + 1$;
end
Output: Trajectories $\{x^i_{t_0:t_T}\}_{i\in A}$ of the unseen features born between time t_0 and t_1 from the truncated $\mathrm{PRF}(\alpha, \beta)$.

additional backward simulation followed by the rejection step in the for loop of Algorithm 4. If a feature that is simulated backwards from time t_1 to t_0 has probability 0 by time t_0, then it is a newborn feature and is accepted with probability 1. On the other hand, if its probability at time t_0 is between 0 and s_{t_0}, the particle belongs to the category of features that were alive and unseen at time t_0. Accepting them with probability $(1 - x(t_0))^{N_{t_0}}$ compensates for the features that were below the truncation level s_{t_0} in Algorithm 3 and were thus not simulated at time t_0. In this way, only the features whose mass is below the slice variables s_t at all times $t \in \{t_0, \ldots, t_T\}$ are not simulated. The exactness of the overall MCMC scheme is preserved by the fact that those features are inactive in all the feature allocation matrices by the definition of slice variable.

Finally note that, for simplicity's sake, Algorithm 4 describes only how to simulate the unseen features that were born between times t_0 and t_1, but the procedure needs to be generalized to account for the features born between any two consecutive time points t_j and t_{j+1}, where $j = 0, \ldots, T-1$. In order to do this, it is sufficient to draw the candidate particles at every time t_{j+1}, with $j = 0, \ldots, T-1$, propagate them backwards until time t_0 and thin them as follows: if their mass exceeds s_t at any $t \in \{t_0, \ldots, t_T\}$, then they are rejected; otherwise, at each backward propagation to time $t \in \{t_0, \ldots, t_{T-1}\}$ they are accepted with probability $(1 - x(t))^{N_t}$.

6. Application: linear-Gaussian likelihood model

The WF-IBP we have described defines a prior over latent feature allocation matrices Z and the corresponding feature probabilities X exhibiting a dependency structure over time. The next step is to relate Z to the observed data by means of a given likelihood model. The first model we explore is the linear-Gaussian likelihood, a very common choice in latent feature models (e.g., Doshi-Velez and Ghahramani, 2009; Griffiths and Ghahramani, 2011; Gershman et al., 2015).

Assume that the collection of observations O_t at time $t = t_0, \ldots, t_T$ is in the form of an $N \times D$ matrix generated by the matrix product $O_t = Z_t \times A + \epsilon_t$. Z_t is the $N \times K$ binary matrix of feature assignments at time t and A is a $K \times D$ factor matrix whose rows represent the feature parameters ρ. The matrix product is the way Z_t determines which features are active in each observation, and ϵ_t is a $N \times D$ Gaussian noise matrix, whose entries are assumed to be distributed as independent $\mathcal{N}(0, \sigma^2_X)$. A typical inference problem is to infer both the feature allocation matrices Z and the factor matrix A. In order to achieve this, we place on each element of A an independent prior $\mathcal{N}(0, \sigma^2_A)$ and on the hyper-parameter σ^2_A an inverse-gamma prior $\Gamma^{-1}(1,1)$. This choice of priors is convenient as it is easy to obtain the posterior distributions of σ^2_A and A (for the case $T = 1$, see Doshi-Velez and Ghahramani, 2009).

For simplicity of notation, consider a fixed number of features K. Denote by $\bar{Z}$ the $TN \times K$ matrix obtained by concatenating the feature matrices Z vertically, and by $\bar{O}$ the $TN \times D$ matrix obtained by combining the observations $\{O_t\}_{t=t_0}^{t_T}$ in the same way. The posterior of A is matrix Gaussian with the following mean μ^A (a $K \times D$ matrix) and, for each column of A, the following covariance matrix Σ^A (a $K \times K$ matrix).

$$\mu^A = \left(\bar{Z}^T\bar{Z} + \frac{\sigma^2_X}{\sigma^2_A} I\right)^{-1} \bar{Z}^T\bar{O}$$
$$\Sigma^A = \sigma^2_X \left(\bar{Z}^T\bar{Z} + \frac{\sigma^2_X}{\sigma^2_A} I\right)^{-1}.$$

By conjugacy, the posterior distribution for σ^2_A is still inverse gamma with updated parameters, namely

$$\sigma^2_A \sim \Gamma^{-1}\left(1 + \frac{1}{2}KD, 1 + \frac{1}{2}\sum_k \sum_d A^2_{kd}\right).$$

6.1 Simulations and results

We tested the WF-IBP combined with a linear-Gaussian likelihood on a variety of synthetic data sets. Starting from the fixed-K approximation, we generated $N = 50$ observations at each of 40 equally-spaced time points as in the linear-Gaussian model. The true factor matrix A contained $K = 3$ latent features in the form of binary vectors of length $D = 30$. Their probability of being active was determined continuously over time by three independent W-F$(1,1)$ diffusions, simulated for 0.01 diffusion time-units between every two consecutive time points. The resulting observations were corrupted by a large amount of noise ($\sigma_X = 0.5$). 1000 iterations of the overall algorithm were performed, choosing a

burn-in period of 100 iterations and setting the time-units and drift parameters of the W-F diffusion equal to the true ones in the PG update. As ground truth was available, we were able to test the ability of the algorithm to recover the true feature allocation matrices, the latent feature parameters and their probabilities over time.

Figure 4-top-left compares the true underlying feature matrices at times $t = \{1, 14, 27, 40\}$ in terms of the most frequently active features in the posterior mean matrices, where a feature is set to be active if that is the case in more than half of the samples of the Markov chain. The resulting mean matrices almost perfectly match the true underlying feature matrices. Figure 4-top-right compares the trajectories over time of the true feature probabilities with the inferred ones. The latter tend to be less than two standard deviations away from the former, meaning that the true feature trajectories are closely tracked. Figure 4-bottom-left compares the three features represented by the true factor matrix A and the ones in the posterior mean matrix $\hat{A}$, showing that the algorithm was able to recover accurately the hidden features underlying the noisy observations. Figure 4-bottom-right plots the log-likelihood at each iteration, showing that the algorithm converged quickly, namely in fewer than 50 iterations.

Then, we tested the ability of the slice sampler-based algorithm to recover the correct number of latent features when given a similar set of synthetic data, this time consisting of 4 latent features evolving over 6 time points. The algorithm was initialized with one feature and run for 3300 iterations with a burn-in period of 1000 iterations. As in the finite case, the true underlying feature allocation matrices and feature probabilities were closely recovered as illustrated by the top row of Figure 5. The bottom row of Figure 5 shows that the features were reconstructed accurately and their correct number detected in about 700 iterations.

Finally, we focused on the ability of Particle Gibbs to track the feature trajectories in a wider variety of settings and under model misspecification. We generated a set of feature allocation matrices obtained by letting three feature probabilities evolve over time, first by simulating standard W-F diffusions and then by introducing jumps and spikes. Figure 6 confirms the robustness of the algorithm in the presence of mismatches between the W-F diffusion and the true process determining the feature trajectories. We simulated a set of 20 feature allocation matrices under an increasing amount of mismatch, first by introducing a jump of increasing size from time 10 to 11, and then by adding a spike of increasing sharpness at time 10. Figure 6-left shows that, although larger jump sizes lead to larger discrepancies between the true and inferred trajectories around the jump, the former are always less than two standard deviations away from the latter. Figure 6-right shows that in all three cases the algorithm is able to detect the presence of the spike at time 10. Figure 7 illustrates the performance of the algorithm on a decreasing number of observations. The results show that the posterior mean of the target distribution closely corresponds to the true feature probabilities and, given a sufficient number of observations, always fall within the interval given by two standard deviations about the posterior mean.

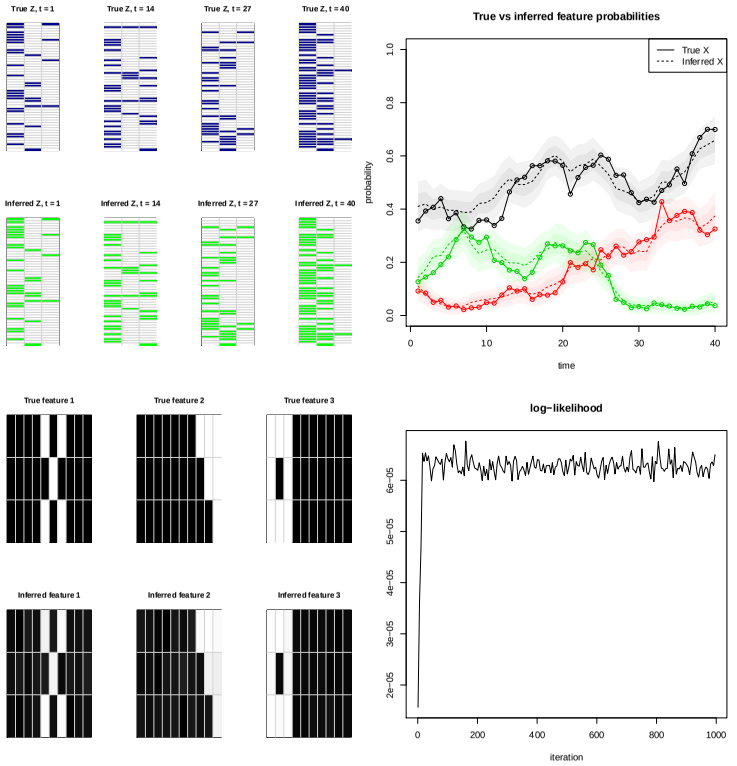


Figure 4: Fixed-K approximation. Top-Left: Subset of true feature allocation matrices vs inferred ones. Top-Right: True vs inferred feature probabilities over time (the dark and the light shaded areas respectively indicate one and two standard deviations about the posterior mean). Bottom-Left: True vs inferred features (black and white entries respectively correspond to 0 and 1, while the shades of grey to the values in between). Bottom-Right: Convergence of the log-likelihood of the observed data.

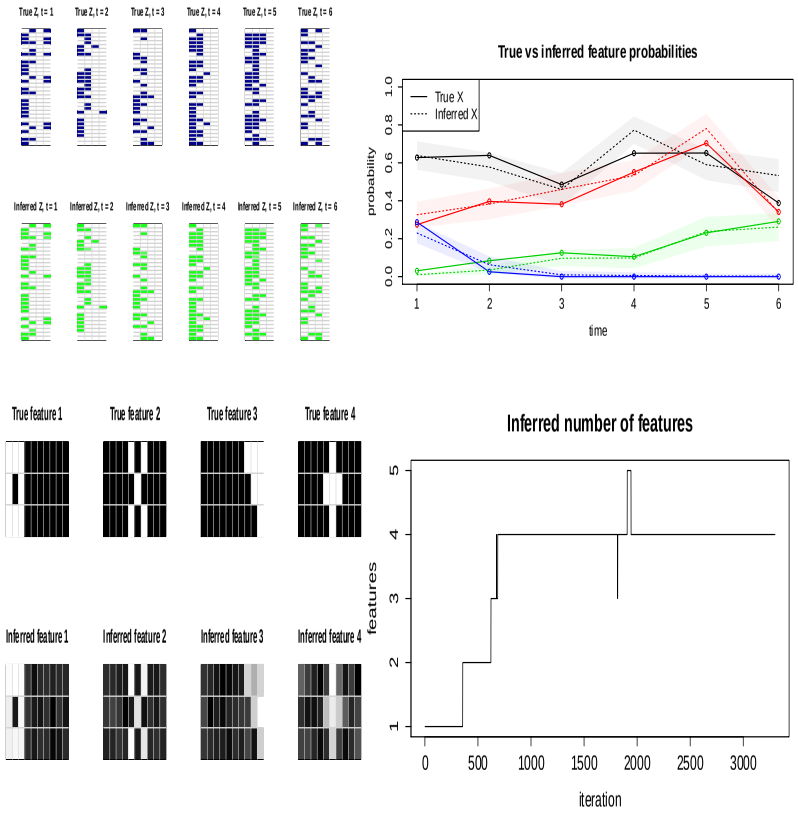


Figure 5: WF-IBP. Top-Left: True vs inferred feature allocation matrices. Top-Right: True vs posterior mean feature trajectories (the shaded areas represent one standard deviation). Bottom-Left: Comparison between true and inferred features. Bottom-Right: Convergence to the true number of features.

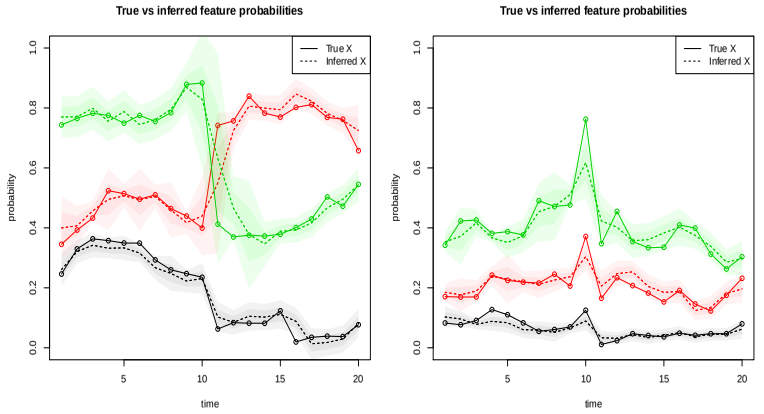


Figure 6: Comparison between the posterior means and the true feature probabilities under an increasing amount of model misspecification. The dark and light shaded areas respectively correspond to 1 and 2 standard deviations about the posterior mean. Left: Jump of increasing size between times 10 and 11. Right: Spike of increasing sharpness at time 10.

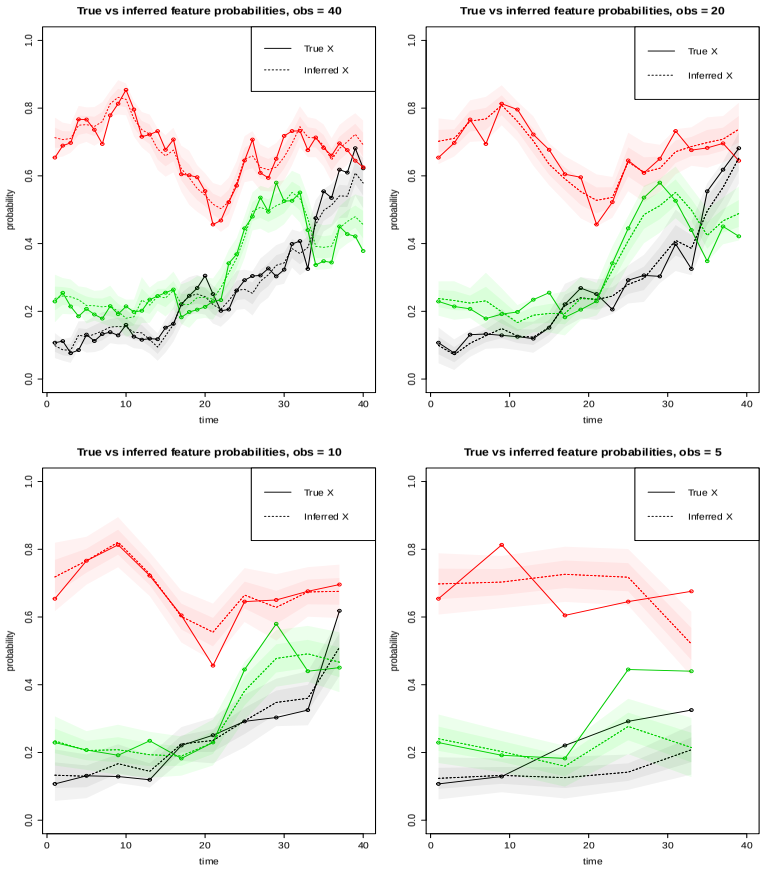


Figure 7: Comparison between true feature probabilities and posterior means obtained via Particle Gibbs for a varying number of observations. The dark and light shaded areas respectively correspond to 1 and 2 standard deviations about the posterior mean.

7. Topic modeling application

In this section we apply the WF-IBP to the modeling of corpora of time-stamped text documents. This is a natural application as documents can be seen as arising from an unknown number of latent topics whose popularity is evolving over time. A related model to achieve this goal is described in Blei and Lafferty (2006), where a Gaussian state space model captures the evolution of topics in such a way that both the content of topics and their proportions evolve over time. This work has had a great impact in the topic modeling community, and anticipates a number of directions for future work that we address here. Specifically their model is parametric, in that the number of topics K is fixed and needs to be pre-specified. The authors claim that it would desirable to drop this assumption to have a more flexible model and, in particular, foresee a process involving births and deaths of topics. The WF-IBP topic model elegantly achieves these goals. Unlike Blei and Lafferty (2006) we focus on the evolution of topic probabilities rather than topic contents, noting that the modeling of time-varying topic contents is orthogonal to our work and could be incorporated into the WF-IBP in future developments. Another class of models called Dirichlet processes aim at modeling the evolution of topics in a time-dependent and nonparametric way. Some of these models, however, assume the evolution of topic probabilities to be unimodal (e.g., Rao and Teh, 2009), while others are HDP-based (Ahmed and Xing, 2012; Dubey et al., 2013) and implicitly assume a positive correlation between the probability of a topic being active and the proportion of that topic within each document. Coupling topic proportions and topic probabilities is undesirable as rare topics may account for a large proportion of words in the few documents in which they appear. Our nonparametric topic model decouples the probability of a topic and its proportion within documents and offers a flexible way to model topic evolutions over time. We achieve this by incorporating time-dependency into the focused topic model presented in Williamson et al. (2010b), which makes use of the IBP to select the finite number of topics that each document treats.

7.1 The WF-IBP topic model

First consider the case in which the number of topics K underlying the corpus of seen documents is known. Define topics as probability distributions over a dictionary of D words and model them as $(\rho_k)_{k=1}^K \overset{iid}{\sim} \text{Dirichlet}(\bar{\eta})$, given a vector $\bar{\eta}$ of length D. Let ρ be the resulting vector and assume the components of $\bar{\eta}$ to be all equal to a constant $\eta > 0$. Consider the usual setting in which the time-dependent popularity of topic (feature) k is denoted by X_k and the binary variables Z_{ikt} indicate whether document i contains topic k at time t. Then, for all $t = t_0, \dots, t_T$ and $k = 1, \dots, K$, sample

$$\begin{aligned}
\theta_{it} \mid \{Z_{it} = z_{it}, \phi_t = \phi'_t\} &\sim \text{Dirichlet}(z_{it} \circ \phi'_t), \quad \forall i = 1, \dots, N_t, \\
\phi_{kt} &\sim \text{Gamma}(\gamma, 1), \\
(Z_{ikt})_{i=1}^{N_t} \mid \{X_k = x_k(t)\} &\overset{iid}{\sim} \text{Bernoulli}(x_k(t)), \\
X_k &\sim \text{WF}\left(\frac{\alpha\beta}{K}, \beta\right),
\end{aligned}$$

where ϕ_{kt} is the kth component of ϕ_t, a K-long vector of topic proportions, and θ_{it} the ith row of θ_t, a $N_t \times K$ matrix with the distributions over topics for each document at

time t. The operation $z_{it} \circ \phi_t$ stands for the Hadamard product between z_{it} and ϕ_t and the Dirichlet is defined over the positive components of the resulting vector. While the topic allocation matrix Z_t encodes which subset of the K topics appears in each document at time t, the variables ϕ_{kt} are related to the proportion of words that topic k explains within each document. Unlike HDP-based models, these two quantities are here modeled independently.

For every document $i = 1, \dots, N_t$, draw the total number of words from a negative-binomial $W_{it} \sim \text{NB}(\sum_k z_{ikt}\phi_{kt}, 1/2)$ and, for each word w_{ilt}, $l = 1, \dots, W_{it}$, sample first the topic assignment

$$a_{ilt} \mid \{\theta_{it} = \theta'_{it}\} \sim \text{Categorical}(\theta'_{it})$$

and then the word

$$w_{ilt} \mid \{a_{ilt} = a'_{ilt}, \rho = \rho'\} \sim \text{Categorical}(\rho'_{a'_{ilt}}).$$

Assume now that the number of potential topics K needs to be learned from the data. The nonparametric extension of this model is easily obtained by replacing the process generating the topic allocation matrices with the WF-IBP, so that topics arise as in the PRF and evolve as independent WF$(0, \beta)$. The feature allocation matrices can be drawn as described in Section 8. In this way, we obtain a time-dependent extension of the IBP compound Dirichlet process presented in Williamson et al. (2010b).

7.2 Posterior inference

In order to infer the latent variables of the model, it is convenient to integrate out the parameters ρ and θ. This can be done easily thanks to the conjugacy between the Dirichlet and the Categorical distribution. In this way, we can run a Gibbs sampler for posterior inference only over the remaining latent variables and are able to follow the derivation of conditionals given by Williamson et al. (2010a). Note that, in our case, we have introduced the slice variable and do not integrate out the topic allocation matrix. Denote by W the complete set of words and by A the complete set of topic assignments a_{ilt} for all times $t = t_0, \dots, t_T$, documents $i = 1, \dots, N_t$ and words $l = 1, \dots, W_{it}$. Denote by S_t the slice variable and by W_t the complete set of words at time t. The conditional distributions that we need to sample from for all times $t = t_0, \dots, t_T$ are

$$\begin{aligned}
&p(A_t \mid Z_t, w, \phi_t),\\
&p(\phi_t, \gamma \mid A_t, W_t, Z_t),\\
&p(S_t \mid Z_t, X(t)),\\
&p(Z_t \mid A_t, X(t), \phi_t, S_t).
\end{aligned}$$

Conditioning on all the other topic assignments a_{-il}, each topic assignment a_{ilt} can be sampled from

$$p(a_{ilt} = k \mid a_{-il}, Z_t, W, \phi_t) \propto (n_k^{w_{il}} + \eta)\frac{n_{kt}^i + \phi_{kt} z_{ikt}}{n_k + \eta D - 1},$$

where $n_k^{w_{il}}$ denotes the number of times that word w_{il} has been assigned to topic k excluding assignment a_{il}, n_{kt}^i the number of words assigned to topic k in document i excluding assignment a_{il} and n_k the total number of words assigned to topic k.

After placing a hyper-prior on $p(\gamma)$, we can sample ϕ and γ via a Metropolis-Hastings step. Indeed, we know that

$$p(\phi_{kt}, \gamma \mid A, Z_t) \propto \frac{\phi_{kt}^{\gamma-1} e^{-\phi_{kt}}}{\Gamma(\gamma)} p(\gamma) \prod_{i=1}^{N_t} \frac{\Gamma(\phi_{kt} z_{ikt} + n_{kt}^i)}{\Gamma(\phi_{kt} z_{ikt}) n_{kt}^i! 2^{\phi_{kt} z_{ikt} + n_{kt}^i}}.$$

Conditioning on Z_t, the slice variable is sampled according to its definition:

$$p(S_t \mid Z_t, X(t)) = \frac{1}{x^{min}(t)} 1_{s_t}([0, x^{min}(t)]),$$

where $x^{min}(t)$ is the minimum among the probabilities of the active topics at time t. As for the feature allocation matrices Z, we sample only the finite number of its components whose topic probability $x_k(t)$ is greater than the slice variable S_t. Assume we are sampling each entry Z_{ikt} sequentially and denote respectively by $x_1^{min}(t)$ and $x_0^{min}(t)$ the minimum active topic probability in the cases $Z_{ikt} = 1$ and $Z_{ikt} = 0$. Let n_{ikt} denote the total number of words assigned to topic k in document i at time t. Then we have that

$$p(Z_{ikt} = 1 \mid A, x_k(t), \phi_{kt}) = \begin{cases} 1, & \text{if } n_{ikt} > 0 \\ \frac{x_k(t) x_0^{min}(t)}{x_k(t) x_0^{min}(t) + 2^{\phi_{kt}}(1 - x_k(t)) x_1^{min}(t)}, & \text{if } n_{ikt} = 0. \end{cases}$$

$$p(Z_{ikt} = 0 \mid A, x_k(t), \phi_{kt}) = \begin{cases} 0, & \text{if } n_{ikt} > 0 \\ \frac{2^{\phi_{kt}}(1 - x_k(t)) x_1^{min}(t)}{x_k(t) x_0^{min}(t) + 2^{\phi_{kt}}(1 - x_k(t)) x_1^{min}(t)}, & \text{if } n_{ikt} = 0. \end{cases}$$

The full conditional distributions presented so far are derived in Appendix C. As in the linear-Gaussian case, when considering the probability of setting $Z_{ikt} = 1$ for a feature that is currently inactive across all observations at time t, it is necessary to jointly propose a new value ϕ_{kt} by drawing it from its prior distribution Gamma$(\gamma, 1)$. Finally, inference on the trajectories of the topic probabilities is a direct application of the Particle Gibbs and thinning scheme outlined for the general case.

7.3 Topic model: simulation and results

At each of 4 time points, a small corpus of $N = 30$ documents was simulated by selecting up to $K = 4$ latent topics for each document and picking words from a dictionary of $D = 100$ words. The hyper-parameter of the Dirichlet prior over words was chosen to be a vector with components equal to $\eta = 0.1$, a Gamma(5,1) hyper-prior was placed on γ and we let topic probabilities evolve as independent W-F$(1, 1)$ diffusions with 0.1 diffusion time-units between each observation. Assume K is known and focus on inference over the remaining parameters. Fix the time-units and drift parameters of the W-F diffusion to their true values in the PG update. We ran the Gibbs sampler for 3000 iterations with a burn-in period of 300 iterations. The log-likelihood converged in about 500 iterations (Figure 8)

and the algorithm was able to infer closely the latent topic allocation matrices (Figure 9-left). The percentage of words assigned to the correct topics were 81% at t_1, 82% at t_2, 83% at t_3 and 85% at t_4.

A Monte Carlo estimate for the probability of word $w = 1, \ldots, D$ under topic k is

$$\hat{\rho}_{kw} = \frac{n_k^w + \eta}{n_k + D\eta},$$

where n_k^w is the number of times word w has been assigned to topic k, n_k is the total number of words assigned to topic k and D is the number of words in the dictionary. A Monte Carlo estimate for the probability of topic k in document i is

$$\hat{\theta}_{ikt} = \frac{n_{ikt} + z_{ikt}\phi_{kt}}{\sum_k (n_{ikt} + z_{ikt}\phi_{kt})}. \tag{11}$$

These two quantities are given by the posterior mean of the Dirichlet distribution under a categorical likelihood. $\hat{\rho}_{kw}$ has been used to plot the posterior distribution over words in Figure 9-right. These results confirm the ability of the algorithm to recover ground truth and provide useful information both at word and topic level.

Finally, we tested the ability of the algorithm to reconstruct topics when presented with a decreasing number of observed documents. 100 documents were simulated at each of two time points as described above. Figure 10 shows how well the probability of the 10 most likely words of the first topic was reconstructed for $N = 60$, 40, 20 and 10 observed documents per time point. As expected, the more documents are observed the more accurate the reconstruction of topics is, with a drop in performance when only 10 documents per time point are observed.

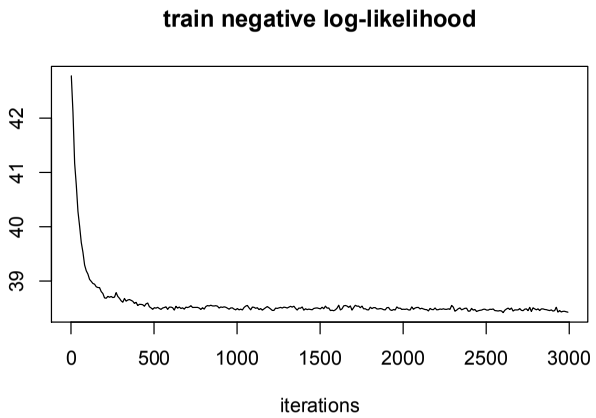


Figure 8: Convergence of the train negative log-likelihood.

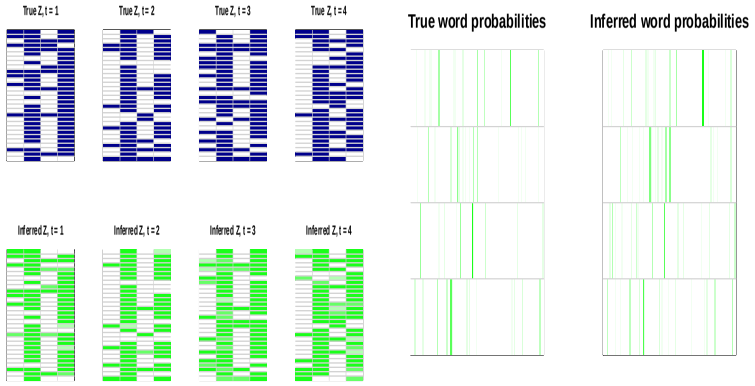


Figure 9: Left: Comparison between the true and the posterior mean topic allocation matrices at each time. Right: True vs inferred distributions over words for each topic. Each row is a topic ($K = 4$) and each column is a word from the dictionary ($D = 100$) (the darker the green, the larger the probability of the corresponding word).

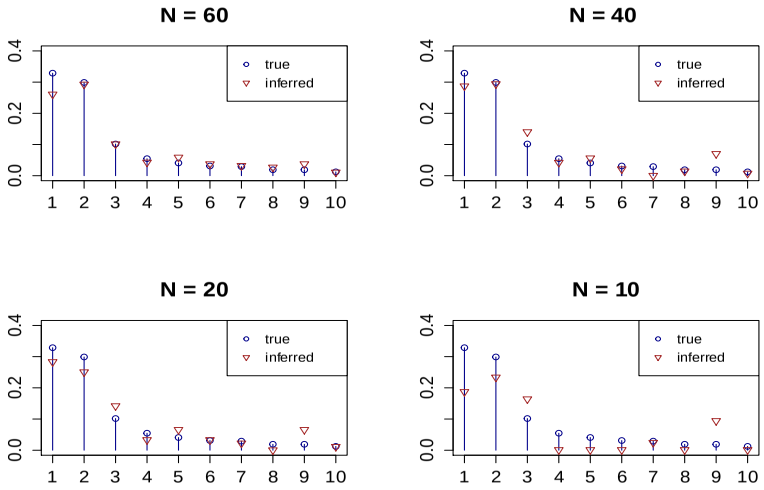


Figure 10: Comparison, given different numbers N of observed documents, between the true probabilities of the 10 most likely words within a given topic and the inferred probabilities of those words for that topic. The more documents are observed, the better the reconstruction of the topic is.

7.3.1 Comparison: static model and hierarchical model

We now demonstrate the advantages of modeling time-dependency by comparing our model with two alternative versions. First, we consider a static counterpart where time is not modeled and thus information about the time-stamp of documents is not exploited. Second, we consider a hierarchical model where feature probabilities at each time point are distributed as conditionally independent beta processes given a lower-layer beta process (Thibaux and Jordan, 2007); note that in this model observations at different time points are modeled separately, but information on the order of the time points is ignored.

In particular, we investigate whether incorporating time into the model improves upon test-set perplexity, a measure widely used in topic modeling settings that assesses the ability of topic models to generalize to unseen data. Given the model parameters Φ, perplexity on documents $D_{test} := \{d_i\}_{i=1}^{M}$ is defined as

$$\text{perplexity}(D_{test} \mid \Phi) = \exp\left(-\frac{\sum_{i=1}^{M} \log p(d_i \mid \Phi)}{\sum_{i=1}^{M} W_i}\right),$$

where W_i denotes the number of words in document d_i. As we assume that words within each document are drawn independently given the model parameters Φ, the probability of document d_i can be computed as

$$p(d_i \mid \Phi) = \prod_{l=1}^{W_i} p(w_{il} \mid \Phi),$$

where

$$p(w_{il} \mid \Phi) = \sum_{k=1}^{K} \theta_{ik}\rho_{kl},$$

recalling that θ_{ik} is the probability of a generic word belonging to topic k in document i and ρ_{kl} is the probability of word l under topic k. These two quantities can be approximated at each iteration of the MCMC algorithm by their current values $\hat{\theta}_{ik}^{(s)}$ and $\hat{\rho}_{kl}^{(s)}$, so that we can approximate the probability of each word by averaging over S samples of the Markov Chain.

$$\hat{p}(w_{il} \mid \Phi) = \frac{1}{S}\sum_{s=1}^{S}\sum_{k=1}^{K} \hat{\theta}_{ik}^{(s)}\hat{\rho}_{kl}^{(s)}.$$

Note that the perplexity is inversely proportional to the likelihood of the data and thus lower values indicate better performance. Chance performance, namely assuming each word to be picked uniformly at random from the dictionary, yields a perplexity equal to the size D of the dictionary.

Different percentages of words were held-out and the model was trained on the remaining data. Testing the model on held-out words is a way to avoid comparing different hyper-parameters, as different treatments of the hyper-parameters could strongly affect the results (Asuncion et al., 2009). For all three models, a dictionary of $D = 1000$ words was used

to generate 30 documents at each of 9 time points. The number of features was fixed to 4 and the algorithms were run 3000 iterations with a burn-in period of 300 iterations. Even though all three algorithms approximately recover the true topic allocations matrices, incorporating time leads to a closer match. This can be measured, for instance, by the Frobenius norm of the difference between each true and inferred Z_t. Table 1 shows that at each time the dynamic model leads to a lower discrepancy with the true topic allocation matrix. Figure 11 shows the posterior trajectories of topic probabilities inferred by the dynamic model and compares them with the posterior feature probabilities inferred by the hierarchical model and the constant values inferred by the static model. It can be observed that the trajectories inferred by the dynamic model are both smoother and closer to the ground truth. Finally, Figure 12 compares the test-set perplexity of the three models (recall that in this case chance performance results in a perplexity of $D = 1000$). As expected, although the hierarchical version performs better than the static counterpart that neglects time-stamp information, it is outperformed by our dynamic model. Indeed, while in the WF-IBP observations that are closer in time exhibit a stronger dependency, the HBP does not explicitly impose an ordering of the time points. The results show that having a suitable model for time dependencies improves on the ability to recover ground truth as well as to generalize to unseen data.

Table 1: Frobenius norm of the difference with the true Z_t. The lower, the better.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
Dynamic	**1.78**	**1.37**	**0.72**	**2.99**	**3.24**	**2.01**	**2.59**	**2.45**	**2.88**
Hierarchical	1.94	1.51	2.80	3.11	3.52	2.91	2.65	2.59	3.24
Static	3.63	4.23	2.84	3.01	3.55	5.27	4.89	3.31	5.90

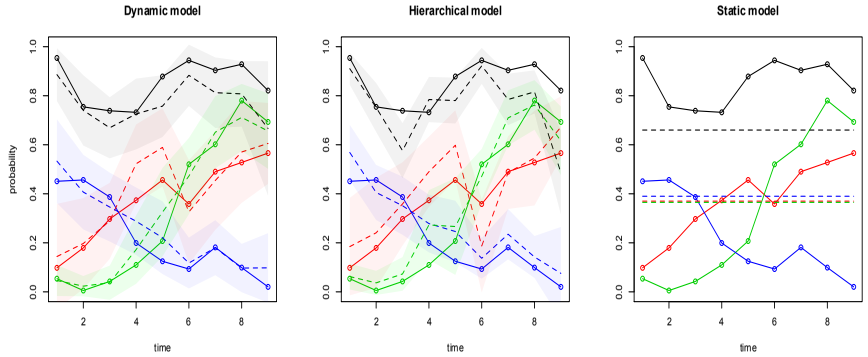


Figure 11: Comparison between true and inferred feature probabilities (respectively continuous and dotted lines) in our fixed-K topic model (left), in the hierarchical version (center) and in the static version (right).

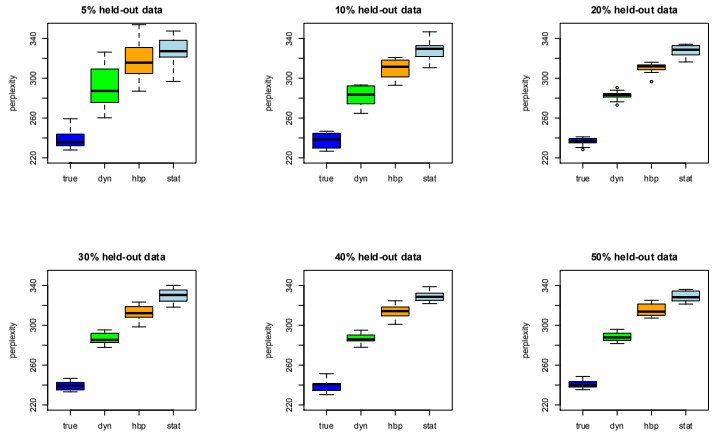


Figure 12: Boxplots of test-set perplexity for different percentages of held-out data for the true model (true), our dynamic model (dyn), the hierarchical version (hbp) and the static version (stat). Each boxplot was obtained by computing the perplexity after holding-out 10 different random subsets of words in the data. Lower values indicate better performance.

7.3.2 Real-world data experiments

We used the WF-IBP topic model to explore the data set consisting of the full text of 5811 NIPS conference papers published between 1987 to 2015.[1] We pre-processed the data and removed words appearing more than 5000 times or fewer than 250 times. The remaining number of word tokens was 4 728 892 with a vocabulary size of 348 672 unique words. Our goal was to discover what topics appear in the corpus and to track the evolution of their popularity over these 29 years.

We set the hyperparameters α and β equal to 1 and the time step to 0.12 diffusion time-units per year so as to reflect realistic evolutions of topic popularity. The Markov chain was run for 2000 iterations with a burn-in period of 200 iterations, setting $\eta = 0.001$ and placing a Gamma(5,1) hyper-prior on γ.

Qualitative results One of the qualitative advantages of modeling time dependency explicitly is that interesting insights into the evolution of topics underlying large collections of documents can be obtained automatically, and uncertainty in the predictions naturally incorporated. The 12 most likely words of 32 topics found in the corpus together with the evolution of their topic proportions are given in Figure 13, where the shaded areas

1. The data set is available at `https://archive.ics.uci.edu/ml/datasets/NIPS+Conference+Papers+1987-2015`.

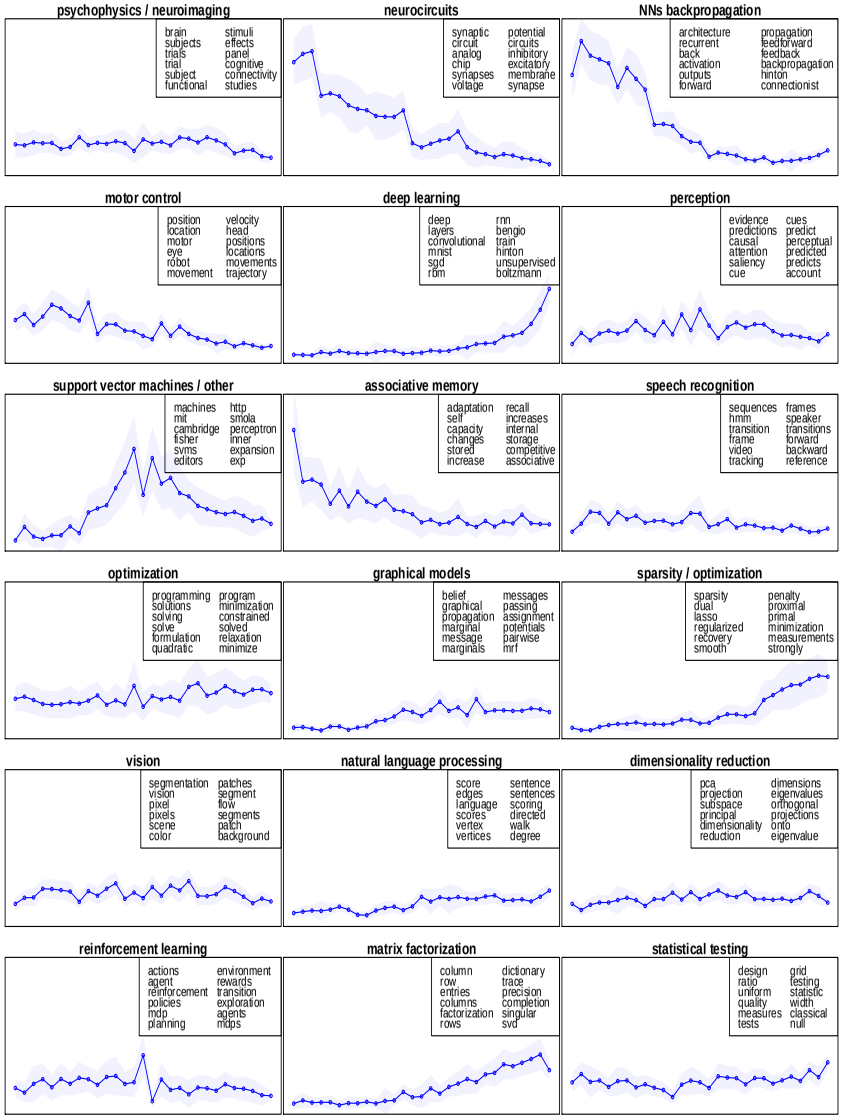
represent one standard deviation around the posterior means. As topics are defined by their distribution over words, it is possible to label them by looking at their most likely words. We observe that, with very few exceptions, the topics detected in the corpus are meaningful and easily interpretable. Figure 14 compares how the popularity of three different approaches to machine learning evolved over time. The results indicate that standard neural networks ('NNs backpropagation') were extremely popular until the early 90s. After this, they went through a steady decline, only to increase in popularity later on. This confirms the well known fact that NNs were largely forsaken in the machine learning community in the late 90s (see LeCun et al., 2015). On the other hand, it can be observed that the popularity of deep architectures and convolutional neural networks ('deep learning') steadily increased over these 29 years, to the point that deep learning became the most popular among all topics in NIPS 2015.

Another key benefit of the WF-IBP over alternative nonparametric topic models is that the overall probability of topics and their proportion within documents are modeled separately, which allows rare topics to be the predominant subject within a few documents. This can be observed by comparing the rarest topic probabilities with the corresponding within-document topic proportions. In a number of documents WF-IBP reveals that the predominant topic is among the rarest topics in the corresponding year, such as in "Text Classification using String Kernels" (information retrieval),"Playing is Believing: The Role of Beliefs in Multi-Agent Learning" (game theory), and "Relative Density Nets: A New Way to Combine Backpropagation with HMMs" (speech recognition).

We then compared the document representations learned by WF-IBP with the ones obtained by Dynamic Topic Models (DTM) (Blei and Lafferty, 2006). One of the benefits of WF-IBP is that it provides sparse and less noisy representations. This is due to the fact that, while DTM assigns a positive probability to all topics within each document, the WF-IBP selects only a subset of topics with positive probability via the feature allocation matrices Z. Recall that DTM requires fixing the number of topics K a priori, hence in our experiments we set $K = 50$. For instance, "Recursive Training of 2D-3D Convolutional Networks for Neuronal Boundary Prediction", "Exploring Models and Data for Image Question Answering" and "Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering" are respectively assigned to deep learning, image recognition and NLP by both models; however, they are respectively explained by 9,8 and 10 topics in WF-IBP, as opposed to 50 in DTM. While it is possible to order the topic proportions in DTM and only consider the ones greater than an arbitrary threshold, WF-IBP automatically sets the probability of irrelevant topics to 0 and offers a more interpretable representation.

Quantitative results We then compared the predictive performance of our fixed-K approximation with its static and hierarchical counterparts. Recall that time-stamps are not used in the static model and their ordering is neglected in the hierarchical model. The results in Figure 15 were obtained by holding out different percentages of words (50%, 60%, 70% and 80%) from all the papers published in 1999 and by training the model over the papers published in the time range 1987-1999. The goal was to investigate whether incorporating time dependency improves the predictions on future documents at time $t+1$ when given the documents up to time t. The held-out words were then used to compute the test-set perplexity after 5 repeated runs with random initializations (the error bars represent

one standard deviation). The dynamic model led to consistently better results, especially as the number of held-out words was increased. This follows from the fact that, the less training data is available in the year in which the models are tested, the more important capturing time-dependence to yield sensible predictions.



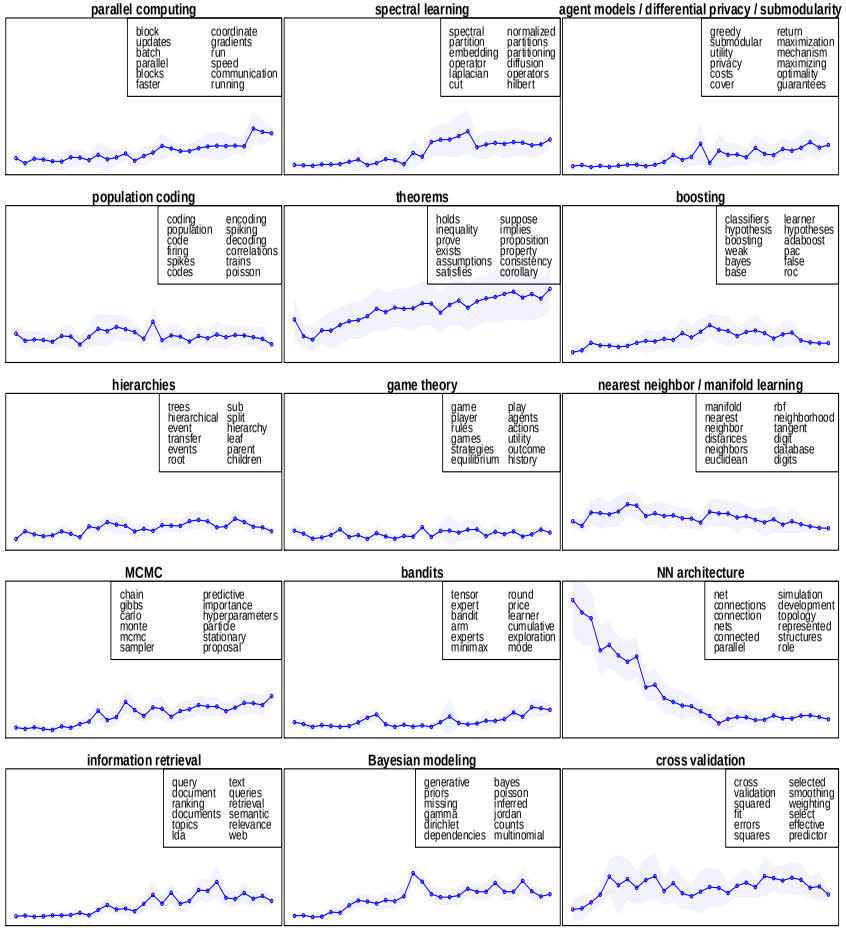


Figure 13: Posterior topic proportions over the years 1987-2015 and 12 most likely words for each topic (NIPS data set).

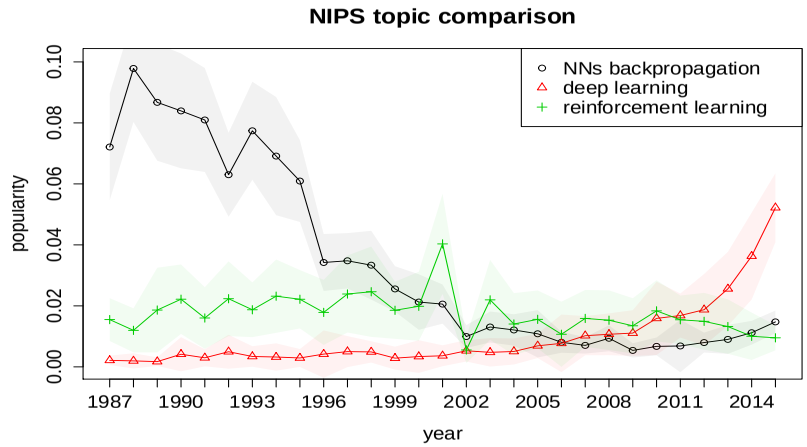


Figure 14: Comparison of a set of posterior topic proportions over the years 1987-2015 (NIPS data set).

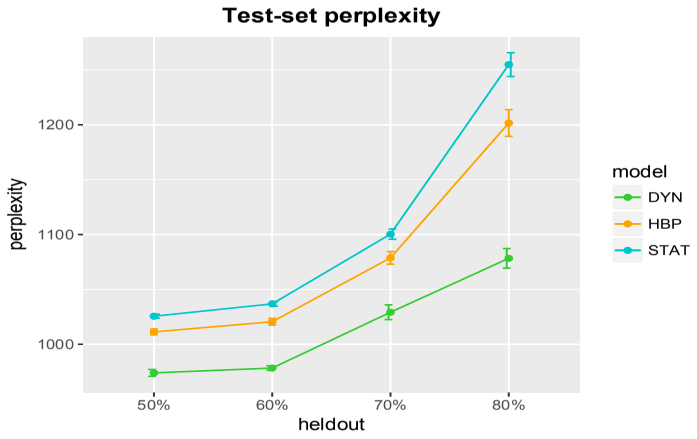


Figure 15: Comparison between the test-set perplexity of our dynamic model (DYN) against its hierarchical (HBP) and static (STAT) counterparts after holding out different percentages of words (NIPS data set). The dynamic model consistently outperforms the other two models, with a substantial difference when the percentage of held-out words is large.

8. Conclusion

We have presented a new framework for generating dependent IBPs by means of a novel time-evolving beta process, whereby feature probabilities evolve over time and are marginally distributed as in the beta process. At each time point items are exchangeable, and the two-parameter IBP is recovered. The key insight has been building on the PRF from population genetics to derive a suitable model for the prevalence and evolution of features over continuous time. We have developed an interesting MCMC framework for exact posterior inference with this model, and presented an alternative finite-dimensional approximation where the number of features is fixed.

As an application of the WF-IBP, we have described a time-dependent focused topic model that builds on Williamson et al. (2010b). The WF-IBP topic model allows for a flexible evolution of the popularity of an unknown number of topics over time, and compares favorably to HDP models by decoupling topic probabilities and within-document topic proportions. We have used our model to explore the data set consisting of the full text of NIPS conference papers from 1987 to 2015 and obtained an interesting visualization of how the popularity of the underlying topics evolved over these 29 years. In addition, test-set perplexity results have shown that incorporating time also improves on the predictive performance of the model.

A number of directions for future work are open. First, as $K \to \infty$ the fixed-K approximation marginally converges to the infinite model, and simulations showed that their dynamics are remarkably similar; further work could formally investigate the exact relationship between the two dynamics. Second, the current MCMC framework could be generalized to include inference on the IBP parameters and the W-F diffusion time step. Third, an extension of this work could modify the PRF by letting features evolve according to a more general W-F diffusion with selection and recombination, which would allow for feature-specific drifts in popularity and the coupled evolution of different features, respectively. Finally, our novel time-dependent beta process is a general construction with applications not limited to topic modeling. Different data and likelihood models could be explored following our work, for applications such as the modeling of time-evolving social networks or gene expression patterns.

Acknowledgments

The authors would like to thank the anonymous reviewers and the editor for their insightful and constructive feedback. Valerio Perrone is supported by EPSRC [EP/L016710/1]. Paul Jenkins is supported in part by EPSRC [EP/L018497/1]. Dario Spanò is supported in part by CRiSM, an EPSRC-HEFCE UK grant. Yee Whye Teh is supported by EPSRC for research funding under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617071.

Appendix A. Proof of Theorem 1

The mean density of a PRF with immigration parameter α is

$$l(x) = \alpha m(x)dx,$$

where

$$m(x) = \frac{e^{I(x)}}{\sigma^2(x)}, \quad \text{with } I(x) := \int_0^x \frac{2\gamma(y)}{\sigma^2(y)}dy, \tag{12}$$

is the speed density of the process (see Griffiths, 2003). γ and σ are the drift and diffusion terms as defined in (2) and (3). Plugging (2) with $\mu = 0$ and (3) into the integral, we have

$$I(x) = \beta \ln(1-x)$$

so that

$$m(x) = \frac{(1-x)^\beta}{x(1-x)} = x^{-1}(1-x)^{\beta-1}.$$

It follows that

$$l(x) = \alpha m(x)dx = \alpha x^{-1}(1-x)^{\beta-1}dx$$

is the resulting mean density, which completes the proof.

Remark 2 *When $\mu, \beta = 0$ it is necessary to condition each diffusion on hitting the boundary 0 before 1 in reverse time, which leads to an extra term in the analogous result in Sawyer and Hartl (1992).*

Appendix B. Simulating from the model

Consider the problem of simulating from the model, namely generating feature probabilities and the corresponding feature allocation matrices at a discrete set of time points.

Simulating X Set a truncation level $u > 0$ and consider the task of simulating features whose probability is above the threshold u at two times t_0 and t_1. As we know how to simulate marginally from the beta process, we can first generate the feature probabilities above u at time t_0 and let them evolve independently to time t_1. This yields features whose probability is greater than u at time t_0, meaning that we are still missing those features whose probability is below u at time t_0. To simulated these, we proceed as follows: we generate these features by drawing them from the beta process at time t_1 and propagate them backwards to time t_0. Finally, in order not to double-count features, all features that in the reverse simulation have probability greater than u at time t_0 have to be rejected.

Now translate these ideas into the following sampling scheme. At time t_0, sample from a truncated version of the PRF, namely from a Poisson process on $[u, 1)$ with rate measure $\alpha x^{-1}(1-x)^{\beta-1}dx$. This can be done, for instance, via an adaptive thinning scheme as described in Ogata (1981). As the truncation level u eliminates the point zero which has an infinite mass, this sampling procedure yields an almost surely finite number of samples. Denote by $\mathcal{K}$ the resulting set of feature indices and proceed as follows.

1. For all $k \in \mathcal{K}$, simulate $X_k(t) \mid \{X_k(t_0) = x_k(t_0)\} \sim \text{WF}(0, \beta)$ for $t \in [t_0, t_1]$.

2. At time t_1, sample the candidate newborn features $X(t_1)$ from the truncated PRF as above. Let $\mathcal{L}$ denote the resulting set of features.

3. For all $l \in \mathcal{L}$, simulate $X_l(t) \mid \{X_l(t_1) = x_l(t_1)\} \sim \text{WF}(0, \beta)$ backwards for $t \in [t_1, t_0]$ and remove from $\mathcal{L}$ the indices in the set $\{l : x_l(t_0) \geq u\}$.

4. Generate $Z_{ikt} \mid \{X_k(t) = x_k(t)\} \overset{iid}{\sim} \text{Bernoulli}(x_k(t))$, for $t = t_0, t_1$, $\forall k \in \mathcal{K} \cup \mathcal{L}$ and $\forall i = 1, \ldots, N_t$.

As mentioned previously, the idea behind steps 2 and 3 is to compensate for the features with probability smaller than u that were discarded when generating features at time t_0.

This construction generalizes to a set of time points $t = t_0, \ldots, t_T$, observing that at a given time $t_{t^*} \in \{t_1, \ldots, t_T\}$ step 3 needs to be modified by simulating the W-F diffusions backwards to time t_0 and removing from $\mathcal{L}$ the indices such that $\exists t \in \{t_0, \ldots, t_{t^*-1}\}$ such that $x_l(t) \geq u$.

Simulating Z and the underlying X Consider now the more complex task of simulating both the feature allocation matrices Z and the features X appearing in them. First note that, although the PRF describes the evolution of an infinite number of features, we can sample the feature allocation matrices Z_{t_0} and Z_{t_1} at two times t_0 and t_1 exactly, as a property of the IBP is that the number of observed features is almost surely finite (Griffiths and Ghahramani, 2011). It is then possible to sample the features that are active in at least one object at times t_0 and t_1 and the corresponding allocation matrices Z_{t_0} and Z_{t_1} as follows. First, draw Z_{t_0} from the IBP, and use its realisation to draw the posterior probabilities $X(t_0)$ of the features seen in Z_{t_0} as in the posterior beta process. Then, simulate from the W-F diffusion to propagate these features to time t_1 and generate Z_{t_1} using these feature probabilities. We are now only missing the columns of Z_{t_1} corresponding to the features that were seen at time t_1 but not at time t_0. To add those columns, first draw a candidate $Z^C_{t_1}$ from the IBP and the corresponding feature probabilities; then, simulate these candidate features backwards to time t_0 and accept them with probability $(1 - X(t_0))^{N_{t_0}}$ to account for the fact that they were not seen at time t_0. The columns of $Z^C_{t_1}$ corresponding to the rejected features are deleted. Translating these ideas into an algorithm, consider the following steps.

1. Draw $Z_{t_0} \sim \text{IBP}(\alpha, \beta)$ and index the resulting columns as $1, \ldots, K_1$.

2. For $k = 1, \ldots, K_1$ draw the corresponding feature probabilities

$$X_k(t_0) \mid \{Z_{t_0} = z_{t_0}\} \sim \text{Beta}\left(n_{kt_0}, \beta + N_{t_0} - n_{kt_0}\right).$$

3. For $k = 1, \ldots, K_1$ simulate $X_k(t) \mid \{X_k(t_0) = x_k(t_0)\} \sim \text{WF}(0, \beta)$ for $t \in [t_0, t_1]$ and set $X_k(t_1) = x_k(t_1)$.

4. Sample $Z^1_{t_1}$ by drawing each component $Z^1_{ikt_1} \mid \{X(t_1) = x(t_1)\} \overset{iid}{\sim} \text{Bernoulli}(x_k(t_1))$, where $k = 1, \ldots, K_1$ and $i = 1, \ldots, N_{t_1}$.

Then, to sample the features that are active only at time t_1, add the following steps.

5. Draw a candidate $Z^{\mathrm{C}}_{t_1} \sim \mathrm{IBP}(\alpha, \beta)$ and index the resulting columns as $K_1+1, \ldots, K_2$.
6. For $k = K_1+1, \ldots, K_2$ draw the corresponding candidate feature probabilities
$$X^{\mathrm{C}}_k(t_1) \mid \{Z^{\mathrm{C}}_{t_1} = z^{\mathrm{C}}_{t_1}\} \sim \mathrm{Beta}\left(n_{kt_1}, \beta + N_{t_1} - n_{kt_1}\right).$$
7. For $k = K_1+1, \ldots, K_2$, simulate $X_k(t) \mid \{X_k(t_1) = x_k(t_1)\} \sim \mathrm{WF}(0, \beta)$ backwards for $t \in [t_1, t_0]$ and set $X_k(t_0) = x_k(t_0)$.
8. Accept the candidate columns of $Z^C_{t_1}$ with probability $(1 - x^C_k(t_0))^{N_{t_0}}$ and let Z_{t_1} be the matrix obtained by the union of the columns of $Z^1_{t_1}$ with the accepted columns of $Z^C_{t_1}$.

Note that the rejection in the last step is a way to account for the fact that we are considering features that are active for the first time at time t_1. When considering a general set of time points $t_0, \ldots, t_T$, it is necessary to account for the features that are active for the first time at each time $t_1, \ldots, t_T$. In this more general case, features seen for the first time at time t_{t^*} need to be accepted with probability $\prod_{t=t_0}^{t_{t^*-1}} (1 - x^C_k(t))^{N_t}$, as they were not seen in any of the feature allocation matrices at the time points before t^*.

Appendix C. Derivation of full conditionals

As observed in Williamson et al. (2009), in this topic modeling setting there are two equivalent ways of generating documents. Either the total number of words is sampled from a negative binomial $\mathrm{NB}(\sum_k z_{ikt}\phi_{kt}, 1/2)$ and then the topic and word assignments are drawn, or the number of words generated by each topic is drawn from $\mathrm{NB}(z_{ikt}\phi_{kt}, 1/2)$ and then the word assignments are picked. Following Williamson et al. (2009) closely, we make use of the latter construction to derive the full conditional distributions for the Gibbs sampler of the WF-IBP topic model.

Full conditional of a_{il} Recall that $a_{ilt} = k$ indicates that the lth word in document i at time t is assigned to topic k. We have
$$\begin{aligned} p(a_{ilt} = k \mid a_{-il}, Z_t, w_{ilt}, \phi_t) &\propto p(w_{ilt} \mid a_{ilt} = k) p(a_{ilt} = k \mid a_{-il}, z_{ikt}, \phi_{kt}) \\ &\propto p(w_{ilt} \mid a_{ilt} = k)(n^i_{kt} + \phi_{kt} z_{ikt}), \end{aligned}$$
where the last step is given by integrating out θ_{it}, namely the distribution over topics in document i at time t, and using the Dirichlet-Categorical conjugacy. Recall that n^i_{kt} denotes the number of words assigned to topic k in document i at time t and that n_{kt} denotes the total number of words assigned to topic k at time t, both excluding the assignment a_{il}. Similarly, integrating out the parameter ρ_k representing the distribution over words of topic k and using the Dirichlet-Categorical conjugacy, we have that
$$p(w_{ilt} \mid a_{ilt} = k) = \frac{(n^{w_{il}}_k + \eta)}{n_k + \eta D - 1},$$
which, plugged into the previous equation, gives the desired full conditional.

Full conditional of ϕ_k and γ We have that
$$\begin{aligned} p(\phi_{kt}, \gamma \mid n_{kt}, X(t), Z_t) &\propto p(\phi_{kt}, \gamma, n_{kt}, X(t), Z_t) \\ &\propto p(\phi_{kt} \mid \gamma) P(\gamma) P(n_{kt} \mid Z_t, \phi_{kt}), \end{aligned} \tag{13}$$
where
$$p(n_{kt} \mid Z_t, \phi_{kt}) = \prod_{i=1}^{N_t} p(n^i_{kt} \mid Z_{ikt}, \phi_{kt}) = \prod_{i=1}^{N_t} \mathrm{NB}(n^i_{kt}; Z_{ikt}\phi_{kt}, 1/2).$$
Note that $p(\phi_{kt} \mid \gamma)$ is distributed according to its prior $\mathrm{Gamma}(\gamma, 1)$ and γ according to a chosen hyper-prior. The result follows immediately by plugging these three distributions into (13).

Full conditional of Z_{ikt} Recall that n_{ikt} denotes the total number of words assigned to topic k in document i at time t. If $n_{ikt} > 0$, then the corresponding entry Z_{ikt} is active with probability 1. If $n_{ikt} = 0$, we have
$$p(Z_{ikt} = 1 \mid Z_{-(ik)t}, n_{ikt} = 0, X_k(t), \phi_{kt}, S_t) = \frac{p(Z_{ikt} = 1, Z_{-(ik)t}, n_{ikt} = 0, X_k(t), \phi_{kt}, S_t)}{p(Z_{-(ik)t}, n_{ikt} = 0, X_k(t), \phi_{kt}, S_t)}.$$
The numerator is equal to
$$\begin{aligned} p(n_{ikt} = 0 \mid Z_{ikt} = 1, \phi_{kt}) p(S_t \mid Z_{ikt} = 1, Z_{-(ik)t}) p(Z_{ikt} = 1 \mid X_k(t)) p(\phi_{kt}, X_k(t), Z_{-(ik)t}) = \\ \mathrm{NB}(0; \phi_{kt}, 1/2) \frac{1}{x^{min}_1(t)} x_k(t) p(\phi_{kt}, X_k(t), Z_{-(ik)t}) \end{aligned}$$
Denoting by C the product of all terms not depending on z_{ikt}, we have
$$p(Z_{ikt} = 1 \mid Z_{-(ik)t}, n_{ikt} = 0, X_k(t), \phi_{kt}, S_t) = C \frac{1}{2^{\phi_{kt}}} \frac{1}{x^{min}_1(t)} x_k(t). \tag{14}$$
By the same token, we have
$$p(Z_{ikt} = 0 \mid Z_{-(ik)t}, n_{ikt} = 0, X_k(t), \phi_{kt}, S_t) = C \frac{1}{x^{min}_0(t)} (1 - x_k(t)). \tag{15}$$
As the two probabilities must sum to 1, we have that
$$C = \frac{2^{\phi_{kt}} x^{min}_1(t) x^{min}_0(t)}{x^{min}_0(t) x_k(t) + 2^{\phi_{kt}} x^{min}_1(t)(1 - x_k(t))},$$
which, plugged into equations 14 and 15, gives the result.

References

A. Ahmed and E. P. Xing. Timeline: A Dynamic Hierarchical Dirichlet Process Model for Recovering Birth/Death and Evolution of Topics in Text Stream. *UAI*, abs/1203.3463, 2012.

A. Amei and S. Sawyer. A time-dependent Poisson random field model for polymorphism within and between two related biological species. *Annals of Applied Probability*, 20(5): 1663–1696, 2010.

A. Amei and S. Sawyer. Statistical inference of selection and divergence from a time-dependent poisson random field model. *PLoS ONE*, 7(4):e34413, 2012.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistal Society B*, 72:269–342, 2010.

A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On Smoothing and Inference for Topic Models. *UAI*, pages 27–34, 2009.

D. M. Blei and J. D. Lafferty. Dynamic topic models. *ICML*, pages 113–120, 2006.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet Allocation. *JMLR*, 3:993–1022, 2003.

A. R. Boyko, S. H. Williamson, A. R. Indap, J. D. Degenhardt, R. D. Hernandez, K. E. Lohmueller, M. D. Adams, S. Schmidt, J. J. Sninsky, S. R. Sunyaev, T. J. White, R. Nielsen, A. G. Clark, and C. D. Bustamante. Assessing the evolutionary impact of amino acid mutations in the human genome. *PLoS Genet*, 4(5), 2008.

C. D. Bustamante, J. Wakeley, S. Sawyer, and D. L. Hartl. Directional Selection and the Site-Frequency Spectrum. *Genetics*, 159(4):1779–1788, 2001.

C. D. Bustamante, R. Nielsen, and D. L. Hartl. Maximum likelihood and Bayesian methods for estimating the distribution of selective effects among classes of mutations using DNA polymorphism data. *Theoretical Population Biology*, 63(2):91–103, 2003. ISSN 0040-5809.

C. E. Dangerfield, D. Kay, S. MacNamara, and K. Burrage. A boundary preserving numerical algorithm for the Wright-Fisher model with mutation. *BIT Numerical Mathematics*, 52(2):283–304, 2012.

F. Doshi-Velez and Z. Ghahramani. Accelerated Sampling for the Indian Buffet Process. *ICML*, pages 273–280, 2009.

A. Dubey, A. Hefny, S. Williamson, and E. P. Xing. A Nonparametric Mixture Model for Topic Modeling over Time. *SDM*, pages 530–538, 2013.

S. N. Ethier and T. G. Kurtz. *Markov processes: characterization and convergence*. Wiley series in probability and mathematical statistics. J. Wiley & Sons, 1986. ISBN 0-471-08186-8.

W. J. Ewens. *Mathematical Population Genetics*. Springer-Verlag, Berlin, 2004.

S. Gershman, P. I. Frazier, and D. M. Blei. Distance Dependent Infinite Latent Feature Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):334–345, 2015.

Z. Ghahramani, T. L. Griffiths, and P. Sollich. Bayesian nonparametric latent feature models. *Bayesian Statistics*, pages 201–225, 2007.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schlkopf, and A. J. Smola. A Kernel Method for the Two-Sample-Problem. *NIPS*, pages 513–520, 2006.

R. C. Griffiths. The frequency spectrum of a mutation, and its age, in a general diffusion model. *Theoretical Population Biology*, 64(2):241–251, 2003.

T. L. Griffiths and Z. Ghahramani. The Indian Buffet Process: An Introduction and Review. *JMLR*, 12:1185–1224, 2011.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101:5228–5235, 2004.

R. N. Gutenkunst, R. D. Hernandez, S. H. Williamson, and C. D. Bustamante. Inferring the Joint Demographic History of Multiple Populations from Multidimensional SNP Frequency Data. *PLoS Genet*, 5(10), 2009.

D. L. Hartl, E. N. Moriyama, and S. A. Sawyer. Selection intensity for codon bias. *Genetics*, pages 227–234, 1994.

P. A. Jenkins and D. Spanò. Exact simulation of the Wright-Fisher diffusion. *Annals of Applied Probability*, 2017. To appear.

S. Karlin and H. M. Taylor. *A Second Course in Stochastic Processes*. Academic Press, 1981.

J. F. C. Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1): 59–78, 1967.

Y. LeCun, Y. Bengio, and G. E. Hinton. Deep learning. *Nature*, 521:436–444, 2015.

K. T. Miller, T. L. Griffiths, and M. I. Jordan. The Phylogenetic Indian Buffet Process: A Non-Exchangeable Nonparametric Prior for Latent Features. *UAI*, abs/1206.3279: 403–410, 2012.

Y. Ogata. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–30, 1981.

V. Rao and Y. W. Teh. Spatial Normalized Gamma Processes. *NIPS*, pages 1554–1562, 2009.

S. A. Sawyer and D. L. Hartl. Population genetics of polymorphism and divergence. *Genetics*, 132(4):1161–1176, 1992.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Y. W. Teh, D. Görür, and Z. Ghahramani. Stick-breaking Construction for the Indian Buffet Process. *AISTATS*, 11:556–563, 2007.

R. Thibaux and M. I. Jordan. Hierarchical beta processes and the Indian buffet process. *AISTATS*, 2:564–571, 2007.

S. G. Walker. Sampling the Dirichlet mixture model with slices. *Communications in Statistics - Simulation and Computation*, 36(1):45–54, 2007.

S. Williamson, C. Wang, K. Heller, and D. Blei. Focused Topic Models. *NIPS Workshop on Applications for Topic Models: Text and Beyond*, pages 1–4, 2009.

S. Williamson, P. Orbanz, and Z. Ghahramani. Dependent Indian Buffet Processes. *AISTATS*, pages 924–931, 2010a.

S. Williamson, C. Wang, K. A. Heller, and D. M. Blei. The IBP Compound Dirichlet Process and its Application to Focused Topic Modeling. *ICML*, pages 1151–1158, 2010b.

S. H. Williamson, R. Hernandez, A. Fledel-Alon, L. Zhu, R. Nielsen, and C. D. Bustamante. Simultaneous inference of selection and population growth from patterns of variation in the human genome. *PNAS*, 102(22):7882–7887, 2005.

M. Zhou, H. Yang, G. Sapiro, D. B. Dunson, and L. Carin. Dependent Hierarchical Beta Process for Image Interpolation and Denoising. *AISTATS*, 15:883–891, 2011.

Journal of Machine Learning Research 18 (2017) 1-33 Submitted 11/16; Revised 11/17; Published 11/17

Gap Safe Screening Rules for Sparsity Enforcing Penalties

Eugene Ndiaye EUGENE.NDIAYE@TELECOM-PARISTECH.FR
LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France

Olivier Fercoq OLIVIER.FERCOQ@TELECOM-PARISTECH.FR
LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France

Alexandre Gramfort ALEXANDRE.GRAMFORT@INRIA.FR
Inria, Université Paris-Saclay, 91120, Palaiseau, France
LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France

Joseph Salmon JOSEPH.SALMON@TELECOM-PARISTECH.FR
LTCI, Télécom ParisTech, Université Paris-Saclay, 75013, Paris, France

Editor: David Wipf

Abstract

In high dimensional regression settings, sparsity enforcing penalties have proved useful to regularize the data-fitting term. A recently introduced technique called *screening rules* propose to ignore some variables in the optimization leveraging the expected sparsity of the solutions and consequently leading to faster solvers. When the procedure is guaranteed not to discard variables wrongly the rules are said to be *safe*. In this work, we propose a unifying framework for generalized linear models regularized with standard sparsity enforcing penalties such as ℓ_1 or ℓ_1/ℓ_2 norms. Our technique allows to discard safely more variables than previously considered safe rules, particularly for low regularization parameters. Our proposed Gap Safe rules (so called because they rely on duality gap computation) can cope with any iterative solver but are particularly well suited to (block) coordinate descent methods. Applied to many standard learning tasks, Lasso, Sparse-Group Lasso, multi-task Lasso, binary and multinomial logistic regression, etc., we report significant speed-ups compared to previously proposed safe rules on all tested data sets.

Keywords: Convex optimization, screening rules, Lasso, multi-task Lasso, sparse logistic regression, Sparse-Group Lasso

1. Introduction

The computational burden of solving high dimensional regularized regression problem has led to a vast literature on improving algorithmic solvers in the last two decades. With the increasing popularity of ℓ_1-type regularization ranging from the Lasso (Tibshirani, 1996) or group-Lasso (Yuan and Lin, 2006) to regularized logistic regression and multi-task learning, many algorithmic methods have emerged to solve the associated optimization problems (Koh et al., 2007; Bach et al., 2012). Although for the simple ℓ_1 regularized least square a specific algorithm (*e.g.,* the LARS (Efron et al., 2004)) can be considered, for more general formulations, penalties, and possibly larger dimensions, (block) coordinate descent has proved to be an efficient strategy (Friedman et al., 2010).

©2017 Ndiaye *et al.* .

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-577.html.

Our main objective in this work is to propose a technique that can speed-up any iterative solver for such learning problems, and that is particularly well suited for (block) coordinate descent method as this type of method can easily ignore useless coordinates[1].

The *safe rules* introduced by El Ghaoui et al. (2012) for generalized ℓ_1 regularized problems, is a set of rules allowing to eliminate features whose associated coefficients are guaranteed to be zero at the optimum, even before starting any algorithm. Relaxing the safe rule, one can obtain some additional speed-up at the price of possible mistakes. Such heuristic strategies, called *strong rules* by Tibshirani et al. (2012) reduce the computational cost using an active set strategy, but require multiple post-processing to check for features possibly wrongly discarded.

Another road to speed-up screening method has been pursued following the introduction of *sequential safe rules* (El Ghaoui et al., 2012; Wang et al., 2012; Xiang et al., 2014; Wang et al., 2014). The idea is to improve the screening thanks to the computation done for a previous regularization parameter as in homotopy/continuation methods. This scenario is particularly relevant in machine learning, where one computes solutions over a grid of regularization parameters, so as to select the best one, *e.g.,* by cross-validation. Nevertheless, the aforementioned methods suffer from the same problem as strong rules, since relevant features can be wrongly disregarded. Indeed, sequential rules usually rely on the exact knowledge of certain theoretical quantities that are only known approximately. Especially, for such rules to work one needs the exact dual optimal solution from the previous regularization parameter, a quantity (almost) never available to the practitioner.

The recent introduction of *dynamic safe rules* by Bonnefoy et al. (2015, 2014) has opened a new promising venue by performing variable screening, not only before the algorithm starts, but also along the iterations. This screening strategy can be applied for any standard optimization algorithm such as FISTA (Beck and Teboulle, 2009), primal-dual (Chambolle and Pock, 2011), augmented Lagrangian (Boyd et al., 2011). Yet, it is particularly relevant for strategies that can benefit from support reduction or active sets (Kowalski et al., 2011; Johnson and Guestrin, 2015), such as coordinate-descent (Fu, 1998; Friedman et al., 2007, 2010).

This paper contains a synthesis and a unified presentation of the methods introduced first for the Lasso in (Fercoq et al., 2015) and then for ℓ_1/ℓ_2 norms in (Ndiaye et al., 2015) as well as for Sparse-Group Lasso in (Ndiaye et al., 2016b). Our so-called Gap Safe rules (because the screening rules rely on duality gap computations), improved on dynamic safe rules for a broad class of learning problems with the following benefits:

- Gap Safe rules are easy to insert in existing solvers,
- they are proved to be safe and unify sequential and dynamic rules,
- they lead to improved speed-ups in practice *w.r.t.* previously known safe rules,
- they achieve fast variable identifications[2].

1. By construction a method like LARS (Efron et al., 2004), that applies only to the Lasso case, cannot beneficiate from screening rules.
2. more precisely, we identify faster the equicorrelation set, see Theorem 14

Furthermore, it is worth noting that strategies also leveraging dual gap computations have recently been considered to safely discard irrelevant coordinates: Shibagaki et al. (2016) have considered screening rules for learning tasks with both feature sparsity and sample sparsity, such as for ℓ_1-regularized SVM. In this case, some interesting developments have been proposed, namely *safe keeping* strategies, which allow to identify features and samples that are guaranteed to be active. Constrained convex problems such as minimum enclosing ball can also be included as shown in Raj et al. (2016). The Blitz algorithm by Johnson and Guestrin (2015) aims to speed up working set methods using duality gaps computations; significant gains were also obtained in limited-memory and distributed settings.

We introduce the general framework of Gap Safe screening rules in Section 3; in Section 4 we instantiate them on various strategies including static, dynamic and sequential ones, and show how the Gap Safe methodology can encompass all of them. The converging nature of the Gap Safe rules is also discussed. In Section 5, we investigate the specific form of our rules for standard machine learning models: Lasso, Group Lasso, Sparse-Group Lasso, logistic regression with ℓ_1 regularization, etc. Section 6 reports a comprehensive set of experiments on four different learning problems using either dense or sparse data. Results demonstrate the systematic gain in computation time of the Gap Safe rules.

2. Notation and Background on Optimization

For any integer $d \in \mathbb{N}$, we denote by $[d]$ the set $\{1, \dots, d\}$ and by $Q^\top$ the transpose of a matrix Q. Our observation vector is $y \in \mathbb{R}^n$ where n represents the number of samples. The design matrix $X = [x_1, \dots, x_n]^\top \in \mathbb{R}^{n \times p}$ has p explanatory variables (or features) column-wise, and n observations row-wise. For a norm Ω, we write $\mathcal{B}_\Omega$ the associated unit ball, $\|\cdot\|_2$ is the ℓ_2 norm (and $\langle \cdot, \cdot \rangle$ for the associated inner product), $\|\cdot\|_1$ is the ℓ_1 norm, and $\|\cdot\|_\infty$ is the ℓ_∞ norm. The ℓ_2 unit ball is denoted by $\mathcal{B}_2$ (or simply $\mathcal{B}$) and we write $\mathcal{B}(\theta, r)$ the ℓ_2 ball with center θ and radius r. For a vector $\beta \in \mathbb{R}^p$, we denote by $\mathrm{supp}(\beta)$ the support of β (*i.e.,* the set of indices corresponding to non-zero coefficients) and by $\|\mathrm{B}\|_F^2 = \sum_{j=1}^p \sum_{k=1}^q \mathrm{B}_{j,k}^2$ the Frobenius norm of a matrix $\mathrm{B} \in \mathbb{R}^{p \times q}$. We denote $(t)_+ = \max(0, t)$ and $\Pi_{\mathcal{C}}(\cdot)$ the projection operator over a closed convex set $\mathcal{C}$. The soft-thresholding operator ST_τ (at level $\tau \geqslant 0$) is defined for any $x \in \mathbb{R}^d$ by $[\mathrm{ST}_\tau(x)]_j = \mathrm{sign}(x_j)(|x_j| - \tau)_+$.

The parameter to recover is a vector $\beta = (\beta_1, \dots, \beta_p)^\top$ admitting a group structure. A group of features is a subset $g \subset [p]$ and n_g is its cardinality. The set of groups is denoted by $\mathcal{G}$ and we focus only on non-overlapping groups[3] that form a partition of the set $[p]$. We denote by β_g the vector in $\mathbb{R}^{n_g}$ which is the restriction of β to the indices in g. We write $[\beta_g]_j$ the j-th coordinate of β_g. We also use the notation $X_g \in \mathbb{R}^{n \times n_g}$ to refer to the sub-matrix of X assembled from the columns with indices $j \in g$ and X_j when the groups are a single feature, *i.e.,* when $g = \{j\}$.

Some elements of convex analysis used in the following sections are introduced here. For a function $f : \mathbb{R}^d \to [-\infty, +\infty]$, the Fenchel-Legendre transform[4] of f, is the function $f^* : \mathbb{R}^d \to [-\infty, +\infty]$ defined by $f^*(u) = \sup_{z \in \mathbb{R}^d} \langle z, u \rangle - f(z)$. The sub-differential of a function f at a point x is denoted by $\partial f(x)$. For a norm Ω over $\mathbb{R}^d$, its dual norm is written Ω^D and is defined for any $u \in \mathbb{R}^d$ by $\Omega^D(u) = \max_{\Omega(z) \leqslant 1} \langle z, u \rangle$. Note that in the case of a group-decomposable norm, one can check that $\Omega^D(\beta) = \max_{g \in \mathcal{G}} \Omega_g^D(\beta_g)$ and $\partial \Omega(\beta) = \Pi_{g \in \mathcal{G}} \partial \Omega_g(\beta_g)$.

We remind below useful standard results from convex analysis:

Proposition 1 (Fermat's Rule) *(see (Bauschke and Combettes, 2011, Proposition 26.1) for a more general result) For any convex function $f : \mathbb{R}^d \to \mathbb{R}$:*

$$x^\star \in \operatorname*{arg\,min}_{x \in \mathbb{R}^d} f(x) \iff 0 \in \partial f(x^\star). \tag{1}$$

Proposition 2 (Subdifferential of a Norm) *(see (Bach et al., 2012, Proposition 1.2)) The sub-differential of the norm Ω at x, is given by*

$$\partial \Omega(x) = \begin{cases} \{z \in \mathbb{R}^d : \Omega^D(z) \leqslant 1\} = \mathcal{B}_{\Omega^D}, & \text{if } x = 0, \\ \{z \in \mathbb{R}^d : \Omega^D(z) = 1 \text{ and } z^\top x = \Omega(x)\}, & \text{otherwise.} \end{cases} \tag{2}$$

3. Gap Safe Framework

We propose to estimate the vector of parameters β by solving

$$\hat{\beta}^{(\lambda)} \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} P_\lambda(\beta), \text{ for } P_\lambda(\beta) := F(\beta) + \lambda \Omega(\beta) := \sum_{i=1}^n f_i(x_i^\top \beta) + \lambda \Omega(\beta) \ , \tag{3}$$

where all $f_i : \mathbb{R} \mapsto \mathbb{R}$ are convex and differentiable functions with $1/\gamma$-Lipschitz gradient and $\Omega : \mathbb{R}^p \mapsto \mathbb{R}_+$ is a norm that is group-decomposable, *i.e.,* $\Omega(\beta) = \sum_{g \in \mathcal{G}} \Omega_g(\beta_g)$ where each Ω_g is a norm on $\mathbb{R}^{n_g}$. The λ parameter is a non-negative constant controlling the trade-off between the data fitting term and the regularization term. Popular instantiations of problems of the form (3) are detailed in Section 5.

Theorem 3 *A dual formulation of the optimization problem defined in* (3) *is given by*

$$\hat{\theta}^{(\lambda)} = \operatorname*{arg\,max}_{\theta \in \Delta_X} - \sum_{i=1}^n f_i^*(-\lambda \theta_i) =: D_\lambda(\theta), \tag{4}$$

where $\Delta_X = \{\theta \in \mathbb{R}^n : \forall g \in \mathcal{G}, \Omega_g^D(X_g^\top \theta) \leqslant 1\}$. Moreover, the Fermat's rule reads:

$$\forall i \in [n], \ \hat{\theta}_i^{(\lambda)} = -\nabla f_i(x_i^\top \hat{\beta}^{(\lambda)})/\lambda \quad (\textbf{\textit{link equation}}), \tag{5}$$

$$\forall g \in \mathcal{G}, \ X_g^\top \hat{\theta}^{(\lambda)} \in \partial \Omega_g(\hat{\beta}_g^{(\lambda)}) \quad (\textbf{\textit{sub-differential inclusion}}). \tag{6}$$

For any $\theta \in \mathbb{R}^n$ let us introduce $G(\theta) := [\nabla f_1(\theta_1), \dots, \nabla f_n(\theta_n)]^\top \in \mathbb{R}^n$. Then the primal/dual link equation can be written $\hat{\theta}^{(\lambda)} = -G(X\hat{\beta}^{(\lambda)})/\lambda$.

Contrarily to the primal, the dual problem has a unique solution under our assumption on the f_i's. Indeed, the dual function is strongly concave, hence strictly concave.

3. Overlapping groups could be treated as well, some insight is given in Section 5.3

4. Such a transform is often referred to as the (convex) conjugate of a (convex) function

3.1 Safe Screening Rules

Following the seminal work by El Ghaoui et al. (2012) screening techniques have emerged as a way to exploit the known sparsity of the solution by discarding features prior to starting a sparse solver. Such techniques are referred to in the literature as *safe rules* when they screen out coefficients guaranteed to be zero in the targeted optimal solution. Zeroing those coefficients allows to focus exclusively on the non-zero ones (likely to represent signal) and helps reducing the computational burden.

One well known extreme is the following: for $\lambda > 0$ large enough, 0 is the unique solution of (3). Indeed,

$$0 \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, F(\beta) + \lambda\Omega(\beta) \overset{(1)}{\iff} 0 \in \{\nabla F(0)\} + \lambda \partial\Omega(0) \overset{(2)}{\iff} \Omega^D(\nabla F(0)) \leqslant \lambda.$$

Hence we recall the first "naive" screening rule, stating that for large values of the regularization parameter, all features can be discarded.

Proposition 4 (Critical Parameter: $\lambda_{\max}$) *For any $\lambda > 0$,*

$$0 \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, P_\lambda(\beta) \iff \lambda \geqslant \lambda_{\max} := \Omega^D(\nabla F(0)) = \Omega^D(X^\top G(0)) \ .$$

So from now on, we will only focus on the case where $\lambda < \lambda_{\max}$. In this case, screening rules rely on a direct consequence of Fermat's rule (6). If $\hat\beta_g^{(\lambda)} \neq 0$, then $\Omega_g^D(X_g^\top \hat\theta^{(\lambda)}) = 1$ thanks to Equation (2). Since $\hat\theta^{(\lambda)} \in \Delta_X$, it implies, by contrapositive, that if $\Omega_g^D(X_g^\top \hat\theta^{(\lambda)}) < 1$ then $\hat\beta_g^{(\lambda)} = 0$. This relation means that the g-th group can be discarded whenever $\Omega_g^D(X_g^\top \hat\theta^{(\lambda)}) < 1$. However, since $\hat\theta^{(\lambda)}$ is **unknown** — unless $\lambda > \lambda_{\max}$, in which case $\hat\theta^{(\lambda)} = G(0)/\lambda$ — this rule is of limited use. Fortunately, it is often possible to construct a set $\mathcal{R} \subset \mathbb{R}^n$, called a *safe region*, that contains $\hat\theta^{(\lambda)}$. This observation leads to the following result.

Proposition 5 (Safe Screening Rule El Ghaoui et al. (2012)) *If $\hat\theta^{(\lambda)} \in \mathcal{R}$, and $g \in \mathcal{G}$:*

$$\max_{\theta \in \mathcal{R}} \Omega_g^D(X_g^\top \theta) < 1 \Longrightarrow \Omega_g^D(X_g^\top \hat\theta^{(\lambda)}) < 1 \Longrightarrow \hat\beta_g^{(\lambda)} = 0 \ . \tag{7}$$

The so-called *safe screening* rule consists in removing the g-th group from the problem whenever the previous test is satisfied, since then $\hat\beta_g^{(\lambda)}$ is guaranteed to be zero. Should $\mathcal{R}$ be small enough to screen many groups, one can observe considerable speed-ups in practice as long as the testing can be performed efficiently. A natural goal is to find safe regions as narrow as possible: smaller safe regions can only increase the number of screened out variables. To have useful screening procedures one needs:

- the safe region $\mathcal{R}$ to be as small as possible (and to contain $\hat\theta^{(\lambda)}$),
- the computation of the quantity $\max_{\theta \in \mathcal{R}} \Omega_g^D(X_g^\top \theta)$ to be cheap.

The later means that safe regions should be simple geometric objects, since otherwise, evaluating the test could lead to a computational burden limiting the benefits of screening.

3.2 Gap Safe Regions

Various shapes have been considered in practice for the safe region $\mathcal{R}$ such as balls (El Ghaoui et al., 2012), domes (Fercoq et al., 2015) or more refined sets (see Xiang et al. (2014) for a survey). Here we consider for simplicity the so-called "sphere regions" (following the terminology introduced by El Ghaoui et al. (2012)) choosing a ball $\mathcal{R} = \mathcal{B}(\theta, r)$ as a safe region. Thanks to the triangle inequality, we have:

$$\max_{\theta \in \mathcal{B}(\theta, r)} \Omega_g^D(X_g^\top \theta) \leqslant \Omega_g^D(X_g^\top \theta) + \max_{\theta \in \mathcal{B}(\theta, r)} \Omega_g^D(X_g^\top(\theta - \theta)),$$

and denoting $\Omega_g^D(X_g) := \sup_{u \neq 0} \frac{\Omega_g^D(X_g^\top u)}{\|u\|_2}$ the operator norm of X_g associated to $\Omega_g^D(\cdot)$, we deduce from Proposition 7 the screening rule for the g-th group:

$$\text{Safe sphere test:} \qquad \text{If} \quad \Omega_g^D(X_g^\top \theta) + r\Omega_g^D(X_g) < 1, \quad \text{then} \quad \hat\beta_g^{(\lambda)} = 0 \ . \tag{8}$$

3.2.1 Finding a Center

To create a useful center for a safe sphere, one needs to be able to create dual feasible points, *i.e.,* points in the dual feasible set Δ_X. One such point is $\theta_{\max} := -G(0)/\lambda_{\max}$ which leads to the original static safe rules proposed by El Ghaoui et al. (2012). Yet, it has a limited interest, being helpful only for a small range of (large) regularization parameters λ, as discussed in Section 4.1. A more generic way of creating a dual point that will be key for creating our safe rules is to rescale any point $z \in \mathbb{R}^n$ such that it is in the dual set Δ_X. The rescaled point is denoted by $\Theta(z)$ and is defined by

$$\Theta(z) := \begin{cases} z, & \text{if } \Omega^D(X^\top z) \leqslant 1, \\ \frac{z}{\Omega^D(X^\top z)}, & \text{otherwise.} \end{cases} \tag{9}$$

This choice guarantees that $\forall z \in \mathbb{R}^n, \Theta(z) \in \Delta_X$. A candidate often considered for computing a dual point is the (generalized) residual term $z = -G(X\beta)/\lambda$. This choice is motivated by the primal-dual link equation 5 *i.e.,* $\hat\theta^{(\lambda)} = -G(X\hat\beta^{(\lambda)})/\lambda$.

3.2.2 Finding a Radius

Now that we have seen how to create a center candidate for the sphere, we need to find a proper radius, that would allow the associated sphere to be safe. The following theorem proposes a way to obtain a radius using the duality gap. The quantity

$$\mathrm{Gap}_\lambda(\beta, \theta) := P_\lambda(\beta) - D_\lambda(\theta) \tag{10}$$

is often referred to as the duality gap in the convex optimization literature, hence the name of our proposed Gap Safe framework. This quantity is also a useful tool when designing a stopping criterion: noting that for any $\beta \in \mathbb{R}^p, \theta \in \Delta_X$, $P_\lambda(\beta) - P_\lambda(\hat\beta^{(\lambda)}) \leqslant \mathrm{Gap}_\lambda(\beta, \theta)$, it suffices to find a primal-dual pair with a duality gap smaller than ϵ to ensure an ϵ-accuracy primal solution for Problem 3.

Theorem 6 (Gap Safe Sphere) *Assuming that F has $1/\gamma$-Lipschitz gradient, we have*

$$\forall \beta \in \mathbb{R}^p, \forall \theta \in \Delta_X, \quad \left\| \hat\theta^{(\lambda)} - \theta \right\|_2 \leqslant \sqrt{\frac{2\mathrm{Gap}_\lambda(\beta, \theta)}{\gamma \lambda^2}} \ . \tag{11}$$

Hence the set $\mathcal{R} = \mathcal{B}(\theta, \sqrt{2\mathrm{Gap}_\lambda(\beta,\theta)/\gamma\lambda^2})$ *is a safe region for any* $\beta \in \mathbb{R}^n$ *and* $\theta \in \Delta_X$.

Proof Remember that $\forall i \in [n], f_i$ is differentiable with a $1/\gamma$-Lipschitz gradient. As a consequence, $\forall i \in [n], f_i^*$ is γ-strongly convex (Hiriart-Urruty and Lemaréchal, 1993, Theorem 4.2.2, p. 83) and so the dual function D_λ is $\gamma\lambda^2$-strongly concave:

$$\forall(\theta_1,\theta_2) \in \mathbb{R}^n \times \mathbb{R}^n, \quad D_\lambda(\theta_2) \leqslant D_\lambda(\theta_1) + \langle \nabla D_\lambda(\theta_1), \theta_2 - \theta_1 \rangle - \frac{\gamma\lambda^2}{2} \|\theta_1 - \theta_2\|_2^2 \ .$$

Specifying the previous inequality for $\theta_1 = \hat{\theta}^{(\lambda)}, \theta_2 = \theta \in \Delta_X$, one has

$$D_\lambda(\theta) \leqslant D_\lambda(\hat{\theta}^{(\lambda)}) + \langle \nabla D_\lambda(\hat{\theta}^{(\lambda)}), \theta - \hat{\theta}^{(\lambda)} \rangle - \frac{\gamma\lambda^2}{2} \left\| \hat{\theta}^{(\lambda)} - \theta \right\|_2^2 \ .$$

By definition, $\hat{\theta}^{(\lambda)}$ maximizes D_λ on Δ_X, so, $\langle \nabla D_\lambda(\hat{\theta}^{(\lambda)}), \theta - \hat{\theta}^{(\lambda)} \rangle \leqslant 0$. This implies

$$D_\lambda(\theta) \leqslant D_\lambda(\hat{\theta}^{(\lambda)}) - \frac{\gamma\lambda^2}{2} \left\| \hat{\theta}^{(\lambda)} - \theta \right\|_2^2 .$$

By weak duality $\forall \beta \in \mathbb{R}^p, D_\lambda(\hat{\theta}^{(\lambda)}) \leqslant P_\lambda(\beta)$, hence $\forall \beta \in \mathbb{R}^p, \forall \theta \in \Delta_X, D_\lambda(\theta) \leqslant P_\lambda(\beta) - \frac{\gamma\lambda^2}{2}\|\hat{\theta}^{(\lambda)} - \theta\|_2^2$ and the conclusion follows. ■

Remark 7 *To build a Gap Safe region as in Equation* (11)*, we only need strong convexity in the dual which is equivalent to smoothness of the loss function whereas the screening property* (7)*, requires group separability of norms. Hence our framework of Gap Safe screening rule automatically applies for a large class of problems.*

Remark 8 *During the review process, we became aware of a possible improvement for the radius Johnson and Guestrin (2016). In the Blitz framework, their approach leads to a potentially smaller radius, using a strongly concave upper bound of the dual function whose maximum is known. In our framework, this can be used to improve the safe radius by a* $\sqrt{2}$ *factor in the static case. This is unclear to us whether this can be done for the sequential and dynamic version. For the SVM problem Zimmert et al. (2015) got the same improvement by writing the duality gap as a function of primal variables only.*

3.2.3 Safe Active Set

Note that any time a safe rule is performed thanks to a safe region $\mathcal{R} = \mathcal{B}(\theta, r)$, one can associate a *safe active set* $\mathcal{A}_{\theta,r}$, consisting of the features that cannot be removed yet by the test in Equation (8). Hence, the safe active set contains the true support of $\hat{\beta}^{(\lambda)}$.

Definition 9 (Safe Active Set) *For a center* $\theta \in \Delta_X$ *and a radius* $r \geqslant 0$*, the safe (sphere) active set consists of the variables not eliminated by the associated (sphere) safe rule,* i.e.,

$$\mathcal{A}(\theta, r) := \{g \in \mathcal{G} : \Omega_g^D(X_g^\top \theta) + r\Omega_g^D(X_g) \geqslant 1\} \ . \tag{12}$$

When choosing $z = -G(X\beta)/\lambda$ as proposed in Section 3.2.1 as the current residual, the computation of $\theta = \Theta(z)$ in Equation (9) involves the computation of $\Omega^D(X^\top z)$. A straightforward implementation would cost $\mathcal{O}(np)$ operations. This can be avoided: when using a safe rule one knows that the index achieving the maximum for this norm is in $\mathcal{A}(\theta, r)$. Indeed, by construction of the safe active set, it is easy to see that $\Omega^D(X^\top z) = \max_{g \in \mathcal{A}(\theta,r)} \Omega_g^D(X_g^\top z)$. In practice the evaluation of the dual gap is therefore $\mathcal{O}(nq)$ where q is the size of $\mathcal{A}(\theta, r)$. In other words, using a safe screening rule also speeds up the evaluation of the stopping criterion.

3.3 Outline of the Algorithm

When designing a supervised learning algorithm with sparsity enforcing penalties, the tuning of the parameter λ in Problem (3) is crucial and is usually done by cross-validation which requires evaluation over a grid of parameter values. A standard grid considered in the literature is $\lambda_t = \lambda_{\max} 10^{-\delta t/(T-1)}$ with a small δ, say $\delta = 10^{-2}$ or 10^{-3}, see for instance (Bühlmann and van de Geer, 2011)[2.12.1] or the *glmnet* package (Friedman et al., 2010). The parameter δ has an important influence on the computational burden: computing time tends to increase for small λ, the primal iterates being less and less sparse, and the problem to solve more and more ill-posed. It is customary to start from the largest regularizer $\lambda_0 = \lambda_{\max}$ and then to perform iteratively the computation of $\hat{\beta}^{(\lambda_t)}$ after the one of $\hat{\beta}^{(\lambda_{t-1})}$. This leads to computing the models in the order of increasing complexity: this allows important speed-up by benefiting of warm start strategies.

Here we propose a simple pathwise algorithm divided in two step:

- **Active warm start**: improve solver initialization by solving the problem restricted to an initial estimation of the support based on sequential informations along the regularization path (see Section 4.4 for details on the various strategies investigated).
- **Dynamic Gap Safe Screening**: use the informations gained during the iterations of the algorithm to obtain a smaller safe region therefore a greater elimination of inactive variables (see Section 4.3).

We summarize our strategy for solving the problem given by Equation (3) in Algorithm 1 and 2. The notation Solver $(\ldots)$ refers to any numerical solver that produces an approximation of the solution of (3) and SolverUpdate $(\ldots)$ is the updating scheme of the current vector along the iterations[5]. We consider solvers that can use a (primal) warm start point.

4. Screening strategies and theoretical analysis

We now describe the simplest safe rule strategy, which we refer to as the static strategy.

4.1 Static Safe Rules

The first static safe rule, introduced by El Ghaoui et al. (2012), discards variables before any computation thanks to Proposition 4. Here, the (safe) sphere is fixed once and for all,

5. For our experiments we have focused on (block) coordinate descent solvers

Algorithm 1 Pathwise algorithm with active warm start

Input : $X,\ y,\ \epsilon,\ K,\ f^{\text{ce}},\ (\lambda_t)_{t\in[T-1]}$
for $t \in [T-1]$ **do**
  $\beta = \check{\beta}^{(\lambda_{t-1})}$ and // Get previous ϵ-solution
  Get an initial (safe or not) support estimator $\mathcal{S} = \mathcal{S}(\check{\beta}^{(\lambda_{t-1})})$
  $\beta_{\mathcal{S}} = \text{Solver}\,(X_{\mathcal{S}},\ y,\ \beta_{\mathcal{S}},\ \epsilon,\ K,\ f^{\text{ce}},\ \lambda_t)$ // Active warm start
  $\check{\beta}^{(\lambda_t)} = \text{Solver}\,(X,\ y,\ \beta,\ \epsilon,\ K,\ f^{\text{ce}},\ \lambda_t)$ // Solve over all variables
Output: $\left(\check{\beta}^{(\lambda_t)}\right)_{t\in[T-1]}$

Algorithm 2 Iterative solver with GAP safe rules: $\text{Solver}\,(X,\ y,\ \beta,\ \epsilon,\ K,\ f^{\text{ce}},\ \lambda)$

Input : $X,\ y,\ \beta,\ \epsilon,\ K,\ f^{\text{ce}},\ \lambda$ // Warm start is authorized here through β
for $k \in [K]$ **do**
  if $k \mod f^{\text{ce}} = 1$ **then**
    Compute a dual variable $\theta = -G(X\beta)/\max(\lambda, \Omega^D(X^\top G(X\beta)))$
    Stop if $\text{Gap}_\lambda(\beta,\theta) \leqslant \epsilon$
    $r = \sqrt{\frac{2\text{Gap}_\lambda(\beta,\theta)}{\gamma\lambda^2}}$ // Get Gap Safe radius as in Equation (11)
    $\mathcal{A} = \left\{g \in \mathcal{G} : \Omega_g^D(X_g^\top\theta) + r\Omega_g^D(X_g) \geqslant 1\right\}$ // Get Safe active set as in Equation (12)
  $\beta_{\mathcal{A}} = \text{SolverUpdate}\,(X_{\mathcal{A}},\ y,\ \beta_{\mathcal{A}},\ \lambda)$ // Solve on current Safe active set
Output: $\check{\beta}^{(\lambda)}$

hence the name static. The static rule reads:

$$\begin{aligned}
\underline{\text{Static sphere rule}}: &\quad \text{If} \quad \Omega_g^D(X_g^\top\theta_{\max}) + r_{\max}\Omega_g^D(X_g) < 1, \quad \text{then} \quad \hat{\beta}_g^{(\lambda)} = 0\ ,\\
\text{Center}: &\quad \theta_{\max} := -G(0)/\lambda_{\max}\ ,\\
\text{Radius}: &\quad r_{\max} := \sqrt{2\text{Gap}_\lambda(0,\theta_{\max})/\gamma\lambda^2}\ .
\end{aligned}$$

There is a threshold λ_{critic} such that for any λ smaller than λ_{critic} the test from the Static sphere rule can never be satisfied. This phenomenon appears clearly in the numerical experiments presented in Section 6. In simple cases a closed form for λ_{critic} can even be provided. For instance, in the case of the Group Lasso, El Ghaoui et al. (2012) proposed to use $r_{\max} = \left|\frac{1}{\lambda} - \frac{1}{\lambda_{\max}}\right| \|y\|_2$, and simple calculation gives:

$$\lambda_{\text{critic}} := \lambda_{\max} \times \min_{g\in\mathcal{G}} \frac{\|y\|_2\, \Omega_g^D(X_g)}{\lambda_{\max} + \|y\|_2\, \Omega_g^D(X_g) - \Omega_g^D(X_g G(0))}\ .$$

4.2 Sequential Safe Rules

Provided that the λ's are close enough along the regularization parameters, knowing an estimate of $\hat{\beta}^{(\lambda_{t-1})}$ gives a clever initialization to compute $\hat{\beta}^{(\lambda_t)}$. To initialize the solver for a new λ_t, a natural choice is to set the primal variable equal to $\check{\beta}^{(\lambda_{t-1})}$, an approximation of $\hat{\beta}^{(\lambda_{t-1})}$ output by the solver (at a prescribed precision). This popular strategy is referred to as "warm start" in the literature (Friedman et al., 2007). On top of this standard strategy,

one can reuse prior dual information to improve the screening as well (El Ghaoui et al., 2012). This leads to the sequential strategy to screen for a new λ_t:

$$\begin{aligned}
\underline{\text{Sequential sphere rule}}: &\quad \text{If} \quad \Omega_g^D(X_g^\top\check{\theta}^{(\lambda_{t-1})}) + r_t\Omega_g^D(X_g) < 1, \quad \text{then} \quad \hat{\beta}_g^{(\lambda_t)} = 0\ ,\\
\text{Center}: &\quad \check{\theta}^{(\lambda_{t-1})} := \Theta(-G(X\check{\beta}^{(\lambda_{t-1})})/\lambda_{t-1})\ ,\\
\text{Radius}: &\quad r_t := \sqrt{2\text{Gap}_{\lambda_t}(\check{\beta}^{(\lambda_{t-1})}, \check{\theta}^{(\lambda_{t-1})})/\gamma\lambda_t^2}\ .
\end{aligned}$$

Remark 10 *Previous works in the literature (Wang et al., 2012; Wang and Ye, 2014; Wang et al., 2014; Lee and Xing, 2014) proposed sequential safe rules, though they were generally used in an unsafe way in practice. Indeed, such rules relied on the exact knowledge of $\hat{\theta}^{(\lambda_{t-1})}$ to screen out coordinates of $\hat{\beta}^{(\lambda_t)}$. Unfortunately, it is impossible to obtain such a point*[6] *since it is the solution of an optimization problem typically solved by an iterative solver, hence such it is only known up to a limited precision. By ignoring such inaccuracy in the knowledge of $\hat{\theta}^{(\lambda_{t-1})}$ one can wrongly eliminate variables that do belong to the support of $\hat{\beta}^{(\lambda_t)}$. Without a posteriori checking the screened out features, this could prevent the algorithm from converging, as shown in (Ndiaye et al., 2016b, Appendix B).*

4.3 Dynamic Safe Rules

Another road to speed up solvers using screening rules was proposed by Bonnefoy et al. (2014, 2015) under the name "dynamic safe rules". For a fixed λ, it consists in performing screening along with the iterations of the optimization algorithm used to solve Problem (3). Denoting by k the iteration number, they introduced a rule for the Lasso that consists of a safe sphere with center y/λ and radius $\|y/\lambda - \theta_k\|$, where θ_k is a current dual feasible point.

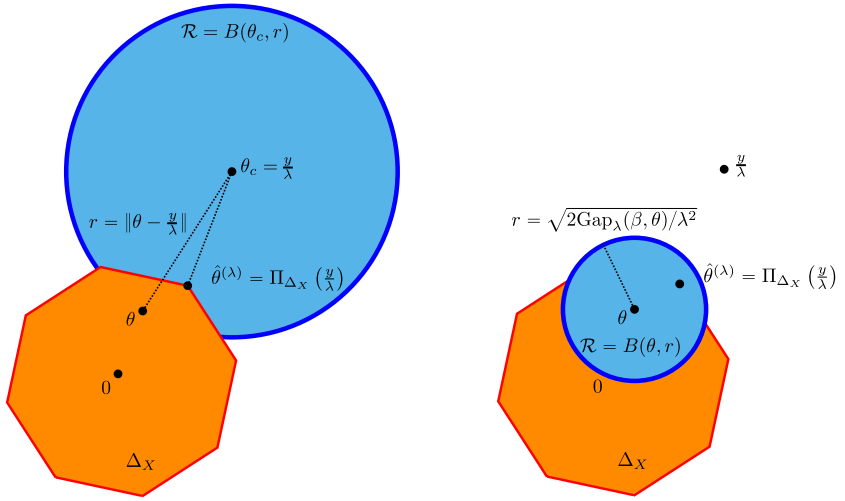
Let us consider a sequence (β_k) that converges to a primal solution $\hat{\beta}^{(\lambda)}$. For creating a dual feasible point, we apply the rescaling introduced in Equation (9) to $z = -G(X\beta_k)/\lambda$ and the dynamic strategy can be summarized by

$$\begin{aligned}
\underline{\text{Dynamic sphere rule}}: &\quad \text{If} \quad \Omega_g^D(X_g^\top\theta_k) + r_k\Omega_g^D(X_g) < 1, \quad \text{then} \quad \hat{\beta}_g^{(\lambda)} = 0\ ,\\
\text{Center}: &\quad \theta_k := \Theta(-G(X\beta_k)/\lambda) \text{ defined thanks to (9)}\ ,\\
\text{Radius}: &\quad r_k := \sqrt{2\text{Gap}_\lambda(\beta_k,\theta_k)/\gamma\lambda^2}\ .
\end{aligned}$$

In practice the computation of the duality gap can be expensive due to the matrix vector operations needed to compute $X^\top G(X\beta_k)$. For instance in the Lasso case, a dual gap computation requires almost as much computation as a full pass of coordinate descent over the data. Hence, it is recommended to evaluate the dynamic (safe) rule only every few passes over the data set. In all our experiments, we have set this screening frequency parameter to $f^{ce} = 10$.

Note that contrary to the original dynamic screening rules proposed by Bonnefoy et al. (2014, 2015), the Gap Safe rules we introduced are converging in the sense that our safe regions converge to the singleton $\{\hat{\theta}^{(\lambda)}\}$ (see Section 4 for more details). Indeed, their proposed safe sphere was centered on y/λ, and their radius can only be greater than $\|y/\lambda - \hat{\theta}^{(\lambda)}\|$ in the Lasso case they consider. We provide a visual comparison in Figure 1.

6. Except for $\lambda_0 = \lambda_{\max}$, where $\theta_{\max} := -G(0)/\lambda_{\max}$ can be computed exactly



(a) Bonnefoy *et al.* (2014) safe region

(b) Gap Safe region

Figure 1: Illustration of safe region differences between Bonnefoy *et al.* (2014) and Gap Safe strategies for the Lasso case; note that $\gamma = 1$ in this case. Here β is a primal point, θ is a dual feasible point (the feasible region Δ_X is in orange, while the respective safe balls $\mathcal{R}$ are in blue).

4.4 Active Warm Start

An another variant to further reduce running time in the *active warm start*, recently introduced by Ndiaye et al. (2016a) for speeding-up concomitant Lasso computations. Instead of simply leveraging the previous primal solution, the *active warm start* strategy also makes use of the previous safe active set $\mathcal{A}(\theta_{t-1}, r_{t-1})$, with $\theta_{t-1} = \check{\theta}^{(\lambda_{t-1})}$ and $r_{t-1} = r_{\lambda_{t-1}}(\check{\beta}^{(\lambda_{t-1})}, \check{\theta}^{(\lambda_{t-1})})$. The idea is to take as a new primal warm start point, the (approximate) minimizer of P_{λ_t} under the additional constraint that its support is included in the safe active set $\mathcal{A}(\theta_{t-1}, r_{t-1})$ *i.e.,*

$$\widetilde{\beta}^{(\lambda_{t-1}, \lambda_t)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, F(\beta) + \lambda_t \Omega(\beta) \text{ s.t. } \text{supp}(\beta) \subseteq \mathcal{A}(\theta_{t-1}, r_{t-1}) \ . \tag{13}$$

In (13), we still choose $\check{\beta}^{(\lambda_{t-1})}$ as a standard warm start initialization with the same number of inner loops and/or accuracy as in (3) (to avoid the multiplication of parameters to be set by the user). Note that un-safe estimators of the active set can be used as for active warm start. In practice, we can use the (un-safe) strong active set provided by the Strong rules introduced by Tibshirani et al. (2012). This *Strong warm start* strategy is detailed in Section 4.6.

4.5 Theoretical Analysis

Dynamic safe screening rules have practical benefits since they increase the number of screened out variables as the algorithm proceeds. In this section, it is shown that Gap Safe rules allow to have sharper and sharper dual regions along the iterations, accelerating support identification. Before this, the following proposition states that if one relies on a primal converging algorithm, then the dual sequence we propose is also converging. Note that the convergence is maintained to the same primal solution when the primal solution is non-unique.

Proposition 11 (Convergence of the Dual Points) *Let β_k be a current estimate of a primal solution $\hat{\beta}^{(\lambda)}$ and $\theta_k = \Theta(-G(X\beta_k)/\lambda)$ be the current estimate of $\hat{\theta}^{(\lambda)}$. Then, $\lim_{k\to+\infty} \beta_k = \hat{\beta}^{(\lambda)}$ implies $\lim_{k\to+\infty} \theta_k = \hat{\theta}^{(\lambda)}$.*

Proof Let $\alpha_k = \max(\lambda, \Omega^D(X^\top G(X\beta_k)))$, we have:

$$\left\|\theta_k - \hat{\theta}^{(\lambda)}\right\|_2 = \left\|\frac{G(X\hat{\beta}^{(\lambda)})}{\lambda} - \frac{G(X\beta_k)}{\alpha_k}\right\|_2 \leqslant \left|\frac{1}{\lambda} - \frac{1}{\alpha_k}\right| \|G(X\beta_k)\|_2 + \frac{\left\|G(X\hat{\beta}^{(\lambda)}) - G(X\beta_k)\right\|_2}{\lambda}.$$

If $\beta_k \to \hat{\beta}^{(\lambda)}$, then $\alpha_k \to \max(\lambda, \Omega^D(X^\top G(X\hat{\beta}^{(\lambda)}))) = \max(\lambda, \lambda\Omega^D(X^\top\hat{\theta}^{(\lambda)})) = \lambda$ since $G(X\hat{\beta}^{(\lambda)}) = -\lambda\hat{\theta}^{(\lambda)}$ thanks to the link-equation (5) and since $\hat{\theta}^{(\lambda)}$ is feasible, $\Omega^D(X^\top\hat{\theta}^{(\lambda)}) \leqslant 1$. Hence, both terms in the previous inequality converge to zero. ■

Let us now describe the notion of converging safe regions and converging safe rules introduced in (Fercoq et al., 2015, Definition 1).

Definition 12 *Let $(\mathcal{R}_k)_{k\in\mathbb{N}}$ be a sequence of closed convex sets in $\mathbb{R}^n$ containing $\hat{\theta}^{(\lambda)}$. It is a converging sequence of safe regions if the diameters of the sets converge to zero. The associated safe screening rules are referred to as converging.*

When $\theta_k = \Theta(-G(X\beta_k)/\lambda)$, Proposition 11 guarantees that Gap Safe spheres are converging. Indeed, the sequence of radius $r_k = (2\text{Gap}_\lambda(\beta_k, \theta_k)/(\gamma\lambda^2))^{1/2}$ converges to 0 with k by strong duality, hence the sequence $\mathcal{B}(\theta_k, r_k)$ converges to $\{\hat{\theta}^{(\lambda)}\}$ which means that the proposed Gap Safe sphere is asymptotically optimal.

We now prove that one recovers a specific set, called the *equicorrelation set* in finite time with any converging strategy.

Definition 13 *The equicorrelation set is defined as* $\mathcal{E}_\lambda := \left\{g \in \mathcal{G} : \Omega_g^D(X_g^\top\hat{\theta}^{(\lambda)}) = 1\right\}$.

Indeed, the following proposition asserts that after a finite number of steps, the equicorrelation set $\mathcal{E}_\lambda$ is exactly identified. Such a property is sometimes referred to as finite identification of the support (Liang et al., 2014) and is summarized in the following proposition. Yet, note that the (primal) optimal support can be strictly smaller than the equicorrelation set, see Tibshirani (2013). For clarity, links between optimal support, sure active sets, equicorrelation set are illustrated in Figure 2.

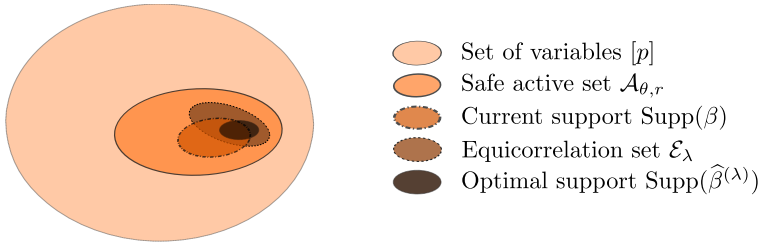


Figure 2: Illustration of the inclusions between several remarkable sets: supp$(\beta) \subseteq \mathcal{A}(\theta, r) \subseteq [p]$ and supp$(\hat{\beta}^{(\lambda)}) \subseteq \mathcal{E}_\lambda \subseteq \mathcal{A}(\theta, r) \subseteq [p]$, where β, θ is a primal/dual pair.

Proposition 14 (Identification of the Equicorrelation Set)
For any sequence of converging safe active set $(\mathcal{A}(\theta_k, r_k))_{k\in\mathbb{N}}$, we have $\lim_{k\to\infty} \mathcal{A}(\theta_k, r_k) = \mathcal{E}_\lambda$. More precisely, there exists an integer $k_0 \in \mathbb{N}$ such that $\mathcal{A}(\theta_k, r_k) = \mathcal{E}_\lambda$ for all $k \geqslant k_0$.

Proof We proceed by double inclusion. First remark that $\mathcal{E}_\lambda = \mathcal{A}(\hat{\theta}^{(\lambda)}, \hat{r}_\lambda)$ where $\hat{r}_\lambda := 2(P_\lambda(\hat{\beta}^{(\lambda)}) - D_\lambda(\hat{\theta}^{(\lambda)}))/\gamma\lambda^2 = 0$ (thanks to strong duality), so for all $k \in \mathbb{N}$, we have $\mathcal{E}_\lambda \subseteq \mathcal{A}(\theta_k, r_k)$.

Reciprocally, suppose that there exists a non active group $g \in \mathcal{G}$ *i.e.,* $\Omega_g^D(X_g^\top \hat{\theta}^{(\lambda)}) < 1$ that remains in the active set $\mathcal{A}(\theta_k, r_k)$ for all iterations *i.e.,* $\forall k \in \mathbb{N}$, $\Omega_g^D(X_g^\top \theta_k) + r_k \Omega_g^D(X_g) \geqslant 1$. Since $\lim_{k\to\infty} \theta_k = \hat{\theta}^{(\lambda)}$ and $\lim_{k\to\infty} r_k = 0$, we obtain $\Omega_g^D(X_g^\top \hat{\theta}^{(\lambda)}) \geqslant 1$ by passing to the limit. Hence, by contradiction, there exits an integer $k_0 \in \mathbb{N}$ such that $\mathcal{E}_\lambda^c \subseteq \mathcal{A}(\theta_{k_0}, r_{k_0})^c$. ■

4.6 Alternative Strategies: a Brief Survey

4.6.1 The Seminal Safe Regions

The first Safe Screening rules introduced by El Ghaoui et al. (2012) can be generalized to Problem (3) as follows. Take $\hat{\theta}^{(\lambda_0)}$ the optimal solution of the dual problem (4) with a regularization parameter λ_0. Since $\hat{\theta}^{(\lambda)}$ is optimal for problem (4) one obtains $\hat{\theta}^{(\lambda)} \in \{\theta : D_\lambda(\theta) \geqslant D_\lambda(\hat{\theta}^{(\lambda_0)})\}$. This set was proposed as a safe region by El Ghaoui et al. (2012).

In the regression case (where $f_i(z) = (y_i - z)^2/2$), it is straightforward to see that it corresponds to the safe sphere $\mathcal{B}(y/\lambda, \|y/\lambda - \hat{\theta}^{(\lambda_0)}\|_2)$.

4.6.2 ST3 and Dynamic ST3

Following (7), the safe sphere test given by Equation (8) is more efficient when θ is near $\hat{\theta}^{(\lambda)}$ and r close to zero. This motivated the following improvements.

A refined sphere rule can be obtained in the regression case by exploiting geometric informations in the dual space. This method was originally proposed in Xiang et al. (2011) and extended in Bonnefoy et al. (2014) with a dynamic refinement of the safe region.

Let $g_\star \in \arg\max_{g\in\mathcal{G}} \Omega_g^D(X^\top y)$ (note that $\Omega_{g_\star}^D(X^\top y) = \lambda_{\max}$), and let us define

$$\mathcal{V}_\star := \{\theta \in \mathbb{R}^n : \Omega_{g_\star}^D(X_{g_\star}^\top \theta) \leqslant 1\} \text{ and } \mathcal{H}_\star := \{\theta \in \mathbb{R}^n : \Omega_{g_\star}^D(X_{g_\star}^\top \theta) = 1\}.$$

We assume that the dual norm is differentiable at $X_{g_\star}^\top y/\lambda_{\max}$ (which is true in all the cases presented in Section 5). Let $\eta := X_{g_\star} \nabla \Omega_{g_\star}^D(X_{g_\star}^\top y/\lambda_{\max})$ be the vector normal to $\mathcal{V}_\star$ at $y/\lambda_{\max}$ and define

$$\theta := \Pi_{\mathcal{H}_\star}\left(\frac{y}{\lambda}\right) = \frac{y}{\lambda} - \frac{\langle \frac{y}{\lambda}, \eta\rangle - 1}{\|\eta\|_2^2}\eta \text{ and } r_\theta := \sqrt{\left\|\frac{y}{\lambda} - \theta\right\|_2^2 - \left\|\frac{y}{\lambda} - \theta\right\|_2^2},$$

where $\theta \in \Delta_X$ is any dual feasible vector. Following the proof in (Ndiaye et al., 2016b, Appendix D), one can show that $\hat{\theta}^{(\lambda)} \in \mathcal{B}(\theta, r_\theta)$. The special case where $\theta = y/\lambda_{\max}$ corresponds to the original ST3 introduced in Xiang et al. (2011) for the Lasso. A further improvement can be obtained by choosing dynamically $\theta = \theta_k$ along the iterations of an algorithm, this strategy corresponding to DST3 introduced in Bonnefoy et al. (2014, 2015) for the Lasso and Group Lasso, and in Ndiaye et al. (2016b) for the Sparse-Group Lasso.

4.6.3 Dual Polytope Projection

In the regression case, Wang et al. (2012) explore other geometric properties of the dual solution. Their method is based on the non-expansiveness of projection operators[7]. Indeed, for $\hat{\theta}^{(\lambda)}$ (resp. $\hat{\theta}^{(\lambda_0)}$)) being optimal dual solution of (4) with parameter λ (resp. λ_0), one has: $\|\hat{\theta}^{(\lambda)} - \hat{\theta}^{(\lambda_0)}\|_2 = \|\Pi_{\Delta_X}(y/\lambda) - \Pi_{\Delta_X}(y/\lambda_0)\|_2 \leqslant \|y/\lambda - y/\lambda_0\|_2$ and hence $\hat{\theta}^{(\lambda)} \in \mathcal{B}(\hat{\theta}^{(\lambda_0)}, \|y/\lambda - y/\lambda_0\|_2)$. Unfortunately, those regions are intractable since they required the *exact* knowledge of the optimal solution $\hat{\theta}^{(\lambda_0)}$ which is not available in practice (except for $\lambda_0 = \lambda_{\max}$). It may lead to un-safe screening rules as discussed in Remark 10.

Remark 15 *The preceding spheres are mainly based on the fact that $\hat{\theta}^{(\lambda)} = \Pi_{\Delta_X}(y/\lambda)$ which is limited to the regression case. Thus, those methods are not appropriate for more general data fitting term which greatly reduces the scope of such rules.*

Remark 16 *The radius of the regions above do not converge to zero even in the dynamic case (DST3), and the (fixed) center of the preceding sphere can be far from $\hat{\theta}^{(\lambda)}$ when λ gets small. Thus, those regions are not converging and are irrelevant for dynamic screening.*

4.6.4 Strong rules

The Strong rules were introduced in Tibshirani et al. (2012) as a *heuristic* extension of the safe rules. It consists in relaxing the *safe* properties to discard features more aggressively, and can be formalized as follows. Assume that the gradient of the data fitting term ∇F is group-wise non-expansive w.r.t. the dual norm $\Omega_g^D(\cdot)$ along the regularization path *i.e.,* that

7. The authors also proved an enhanced version of this safe region by using the firm non-expansiveness of the projection operator.

for any $g \in \mathcal{G}$, any $\lambda > 0, \lambda' > 0$, $\Omega_g^D\big(\nabla_g F(\hat{\beta}^{(\lambda)}) - \nabla_g F(\hat{\beta}^{(\lambda')})\big) \leqslant |\lambda - \lambda'|$. When choosing two regularization parameters such that $\lambda < \lambda'$ one has:

$$\begin{aligned}\lambda\Omega_g^D\left(X_g^\top\hat{\theta}^{(\lambda)}\right) = \Omega_g^D\left(\nabla_g F(\hat{\beta}^{(\lambda)})\right) &\leqslant \Omega_g^D\left(\nabla_g F(\hat{\beta}^{(\lambda')})\right) + \Omega_g^D\left(\nabla_g F(\hat{\beta}^{(\lambda)}) - \nabla_g F(\hat{\beta}^{(\lambda')})\right)\\ &\leqslant \Omega_g^D\left(\nabla_g F(\hat{\beta}^{(\lambda')})\right) + |\lambda - \lambda'|\\ &= \lambda'\Omega_g^D\left(X_g^\top\hat{\theta}^{(\lambda')}\right) + \lambda' - \lambda.\end{aligned}$$

Combining this with the screening rule (7), one obtains:

$$\Omega_g^D\left(X_g^\top\hat{\theta}^{(\lambda')}\right) < \frac{2\lambda - \lambda'}{\lambda'} \Longrightarrow \Omega_g^D\left(X_g^\top\hat{\theta}^{(\lambda)}\right) < 1 \Longrightarrow \hat{\beta}_g^{(\lambda)} = 0. \tag{14}$$

The set of variables not eliminated is called the *strong active set* and is defined as:

$$\mathcal{STG}(\hat{\theta}^{(\lambda')}, \lambda, \lambda') := \left\{g \in \mathcal{G} : \Omega_g^D\left(X_g^\top\hat{\theta}^{(\lambda')}\right) \geqslant \frac{2\lambda - \lambda'}{\lambda'}\right\}. \tag{15}$$

Note that Strong rules are un-safe because the non-expansiveness condition on the (gradient of the) data fitting term is usually not satisfied without stronger assumptions on the design matrix X; see discussion in (Tibshirani et al., 2012, Section 3). It requires the exact knowledge of $\hat{\theta}^{(\lambda')}$ which is not available in practice. Using such rules, the authors advised to check the KKT condition[8] a posteriori, to avoid removing wrongly some features.

To overcome this limitation, we propose to use the strong active set $\mathcal{STG}(\hat{\theta}^{(\lambda_{t-1})}, \lambda_t, \lambda_{t-1})$ defined by Equation (15) for an active warm start strategy (cf. Section 4.4). We compare below this strategy with the one using $\mathcal{A}(\theta_{t-1}, r_{t-1})$ in Equation (13) as initial active set. A similar strategy is also used in the "big lasso" package by Zeng and Breheny (2017) as a hybrid screening strategy that *"alleviates the computational burden of KKT post-convergence checking for the strong rules by not checking features that can be safely eliminated"*. However, our warm start strategy (active or strong) does not require post-processing steps.

4.6.5 Correlation Based Rule

Previous works in statistics have proposed various model-based screening methods to select important variables. Those methods discard variables with small correlation between the features and response variables. For instance Sure Independence Screening (SIS) by Fan and Lv (2008) reads: for a chosen critical threshold γ (such that the number of selected variables is smaller than a prescribed proportion of the features),

$$\text{If } \Omega_g^D(X_g^\top y) < \gamma \text{ then remove } X_g \text{ from the problem.}$$

8. The post-processing for the Lasso (with the notation from Section 5.1) adds back variables violating the approximated KKT conditions

$$KKT_\epsilon : \begin{cases} |X_j^\top\theta| \leqslant 1 + \epsilon, & \text{if } \beta_j = 0,\\ |X_j^\top\theta - \operatorname{sign}(\beta_j)| \leqslant \epsilon, & \text{if } \beta_j \neq 0.\end{cases} \tag{16}$$

One can show that $\operatorname{Gap}_\lambda(\beta, \theta) \leqslant (1 - \lambda/\alpha)^2 \|y - X\beta\|^2/2 + \lambda\epsilon\|\beta\|_1$ where $\alpha = \max(\lambda, \|X^\top(y - X\beta)\|_\infty)$. Hence choosing $\epsilon = \epsilon'/P_\lambda(\beta) - (1 - \lambda/\alpha)^2$ imply an ϵ'-duality gap.

	Lasso	Multi-task regr.	Logistic regr.	Multinomial regr.
$f_i(z)$	$\frac{(y_i - z)^2}{2}$	$\frac{\|Y_i - z\|^2}{2}$	$\log(1 + \mathrm{e}^z) - y_i z$	$\log\left(\sum_{k=1}^q \mathrm{e}^{z_k}\right) - \sum_{k=1}^q Y_{i,k} z_k$
$f_i^*(u)$	$\frac{(u+y_i)^2 - y_i^2}{2}$	$\frac{\|u+y_i\|^2 - \|Y_i\|_2^2}{2}$	$\mathrm{Nh}(u + y_i)$	$\mathrm{NH}(u + Y_i)$
$G(\theta)$	$\theta - y$	$\theta - Y$	$\frac{\mathrm{e}^\theta}{1+\mathrm{e}^\theta} - y$	$\mathrm{RowNorm}(\mathrm{e}^\theta) - Y$
γ	1	1	4	1

	ℓ_1	ℓ_1/ℓ_2	$\ell_1 + \ell_1/\ell_2$
$\Omega(\beta)$	$\|\beta\|_1$	$\sum_{g\in\mathcal{G}} \|\beta_g\|_2$	$\tau\|\beta\|_1 + (1-\tau)\sum_{g\in\mathcal{G}} w_g\|\beta_g\|_2$
$\Omega^D(\xi)$	$\max_{j\in[p]} \lvert\xi_j\rvert$	$\max_{g\in\mathcal{G}} \|\xi_g\|_2$	$\max_{g\in\mathcal{G}} \frac{\|\xi_g\|_{\epsilon_g}}{\tau + (1-\tau)w_g}$, $\epsilon_g := \frac{(1-\tau)w_g}{\tau + (1-\tau)w_g}$

Table 1: Useful ingredients for computing Gap Safe rules. We have used lower case to indicate when the parameters are vectors or not (following the notation used in Section 5.5 and 5.6). The function RowNorm consists in normalizing a (non-negative) matrix row-wise, such that each row sums to one. The details for computing the ϵ-norm $\|\cdot\|_{\epsilon_g}$ is given in Proposition 17.

It is a marginal oriented variable selection method and it is worth noting that SIS can be recast as a static sphere test in linear regression scenarios:

$$\text{If } \Omega_g^D(X_g^\top y) < \gamma = \lambda\left(1 - r\Omega_g^D(X_g)\right) \text{ then } \hat{\beta}_g^{(\lambda)} = 0 \,(\text{remove } X_g).$$

Other refinements can also be found in the literature such as iterative screening (ISIS) (Fan and Lv, 2008), that bears some similarities with dynamic sphere safe tests.

5. Gap Safe Rule for Popular Estimators

We now detail how our results apply to relevant supervised learning problems. A summary synthesizing the different learning task we are addressing is given in Table 1.

5.1 Lasso

For the Lasso estimator (Tibshirani, 1996), the data-fitting term is the standard least square, *i.e.*, $F(\beta) = \|y - X\beta\|_2^2/2 = \sum_{i=1}^n (y_i - x_i^\top\beta)^2/2$ (meaning that $f_i(z) = (y_i - z)^2/2$). The regularization term enforces sparsity at the feature level and is defined by

$$\Omega(\beta) = \|\beta\|_1 \quad \text{and} \quad \Omega^D(\xi) = \|\xi\|_\infty = \max_{j\in[p]} |\xi_j|.$$

5.2 Group Lasso

For the Group Lasso estimator (Yuan and Lin, 2006), the data-fitting term is the same $F(\beta) = \|y - X\beta\|_2^2/2$ but the penalty considered enforces group sparsity. Hence, we consider

the norm $\Omega(\beta) = \Omega_w(\beta)$, often referred to as an ℓ_1/ℓ_2 norm, defined by:

$$\Omega_w(\beta) := \sum_{g\in\mathcal{G}} w_g \left\|\beta_g\right\|_2 \quad \text{and} \quad \Omega_w^D(\xi) := \max_{g\in\mathcal{G}} \frac{\|\xi_g\|_2}{w_g} \ ,$$

where $w = (w_g)_{g\in\mathcal{G}}$ are some weights satisfying $w_g > 0$ for all $g \in \mathcal{G}$.

5.3 Sparse-Group Lasso

In the Sparse-Group Lasso case, we also have $\beta \in \mathbb{R}^p$ and $F(\beta) = \|y - X\beta\|_2^2/2$ but the regularization $\Omega(\beta) = \Omega_{\tau,w}(\beta)$ is defined by

$$\Omega_{\tau,w}(\beta) := \tau\|\beta\|_1 + (1-\tau)\sum_{g\in\mathcal{G}} w_g \left\|\beta_g\right\|_2 ,$$

for $\tau \in [0,1], w = (w_g)_{g\in\mathcal{G}}$ with $w_g \geqslant 0$ for all $g \in \mathcal{G}$. Note that we recover the Lasso if $\tau = 1$, and the Group Lasso if $\tau = 0$; the case where $w_g = 0$ for some $g \in \mathcal{G}$ together with $\tau = 0$ is excluded ($\Omega_{\tau,w}$ is not a norm in such a case). This estimator was introduced by Simon et al. (2013) to enforce sparsity both at the feature and at the group level, and was used in different applications such as brain imaging in Gramfort et al. (2013) or in genomics in Peng et al. (2010). Other hierarchical norms have also been proposed in Sprechmann et al. (2010) or Jenatton et al. (2011) and could be handled in our framework modulo additional technical details.

For the Sparse-Group Lasso, the geometry of the dual feasible set Δ_X is more complex (*cf.* Figure 3 for a comparison w.r.t. Lasso and Group Lasso). As a consequence, additional geometrical insights are needed to derive efficient safe rules, especially to compute the dual norm required by Equation (9) and the computation of the safe screening rules given in (7).

We now introduce the ϵ-norm (denoted $\|\cdot\|_\epsilon$) as it has a connection with the Sparse-Group Lasso norm $\Omega_{\tau,w}$. The ϵ-norm was first proposed in Burdakov (1988) for other purposes (see also Burdakov and Merkulov (2001)). For any $\epsilon \in [0,1]$ and any $x \in \mathbb{R}^d$, $\|x\|_\epsilon$ is defined as the unique nonnegative solution ν of the following equation (for $\epsilon = 0$, we define $\|x\|_{\epsilon=0} := \|x\|_\infty$):

$$\sum_{i=1}^{d} (|x_i| - (1-\epsilon)\nu)_+^2 = (\epsilon\nu)^2. \tag{17}$$

Using soft-thresholding, this is equivalent to solve in ν the equation $\|\mathrm{ST}_{(1-\epsilon)\nu}(x)\|_2 = \epsilon\nu$. Moreover, its dual norm is given by; see (Burdakov and Merkulov, 2001, Equation (42)):

$$\|\xi\|_\epsilon^D = \epsilon\|\xi\|_2^D + (1-\epsilon)\|\xi\|_\infty^D = \epsilon\|\xi\|_2 + (1-\epsilon)\|\xi\|_1 \ . \tag{18}$$

This allows to express the Sparse-Group Lasso norm $\Omega_{\tau,w}$ using the dual ϵ-norm. We now derive an explicit formulation for the dual norm of the Sparse-Group Lasso, originally proposed in (Ndiaye et al., 2016b, Prop. 4):

Proposition 17 *For all groups g in $\mathcal{G}$, let us introduce $\epsilon_g := \dfrac{(1-\tau)w_g}{\tau + (1-\tau)w_g}$. Then, the Sparse-Group Lasso norm satisfies the following properties: for any β and ξ in $\mathbb{R}^p$,*

$$\Omega_{\tau,w}(\beta) = \sum_{g\in\mathcal{G}} (\tau + (1-\tau)w_g)\|\beta_g\|_{\epsilon_g}^D \quad \textit{and} \quad \Omega_{\tau,w}^D(\xi) = \max_{g\in\mathcal{G}} \frac{\|\xi_g\|_{\epsilon_g}}{\tau + (1-\tau)w_g}.$$

$$\mathcal{B}_{\Omega_{t,w}^D} = \left\{\xi \in \mathbb{R}^p : \forall g \in \mathcal{G}, \|\mathrm{ST}_\tau(\xi_g)\|_2 \leqslant (1-\tau)w_g\right\}.$$

$$\partial\Omega_{\tau,w}(\beta) = \left\{z \in \mathbb{R}^p : \forall g \in \mathcal{G}, z_g \in \tau\partial\|\cdot\|_1(\beta_g) + (1-\tau)w_g\partial\|\cdot\|_2(\beta_g)\right\}.$$

Hence the dual feasible set is given by

$$\begin{aligned}\Delta_X &= \left\{\theta \in \mathbb{R}^n : \forall g \in \mathcal{G}, \|\mathrm{ST}_\tau\left(X_g^\top\theta\right)\|_2 \leqslant (1-\tau)w_g\right\} \\ &= \left\{\theta \in \mathbb{R}^n : \forall g \in \mathcal{G}, \|X_g^\top\theta\|_{\epsilon_g} \leqslant \tau + (1-\tau)w_g\right\}.\end{aligned}$$

Remark 18 *Computing the dual norm of the Sparse-Group Lasso involves solving for each group $g \in \mathcal{G}$, a problem similar to the one given in (17) which has a quadratic complexity. To overcome this difficulty, an efficient algorithm relying on sorting techniques was proposed in (Ndiaye et al., 2016b, Prop. 5) to perform exact dual norm evaluation.*

The Sparse-Group Lasso benefits from two levels of screening: the safe rules can detect both group-wise zeros and coordinate-wise zeros in the remaining groups: for any group g in $\mathcal{G}$ and any safe sphere $\mathcal{B}(\theta, r)$, Equation (7) and the sub-differential of the Sparse-Group Lasso norm in Proposition 17 give (a detailed proof is given in (Ndiaye et al., 2016b, Appendix C))

$$\textbf{\textit{Group level safe screening rule:}} \qquad \max_{\theta\in\mathcal{B}(\theta,r)} \frac{\|X_g^\top\theta\|_{\epsilon_g}}{\tau + (1-\tau)w_g} < 1 \Rightarrow \hat{\beta}_g^{(\lambda)} = 0.$$

$$\textbf{\textit{Feature level safe screening rule:}} \qquad \forall j \in g, \max_{\theta\in\mathcal{B}(\theta,r)} |X_j^\top\theta| < \tau \Rightarrow \hat{\beta}_j^{(\lambda)} = 0.$$

Noting that $\|\mathrm{ST}_\tau(x)\|_2 = (1-\tau)w_g \iff \|x\|_{\epsilon_g} = \tau + (1-\tau)w_g$, the above screening test on the group level can be rewritten as

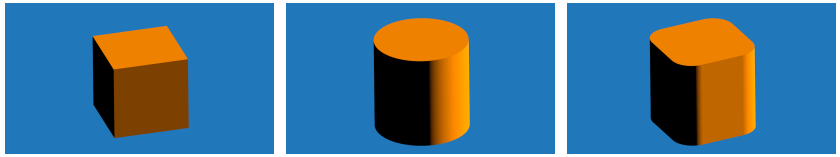
$$\max_{\theta\in\mathcal{B}(\theta,r)} \left\|\mathrm{ST}_\tau(X_g^\top\theta)\right\|_2 < (1-\tau)w_g \Rightarrow \hat{\beta}_g^{(\lambda)} = 0.$$

The advantage of this formulation is that one can easily derive a "tight" upper-bound of the non-convex optimization problem in the left hand side of the preceding test. Indeed, we have $\mathrm{ST}_\tau(x) = x - \Pi_{\tau\mathcal{B}_\infty}(x)$ which brings us finally into a geometric problem easier to solve. We recall from (Ndiaye et al., 2016b, Prop. 1) that for any center $\theta \in \Delta_X$, any group $g \in \mathcal{G}$ and any $j \in g$, we have the following upper-bound

$$\max_{\theta\in\mathcal{B}(\theta,r)} |X_j^\top\theta| \leqslant |X_j^\top\theta| + r\left\|X_j\right\|_2 ,$$

$$\max_{\theta\in\mathcal{B}(\theta,r)} \left\|\mathrm{ST}_\tau(X_g^\top\theta)\right\|_2 \leqslant T_g := \begin{cases} \left\|\mathrm{ST}_\tau(X_g^\top\theta)\right\|_2 + r\left\|X_g\right\|_2 , & \text{if } \left\|X_g^\top\theta\right\|_\infty > \tau, \\ (\left\|X_g^\top\theta\right\|_\infty + r\left\|X_g\right\|_2 - \tau)_+, & \text{otherwise.} \end{cases}$$

Hence we derive the two level of safe screening rule:



(a) Lasso dual ball $\mathcal{B}_{\Omega^D}$ for $\Omega^D(\theta) = \|\theta\|_\infty$.

(b) Group Lasso dual ball $\mathcal{B}_{\Omega^D}$ for $\Omega^D(\theta) = \max(\sqrt{\theta_1^2 + \theta_2^2}, |\theta_3|)$.

(c) Sparse-Group Lasso dual ball $\mathcal{B}_{\Omega^D} = \{\theta : \forall g \in \mathcal{G}, \|\mathrm{ST}_\tau(\theta_g)\|_2 \leqslant (1-\tau)w_g\}$.

Figure 3: Lasso, Group Lasso and Sparse-Group Lasso dual unit balls: $\mathcal{B}_{\Omega^D} = \{\theta : \Omega^D(\theta) \leqslant 1\}$. For the illustration, the group structure is chosen such that $\mathcal{G} = \{\{1,2\},\{3\}\}$, *i.e.,* $g_1 = \{1,2\}, g_2 = \{3\}$, $n = p = 3$, $w_{g_1} = w_{g_2} = 1$ and $\tau = 1/2$.

Proposition 19 (Safe screening rule for the Sparse-Group Lasso)

Group level screening: $\forall g \in \mathcal{G}, \quad$ *if* $T_g < (1-\tau)w_g, \quad$ *then* $\hat{\beta}_g^{(\lambda)} = 0.$

Feature level screening: $\forall g \in \mathcal{G}, \forall j \in g,$ *if* $|X_j^\top \theta| + r\|X_j\|_2 < \tau, \quad$ *then* $\hat{\beta}_j^{(\lambda)} = 0.$

In the same spirit than Proposition 14, for any safe region $\mathcal{R}$, *i.e.,* a set containing $\hat{\theta}^{(\lambda)}$, we define two levels of active sets, one for the group level and one for the feature level:

$$\mathcal{A}_{\mathrm{gp}}(\mathcal{R}) := \{g \in \mathcal{G}, \max_{\theta \in \mathcal{R}} \left\|\mathrm{ST}_\tau(X_g^\top \theta)\right\|_2 \geqslant (1-\tau)w_g\},$$
$$\mathcal{A}_{\mathrm{ft}}(\mathcal{R}) := \bigcup_{g \in \mathcal{A}_{\mathrm{gp}}(\mathcal{R})} \{j \in g : \max_{\theta \in \mathcal{R}} |X_j^\top \theta| \geqslant \tau\}.$$

If one considers sequence of converging regions, then the next proposition (see (Ndiaye et al., 2016b, Prop. 3)) states that we can identify in finite time the optimal active sets defined as follows:

$$\mathcal{E}_{\mathrm{gp}} := \left\{g \in \mathcal{G} : \left\|\mathrm{ST}_\tau(X_g^\top \hat{\theta}^{(\lambda)})\right\|_2 = (1-\tau)w_g\right\}, \ \mathcal{E}_{\mathrm{ft}} := \bigcup_{g \in \mathcal{E}_{\mathrm{gp}}} \left\{j \in g : |X_j^\top \hat{\theta}^{(\lambda)}| \geqslant \tau\right\}.$$

Proposition 20 *Let $(\mathcal{R}_k)_{k\in\mathbb{N}}$ be a sequence of safe regions whose diameters converge to 0. Then, $\lim_{k\to\infty} \mathcal{A}_{gp}(\mathcal{R}_k) = \mathcal{E}_{gp}$ and $\lim_{k\to\infty} \mathcal{A}_{ft}(\mathcal{R}_k) = \mathcal{E}_{ft}$.*

5.4 ℓ_1 Regularized Logistic Regression

Here, we consider the formulation given in (Bühlmann and van de Geer, 2011, Chapter 3) for the two-class logistic regression. In such a context, one observes for each $i \in [n]$ a class label $l_i \in \{1,2\}$. This information can be recast as $y_i = \mathbb{1}_{\{l_i=1\}}$ (where $\mathbb{1}$ is the indicator function), and it is then customary to minimize (3) where

$$F(\beta) = \sum_{i=1}^n \left(-y_i x_i^\top \beta + \log\left(1 + \exp\left(x_i^\top \beta\right)\right)\right), \tag{19}$$

with $f_i(z) = -y_i z + \log(1 + \exp(z))$, and the penalty is simply the ℓ_1 norm: $\Omega(\beta) = \|\beta\|_1$. Let us introduce Nh, the (binary) negative entropy function defined by:

$$\mathrm{Nh}(x) = \begin{cases} x\log(x) + (1-x)\log(1-x), & \text{if } x \in [0,1] \ , \\ +\infty, & \text{otherwise} \ . \end{cases} \tag{20}$$

We use the convention $0\log(0) = 0$, and one can check that $f_i^*(z_i) = \mathrm{Nh}(z_i + y_i)$ and $\gamma = 4$.

Remark 21 *We have privileged the formulation with the label $y \in \{0,1\}^n$ instead of $y \in \{+1,-1\}^n$ in order to be consistent with the multinomial cases below. One can simply switch from one formulation to the other thanks to the mapping $\tilde{y} = 2y - 1$.*

5.5 ℓ_1/ℓ_2 Multi-task Regression

The multi-task Lasso is a regression problem where the parameters form a matrix $\mathrm{B} \in \mathbb{R}^{p\times q}$. Denoting n the number of observations for each task $k \in [q]$, it is defined as

$$\min_{\mathrm{B}\in\mathbb{R}^{p\times q}} \frac{1}{2}\|Y - X\mathrm{B}\|_F^2 + \lambda \sum_{j=1}^p \|\mathrm{B}_{j,:}\|_2, \tag{21}$$

where $X \in \mathbb{R}^{n\times p}$ and $Y \in \mathbb{R}^{n\times q}$. Here we assume that the explanatory variables X are shared among the tasks however the Gap Safe rules would readily apply to the non-shared design formulation as in Lee et al. (2010) or in Liu et al. (2009) since the loss is still smooth (*cf.* Remark 7).

Introducing the vec operator that vectorizes a matrix by stacking its columns to form a column vector, and the Kronecker product $\otimes$ of two matrices, the multi-task Lasso can be rewritten as a special case of Group Lasso. In fact, we have n class of observations $c_i = (i + (k-1)n)_{k\in[q]}$ of size q for each $i \in [n]$ (the overall number of observations is $n' = nq$) and p groups $g_j = (j + (k-1)p)_{k\in[q]}$ such that $|g_j| = q$ for $j \in [p]$. The design matrix $\tilde{X} = I_q \otimes X \in \mathbb{R}^{n'\times p'} = \mathbb{R}^{nq\times pq}$ is a q-block diagonal matrix defined as $\tilde{X} = \mathrm{diag}(X,\ldots,X)$, $y = \mathrm{vec}(Y)$ and $\beta = \mathrm{vec}(\mathrm{B})$, we have:

$$\min_{\beta\in\mathbb{R}^{p'}} \frac{1}{2}\sum_{i=1}^n \left\|y_{c_i} - \tilde{x}_i^\top \beta\right\|_2^2 + \lambda \sum_{j=1}^p \left\|\beta_{g_j}\right\|_2, \tag{22}$$

i.e., $f_i(z) = \|y_{c_i} - z\|_2^2/2$. The advantage of this formulation is that it can be concisely written using the matrix forms of y and β, without the need to actually construct the large matrix X'. This is particularly appealing for the implementation.

In signal processing, this model is also referred to as the Multiple Measurement Vector (MMV) problem. It allows to jointly select the same features for multiple regression tasks, see (Argyriou et al., 2006, 2008; Obozinski et al., 2010). This estimator has been used in various applications such as prediction of the location of a protein within a cell (Xu et al., 2011) or in neuroscience (Gramfort et al., 2012), for instance to diagnose Alzheimer's disease (Zhang et al., 2012).

5.6 ℓ_1/ℓ_2 Multinomial Logistic Regression

We adapt the formulation given in (Bühlmann and van de Geer, 2011, Chapter 3) for the multinomial regression. In such a context, one observes for each $i \in [n]$ a class label $l_i \in [q]$. This information can be recast into a matrix $Y \in \mathbb{R}^{n\times q}$ filled by 0's and 1's: $Y_{i,k} = \mathbb{1}_{\{l_i=k\}}$ (where $\mathbb{1}$ is the indicator function). In the same spirit as for the multi-task Lasso, a matrix $\mathrm{B} \in \mathbb{R}^{p\times q}$ is formed by q vectors encoding the hyperplanes for the linear classification. Thus the multinomial ℓ_1/ℓ_2 regularized regression reads:

$$\min_{\mathrm{B}\in\mathbb{R}^{p\times q}} \sum_{i=1}^{n} \left(\sum_{k=1}^{q} -Y_{i,k} x_i^\top \mathrm{B}_{:,k} + \log\left(\sum_{k=1}^{q} \exp\left(x_i^\top \mathrm{B}_{:,k}\right)\right)\right) + \lambda \sum_{j=1}^{p} \|\mathrm{B}_{j,:}\|_2 \,. \tag{23}$$

Using a similar reformulation as in Section 5.5 *i.e.,* defining $c_i = (i + (k-1)n)_{k\in[q]}$ for each $i \in [n]$ and $g_j = (j + (k-1)p)_{k\in[q]}$ for each $j \in [p]$, the ℓ_1/ℓ_2 multinomial logistic regression can be cast into our framework as:

$$\min_{\beta\in\mathbb{R}^{p'}} \sum_{i=1}^{n} f_i\left(\tilde{x}_i^\top \beta\right) + \lambda \sum_{j=1}^{p} \left\|\beta_{g_j}\right\|_2 , \tag{24}$$

with $f_i : \mathbb{R}^q \to \mathbb{R}$ such that $f_i(z) = -y_{c_i}^\top z + \log\left(\sum_{k=1}^{q} \exp\left(z_k\right)\right)$. Note that generalizing (3) to functions $f_i : \mathbb{R}^q \to \mathbb{R}$ does not bear difficulties, see Ndiaye et al. (2015).

Let us introduce NH, the negative entropy function defined by

$$\mathrm{NH}(x) = \begin{cases} \sum_{i=1}^{q} x_i \log(x_i), & \text{if } x \in \Sigma_q = \{x \in \mathbb{R}_+^q : \sum_{i=1}^{q} x_i = 1\}, \\ +\infty, & \text{otherwise.} \end{cases} \tag{25}$$

We use the convention $0\log(0) = 0$, and one can check that $f_i^*(z) = \mathrm{NH}(z + Y_i)$ and $\gamma = 1$.

For multinomial logistic regression, D_λ implicitly encodes the additional constraint $\theta \in \operatorname{dom} D_\lambda = \{\theta' \in \mathbb{R}^n \ : \ \forall i \in [n], -\lambda\theta'_{c_i} + y_{c_i} \in \Sigma_q\}$ where Σ_q is the q dimensional simplex, see Equation (25). By the dual scaling Equation (9), we have:

$$\theta = \Theta\left(\frac{-G(\tilde{X}\beta)}{\lambda}\right) = \frac{R}{\max(\lambda, \Omega^D(X^\top R))}, \text{ with } R = y - \mathrm{RowNorm}(\exp(\tilde{X}\beta))$$

where the function RowNorm consists in normalizing a (non-negative) matrix row-wise, such that each row sums to one. Thus for any $i \in [n]$ and $\alpha := \lambda/\max(\lambda, \Omega^D(X^\top R)) \in [0, 1]$,

$$-\lambda\theta_{c_i} + y_{c_i} = (1-\alpha)y_{c_i} + \alpha\,\mathrm{RowNorm}(\exp(\tilde{x}_i^\top\beta))$$

which is a convex combination of elements in Σ_q. Hence the dual scaling (9) preserves this additional constraint.

Remark 22 *The intercept has been neglected in our models for simplicity. The Gap Safe framework can also handle such a feature to the cost of more technical details (by adapting the results from Koh et al. (2007) for instance). However, in practice, the intercept can be handled in the present formulation by adding a constant column to the design matrix X. The intercept is then regularized. However, if the constant is set high enough, regularization is small and experiments show that it has little to no impact for high-dimensional problems. This is the strategy used in the* Liblinear *package by Fan et al. (2008). Another alternative could be to handle the constant term as is performed by El Ghaoui et al. (2012).*

6. Experiments

In this section we present results obtained with the Gap Safe rules on various data sets. Implementation[9] has been done in Python and Cython (Behnel et al., 2011) for low level critical parts. A coordinate descent algorithm is used with a scaled dual gap stopping criterion *i.e.,* we normalize the targeted accuracy ϵ (in the stopping criterion) in order to have a running time that is independent from the data scaling, *i.e.,* $\epsilon \leftarrow \epsilon\|y\|_2^2$ for the regression cases and $\epsilon \leftarrow \epsilon\min(n_1, n_2)/n$ where n_i is the number of observations in the class i, for the logistic cases.

Note that in the Lasso case, to compare our method with the un-safe *strong rules* by Tibshirani et al. (2012) and with the sequential screening rule such as the *eddp+* by Wang et al. (2012), we have added an approximated KKT post-processing step. We do this following Footnote 8, since they require the previous (exact) dual optimal solution which is not available in practice (Ndiaye et al., 2016b, Appendix B). The same limit holds true for the *TLFre* approach of Wang and Ye (2014) addressing the Sparse-Group Lasso formulation, as well as for the method explored by Lee and Xing (2014) to handle overlapping groups and *slores* by Wang et al. (2014) for the binary logistic regression.

We have compared our method to various known safe screening rules (El Ghaoui et al., 2012; Xiang et al., 2011; Bonnefoy et al., 2014). For the Sparse-Group Lasso, such rules did not exist, so we have proposed natural extensions (Ndiaye et al., 2016b) thanks to exact computation of the dual norm in Proposition 17. For the Lasso estimator, we have also compared our implementation with the *Blitz* algorithm (Johnson and Guestrin, 2015) which combines Gap Safe screening rules, *Prox-Newton* coordinate descent and an active set strategy.

6.1 ℓ_1 Lasso Regression

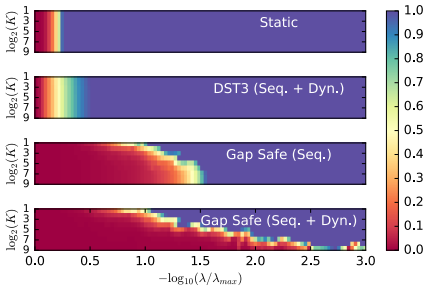
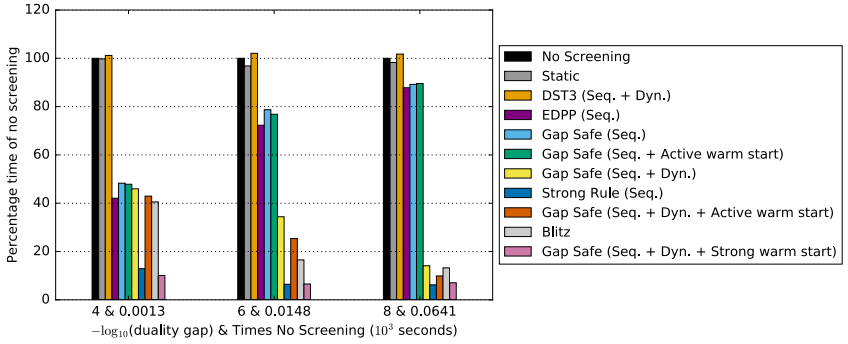
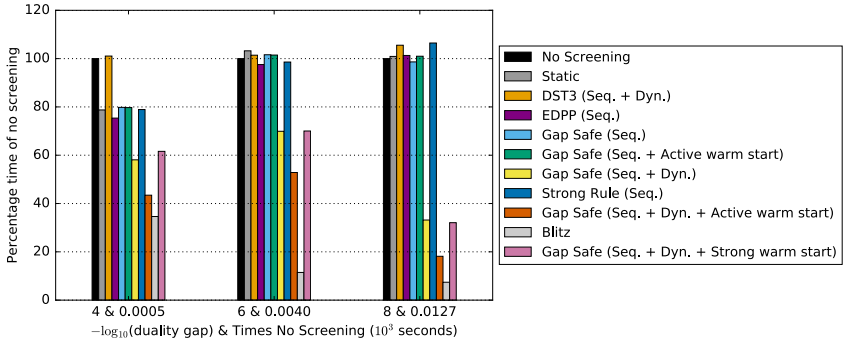


Figure 4: Lasso on the Leukemia (dense data with $n = 72$ observations and $p = 7129$ features). fraction of the variables that are active. Each line corresponds to a fixed number of iterations for which the algorithm is run.

9. The source code can be found in `https://github.com/EugeneNdiaye`.



(a) Dense grid with 100 values of λ.



(b) Coarse grid with 10 values of λ.

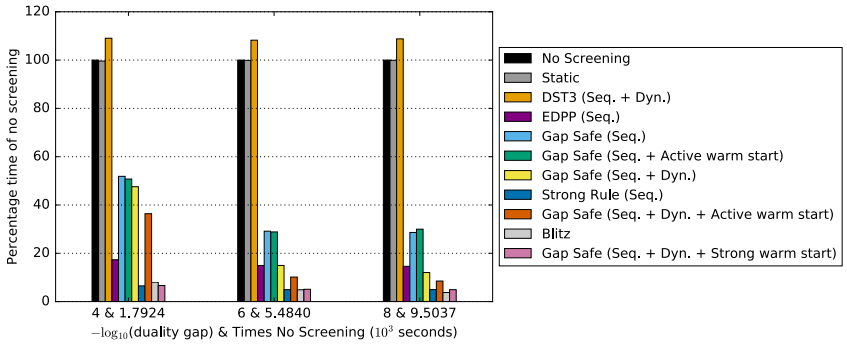
Figure 5: Lasso on the Leukemia (dense data with $n = 72$ observations and $p = 7129$ features). Computation times needed to solve the Lasso regression path to desired accuracy for a grid of λ from $\lambda_{\max}$ to $\lambda_{\max}/10^3$.

We have evaluated the computing time for the Gap Safe rules with and without active warm start, and compared with the static rule El Ghaoui et al. (2012) and the refined dynamic rule DST3 by Xiang et al. (2011), as well as Bonnefoy et al. (2015). We used the classic dense data set Leukemia, and the large sparse financial data set E2006-log1p available from LIBSVM[10]. We have normalized the column of X and standardized y to have zero mean and unit variance.

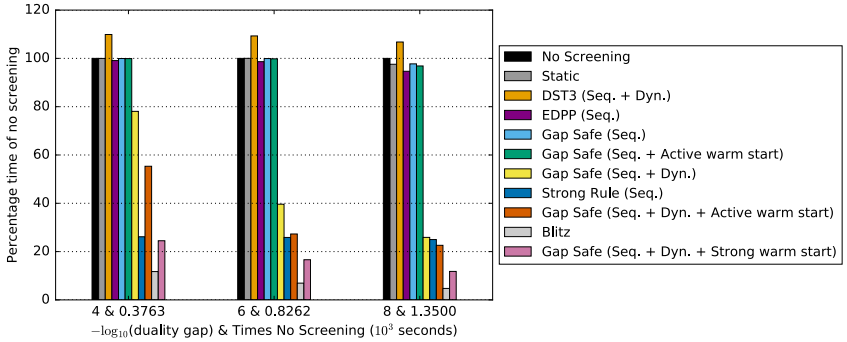
The experiments on Figure 4 focuses on the Leukemia data set. The screening performance for a fixed number of iterations, from 2 to 2^9, is investigated for each λ. It

10. `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

demonstrates that increasing the number of iterations benefits to the dynamic screening rule. Also, the closer the estimate is from the global minimum, the better the screening. This is inline with the results in running time in the benchmark on Figure 5(a). Note that the dynamic Gap Safe rule is the only rule that significantly improves the running time of the Lasso.



(a) Dense grid with 100 values of λ.



(b) Sparse grid with 10 values of λ.

Figure 6: Lasso on financial data E2006-log1p (sparse data with $n = 16087$ observations and $p = 1668737$ features). Computation times needed to solve the Lasso regression path to desired accuracy for a grid of λ from $\lambda_{\max}$ to $\lambda_{\max}/20$.

Results presented in the financial data set in Figure 6 are inline with the results on Leukemia. We observe that the *Blitz* algorithm (Johnson and Guestrin, 2015), also achieves a significant speed-up with gains in the same order of magnitude than our dynamic Gap Safe

implementation combined with active or strong warm start. One advantage of our approach though, is the simplicity to insert it in any iterative algorithm as shown in Algorithm 1 and 2.

To demonstrate the limitations of the strong rules, we report in Figure 5(b) results with a coarse grid with only 10 values of λ from $\lambda_{\max}$ to $\lambda_{\max}/10^3$ such that $2\lambda_t < \lambda_{t-1}$. The strong rules become then useless since the screening test (14) selects all variables, *i.e.*, $\mathcal{STG}(\hat{\theta}^{(\lambda_{t-1})}, \lambda_t, \lambda_{t-1}) = \mathcal{G}$. Overall, the greater the gap between grid points, the lower the benefits of (active) warm start.

In the experiment in Figure 6(b), we have stopped the grid at $\lambda_{\max}/20$ leading to a sparse solution with 1562 active variables. We obtain an important speed-up for both coarse and dense grids demonstrating the consistent efficiency of the active warm start strategy specially in a sparse regime.

Finally, with an extremely coarse grid, we therefore recommend the active warm start with the previous safe active set (which performance is only affected through the initialization point) rather than the strong active set (*cf.* Figure 5(b)).

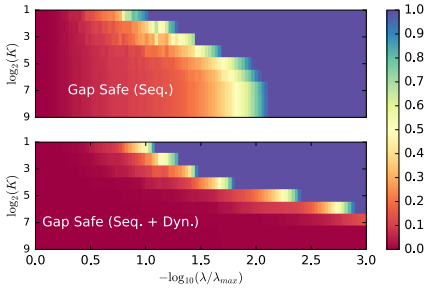
6.2 ℓ_1 Binary Logistic Regression

Results on the Leukemia data set for standard logistic regression are reported in Figure 7. We compare the dynamic strategy of Gap Safe to the sequential strategy. Results demonstrate the clear benefit of the dynamic rule in terms of high number of screened out variables. This is reflected in the graph of running times, which shows that dynamic Gap Safe rule with strong warm start can yield up to a 30× speed-up compared to sequential rule and even more compared to an absence of screening (up to 50× speed-up).

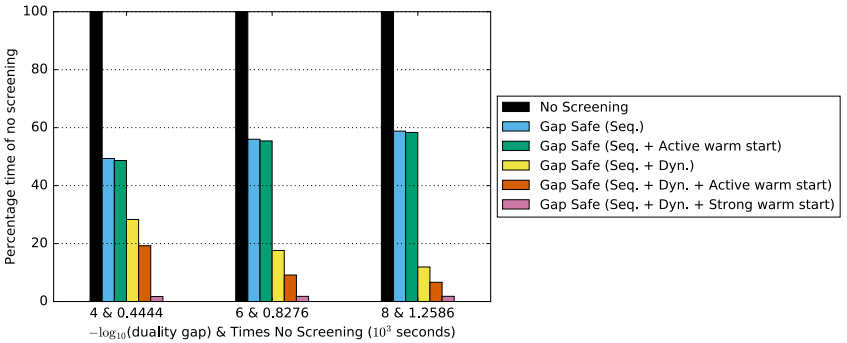
6.3 ℓ_1/ℓ_2 Multi-task Regression

To demonstrate the benefit of the Gap Safe screening rules for a multi-task Lasso problem we have considered neuroimaging data. Electroencephalography (EEG) and magnetoencephalography (MEG) are brain imaging modalities that allow to identify active brain regions. The problem to solve is a multi-task regression problem with squared loss where every task corresponds to a time instant. Using a multi-task Lasso one can constrain the recovered sources to be identical during a short time interval (Gramfort et al., 2012). This corresponds to a temporal stationary assumption. In this experiment we used a joint MEG/EEG data with 301 MEG and 59 EEG sensors leading to $n = 360$. The number of possible sources is $p = 22,494$ and the number of time instants is $q = 20$. With a 1 kHz sampling rate it is equivalent to say that the sources stay the same for 20 ms.

Results are presented in Figure 8. The Gap Safe rule is compared with the dynamic safe rule from Bonnefoy et al. (2015). Figure 8(a) shows the fraction of active variables. It demonstrates that the Gap Safe rule screens out much more variables than the compared methods. Thanks to the converging nature of our rule, the more iterations are performed the more variables are screened out. On Figure 8(b), the computation time confirms the effective speed-up. Our rule significantly improves the computation time for all duality gap tolerance from 10^{-2} to 10^{-8}, especially when accurate estimates are required, *e.g.*, for feature selection.



(a) Fraction of the variables that are active. Each line corresponds to a fixed number of iterations for which the algorithm is run.

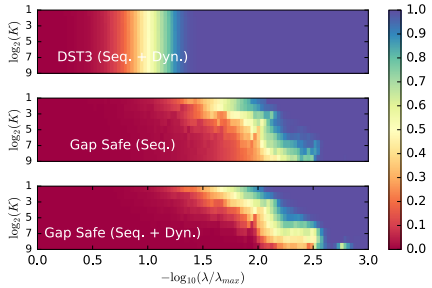


(b) Computation times needed to solve the logistic regression path to desired accuracy with 100 values of λ from $\lambda_{\max}$ to $\lambda_{\max}/10^3$.

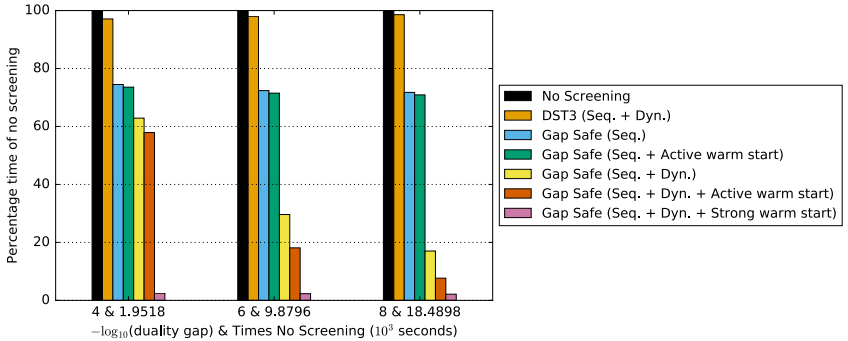
Figure 7: ℓ_1 regularized binary logistic regression on the Leukemia (dense data with $n = 72$ observations and $p = 7129$ features). Sequential and full dynamic screening Gap Safe rules are compared.

6.4 Sparse-Group Lasso Regression

We consider the data set NCEP/NCAR Reanalysis 1 Kalnay et al. (1996) which contains monthly means of climate data measurements spread across the globe in a grid of $2.5^\circ \times 2.5^\circ$ resolutions (longitude and latitude 144×73) from 1948/1/1 to 2015/10/31. Each grid point constitutes a group of 7 predictive variables (*Air Temperature, Precipitable water, Relative humidity, Pressure, Sea Level Pressure, Horizontal Wind Speed* and *Vertical Wind Speed*) whose concatenation across time constitutes our design matrix $X \in \mathbb{R}^{814 \times 73577}$. Such data have therefore a natural group structure, with seven features per group. As target variable $y \in \mathbb{R}^{814}$, we use the values of *Air Temperature* in a neighborhood of Dakar.



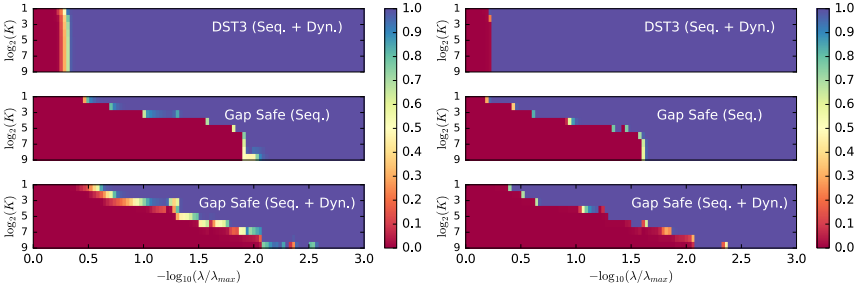
(a) Fraction of active variables as a function of λ and the number of iterations K. The Gap Safe strategy has a much longer range of λ with (red) small active sets.



(b) Computation time to reach convergence using different screening strategies. We have run the algorithm with 100 values of λ from $\lambda_{\max}$ to $\lambda_{\max}/10^3$.

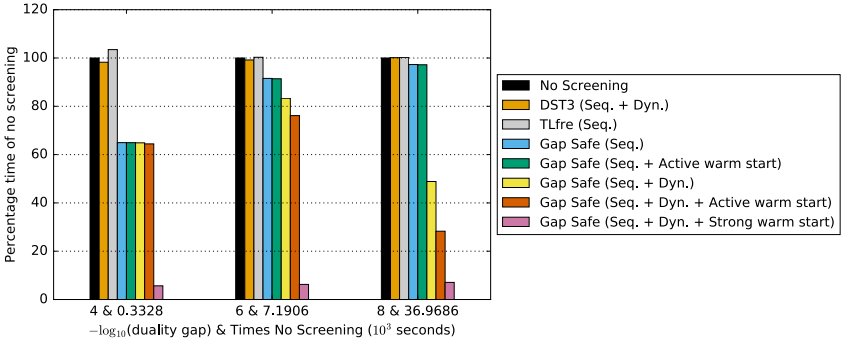
Figure 8: Experiments on MEG/EEG brain imaging data set (dense data with $n = 360$ observations, $p = 22494$ features and $q = 20$ time instants).

For preprocessing, we remove the seasonality (we center the data month by month) and the trend (we remove the linear trend obtained by least squares) present in the data set. We then standardize the data so that each feature has a variance of one. This preprocessing is usually done in climate analysis to prevent some bias in the regression estimates. Similar data have been used in the past by Chatterjee et al. (2012), demonstrating that the Sparse-Group Lasso estimator is well suited for prediction in such climatology applications. Indeed, thanks to the sparsity structure, the estimates delineate via their support some predictive regions at the group level, as well as predictive feature via coordinate-wise screening.



(a) Proportion of active coordinate-wise variables as a function of parameters (λ_t) and the number of iterations K.

(b) Proportion of active group variables as a function of parameters (λ_t) and the number of iterations K.



(c) Time to reach convergence as a function of increasing prescribed accuracy, using various screening strategies and a logarithmic grid from $\lambda_{\max}$ to $\lambda_{\max}/10^{2.5}$.

Figure 9: Sparse-Group Lasso experiments on climate data NCEP/NCAR Reanalysis 1 (dense data with $n = 814$ observations and $p = 73577$ features) with $\tau = 0.4$ chosen by cross-validation.

We choose the parameter τ in the set $\{0, 0.1, \ldots, 0.9, 1\}$ by splitting in half the observations, and run a training-test validation procedure. For each value of τ, we require a duality gap of 10^{-8} on the training set and pick the best one in term of prediction accuracy on the test set. Since the prediction error degrades increasingly for $\lambda \leqslant \lambda_{\max}/10^{-2.5}$, we fix $\delta = 2.5$. We have fixed the weight $w_g = 1$ since all groups have the same size. The computational time benchmark is presented in Figure 9(c). Here also, we observe a significant gain by using a dynamic Gap Safe screening rule, which is further improved by the active warm start.

7. Conclusion

We have proposed a unified presentation of the Gap Safe screening rules for accelerating algorithms solving supervised learning problems under sparsity constraints. The proposed approach applies to many popular estimators that boil down to convex optimization problems where the data fitting term has a Lipschitz gradient and the regularization term is a separable sparsity enforcing function. We have shown that our methodology is more flexible than previously known safe rules as it conveniently unifies both regression and classification settings. The efficiency of the Gap Safe rules along with the new *active /strong warm start* strategies was demonstrated on multiple experiments using real high dimensional data set, suggesting that Gap Safe screening rules are always helpful to speed-up solvers targeting sparse regularization.

Acknowledgments

This work was supported by the ANR THALAMEEG ANR-14-NEUC-0002-01, the NIH R01 MH106174, by the ERC Starting Grant SLAB ERC-YStG-676943, by the Chair Machine Learning for Big Data at Télécom ParisTech and by the Orange/Télécom ParisTech think tank Phi-TAB. We would like to thank the reviewers for their valuable comments which contributed to improve the quality of this paper.

References

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *NIPS*, pages 41–48, 2006.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, New York, 2011.

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31 –39, 2011.

A. Bonnefoy, V. Emiya, L. Ralaivola, and R. Gribonval. A dynamic screening principle for the lasso. In *EUSIPCO*, 2014.

A. Bonnefoy, V. Emiya, L. Ralaivola, and R. Gribonval. Dynamic Screening: Accelerating First-Order Algorithms for the Lasso and Group-Lasso. *IEEE Trans. Signal Process.*, 63 (19):20, 2015.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Springer, Heidelberg, 2011. Methods, theory and applications.

O. Burdakov. A new vector norm for nonlinear curve fitting and some other optimization problems. *33. Int. Wiss. Kolloq. Fortragsreihe "Mathematische Optimierung — Theorie und Anwendungen"*, pages 15–17, 1988.

O. Burdakov and B. Merkulov. On a new norm for data fitting and optimization problems. *Linköping University, Linköping, Sweden, Tech. Rep. LiTH-MAT*, 2001.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, 2011.

S. Chatterjee, K. Steinhaeuser, A. Banerjee, S. Chatterjee, and A. Ganguly. Sparse group lasso: Consistency and climate applications. In *SIAM International Conference on Data Mining*, pages 47–58, 2012.

B. Efron, T. Hastie, I. M. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004. With discussion, and a rejoinder by the authors.

L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination in sparse supervised learning. *J. Pacific Optim.*, 8(4):667–698, 2012.

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.

J. Fan and J. Lv. Sure independence screening for ultrahigh dimensional feature space. *J. Roy. Statist. Soc. Ser. B*, 70(5):849–911, 2008.

O. Fercoq, A. Gramfort, and J. Salmon. Mind the duality gap: safer rules for the lasso. In *ICML*, pages 333–342, 2015.

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

W. J. Fu. Penalized regressions: the bridge versus the lasso. *J. Comput. Graph. Statist.*, 7 (3):397–416, 1998.

A. Gramfort, M. Kowalski, and M. Hämäläinen. Mixed-norm estimates for the M/EEG inverse problem using accelerated gradient methods. *Phys. Med. Biol.*, 57(7):1937–1961, 2012.

A. Gramfort, D. Strohmeier, J. Haueisen, M.S. Hmlinen, and M. Kowalski. Time-frequency mixed-norm estimates: Sparse M/EEG imaging with non-stationary source activations. *NeuroImage*, 70(0):410 – 422, 2013.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. II*, volume 306. Springer-Verlag, Berlin, 1993.

R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *J. Mach. Learn. Res.*, 12:2297–2334, 2011.

T. B. Johnson and C. Guestrin. Blitz: A principled meta-algorithm for scaling sparse optimization. In *ICML*, pages 1171–1179, 2015.

T. B. Johnson and C. Guestrin. Unified methods for exploiting piecewise linear structure in convex optimization. In *NIPS*, pages 4754–4762, 2016.

E. Kalnay, M. Kanamitsu, R. Kistler, W. Collins, D. Deaven, L. Gandin, M. Iredell, S. Saha, G. White, J. Woollen, et al. The NCEP/NCAR 40-year reanalysis project. *Bulletin of the American meteorological Society*, 77(3):437–471, 1996.

K. Koh, S.-J. Kim, and S. Boyd. An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res.*, 8(8):1519–1555, 2007.

M. Kowalski, P. Weiss, A. Gramfort, and S. Anthoine. Accelerating ISTA with an active set strategy. In *OPT 2011: 4th International Workshop on Optimization for Machine Learning*, page 7, 2011.

S. Lee and E. P. Xing. Screening rules for overlapping group lasso. *preprint arXiv:1410.6880v1*, 2014.

S. Lee, J. Zhu, and E. P. Xing. Adaptive multi-task lasso: with application to eqtl detection. In *NIPS*, pages 1306–1314, 2010.

J. Liang, J. Fadili, and G. Peyré. Local linear convergence of forward–backward under partial smoothness. In *NIPS*, pages 1970–1978, 2014.

H. Liu, M. Palatucci, and J. Zhang. Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *ICML*, pages 649–656, 2009.

E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. GAP safe screening rules for sparse multi-task and multi-class models. *NIPS*, pages 811–819, 2015.

E. Ndiaye, O. Fercoq, A. Gramfort, V. Leclere, and J. Salmon. Efficient smoothed concomitant Lasso estimation for high dimensional regression. *Arxiv preprint arXiv:1606.02702*, 2016a.

E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. GAP safe screening rules for Sparse-Group Lasso. *NIPS*, 2016b.

G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

J. Peng, J. Zhu, A. Bergamaschi, W. Han, D.-Y. Noh, J. R. Pollack, and P. Wang. Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *Ann. Appl. Stat.*, 4(1):53–77, 03 2010.

A. Raj, J. Olbrich, B. Gärtner, B. Schölkopf, and M. Jaggi. Screening rules for convex problems. *arXiv preprint arXiv:1609.07478*, 2016.

A. Shibagaki, M. Karasuyama, K. Hatano, and I. Takeuchi. Simultaneous safe screening of features and samples in doubly sparse modeling. In *ICML*, pages 1577–1586, 2016.

N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *J. Comput. Graph. Statist.*, 22(2):231–245, 2013.

P. Sprechmann, I. Ramirez, G. Sapiro, and C. E. Yonina. Collaborative hierarchical sparse modeling. In *Information Sciences and Systems (CISS), 2010 44th Annual Conference on*, pages 1–6. IEEE, 2010.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *J. Roy. Statist. Soc. Ser. B*, 74(2):245–266, 2012.

R. J. Tibshirani. The lasso problem and uniqueness. *Electron. J. Stat.*, 7:1456–1490, 2013.

J. Wang and J. Ye. Two-layer feature reduction for sparse-group lasso via decomposition of convex sets. *arXiv preprint arXiv:1410.4210*, 2014.

J. Wang, P. Wonka, and J. Ye. Lasso screening rules via dual polytope projection. *arXiv preprint arXiv:1211.3966*, 2012.

J. Wang, J. Zhou, J. Liu, P. Wonka, and J. Ye. A safe screening rule for sparse logistic regression. In *NIPS*, pages 1053–1061, 2014.

Z. J. Xiang, H. Xu, and P. J. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *NIPS*, pages 900–908, 2011.

Z. J. Xiang, Y. Wang, and P. J. Ramadge. Screening tests for lasso problems. *arXiv preprint arXiv:1405.4897*, 2014.

Q. Xu, S. J. Pan, H. Xue, and Q. Yang. Multitask learning for protein subcellular location prediction. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 8(3):748–759, 2011.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Roy. Statist. Soc. Ser. B*, 68(1):49–67, 2006.

Y. Zeng and P. Breheny. The biglasso package: A memory- and computation-efficient solver for lasso model fitting with big data in r. *arXiv preprint arXiv:1701.05936*, 2017.

D. Zhang, D. Shen, and Alzheimer's Disease Neuroimaging Initiative. Multi-modal multi-task learning for joint prediction of multiple regression and classification variables in alzheimer's disease. *Neuroimage*, 59(2):895–907, 2012.

J. Zimmert, C. S. de Witt, G. Kerg, and M. Kloft. Safe screening for support vector machines. In *NIPS 2015 Workshop on Optimization in Machine Learning (OPT)*, 2015.

Journal of Machine Learning Research 18 (2017) 1-31 Submitted 10/16; Revised 9/17; Published 11/17

Minimax Filter: Learning to Preserve Privacy from Inference Attacks

Jihun Hamm HAMMJ@CSE.OHIO-STATE.EDU
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH 43210, USA

Editor: Amos Storkey

Abstract

Preserving privacy of continuous and/or high-dimensional data such as images, videos and audios, can be challenging with syntactic anonymization methods which are designed for discrete attributes. Differentially privacy, which uses a more rigorous definition of privacy loss, has shown more success in sanitizing continuous data. However, both syntactic and differential privacy are susceptible to inference attacks, i.e., an adversary can accurately infer sensitive attributes from sanitized data. The paper proposes a novel filter-based mechanism which preserves privacy of continuous and high-dimensional attributes against inference attacks. Finding the optimal utility-privacy tradeoff is formulated as a min-diff-max optimization problem. The paper provides an ERM-like analysis of the generalization error and also a practical algorithm to perform minimax optimization. In addition, the paper proposes a noisy minimax filter which combines minimax filter and differentially-private mechanism. Advantages of the method over purely noisy mechanisms is explained and demonstrated with examples. Experiments with several real-world tasks including facial expression classification, speech emotion classification, and activity classification from motion, show that the minimax filter can simultaneously achieve similar or higher target task accuracy and lower inference accuracy, often significantly lower than previous methods.

Keywords: inference attack, empirical risk minimization, minimax optimization, differential privacy, k-anonymity

1. Introduction

Privacy is an important issue when data collected from or related to individuals are analyzed and released to a third party. In response to growing privacy concerns, various methods for privacy-preserving data publishing have been proposed (see Fung et al. 2010 for a review.) *Syntactic anonymization* methods, such as k-anonymity (Sweeney, 2002) and l-diversity (Machanavajjhala et al., 2007) focus on anonymization of quasi-identifiers and protection of sensitive attributes in static databases. However, it is known that syntactic anonymization is susceptible to several types of attacks such as the DeFinetti attack (Kifer, 2009). An adversary may be able to accurately *infer* sensitive attributes of individuals from insensitive, sanitized attributes. High-dimensional data also poses a challenge for syntactic anonymization methods. For example, k-anonymity is known to be ineffective for high-dimensional sparse databases (Narayanan and Shmatikov, 2008). In addition, syntactic anonymization methods are designed with discrete attributes in mind. However, continuous attributes such as videos, images, or audios cannot be discretized by binning or clustering without loss of information. Besides, conventional categorization of attributes as identifiers, quasi-identifiers, or sensitive information becomes ambiguous with multimedia-type data. For example, an image can be an identifier if it contains the face of the data owner. However, even if the face is blurred, other attributes considered sensitive by the owner such as gender or race can still be recognizable. Furthermore, identifying or sensitive information can be revealed through correlation with other information such as the background or other people in the scene.

Differential privacy (Dwork and Nissim, 2004; Dwork et al., 2006; Dwork, 2006) was proposed to address many weaknesses of syntactic methods (see the discussion by Clifton and Tassa 2013.) Differential privacy has a more formal privacy guarantee than that of syntactic methods, and is applicable to many problems beyond database release (Dwork et al., 2014). In particular, differential privacy can be defined for continuous and/or high-dimensional attributes as well as for functions (Hall et al., 2013). However, similarly to syntactic anonymization, differential privacy is not immune to inference attacks (Cormode, 2011), as differentially privacy only prevents an adversary from gaining *additional* knowledge by inclusion/exclusion of a subject (Dwork et al., 2014), and not from gaining knowledge from released data itself. Therefore, an adversary may still guess sensitive attributes of subjects from differentially-private attributes with confidence.

To preserve privacy of continuous high dimensional data from inference attacks, this paper proposes an approach which differs significantly from previous syntactic or differentially-private approaches. Consider a scenario where a social media user wants to obfuscate all faces in her picture with minimal distortion before posting the picture online. The obfuscation mechanism proposed in the paper is a type of *filtering* of the original features by a non-invertible transformation. How to choose an optimal filter is explained in the following general description. Once the filtered data (e.g., obfuscated pictures) are released, an adversary will try to infer sensitive or identifying attributes from the data in particular using machine learning predictors. Therefore the data owner needs a filter that can minimize the maximum accuracy that any adversary may achieve in predicting the sensitive or identifying attributes. This is an instance of *minimax games* between the data owner and the adversary. The privacy of filtered data is measured by the *expected risk* of adversarial algorithms on specific inference tasks such as identification. However, if privacy is the only goal, near-perfect privacy is achievable with a simple mechanism that sends no or garbage data, which has no utility for any party. To avoid those trivial solutions, (dis)utility of filtered data needs to be considered as the second goal. Disutility can be measured by the amount of distortion of the original data after filtering. Alternatively, if there are particular tasks of interest to be performed on the data by non-adversarial analysts, then again the expected risk of the target tasks can be used as disutility. The two goals—achieving privacy and utility—are often mutually conflicting, and finding an optimal tradeoff between the two is a central question in privacy research (see Related work.) This paper proposes to minimize the difference of two risks by the minimax optimization of (7). The solution to the optimization problem will be referred to as **minimax filter**. In the literature, several methods have been used to solve continuous minimax optimization problems, including the method by Kiwiel (1987) used in Hamm (2015). The paper uses a simpler optimization method based on the classic theorem of Danskin (1967).

©2017 Jihun Hamm.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-501.html.

A notable assumption this paper makes is that the training data for computing an optimal filter are independent of the test data. For example, there are publicly available data sets that can be used to compute minimax filters such as those from the UCI data repository.[1] As the training data set is already public information, the paper considers only the privacy of the subjects who use the filter at test time. A similar assumption was made in Hamm et al. (2016) for knowledge transfer purposes. After the filter is learned from training data, a new test subject can use the filter to obfuscate her data by herself without requiring a third party to collect and process her raw data. Note that this setting is analogous to the setting of local differential privacy (Duchi et al., 2013) where the entity that collects data is not trusted. Since the training procedure can only access empirical risks, the performance of the filter on test data is given in the form of expectation/probability. The paper presents an analysis of generalization error for empirical minimax optimizers in analogy with the analysis of empirical risk minimizers (ERM).

The goal of minimax filter is to prevent inference attacks, and its privacy guarantee is quite different from those of other privacy mechanisms. It is task-dependent and is given in probability or expectation rather than given absolutely, which may be considered weaker than other privacy criteria such as differential privacy. Furthermore, the sanitized data whose sensitive information is filtered out may become unsafe in the future if people's perception of which attribute is sensitive changes over time. Since the goal of minimax filter and the goal of differential privacy are very different, it is natural to ask if the two methods can be combined to take advantages of both methods. Consequently, this paper presents an extension of minimax filter called **noisy minimax filter**, which combines the filter with additive noise mechanism to satisfy the differential privacy criterion. Two methods of combination—preprocessing and postprocessing—are proposed (see Fig. 2.) In the preprocessing approach, minimax filter is applied before perturbation to reduce the sensitivity of transformed data, so that the same level of differential privacy is achieved with less noise. Similar ideas have been utilized before, where data are transformed by Discrete Fourier Transform (Rastogi and Nath, 2010) and by Wavelet Transform (Xiao et al., 2011) before noise is added. In the postprocessing approach, minimax filter is applied after perturbation, and its performance is compared with the preprocessing approach.

Minimax filter and its extensions are evaluated with several real-world tasks including facial expression classification, speech emotion classification, and activity classification from motion. Experiments show that publicly available continuous and high-dimensional data sets are surprisingly susceptible to subject identification attacks, and that minimax filters can reduce the privacy risks to near chance levels without sacrificing utility much. Experiments with noisy minimax filter also yield intuitive results. Differential privacy and resilience to inference attack are indeed different goals, such that using differentially private mechanism alone to achieve the latter requires a large amount of noise that destroys utility of data. In contrast, minimax filters can suppress inference attack with little loss of utility with or without perturbation. Therefore, adding a small amount of noise to the minimax filter can provide a formal differential privacy to a degree and also high on-average task-dependent utility and privacy against inference attacks.

To summarize, the paper has the following contributions.

1. `http://archive.ics.uci.edu/ml/`

- The paper proposes a novel filtering approach which preserves privacy of continuous and high-dimensional attributes against inference attacks. This mechanism is different from previous mechanisms in many ways; in particular, it is a learning-based approach and is task-dependent.
- The paper measures utility and privacy by expected risks, and formulates the utility-privacy tradeoff as a min-diff-max optimization problem. The paper provides an ERM-like analysis of the generalization performance of empirical optimizers.
- The paper presents a practical algorithm which can find minimax filters for a broad family of filters and losses/classifiers. The proposed optimization algorithm and supporting classes can be found on the open-source repository.[2]
- The paper proposes preprocessing and postprocessing approaches to combine minimax filter with noisy mechanisms. The resulting combination can achieve resilience to inference attacks and differential privacy at the same time.
- The paper evaluates proposed algorithms on real-world tasks and compares them with representative algorithms from the literature.

The rest of the paper is organized as follows. Sec. 2 presents related work in the literature. Sec. 3 presents minimax filters and analyzes its generalization performance on test data. Sec. 4 explains the difficulty of solving general minimax problems, and present a simple alternating optimization algorithm. Sec. 5 presents noisy minimax filters and two types of perturbation by additive noise. Sec. 6 evaluates minimax filters with three data sets compared to non-minimax approaches and also evaluates noisy minimax filters under various conditions. Sec. 7 concludes the paper with discussions.

2. Related work

Optimal utility-privacy tradeoff is one of the main goals in privacy research. Utility-privacy tradeoff has particularly been well-studied under differential privacy assumptions (Dwork and Nissim, 2004; Dwork et al., 2006; Dwork, 2006), in the context of the statistical estimation (Smith, 2011; Alvim et al., 2012; Duchi et al., 2013) and learnability (Kasiviswanathan et al., 2011). Other measures of privacy and utility were also proposed. Information-theoretic quantities were proposed by Sankar et al. (2010); Rebollo-Monedero et al. (2010); du Pin Calmon and Fawaz (2012) who analyzed privacy in terms of the rate-distortion theory in communication. One problem with using mutual information or related quantity is that it is difficult to estimate mutual information of high-dimensional and continuous variables in practice without assuming a simple distribution. In contrast, this paper proposes to use classification or regression risks to measure privacy and utility, which is directly computable from data without making assumptions on the distribution. Regarding the use of risks in this paper, classification error-based quantities have been suggested in the literature (Iyengar, 2002; Brickell and Shmatikov, 2008; Li and Li, 2009). However, privacy in those works is measured either by syntactic anonymity or probabilistic divergence which are mainly suitable for discrete attributes. In this paper, privacy and utility are both defined using risks and are therefore directly comparable when defining the tradeoff of the two. Furthermore, the proposed method explicitly preserves privacy against inference attacks, which

2. `https://github.com/jihunhamm/MinimaxFilter`

both syntactic and differentially-private methods are known to be susceptible to (Cormode, 2011).

Most of the aforementioned works focused on the analyses of utility-privacy tradeoff using different measures and assumptions. Few studied efficient algorithms to actively find optimal tradeoff which this paper aims to do. For discrete variables, Krause and Horvitz (2008) showed the NP-hardness of optimal utility-privacy tradeoff in discrete attribute selection, and demonstrated near-optimality of greedy selection. In particular, they used a weighted difference of utility and privacy cost as the joint cost similar to this work. Ghosh et al. (2009) proposed geometric mechanism and linear programming to achieve near-optimal utility for unknown users. Note that the optimization problems with discrete distributions are quite different from the problems involving high-dimensional and/or continuous distributions.

Algorithms for preserving privacy of high-dimensional face images has been proposed previously. Newton et al. (2005) applied k-anonymity to images; Enev et al. (2012) proposed to learn a linear filter using Partial Least Squares to reduce the covariance between filtered data and private labels; Whitehill and Movellan (2012) also proposed a linear filter using the log-ratio of the Fisher's Linear Discriminant Analysis metrics. Xu et al. (2017) presented a related method of preserving privacy of linear predictors using the Augmented Fractional Knapsack algorithm. This paper differs from these in several aspects: it is not limited to linear filters and is applicable to arbitrary differentiable nonlinear filters such as multilayer neural networks; it directly optimizes the utility-privacy risk instead of optimizing heuristic criteria such as covariance differences or LDA log-ratios.

The noisy minimax filter proposed in Sec. 5 bears a resemblance to the work of Rastogi and Nath (2010) and Xiao et al. (2011). Rastogi and Nath (2010) presented a differentially private method of answering queries on time-series data. They used Discrete Fourier Transform to reduce the data dimension and homomorphic encryption to perform distributed noise addition which outperformed the naive noise addition method. Xiao et al. (2011) presented a differentially private range-counting method where they used wavelets to transform the data before adding noise. Effectiveness of the method was analyzed and also demonstrated empirically. The noisy minimax filter presented in this paper, especially the preprocessing approach, is similar in concept to those works in that the combination of data transformation and perturbation is used to enhance utility. However, the transform in this paper (i.e., the minimax filter) is learned from data for specific tasks unlike the Fourier or the Wavelet transform which are data and task independent.

Lastly, the alternating optimization algorithm (Alg. 1) presented in this paper is related to the algorithm proposed by Goodfellow et al. (2014). The algorithm proposed in this paper solves a min-diff-max problem to find an optimal utility-privacy tradeoff, while Goodfellow et al. (2014) solve a minimax problem to learn generative models.

Parts of this paper have appeared in conference proceedings (Hamm, 2015, 2017). New materials in this paper include reformulations of concepts and terms, ERM-like analysis of generalization error, new closed-form examples for minimax optimization, and an alternating optimization algorithm to solve minimax problems.

3. Minimax Filter

In this section, minimax filter is introduced and discussed in detail, and its generalization error is analyzed.

3.1 Formulation

Minimax filter is a non-invertible transformation of raw features/attributes such that the transformed data have optimal utility-privacy tradeoff. Non-invertibility is assumed so that original features are not always recoverable from the filtered data. Let's assume the filter is deterministic; randomize filters will be discussed in Sec. 5. Let $\mathcal{X} \subset \mathbb{R}^D$ be the space of features/attributes as real-valued vectors. Note that discrete attributes can also be represented by real vectors, e.g., by one-hot vector. Let the filter be a map

$$g(x;u) \in G : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^d \tag{1}$$

which is continuous in x and is continuously differentiable w.r.t. the parameter $u \in \mathcal{U}$. Given a filtered output $g(x)$, an adversary can make a prediction $h_p(g(x);v)$ of a private variable y which can be an identifying or sensitive attribute. The prediction function $h_p(g(x);v)$ parameterized by $v \in \mathcal{V}$ is also assumed to be continuous in v and continuously differentiable w.r.t. to the input $g(x)$. The paper proposes to use expected risk to measure the privacy of filtered output against adversarial inference:

$$f_{\text{priv}}(u,v) \triangleq E[l_p(h_p(g(x;u);v),y)], \tag{2}$$

where $l_p(\cdot)$ is a continuously differentiable loss function. From the assumptions above, f_{priv} is continuously differentiable w.r.t. the filter parameter u.

Trivial solutions to maximize privacy already exist, which are the filters that output random or constant numbers independent of actual data. However, such filters have no utility whatsoever for any party. To avoid such trivial solutions, it is necessary to consider the secondary goal of maximizing utility.

Suppose the disutility of filtered data is measured by the distortion of the original data. If $g(x;u)$ is the filter/encoder $\mathcal{X} \to \mathbb{R}^d$, then one can construct the decoder $h(\cdot\,;w) : \mathbb{R}^d \to \mathcal{X}$, such that following reconstruction error

$$f_{\text{util}}(u,w) \triangleq E[\|h_u(g(x;u);w) - x\|^2], \tag{3}$$

is minimized (i.e., $\min_w f_{\text{util}}(u,w)$.)

For another example of utility, let z be a target variable that is of interest to the subjects and analysts such as medical diagnosis of users' data. An analyst can make a prediction $h_u(g(x);w)$ parameterized by $w \in \mathcal{W}$, which is assumed to be continuous in w and continuously differentiable w.r.t. the input $g(x)$. The (dis)utility of the filtered output for a non-adversarial analyst can also be measured by the expected risk

$$f_{\text{util}}(u,w) \triangleq E[l_u(h_u(g(x;u);w),z)], \tag{4}$$

where $l_u(\cdot)$ is a continuously differentiable loss function, such that f_{util} is continuously differentiable w.r.t. the filter parameter u. To facilitate the analysis, the paper assumes

that the constraint sets $\mathcal{W}$, $\mathcal{V}$, and $\mathcal{U}$ are compact and convex subsets of Euclidean spaces such as a ball with a large but finite radius. Along with the assumption that the filter g and the risks f_{priv} and f_{util} are all continuous, min and max values are bounded and attainable. In addition, the solutions to min or max problems are assumed to be in the interior of $\mathcal{W}$, $\mathcal{V}$, and $\mathcal{U}$, enforced by adding appropriate regularization (e.g, $\lambda\|w\|^2$) to the optimization problems if necessary. For this reason, min or max problems that appear in the paper will be treated as unconstrained and the notations $u \in \mathcal{U}$, $v \in \mathcal{V}$, and $w \in \mathcal{W}$ will be omitted.

Having defined the privacy measure and the utility measure, the goal of a filter designer is to find a filter that achieves the following two objectives. The first objective is to *maximize privacy*

$$\max_u \min_v f_{\text{priv}}(u,v) \quad (\text{or equivalently}, \quad \min_u \max_v -f_{\text{priv}}(u,v)) \tag{5}$$

where $\min_v f_{\text{priv}}(u,v)$ represents the risk of the worst (i.e., most capable) adversary: the smaller the risk, the more accurately can she infer the sensitive variable y. As mentioned before, this problem alone has a trivial solution such as a constant filter that outputs zeros. The second objective is to *minimize disutility*

$$\min_u \min_w f_{\text{util}}(u,w) \quad (\text{or equivalently}, \quad \min_u -\max_w -f_{\text{util}}(u,w)) \tag{6}$$

where $\min_w f_{\text{util}}(u,w)$ represents the risk of the best analyst: the smaller the risk, the more accurately can the analyst reconstruct original data x or predict the variable of interest z. To achieve the two opposing goals, we can solve the joint problem of minimizing the weighted sum

$$\min_u \left[\max_v -f_{\text{priv}}(u,v) + \rho \min_w f_{\text{util}}(u,w)\right], \tag{7}$$

or equivalently the weighted difference of max values

$$\min_u \left[\max_v -f_{\text{priv}}(u,v) - \rho \max_w -f_{\text{util}}(u,w)\right]. \tag{8}$$

The constant $\rho > 0$ determines the relative importance of utility versus privacy. For a small $\rho \ll 1$, the problem is close to a trivial privacy-only task, and for a large $\rho \gg 1$, the problem is close to a utility-only task. The solution to (7) or (8) will be referred to as **minimax filter**[3] and is by definition an optimal filter for utility-privacy tradeoff in terms of expected risks given the family of filters $\{g(\cdot\,;u)|u \in \mathcal{U}\}$, the family of private losses/classifiers $\{l_p(h_p(\cdot\,;v))|v \in \mathcal{V}\}$ and the family of utility losses/classifiers $\{l_u(h_u(\cdot\,;w))|w \in \mathcal{W}\}$. Note that the choice of filter and loss/classifier families is very flexible, with the assumption of differentiability only. In practice, almost-everywhere differentiability suffices to use the algorithm in the paper. Fig. 1 shows an example filter/classifier from the class of multilayer neural networks. As an aside, the joint problem may be formulated as minimization of disutility with a hard constraint on privacy risk. When using interior-point methods, the procedure is similar to solving (8) iteratively with an increasing ρ, which is more demanding than minimizing the weighted sum only once as the paper proposes.

3. To be precise, the joint task (7) is a min-diff-max problem and the privacy-only task (5) is a minimax problem. However, both will be referred to as minimax as (7) can be rewritten as a standard minimax problem.

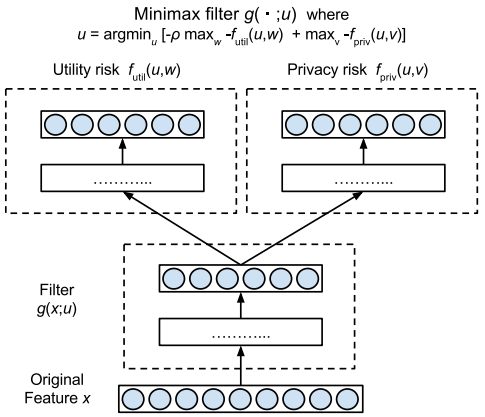


Figure 1: Minimax filter with a filter/classifiers from the class of multilayer neural networks.

3.2 Notes on private and utility tasks

The private variable y can be any attribute which is considered sensitive or identifying. For example, let y be any number or string unique to a person in the data set. Such identifiers are bijective with $\{1,...,S\}$, where S is the total number of subjects, so assume $y \in \{1,...,S\}$. The private task for an adversary is then to predict the subject number y from the filtered data $g(x)$, whose inaccuracy is measured by the expected risk of the identification task. That is, the less accurate the private classifier is, the more anonymous the filtered output is. The identity variable can also be group identifiers, e.g., y is a demographic grouping based on age, sex, ethnicity, etc. Another example of private tasks is to single-out a particular subject among the rest, in which case y is binary: $y = -1$ means 'not the target subject' and $y = 1$ means 'target subject'. To summarize, anonymity of filtered data in this paper means resilience to inference attacks on any variable y that we choose. This unifying approach is convenient since we need not determine whether an attribute is an identifier, a quasi-identifier or a sensitive attribute as in syntactic anonymization. Any information hidden in the continuous high-dimensional features which are relevant to the private variable y—whatever it may be—will be maximally filtered out by construction. Similarly, the target variable z of interest can be any variable that is not the same as the private variable y. In the pathological case where they are the same ($z = y$), the objective (8) becomes

$$\min_u \left[(1-\rho)\max_v -f(u,v)\right] \tag{9}$$

which is either a trivial privacy-only problem when $0 \le \rho < 1$, or a utility-only problem when $\rho > 1$. In general, z and y will be correlated to a certain degree, and the minimax filter will find the best compromise of utility and privacy risks.

Also, private and target tasks need not be classification tasks only. Regression tasks can also be used as a target task, as well as unsupervised tasks that do not require label z. Unsupervised tasks are useful when the target task is unknown or non-specific. For example, (3) measures the expected least-squares error between the original and the reconstructed features.

3.3 Multiple tasks

Extension of the idea in the previous section to multiple private and target tasks is straightforward. Suppose there are N_p private tasks $f^1_{\text{priv}}(u, v_1), ..., f^{N_p}_{\text{priv}}(u, v_{Np})$ associated with private random variables $y^1, ..., y^{Np}$. Note that $f^i_{\text{priv}}(u, v_i) = E[l^i_p(h^i(g(x; u); v_i), y^i)]$. Similarly, suppose there are N_u target tasks $f^1_{\text{util}}(u, w_1), ..., f^{N_u}_{\text{util}}(u, w_{Nu})$ associated with target random variables $z_1, ..., z_{Nu}$. If $\kappa_1, ..., \kappa_{Np}$ are the coefficients representing relative importance of private tasks, and $\rho_1, ..., \rho_{Nu}$ are the coefficients for utility tasks, then the final objective is to solve the following problem

$$\min_u \left[\sum_{i=1}^{N_p} \kappa_i \max_{v_i} -f^i_{\text{priv}}(u, v_i) \; + \; \sum_{i=1}^{N_u} \rho_i \min_{w_i} f^i_{\text{util}}(u, w_i) \right]. \tag{10}$$

Since this can be rewritten as

$$\min_u \left[\max_{v_1,...,v_{Np}} -\sum_{i=1}^{N_p} \kappa_i f^i_{\text{priv}}(u, v_i) \; + \; \min_{w_1,...,w_{Nu}} \sum_{i=1}^{N_u} \rho_i f^i_{\text{util}}(u, w_i) \right], \tag{11}$$

this multiple task problem is nearly identical to the original single task problem (7), with the new utility and privacy tasks defined as

$$\hat{f}_{\text{priv}}(u, v{=}(v_1, ..., v_{Np})) \;\triangleq\; \sum_i \kappa_i f^i_{\text{priv}}(u, v_i), \quad \text{and} \tag{12}$$

$$\hat{f}_{\text{util}}(u, w{=}(w_1, ..., w_{Nu})) \;\triangleq\; \sum_i \rho_i f^i_{\text{util}}(u, w_i), \tag{13}$$

with $\rho = 1$. Using this, it is straightforward to extend the analysis and algorithms developed for single tasks to those for multiple tasks.

3.4 Generalization performance of minimax filter

The proposed privacy mechanism is a learning-based approach. An optimal filter is one that solves the expected risk optimization (7). However, in reality, an optimal filter has to be estimated from finite training samples, and we need a guarantee on the performance of the learned filter on unseen test samples. This section derives generalization bounds for empirical minimax filter similar to the derivation of bounds for empirical risk minimizers (ERM).

The joint problem for expected risks was

$$\min_u \left[\max_v -f_{\text{priv}}(u, v) + \rho \min_w f_{\text{util}}(u, w) \right] = \min_u \left[\max_v -E[l_p(u, v)] + \rho \min_w E[l_u(u, w)] \right]. \tag{14}$$

A joint loss l_J is introduced for convenience:

$$l_J(u, v, w) \triangleq -l_p(u, v) + \rho \, l_u(u, w). \tag{15}$$

Let (u^*, v^*, w^*) be a solution to the expected risk optimization problem:

$$E_D[l_J(u^*, v^*, w^*)] = \min_u \left[\max_v E_D[-l_p(u, v)] + \rho \min_w E_D[l_u(u, w)] \right], \tag{16}$$

where $E_D[\cdot]$ is the expected value w.r.t. the unknown data distribution $P(x, y)$. Similarly, let $(\hat{u}, \hat{v}, \hat{w})$ be a solution to the empirical risk minimax problem:

$$E_S[l_J(\hat{u}, \hat{v}, \hat{w})] = \min_u \left[\max_v E_S[-l_p(u, v)] + \rho \min_w E_S[l_u(u, w)] \right], \tag{17}$$

where the empirical mean $E_S[\cdot]$ for $S = \{(x_1, y_1), \cdots, (x_N, y_N)\}$ is

$$E_S[l(x, y)] \triangleq \frac{1}{N} \sum_{(x,y)\in S} l(x, y). \tag{18}$$

The goal in this analysis is to show that the expected and the empirical optimizers perform equally well in expectation/probability given enough training samples:

$$E_D[l_J(\hat{u}, \hat{v}, \hat{w})] \simeq E_D[l_J(u^*, v^*, w^*)], \;\text{ as }\; N \to \infty. \tag{19}$$

The main result is the Theorem 4 which is proved in the remainder of this section. Let's define optimal parameters $v(u)$ and $w(u)$ given u as

$$v^*(u) \;\triangleq\; \arg\max_v E_D[-l_p(u, v)], \qquad \hat{v}(u) \;\triangleq\; \arg\max_v E_S[-l_p(u, v)], \tag{20}$$

$$w^*(u) \;\triangleq\; \arg\min_w E_D[l_u(u, w)], \qquad \hat{w}(u) \;\triangleq\; \arg\min_w E_S[l_u(u, w)]. \tag{21}$$

One can then write

$$\begin{aligned} E_D[l_J(u^*, v^*, w^*)] &= \min_u \left[E_D[-l_p(u, v^*(u))] + \rho E_D[l_u(u, w^*(u))] \right] && (22) \\ &= \min_u E_D[l_J(u, v^*(u), w^*(u))], && (23) \end{aligned}$$

and similarly

$$E_S[l_J(\hat{u}, \hat{v}, \hat{w})] = \min_u \left[E_S[-l_p(u, \hat{v}(u))] + \rho E_S[l_u(u, \hat{w}(u))] \right] = \min_u E_S[l_J(u, \hat{v}(u), \hat{w}(u))]. \tag{24}$$

From these definitions we have for all u,

$$\begin{aligned} E_D[l_J(u^*, v^*, w^*)] &\le E_D[l_J(u, v^*(u), w^*(u))], && (25) \\ E_S[l_J(\hat{u}, \hat{v}, \hat{w})] &\le E_S[l_J(u, \hat{v}(u), \hat{w}(u))]. && (26) \end{aligned}$$

Also from definition, for all (u, v, w),

$$\begin{aligned} E_D[l_J(u, v, w^*(u))] \;\le\; E_D[l_J(u, v, w)] \;\le\; E_D[l_J(u, v^*(u), w)], && (27) \\ E_S[l_J(u, v, \hat{w}(u))] \;\le\; E_S[l_J(u, v, w)] \;\le\; E_S[l_J(u, \hat{v}(u), w)]. && (28) \end{aligned}$$

These observations imply the following theorem.

Theorem 1 *The risk difference of expected and empirical optimizers is at most twice the largest difference of expected and empirical risks of any set of parameters:*

$$|E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_D[l_J(u^*,v^*,w^*)]| \leq 2 \sup_{u,v,w} |E_D[l_J(u,v,w)] - E_S[l_J(u,v,w)]|\,. \quad (29)$$

Proof The expected risk of empirical risk optimizers $(\hat{u},\hat{v},\hat{w})$ is upper-bounded by the risk of expected risk optimizers (u^*,v^*,w^*) as follows:

$$\begin{aligned}
& E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_D[l_J(u^*,v^*,w^*)] \\
= \;& E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_S[l_J(\hat{u},\hat{v},\hat{w})] - (E_D[l_J(u^*,v^*,w^*)] - E_S[l_J(\hat{u},\hat{v},\hat{w})]) \\
\leq \;& E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_S[l_J(\hat{u},\hat{v},\hat{w})] - (E_D[l_J(u^*,v^*,w^*)] - E_S[l_J(u^*,\hat{v}(u^*),\hat{w}(u^*))]) \quad \text{(from (26))} \\
\leq \;& E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_S[l_J(\hat{u},\hat{v},\hat{w})] - (E_D[l_J(u^*,\hat{v}(u^*),w^*)] - E_S[l_J(u^*,\hat{v}(u^*),\hat{w}(u^*))]) \quad \text{(from (27))} \\
\leq \;& E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_S[l_J(\hat{u},\hat{v},\hat{w})] - (E_D[l_J(u^*,\hat{v}(u^*),w^*)] - E_S[l_J(u^*,\hat{v}(u^*),w^*)]) \quad \text{(from (28))} \\
\leq \;& 2 \sup_{u,v,w} |E_D[l_J(u,v,w)] - E_S[l_J(u,v,w)]|\,.
\end{aligned}$$

The difference can also be lower-bounded as follows:

$$\begin{aligned}
& E_D[l_J(u^*,v^*,w^*)] - E_D[l_J(\hat{u},\hat{v},\hat{w})] \\
= \;& E_D[l_J(u^*,v^*,w^*)] - E_S[l_J(\hat{u},\hat{v},\hat{w})] - (E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_S[l_J(\hat{u},\hat{v},\hat{w})]) \\
\leq \;& E_D[l_J(\hat{u},v^*(\hat{u}),w^*(\hat{u}))] - E_S[l_J(\hat{u},\hat{v},\hat{w})] - (E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_S[l_J(\hat{u},\hat{v},\hat{w})]) \quad \text{(from (25))} \\
\leq \;& E_D[l_J(\hat{u},v^*(\hat{u}),w^*(\hat{u}))] - E_S[l_J(\hat{u},v^*(\hat{u}),\hat{w})] - (E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_S[l_J(\hat{u},\hat{v},\hat{w})]) \quad \text{(from (28))} \\
\leq \;& E_D[l_J(\hat{u},v^*(\hat{u}),\hat{w})] - E_S[l_J(\hat{u},v^*(\hat{u}),\hat{w})] - (E_D[l_J(\hat{u},\hat{v},\hat{w})] - E_S[l_J(\hat{u},\hat{v},\hat{w})]) \quad \text{(from (27))} \\
\leq \;& 2 \sup_{u,v,w} |E_D[l_J(u,v,w)] - E_S[l_J(u,v,w)]|\,.
\end{aligned}$$

■

To bound the RHS of (29), one can use the Rademacher complexity theory (e.g., Lemma 26.2 of Shalev-Shwartz and Ben-David (2014).)

Lemma 2 *Let F be a class of real-valued functions, and let S be a set of N samples $S = \{(x_1,y_1),...,(x_N,y_N)\}$. Then,*

$$E_{S\sim D^N}\left[\sup_{f\in F} |E_D[f] - E_S[f]|\right] \leq 2E_{S\sim D^N}[\mathfrak{R}(F\circ S)], \quad (30)$$

where $\mathfrak{R}(F\circ S)$ is the empirical Rademacher complexity

$$\mathfrak{R}(f\circ S) \triangleq \frac{1}{N} E_{\sigma\sim\{-1,+1\}^N}\left[\sup_{f\in F}\sum_{i=1}^N \sigma_i f(x_i,y_i)\right] \quad (31)$$

for the class of real-valued functions $\{(x,y)\mapsto f(x,y) \;:\; \forall f\in F\}$.

Consider the class of real-valued functions defined from the joint loss (15):

$$\begin{aligned}
l_J\circ H_J\circ S &\triangleq \{(x,y,z)\mapsto l_J(x,y,z;u,v,w) \;:\; u\in\mathcal{U}, v\in\mathcal{V}, w\in\mathcal{W}\} \quad (32)\\
&= \{(x,y,z)\mapsto -l_p(h_v(g_u(x)),y) + \rho\, l_u(h_w(g_u(x)),z) \;:\; u\in\mathcal{U}, v\in\mathcal{V}, w\in\mathcal{W}\}.
\end{aligned}$$

Let $\mathfrak{R}(l_J\circ H_J\circ S)$ denote the empirical Rademacher complexity of the joint loss class. Furthermore, the Rademacher complexity of sum of functions can be upper-bounded by the sum of complexities:

Lemma 3 *The empirical Rademacher complexity of the joint privacy-utility loss is upper-bounded as*

$$\mathfrak{R}(l_J\circ H_J\circ S) \leq \mathfrak{R}(l_p\circ H_p\circ G\circ S) + \rho\,\mathfrak{R}(l_u\circ H_u\circ G\circ S), \quad (\rho>0) \quad (33)$$

where

$$\begin{aligned}
l_p\circ H_p\circ G\circ S &\triangleq \{(x,y,z)\mapsto l_p(h_p(g_u(x)),y) \;:\; u\in\mathcal{U}, v\in\mathcal{V}\}, \quad (34)\\
l_u\circ H_u\circ G\circ S &\triangleq \{(x,y,z)\mapsto l_u(h_u(g_u(x)),z) \;:\; u\in\mathcal{U}, w\in\mathcal{W}\}. \quad (35)
\end{aligned}$$

Proof

$$\begin{aligned}
\mathfrak{R}(l_J\circ H_J\circ S) &= \frac{1}{N}E_\sigma\left[\sup_{u,v,w}\left|\sum_{i=1}^N \sigma_i l_J(x_i,y_i,z_i;u,v,w)\right|\right] \quad (36)\\
&= \frac{1}{N}E_\sigma\left[\sup_{u,v,w}\left|\sum_{i=1}^N \sigma_i(-l_p(x_i,y_i;u,v) + \rho\, l_u(x_i,z_i;u,w))\right|\right] \quad (37)\\
&\leq \frac{1}{N}E_\sigma\left[\sup_{u,v}\left|\sum_{i=1}^N \sigma_i l_p(x_i,y_i;u,v)\right| + \rho\sup_{u,w}\left|\sum_{i=1}^N \sigma_i l_u(x_i,z_i;u,w)\right|\right] \quad (38)\\
&= \mathfrak{R}(l_p\circ H_p\circ G\circ S) + \rho\,\mathfrak{R}(l_u\circ H_u\circ G\circ S). \quad (39)
\end{aligned}$$

■

From Theorem 1 and Lemmas 2 and 3, we get the following generalization bounds in terms of the Rademacher complexity.

Theorem 4

$$\begin{aligned}
& E_{S\sim D^m}\left[|E_D[l_J(u^*,v^*,w^*)] - E_D[l_J(\hat{u},\hat{v},\hat{w})]|\right] \\
\leq \;& 4E_{S\sim D^m}\left[\mathfrak{R}(l_p\circ H_p\circ G\circ S) + \rho\,\mathfrak{R}(l_u\circ H_u\circ G\circ S)\right]. \quad (40)
\end{aligned}$$

A probabilistic bound instead of expected value can also be obtained by applying McDiarmid's inequality, which is omitted.

The Rademacher complexity of privacy and utility losses depends on our choice of loss functions, hypothesis classes, and filter classes. For the simple case of linear filters and linear classifiers, one can compute the complexity using the following lemmas (26.9 and 26.10 from Shalev-Shwartz and Ben-David (2014)):

Lemma 5 *Suppose $\phi:\mathbb{R}\to\mathbb{R}$ is α-Lipschitz, i.e., $|\phi(a)-\phi(b)|\leq\alpha|a-b|$, $\forall a,b\in\mathbb{R}$. Then,*

$$\mathfrak{R}(\phi\circ F) = \alpha\mathfrak{R}(F). \quad (41)$$

Lemma 6 *For the class of linear classifiers* $H = \{x \mapsto w^T x \ : \ \|w\|_2 \leq 1\}$,

$$\mathfrak{R}(H \circ S) \leq \frac{1}{\sqrt{N}} \sup_{x \in S} \|x\|_2. \tag{42}$$

.

From these lemmas and Theorem 4, we have a corollary for a simple case of linear filters and classifiers.

Corollary 7 *Let the loss functions* l_u *and* l_p *be* α*-Lipschitz (e.g.,* $\alpha = 1$ *for logistic regression.) Suppose* U *is a* $d \times D$ *real matrix with a bounded norm* $\|U\|_2 \leq 1$*, and* w *and* v *are vectors with bounded norms (*$\|w\|_2 \leq 1$ *and* $\|v\|_2 \leq 1$*). If the feature domain* $\mathcal{X}$ *is also bounded with a radius* $r = \max_{x \in \mathcal{X}} \|x\|_2$*, then we have*

$$|E_D[l_J(u^*, v^*, w^*)] - E_D[l_J(\hat{u}, \hat{v}, \hat{w})]| \leq \frac{4(1+\rho)\ \alpha\ r}{\sqrt{N}}. \tag{43}$$

Alternatively, one can use the VC dimension to specify the bound.

In any case, the generalization bounds in this section justify the claim that minimax filter can preserve utility-privacy of unseen test data in expectation or probability.

4. Minimax Optimization

This section presents theoretical and numerical solutions of the joint problem (8), which is a variant of unconstrained continuous minimax problems. (See Rustem and Howe (2009) for a review.) The problem (8) can be written in an equivalent form

$$\begin{aligned} \min_u \Phi(u) &= \min_u [\Phi_{\text{priv}}(u) - \rho\ \Phi_{\text{util}}(u)] && (44) \\ &= \min_u [\max_v -f_{\text{priv}}(u, v) - \rho\ \max_w -f_{\text{util}}(u, w)] && (45) \end{aligned}$$

The optimization above is a min-diff-max problem and can be considered as simultaneously solving two subproblems $\min_u[\max_v -f_{\text{priv}}(u,v)]$ and $\min_u[-\max_w -f_{\text{util}}(u,w)]$, but is evidently not the same as summing individual solutions

$$\min_u \Phi(u) \neq \min_u [\max_v -f_{\text{priv}}(u, v)] \ + \ \min_u [-\rho \max_w -f_{\text{util}}(u, w)]. \tag{46}$$

Since the second subproblem $\min_u[-\max_w -f_{\text{util}}(u,w)] = \min_{u,w} f_{\text{util}}(u,w)$ is a standard minimization problem, let's focus only on the first subproblem $\min_u[\max_v -f_{\text{priv}}(u,v)]$ which is a continuous minimax problem. Continuous minimax problems are in general more challenging to solve than standard minimization problems, as the inner optimization $\Phi_{\text{priv}}(u) = \max_v -f_{\text{priv}}$ does not usually have a closed-form solution; when it does, the whole problem can be treated as a standard minimization problem. Furthermore, there can be more than one solution to $\Phi_{\text{priv}}(u) = \max_v -f_{\text{priv}}$. To better understand minimax problems, we look at several examples starting from a simple case where Φ_{priv} and Φ_{util} have closed-form solutions.

4.1 Simple case: eigenvalue problem

Consider finding a minimax filter for the following problem. The filter class is a linear dimensionality reduction ($g(x;u) = U^T x$) parameterized by the matrix $U \in \mathbb{R}^{D \times d}$, and the private and target tasks are least-squares regressions parameterized by the matrices V and W:

$$\begin{aligned} f_{\text{priv}}(U, V) &= \frac{1}{N} \sum_i \|V^T U^T x_i - y_i\|^2, \ \text{and} && (47) \\ f_{\text{util}}(U, W) &= \frac{1}{N} \sum_i \|W^T U^T x_i - z_i\|^2. && (48) \end{aligned}$$

In this case, $\Phi_{\text{priv}}(U) = \max_V -f_p(U,V)$ and $\Phi_{\text{util}}(U) = \max_W -f_u(U,W)$ are both concave problems with closed-form solutions

$$\begin{aligned} \hat{V} &= \arg\min_V f_{\text{priv}} = (U^T C_{xx} U)^{-1} U^T C_{xy} \ \text{and} && (49) \\ \hat{W} &= \arg\min_W f_{\text{util}} = (U^T C_{xx} U)^{-1} U^T C_{xz}, && (50) \end{aligned}$$

where

$$C_{xy} = \frac{1}{N} \sum_i x_i y_i^T, \ \ C_{xz} = \frac{1}{N} \sum_i x_i z_i^T, \ \ \text{and} \ \ C_{xx} = \frac{1}{N} \sum_i x_i x_i^T. \tag{51}$$

The corresponding min values are

$$\begin{aligned} \Phi_{\text{priv}}(U) &= -f_{\text{priv}}(U, \hat{V}) = \text{Tr}\left[(U^T C_{xx} U)^{-1} U^T C_{xy} C_{xy}^T U\right] + \text{const, and} && (52) \\ \Phi_{\text{util}}(U) &= -f_{\text{util}}(U, \hat{W}) = \text{Tr}\left[(U^T C_{xx} U)^{-1} U^T C_{xz} C_{xz}^T U\right] + \text{const.} && (53) \end{aligned}$$

The outer minimization over u is then

$$\begin{aligned} \min_U \Phi(U) &= \min_U \left[\Phi_{\text{priv}}(U) - \rho \Phi_{\text{util}}(U)\right] && (54) \\ &= \min_U \left[-f_{\text{priv}}(U, \hat{V}) + \rho f_{\text{util}}(U, \hat{W})\right] = \min_U \text{Tr}\left[(U^T C_{xx} U)^{-1}\ U^T C_{xyz} U\right] && (55) \end{aligned}$$

where

$$C_{xyz} = C_{xy} C_{xy}^T - \rho C_{xz} C_{xz}^T. \tag{56}$$

The problem (55) can be reformulated as a generalized eigenvalue problem. Let $Q = C_{xx}^{1/2} U$ be a $D \times d$ full-rank matrix. The problem can be rewritten as

$$\min_U \text{Tr}\left[(U^T C_{xx} U)^{-1}\ U^T C_{xyz} U\right] = \min_Q \text{Tr}\left[(Q^T Q)^{-1} Q^T C_{xx}^{-1/2} C_{xyz} C_{xx}^{-1/2} Q\right]. \tag{57}$$

Furthermore, note that min value (55) is invariant to the right multiplication of U by any $d \times d$ nonsingular matrix R. So chose R so that $Q^T Q = R^T U^T C_{xx} U R = I_d$ without loss of generality. Let $A = (C_{xx}^{-1/2})^T C_{xyz} C_{xx}^{-1/2}$, and the minimax problem becomes the following eigenvalue problem:

$$\min_U \Phi(U) = \min_{\{Q \ | \ Q^T Q = I_d\}} \text{Tr}\ Q^T A Q, \tag{58}$$

which is the sum of the d smallest eigenvalues of A which may not be positive semidefinite. Note that this special case problem is quite similar to the objective of Enev et al. (2012):

$$\max_u \ [-\lambda u^T C_{xy}^T C_{xy} u + u^T C_{xz}^T C_{xz} u], \ \text{ s. t. } \ u^T u = 1.$$

The paper also proposes a variant of the eigenvalue problem (58), called **Privacy LDS** which is an analogue of linear discriminant analysis (LDS) for privacy-utility optimization problem. Define the symmetric positive semidefinite matrix C_u as

$$C_u = \sum_{k=1}^{K} N_k (\mu_k - \mu)(\mu_k - \mu)^T, \tag{59}$$

where

$$z \in \{1, ..., K\}, \ \ N_k = \sum_{i=1}^{N} I[z_i = k], \ \ \mu_k = \frac{1}{N_k} \sum_{i=1}^{N} x_i I[z_i = k], \ \text{ and } \ \mu = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{60}$$

Define C_p similarly with

$$y \in \{1, ..., K'\}, \ \ N'_k = \sum_{i=1}^{N'} I[y_i = k], \ \text{ and } \ \mu'_k = \frac{1}{N'_k} \sum_{i=1}^{N'} x_i I[y_i = k]. \tag{61}$$

The proposed Privacy LDS is a linear filter $g(x; U) = U^T x$, where $U = [u_1, ..., u_d]$ is a matrix of top eigenvectors u_i's from the following generalized eigenvalue problem:

$$\max_{\|u\|=1} \ \frac{u^T (C_u + \lambda I) u}{u^T (C_p + \lambda I) u}. \tag{62}$$

This paper uses Privacy LDS as a heuristic to find the initial linear filter before fine-tuning the parameter u using a general optimization method presented in the following sections. Note that this initialization is applicable only to linear filters.

4.2 Saddle-point problem

Continuous minimax problems cannot in general be solved in closed form and require numerical solvers. There is a subclass of continuous minimax problems which are easier to solve than others. Saddle-point problems are minimax problems for which $f(u, v)$ is convex in u and concave in v, such as the following "saddle" problem

$$\min_u \max_v f(u, v) = \min_u \max_v \ [u^2 - v^2]. \tag{63}$$

Analogous to convex problems, $f(u, v)$ has a global optimum (u^*, v^*) which satisfies

$$f(u^*, v) \le f(u^*, v^*) \le f(u, v^*). \tag{64}$$

The convergence rate of a simple subgradient-descent method for saddle-point problems was previously analyzed by Nedić and Ozdaglar (2009). Unfortunately, the minimax problem

$\min_u \max_v -f_{\text{priv}}$ considered in this paper is not a saddle-point problem even for a relatively simple case. Suppose one chooses linear filters, convex differentiable losses (e.g., least-squares, logistic, or exponential losses) and linear classifiers for the problem. Then

$$-f_{\text{priv}}(u, v) = -E[l(g(u); v)] = -E[l(yv^T U^T x)] \tag{65}$$

is the negative expected value of the composition of a convex $l(\cdot)$ and a linear $U^T x$, which is concave in U and is also concave in v, which cannot be a saddle-point problem.

4.3 General problem

A general numerical solution to the optimization (44) is described in this section. Let $f(u, v)$ be a real-valued function $f : \mathcal{U} \times \mathcal{V} \to \mathbb{R}$, where $\mathcal{U}$ and $\mathcal{V}$ are compact subsets of the Euclidean space. Suppose f is jointly continuous and has a continuous partial derivative $\nabla_u f$ w.r.t. the first variable u. The maximum over v

$$\Phi(u) = \max_{v \in \mathcal{V}} f(u, v) \tag{66}$$

has a property that $\Phi(u)$ is in general not differentiable in u even if $f(u, v)$ is (Danskin, 1967). Suppose $V(u)$ is the set of maximizers of f given u:

$$V(u) = \{\hat{v} \in \mathcal{V} \mid f(u, \hat{v}) = \max_{v \in \mathcal{V}} f(u, v)\}. \tag{67}$$

Danskin proved that the directional derivative $D_y \Phi(u)$ in any direction $y \in \mathbb{R}^d$ can be written as the maximum directional derivatives of $f(u, v)$ over all $\hat{v} \in V(u)$:

$$D_y \Phi(u) = \max_{\hat{v} \in V(u)} D_y f(u, \hat{v}), \tag{68}$$

where $D_y f(u, v)$ is the directional derivative of f w.r.t. u. Furthermore, in the case where $V(u)$ is a singleton $\{\hat{v}(u)\}$ for each u, we have

$$D_y \Phi(u) = D_y f(u, \hat{v}(u)). \tag{69}$$

There are several classic minimax optimization algorithms using this property. Suppose $f(u, v)$ is also continuously differentiable w.r.t. v, and $\nabla_u f$ is continuously differentiable w.r.t. v. A first-order method for minimax problems was proposed by Panin (1981) and was later refined by Kiwiel (1987). The latter uses a linear approximation of f at a fixed $\bar{u}$ along the direction q

$$f^l(q, v) = f(\bar{u}, v) + \langle \nabla_u f(\bar{u}, v), \ q \rangle, \tag{70}$$

and uses it to compute the approximate max value

$$\Phi^l(q) = \max_v f^l(q, v). \tag{71}$$

Using this approximation, a line search can be performed along the descent direction q that minimizes the max function $\Phi(\bar{u} + \alpha q)$. In particular, with additional assumptions of Lipschitz continuity of $\nabla_u f$ and compactness of $\mathcal{U}$ and $\mathcal{V}$, Kiwiel's algorithm monotonically decreases f for each iteration and converges to a stationary point u^*, i.e., a point u for

which $\max_v \langle \nabla_u f(u^*, v),\ q \rangle \geq 0$ for all directions q. Previously, Hamm (2015) used Kiwiel's algorithm to solve the optimization problems (44). However, one disadvantage of the method was its slow speed in practice, due to the auxiliary routine of finding the descent direction q at each iteration, described in the supplementary material of Hamm (2015).

Instead, this paper proposes a simple alternating algorithm (Alg. 1) for solving min-diff-max problem based directly on Danskin's theorem. The algorithm only assumes $f_{\text{priv}}(u, v)$ and $f_{\text{util}}(u, w)$ to be jointly continuous and have continuous partial derivatives $\nabla_u f_{\text{priv}}$ and $\nabla_u f_{\text{util}}$. Additionally, if $f_{\text{priv}}(u, v)$ and $f_{\text{util}}(u, w)$ are convex in v and w respectively, then the global minima

$$v_t = \arg\min_v f_{\text{priv}}(u_t, v) \ \text{ and } \ w_t = \arg\min_w f_{\text{util}}(u_t, w) \tag{72}$$

can be found easily, either approximately or accurately. Furthermore, if f_{util} and f_{priv} are strongly convex (e.g., due to regularization), the solutions are unique. Consequently, the descent direction q_t in Alg. 1 is truly the (negative) gradient of $\Phi(u)$ (44) as desired:

$$q_t = \nabla_u f_{\text{priv}}(u, v_t) - \rho \nabla_u f_{\text{util}}(u, w_t) = -\nabla_u \Phi_{\text{priv}}(u) + \rho\ \nabla_u \Phi_{\text{util}}(u) = -\nabla_u \Phi(u). \tag{73}$$

Note that it is still a heuristic for non-convex f_{util} and f_{priv} such as when using neural networks for the filter and/or the classifiers. A related heuristic for minimax problems was proposed by Goodfellow et al. (2014) for learning generative models.

Algorithm 1 Alternating algorithm for min-diff-max

Input: data $\{(x_i, y_i, z_i)\}$, filter g, loss l, classifier h, tradeoff coefficient ρ, max iteration T, learning rates (α_t)
Output: optimal filter parameter u
Begin:
Initialize u_1
for $t = 1, ..., T$ **do**
Solve (approximately)

$$v_t = \arg\min_v f_{\text{priv}}(u_t, v) \ \text{ and } \ w_t = \arg\min_w f_{\text{util}}(u_t, w), \quad \text{where} \tag{74}$$

$$f_{\text{priv}}(u, v) = \frac{1}{N}\sum_{i=1}^{N} l_p(h_p(g(x_i; u); v), y_i) \ \text{ and } \ f_{\text{util}}(u, w) = \frac{1}{N}\sum_{i=1}^{N} l_u(h_u(g(x_i; u); w), z_i). \tag{75}$$

Compute the descent direction by

$$q_t = \nabla_u f_{\text{priv}}(u, v_t) - \rho \nabla_u f_{\text{util}}(u, w_t) \tag{76}$$

Perform line search along q_t and update $u_{t+1} = u_t + \alpha_t \cdot q_t$
Exit if solution converged
end for

The proposed optimization algorithm and supporting classes are implemented in Python and are available on the open-source repository[4].

5. Noisy Minimax Filter

The privacy guarantee that minimax filter provides is very different from that of differentially-private mechanisms. As the filter is learned from training data, its privacy guarantee for test data is given only in expectation/probability. Besides, it is a deterministic mechanism which cannot provide differential privacy. This section presents the *noisy minimax filter* that combines minimax filter with additive noise mechanism to satisfy the differential privacy criterion. Two methods of combination—preprocessing and postprocessing—are proposed and compared. For completeness, the definition of differential privacy is given briefly.

5.1 Differential privacy

A randomized algorithm that takes data $\mathcal{D}$ as input and outputs $\tilde{f}(\mathcal{D})$ is called ϵ-differentially private if

$$Pr(\tilde{f}(\mathcal{D}) \in \mathcal{S}) \leq e^{\epsilon} Pr(\tilde{f}(\mathcal{D}') \in \mathcal{S}) \tag{77}$$

for all measurable $\mathcal{S} \subset \mathcal{T}$ of the output range and for all data sets $\mathcal{D}$ and $\mathcal{D}'$ differing in a single item, denoted by $\mathcal{D} \sim \mathcal{D}'$. That is, even if an adversary knows the whole data set $\mathcal{D}$ except for a single item, she cannot infer much more about the unknown item from the output of the algorithm. A well-known mechanism for turning a non-private function f into a private function $\tilde{f}$ is the perturbation by additive noise. When an algorithm outputs a real-valued vector $f(\mathcal{D}) \in \mathbb{R}^D$, its global sensitivity (Dwork et al., 2006) is defined as

$$S(f) = \max_{\mathcal{D} \sim \mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\| \tag{78}$$

where $\|\cdot\|$ is a norm such as the Euclidean norm. An important result from Dwork et al. (2006) is that the perturbation by additive noise

$$\tilde{f}(\mathcal{D}) = f(\mathcal{D}) + \xi, \tag{79}$$

where ξ has the Laplace-like probability density whose scale parameter is proportional to $S(f)$

$$P(\xi) \propto e^{-\frac{\epsilon}{S(f)}\|\xi\|}, \tag{80}$$

is ϵ-differentially private.

This paper considers local differential privacy (Duchi et al., 2013) of the filter output $g(x)$, that is, perturbation is applied by each subject before $g(x)$ is released to a third party. Let $X = \{x_1, ..., x_S\}$ be a collection of data from S subjects. Then, $X = \{x_1, ..., x_S\}$ and $X' = \{x'_1, ..., x'_S\}$ are defined as neighbors if $x_i = x'_i$ for all $i = 1, ..., S$ except for some $j \in 1, ..., S$. For this subject, x_j and x'_j can be any two samples from the common feature space $\mathcal{X}$ of all subjects. Consequently, a randomized filter $\tilde{g}(\cdot)$ is ϵ-differentially private if for all $x, x' \in \mathcal{X}$ and all measurable $\mathcal{S} \subset \mathcal{T}$ of the output range,

$$Pr(\tilde{g}(x) \in S) \leq e^{\epsilon} Pr(\tilde{g}(x') \in S). \tag{81}$$

4. `https://github.com/jihunhamm/MinimaxFilter`

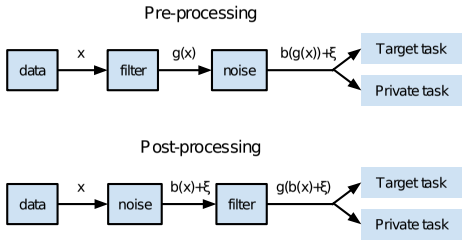


Figure 2: Preprocessing and postprocessing approaches to differentially private minimax filtering.

To use additive noise mechanism (79), the sensitivity (78) of the output $g(\cdot)$ needs to be determined:

$$S(g) = \sup_{x,x' \in \mathcal{X}} \|g(x) - g(x')\|, \tag{82}$$

which is finite if $\mathcal{X}$ is compact and $g(\cdot)$ is continuous. If $\mathcal{X}$ is unbounded, one can directly bound the diameter of $g(\mathcal{X})$ by *bounding functions*. Examples of the bounding function $b : \mathbb{R}^D \to \mathbb{R}^D$ are

1. Hard-bounding by clipping: $b(h) = \min\{1/a,\ 1/\|h\|\} \cdot h$ for some $a > 0$,
2. Soft-bounding by squashing: $b(h) = \tanh(a\|h\|)$ for some $a > 0$, and
3. Normalization after clipping: $b(h) = h/\|h\|$,

where $h = g(x)$ is the filter output of a sample $x \in \mathcal{X}$. Note that these functions enforce the sensitivity $S(b(g))$ to be at most 2, regardless of $\mathcal{X}$ or $g(\cdot)$. The threshold a can be determined from the training data.

5.2 Preprocessing vs postprocessing

Minimax filters can be made locally differentially private using the additive noise mechanism (79) in the signal chain of filtering. The paper proposes two approaches. In the *preprocessing* approach, filtering is performed first and is followed by perturbation. In the *postprocessing* approach, perturbation is applied first and is followed by filtering. Note that preprocessing and postprocessing approaches are similar to output perturbation and input perturbation in Sarwate and Chaudhuri (2013). Fig. 2 shows the signal chains of the two approaches. In preprocessing, the original feature x is first filtered by $g(x)$, and then made differential private by a bounding function and perturbation $b(g(x))+\xi$. In postprocessing, the original feature x is first made differentially private by a bounding function and perturbation $b(x)+\xi$ followed by filtering $g(b(x)+\xi)$. By adding an appropriate amount of noise, both approaches can be made ϵ-differentially private regardless of data distribution. However, when the noisy mechanism is used in conjunction with a minimax filter which is dependent on data

distribution $P(x, y, z)$, preprocessing and postprocessing approaches have different effects that depend on the distribution.

A scenario when preprocessing is preferable to postprocessing is as follows. For the convenience of explanation, let's assume that subject identification is the private task. Let $y(x)$ be the subject identity label of sample x and let $z(x)$ be the target label of sample x for any target task. Define *between-subject diameter* as the max distance of two samples x, x' from different subjects that have the same target label:

$$S_b \triangleq \max_{x,x' \in \mathcal{X}} \|x - x'\| \quad \text{s.t.} \quad y(x) \neq y(x'),\ z(x) = z(x'). \tag{83}$$

Similarly, define *within-subject diameter* as the max distance of two samples x, x' from the same subject that have different target labels:

$$S_w \triangleq \max_{x,x' \in \mathcal{X}} \|x - x'\| \quad \text{s.t.} \quad y(x) = y(x'),\ z(x) \neq z(x'). \tag{84}$$

Also for the purpose of explanation, assume that the filter g is an orthogonal projection onto a lower-dimensional Euclidean space. For a given data set $\mathcal{X}$, if the between-subject diameter is larger than the within-subject diameter ($S_b > S_w$) in the original feature space (Fig. 3a), then minimax filtering can potentially reduce the diameter $S(g) = \max_{x,x'} \|g(x) - g(x')\|$ significantly. This translates to less amount of noise required to achieve the same ϵ-privacy than the amount of noise required before filtering, as the data diameter has shrunk. This will result in better utility of the preprocessing approach over the postprocessing approach where noise is added before filtering. From the same reasoning, if the opposite is true ($S_w > S_b$) (Fig. 3b), then the diameter $S(g)$ after minimax filtering does not change much, and the preprocessing approach may not offer much benefit over the postprocessing approach. However, there are still other differences between the two approaches. This paper assumes that the training data are public information and their privacy is not the primary concern unlike the privacy of test data. However, if we begin to consider the privacy of training data as well, then one should be aware that the learned filters can leak private information, analogous to how the PCA components can leak information about training data (Chaudhuri et al., 2012), and that the filters also need to be sanitized before release. The postprocessing approach makes the whole process simpler. In this case, after each data owner perturbs the data by herself, any subsequent postprocessing, whether it is the process of applying pretrained filters or the process of training minimax filters, does not worsen differential privacy guarantees (Dwork et al., 2014), and therefore the postprocessing approach is a safer choice when the data owners cannot trust the entity that collects training data.

6. Experiments

In this section, the algorithms proposed in the paper are evaluated using three real-world data sets: face data for gender/expression classification, speech data for emotion classification, and motion data for activity classification. Firstly, minimax filters are compared with non-minimax methods in terms of privacy breach vs utility as measured by accuracy of private and target tasks classifiers on test data. Secondly, noisy minimax filters are tested under various conditions using the same data sets.

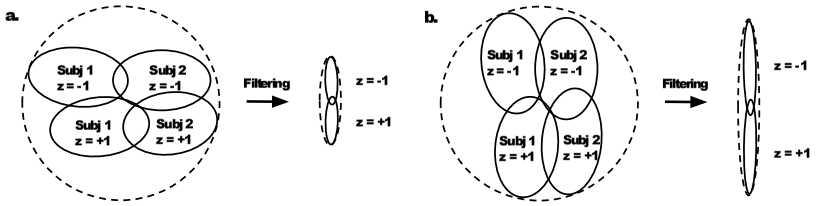


Figure 3: Two example data distributions which have the same data diameter before filtering but have different diameters after filtering. a. an example where between-subject diameter (83) is large. b. an example where within-subject diameter (84) is large.

6.1 Methods

Filters. The following minimax and non-minimax filters are compared.

- Rand: random subspace projection with $g(x;U) = U^T x$, where U is a random full rank $D \times d$ matrix.
- PCA: principal component analysis with $g(x;U) = U^T x$, where U is the eigenvectors corresponding to d largest eigenvalues of $\mathrm{Cov}(x)$.
- PPLS: private partial least squares, using Algorithm 1 from Enev et al. (2012).
- DDD: discriminately decreasing discriminability (DDD) from Whitehill and Movellan (2012) with a mask-type filter from the code[5].
- Minimax 1: linear filter $g(x;U) = U^T x$ where U is computed from Alg. (1).
- Minimax 2: nonlinear filter $g(x)$ from a two-layer sigmoid neural network with of hidden nodes of 20 and 10, computed from Alg. 1.

Remarks. DDD requires analytical solutions to eigenvalue problems which are unavailable for multiclass problems, and is used only in the binary problem with the face database. Also, DDD uses a mask-type filter in the codes, and the dimension d is same as the image size. The dimension d is also irrelevant to nonlinear Minimax filter 2 since it does not use linear dimensionality reduction. The nonlinear filter is pretrained as a stacked denoising autoencoders (Vincent et al., 2008) followed by supervised backpropagation with the target task. **Classifier/loss**. For all experiments, binary or multinomial logistic regression is used a classifier for both utility and privacy risks, where the loss $l(h(g(x;u);v),y)$ is the negative log-likelihood with regularization:

$$l = -v(y)^T g(x;u) + \log(\sum_{k=1}^{K} e^{v(k)^T g(x;u)}) + \frac{\lambda}{2}\sum_{k=1}^{K} \|v(k)\|^2 \tag{85}$$

5. `http://mplab.ucsd.edu/~jake`

where K is the number of classes. The regularization coefficient was $\lambda = 10^{-6}$ and the utility-privacy tradeoff coefficient was $\rho = 10$. The main iteration in Alg. 1 was stopped when the progress was slow, which was between $T = 20 - 200$.

6.2 Data sets

Gender/expression classification from face: The GENKI database (Whitehill and Movellan, 2012) consists of face images with varying poses and facial expressions. The original data set is used unchanged, which has $N = 1740$ training images (50% male and 50% female; 50% smile and 50% non-smile). The test set has 100 images (50 males and 50 females; 50 smiling and 50 non-smiling) not overlapping with the training set. The dimensionality of the original data is $D = 256$, and the filters are tested with $d = 10, 20, 50, 100$. The data set has gender and expression labels but no subject label. Consequently, gender classification is used as the private task and expression classification is used as the target task.

Emotion classification from speech: The ENTERFACE database (Martin et al., 2006) is an audiovisual emotion database of 43 speakers from 14 nations reading predefined English sentences in six induced emotions. From the raw speech signals sampled in 48 KHz, MFCC coefficients are computed using 20 ms windows with 50% overlap and 13 Mel-frequency bands. The mean, max, min, and standard deviation of the MFCC coefficients over the duration of each sentence are computed, resulting in $N = 427$ samples of $D = 52$ dimensional feature vectors from $S = 43$ subjects. Each subject's samples are randomly split to generate training (80%) and test (20%) sets. Average test accuracy over 10 such trials is reported. Filters are tested with $d = 10, 20, 30, 40$. The target task is the binary classification of 'happy' and 'non-happy' emotions from speech, and the privacy task is the multiclass ($S = 43$) subject classification.

Activity classification from motion: The UCI Human Activity Recognition (HAR) data set (Anguita et al., 2012) is a collection of motion sensor data on a smartphone by 30 subjects performing six activities (*walking*, *walking upstairs*, *walking downstairs*, *sitting*, *standing*, *laying*). Various time and frequency domain variables are extracted from the signal, resulting in $N = 10299$ samples of $D = 561$ dimensional features from 30 subjects which are used unchanged. Out of 30 subjects, 15 subjects are chosen randomly. For each domain, each subject's samples are randomly split to generate training (50%) and test (50%) sets. At each trial, the subjects and the training/test sets are randomized, and the average test accuracy over 10 such trials is reported. Filters with dimensions $d = 10, 20, 50, 100$ are used. The target task is the multiclass ($C = 6$) classification of activity, and the privacy task is the multiclass ($S = 15$) subject classification.

6.3 Result 1: Minimax filters

Before any filter is applied, the accuracy of the target tasks with raw data is 0.90 (GENKI), 0.84 (ENTERFACE), and 0.97 (HAR). On the other hand, the accuracy of the private tasks with raw data is 0.90 (GENKI), 0.62 (ENTERFACE), and 0.70 (HAR). The high accuracy of the private tasks (considering the chance level accuracy of 0.5 (GENKI), 0.02 (ENTERFACE), and 0.067 (HAR)) demonstrates that an adversary can accurately infer private variables such as gender and identity from raw data if no filter is used. A simple defense

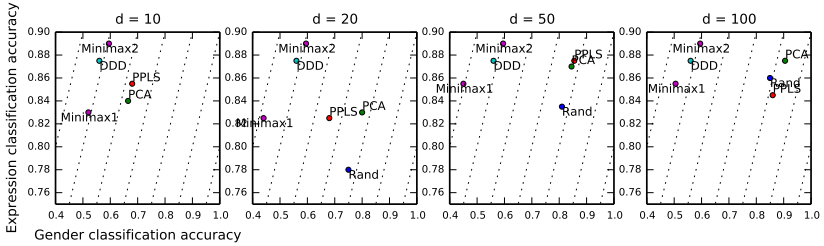


Figure 4: GENKI: Expression classification vs and gender classification from faces.

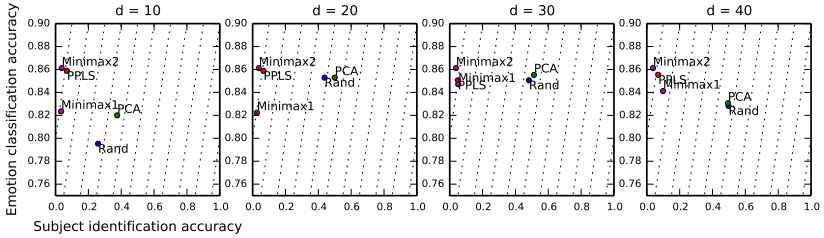


Figure 5: ENTERFACE: Emotion classification vs and subject identification from speech.

against inference attack is to perform dimensionality reduction on the original data, such as Rand and PCA projections. As the dimensionality d decreases from the original value D towards zero, one can expect both the private and the target accuracy to decrease toward the chance level. This trend is indeed the case with both the non-private (Rand,PCA) and the private (PPLS,DDD,Minimax) filters used in the paper. Therefore these filters are evaluated at several different values of the dimensionality d to make fine-grained comparisons of utility-privacy.

Fig. 4 shows the test accuracy with GENKI. The dotted lines are level sets of utility-privacy tradeoff (i.e., target task accuracy - private task accuracy) shown for reference. Minimax 2 achieves the best utility (i.e., most accurate expression classification) and Minimax 1 (linear) achieves the best privacy (i.e., least accurate gender classification). For all dimensions d, Minimax 1 achieves the best utility-privacy compromise (i.e., closest to the top-left corner of the plot), with Minimax 2 and DDD performing similarly. In terms of private task accuracy, Minimax 1 achieves almost the chance level accuracy (0.5), which implies a strong privacy preservation. DDD comes close to Minimax 1, while another private method PPLS is not very successful in preventing the inference of the private variable. As

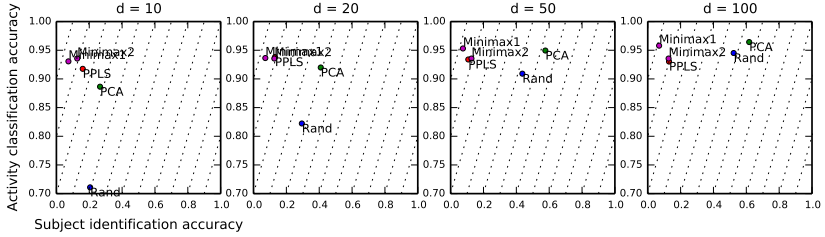


Figure 6: HAR: Activity classification vs subject identification from motion.

expected, non-private methods Rand and PCA also do not reduce the privacy task accuracy. As dimension d increases from 10 to 100, the accuracy of both the target and the private tasks increase (toward the top-right corner of the plot) for PPLS, PCA and Rand, but the value of utility-privacy tradeoff (i.e., target task accuracy - private task accuracy) remains relatively similar even though d changes. Note that d is irrelevant to Minimax 2 and DDD.

Fig. 5 shows the test accuracy of ENTERFACE. Minimax 2 achieves the best utility (i.e., most accurate emotion classification) and the best privacy (i.e., least accurate subject classification) at the same time. PPLS performs well in this task; its private and target task accuracy is close to those of Minimax 2. The private task accuracy of Minimax 2 is near the chance level ($1/S = 0.02$) compared to $0.4 - 0.5$ of non-private methods, suggesting that seemingly harmless statistics (mean, max, min, s.d. of MFCC) are quite susceptible to identification attacks if no privacy mechanism is used. Similar to GENKI, the accuracy of both the target and the private tasks increases with the dimension d for PCA and Rand, and the value of utility-privacy tradeoff remains similar regardless of d.

Fig. 6 shows the test accuracy of HAR. Minimax 1 achieves the best utility (i.e., most accurate activity recognition) and the best privacy (i.e., least accurate subject classification), while Minimax 2 and PPLS performs similarly well. The private task accuracy of Minimax 1 is lower than others close to the chance level ($1/S = 0.067$). The figure also shows that motion data are susceptible ($0.2 - 0.7$) to identification attacks when no privacy mechanism is used. For all dimensions d, Minimax 1 achieves the best compromise of all methods similar to previous experiments. Also the accuracy of both the target and the private tasks roughly increases with d for PCA and Rand, but the value of utility-privacy tradeoff remains similar.

6.4 Result 2: Noisy minimax filters

The same data sets from the previous section are used to demonstrate the effect of noisy mechanism on minimax filters. Four types of noisy filters are compared: PCA-pre, PCA-post, Minimax-pre, and Minimax-post. PCA is chosen as a non-minimax reference filter which preserves the original signal the best in the least mean-squared-error sense. PCA-pre/post means that PCA is applied before/after the perturbation similarly to Minimax-

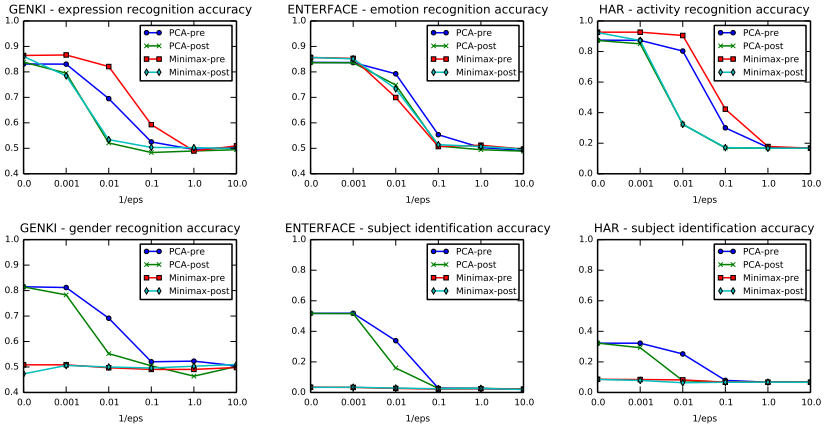


Figure 7: Impact of four noisy filters (PCA-pre/post and Minimax-pre/post) on the accuracy of target and private tasks for three data sets (GENKI, ENTERFACE, HAR), over the range of $\epsilon^{-1} = \{0, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$. Top row is the target task accuracy (higher the better) and bottom row is the private task accuracy (lower the better.) Minimax-pre/post can limit the accuracy of inference attack (bottom row) to almost chance levels regardless of the value of ϵ, while PCA-pre/post requires a significantly high ϵ to prevent inference attacks which also destroy the utility.

pre/post from Fig. 2. For Minimax-pre/post, a linear filter of the same dimension d as PCA-pre/post is used. Tests are performed for the same ranges of dimension d as in Sec 6.3. The results for $d = 20$ with all three data sets are summarized in Fig. 7. Results for different dimensions show similar trends and are summarized in Fig. 8. Optimization of (7) is done similarly to the previous section. All tests are repeated 10 times for different noise samples of (80), for each of 10 random training/test splits.

Fig. 7 shows the following results. Firstly, within each plot, increasing the privacy level from left (ϵ^{-1}=0) to right (ϵ^{-1}=10) lowers the accuracy of both target and private tasks for all filter types and data sets, which is intuitively correct. Secondly, target task accuracy (top row) shows that the four filters are equally accurate with no noise (ϵ^{-1}=0), with Minimax-pre/post slightly more accurate than PCA-pre/post. This observation is consistent with the results in Sec. 6.3. In GENKI and HAR, preprocessing is better than postprocessing for both PCA and Minimax, and Minimax-pre performs the best. In ENTERFACE, preprocessing and postprocessing approaches perform similarly, and all four filters is perform similarly on the target task. This result may be ascribed to the discussion of different data distribution in Sec. 5.2. Thirdly, and most importantly, private task accuracy (bottom row) is quite

different between Minimax-pre/post and non-minimax PCA-pre/post. For both Minimax-pre and Minimax-post, the private task accuracy is almost as low as the chance accuracy of each data set (0.5, 0.03, 0.07) regardless of the noise level ϵ. This demonstrates that minimax filter can prevent inference attacks with little help of noise. In contrast, the non-minimax filters (PCA-pre/post) allow an adversary to infer private variables quite accurately (0.8, 0.5, 0.3) when no noise is used. Preventing such attacks for non-minimax filters requires a significant amount of additive noise (e.g., $\epsilon^{-1} \geq 0.1$) which destroys the utility of data. These results show that differentially privacy is indeed different from privacy against inference attacks and the combination of two methods is beneficial.

7. Conclusion

This work presents a new learning-based mechanism for preventing inference attacks on continuous and high-dimensional data. In this mechanism, a filter transforms continuous and high-dimensional raw features to dimensionality-reduced representations of data. After filtering, information on target tasks remains but information on identifying or sensitive attributes is removed which makes it difficult for an adversary to accurately infer such attributes from the released filtered output. Minimax filters are designed to achieve the optimal utility-privacy tradeoff in terms of expected risks. The paper proves that a filter learned from empirical risks is not far from an ideal filter that is learned from expected risks as the number of samples increases. This property and its dependency on the task make this mechanism quite different from previous mechanisms, including syntactic anonymization and differential privacy. Algorithms for finding minimax filters are presented and evaluated on real-world data sets to show its practical usages. Experiments show that publicly available multisubject data sets are surprisingly susceptible to subject identification attacks, and that even simple linear minimax filters can reduce the privacy risks close to chance level without sacrificing target task accuracy by much.

This work also presents preprocessing and postprocessing approaches to combine minimax privacy and differential privacy. While differential privacy has become a popular criterion of privacy loss, it is not without limitations, in particular against inference attacks as empirically demonstrated in the paper. This leaves room for development of new mechanisms such as the noisy minimax filter presented in the paper, which aims to achieve high on-average utility and protection against inference attacks, and a formal privacy guarantee to a degree. The results from experiments encourage further research on potential benefits of combining different notions and mechanisms of privacy, which is left as future work.

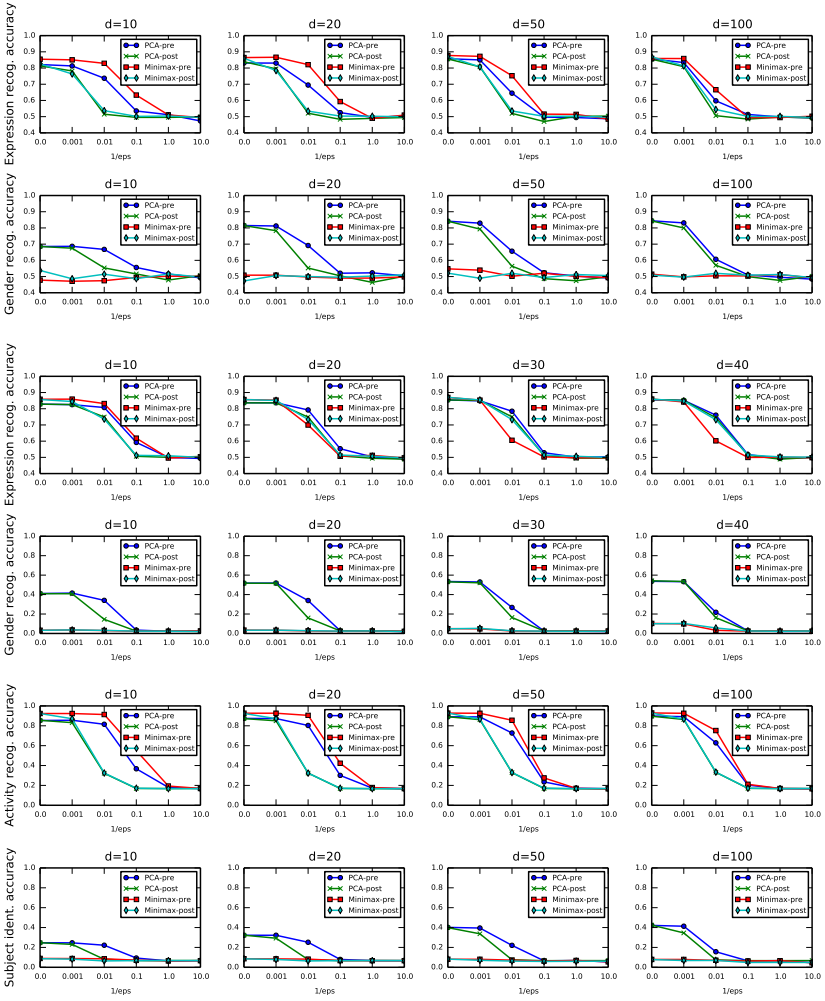


Figure 8: The full result of the performance noisy filters (PCA-pre/post and Minimax-pre/post) on the accuracy of target and private tasks on three data sets: GENKI (1st & 2nd row), ENTERFACE (3rd & 4th row), HAR (5th & 6th row).

Acknowledgments

This research was supported in part by funding from OFRN-C4ISR 2016 and Google Faculty Research Award 2015.

References

Mário S Alvim, Miguel E Andrés, Konstantinos Chatzikokolakis, Pierpaolo Degano, and Catuscia Palamidessi. Differential privacy: on the trade-off between utility and information leakage. In *Formal Aspects of Security and Trust*, pages 39–54. Springer, 2012.

Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L Reyes-Ortiz. Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *Ambient assisted living and home care*, pages 216–223. Springer, 2012.

Justin Brickell and Vitaly Shmatikov. The cost of privacy: destruction of data-mining utility in anonymized data publishing. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 70–78. ACM, 2008.

Kamalika Chaudhuri, Anand Sarwate, and Kaushik Sinha. Near-optimal differentially private principal components. In *Advances in Neural Information Processing Systems*, pages 989–997, 2012.

Chris Clifton and Tamir Tassa. On syntactic anonymity and differential privacy. *Transactions on Data Privacy*, 6(2):161–183, 2013.

Graham Cormode. Personal privacy vs population privacy: learning to attack anonymization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1253–1261. ACM, 2011.

John M Danskin. *The theory of max-min and its application to weapons allocation problems*. Springer, 1967.

Flávio du Pin Calmon and Nadia Fawaz. Privacy against statistical inference. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1401–1408. IEEE, 2012.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 429–438. IEEE, 2013.

Cynthia Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.

Cynthia Dwork and Kobbi Nissim. Privacy-Preserving Data Mining on Vertically Partitioned Databases. In *Proc. CRYPTO*. Springer, 2004.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284. Springer, 2006.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Miro Enev, Jaeyeon Jung, Liefeng Bo, Xiaofeng Ren, and Tadayoshi Kohno. Sensorsift: balancing sensor data privacy and utility in automated face understanding. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 149–158. ACM, 2012.

Benjamin Fung, Ke Wang, Rui Chen, and Philip S Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comp. Surveys (CSUR)*, 42(4):14, 2010.

Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 351–360. ACM, 2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

Rob Hall, Alessandro Rinaldo, and Larry Wasserman. Differential privacy for functions and functional data. *Journal of Machine Learning Research*, 14(Feb):703–727, 2013.

Jihun Hamm. Preserving privacy of continuous high-dimensional data with minimax filters. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Jihun Hamm. Enhancing utility and privacy with noisy minimax filters. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 6389–6393. IEEE, 2017.

Jihun Hamm, Paul Cao, and Mikhail Belkin. Learning privately from multiparty data. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pages 555–563, 2016.

Vijay S Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 279–288. ACM, 2002.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Daniel Kifer. Attacks on privacy and definetti's theorem. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 127–138. ACM, 2009.

Krzysztof C Kiwiel. A direct method of linearization for continuous minimax problems. *Journal of optimization theory and applications*, 55(2):271–287, 1987.

Andreas Krause and Eric Horvitz. A utility-theoretic approach to privacy and personalization. In *AAAI*, volume 8, pages 1181–1188, 2008.

Tiancheng Li and Ninghui Li. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 517–526. ACM, 2009.

Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

Olivier Martin, Irene Kotsia, Benoit Macq, and Ioannis Pitas. The enterface´05 audio-visual emotion database. In *Data Engineering Workshops, 2006. Proceedings. 22nd International Conference on*, pages 8–8. IEEE, 2006.

Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

Elaine M Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. *Knowledge and Data Engineering, IEEE Transactions on*, 17 (2):232–243, 2005.

VM Panin. Linearization method for continuous min-max problem. *Cybernetics and Systems Analysis*, 17(2):239–243, 1981.

Vibhor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 735–746. ACM, 2010.

David Rebollo-Monedero, Jordi Forne, and Josep Domingo-Ferrer. From t-closeness-like privacy to postrandomization via information theory. *Knowledge and Data Engineering, IEEE Transactions on*, 22(11):1623–1636, 2010.

Berc Rustem and Melendres Howe. *Algorithms for worst-case design and applications to risk management*. Princeton University Press, 2009.

Lalitha Sankar, Raj Rajagopalan, and Vincent Poor. An information-theoretic approach to privacy. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1220–1227. IEEE, 2010.

Anand Sarwate and Kamalika Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *Signal Processing Magazine, IEEE*, 30(5):86–94, 2013.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014. ISBN 1107057132, 9781107057135.

Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 813–822. ACM, 2011.

Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of ICML*, pages 1096–1103. ACM, 2008.

Jacob Whitehill and Javier Movellan. Discriminately decreasing discriminability with learned image filters. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2488–2495. IEEE, 2012.

Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering*, 23(8):1200–1214, 2011.

Ke Xu, Tongyi Cao, Swair Shah, Crystal Maung, and Haim Schweitzer. Cleaning the null space: A privacy mechanism for predictors. In *AAAI*, pages 2789–2795, 2017.

Journal of Machine Learning Research 18 (2017) 1-38 Submitted 11/16; Revised 7/17; Published 11/17

Knowledge Graph Completion via Complex Tensor Factorization

Théo Trouillon theo.trouillon@imag.fr
Univ. Grenoble Alpes, 700 avenue Centrale, 38401 Saint Martin d'Hères, France

Christopher R. Dance chris.dance@naverlabs.com
NAVER LABS Europe, 6 chemin de Maupertuis, 38240 Meylan, France

Éric Gaussier eric.gaussier@imag.fr
Univ. Grenoble Alpes, 700 avenue Centrale, 38401 Saint Martin d'Hères, France

Johannes Welbl j.welbl@cs.ucl.ac.uk
Sebastian Riedel s.riedel@cs.ucl.ac.uk
University College London, Gower St, London WC1E 6BT, United Kingdom

Guillaume Bouchard g.bouchard@cs.ucl.ac.uk
Bloomsbury AI, 115 Hampstead Road, London NW1 3EE, United Kingdom
University College London, Gower St, London WC1E 6BT, United Kingdom

Editor: Luc De Raedt

Abstract

In statistical relational learning, knowledge graph completion deals with automatically understanding the structure of large knowledge graphs—labeled directed graphs—and predicting missing relationships—labeled edges. State-of-the-art embedding models propose different trade-offs between modeling expressiveness, and time and space complexity. We reconcile both expressiveness and complexity through the use of complex-valued embeddings and explore the link between such complex-valued embeddings and unitary diagonalization. We corroborate our approach theoretically and show that *all* real square matrices—thus all possible relation/adjacency matrices—are the real part of some unitarily diagonalizable matrix. This results opens the door to a lot of other applications of square matrices factorization. Our approach based on complex embeddings is arguably simple, as it only involves a Hermitian dot product, the complex counterpart of the standard dot product between real vectors, whereas other methods resort to more and more complicated composition functions to increase their expressiveness. The proposed complex embeddings are scalable to large data sets as it remains linear in both space and time, while consistently outperforming alternative approaches on standard link prediction benchmarks.[1]

Keywords: complex embeddings, tensor factorization, knowledge graph, matrix completion, statistical relational learning

1. Introduction

Web-scale knowledge graph provide a structured representation of world knowledge, with projects such as the Google Knowledge Vault (Dong et al., 2014). They enable a wide range of applications including recommender systems, question answering and automated personal agents. The incompleteness of these knowledge graphs—also called knowledge bases—has stimulated research into predicting missing entries, a task known as *link prediction* or *knowledge graph completion*. The need for high quality predictions required by link prediction applications made it progressively become the main problem in statistical relational learning (Getoor and Taskar, 2007), a research field interested in relational data representation and modeling.

Knowledge graphs were born with the advent of the Semantic Web, pushed by the World Wide Web Consortium (W3C) recommendations. Namely, the Resource Description Framework (RDF) standard, that underlies knowledge graphs' data representation, provides for the first time a common framework across all connected information systems to share their data under the same paradigm. Being more expressive than classical relational databases, all existing relational data can be translated into RDF knowledge graphs (Sahoo et al., 2009).

Knowledge graphs express data as a directed graph with labeled edges (relations) between nodes (entities). Natural redundancies between the recorded relations often make it possible to fill in the missing entries of a knowledge graph. As an example, the relation `CountryOfBirth` could not be recorded for all entities, but it can be inferred if the relation `CityOfBirth` is known. The goal of link prediction is the automatic discovery of such regularities. However, many relations are non-deterministic: the combination of the two facts `IsBornIn(John,Athens)` and `IsLocatedIn(Athens,Greece)` does not always imply the fact `HasNationality(John,Greece)`. Hence, it is natural to handle inference probabilistically, and jointly with other facts involving these relations and entities. To this end, an increasingly popular method is to state the knowledge graph completion task as a 3D binary tensor completion problem, where each tensor slice is the adjacency matrix of one relation in the knowledge graph, and compute a decomposition of this partially-observed tensor from which its missing entries can be completed.

Factorization models with low-rank embeddings were popularized by the Netflix challenge (Koren et al., 2009). A partially-observed matrix or tensor is decomposed into a product of embedding matrices with much smaller dimensions, resulting in fixed-dimensional vector representations for each entity and relation in the graph, that allow completion of the missing entries. For a given fact *r(s,o)* in which the subject entity s is linked to the object entity o through the relation r, a score for the fact can be recovered as a multilinear product between the embedding vectors of s, r and o, or through more sophisticated composition functions (Nickel et al., 2016a).

Binary relations in knowledge graphs exhibit various types of patterns: hierarchies and compositions like `FatherOf`, `OlderThan` or `IsPartOf`, with strict/non-strict orders or preorders, and equivalence relations like `IsSimilarTo`. These characteristics maps to different combinations of the following properties: reflexivity/irreflexivity, symmetry/antisymmetry and transitivity. As described in Bordes et al. (2013a), a relational model should (i) be able to learn all combinations of such properties, and (ii) be linear in both time and memory in order to scale to the size of present-day knowledge graphs, and keep up with their growth.

A natural way to handle any possible set of relations is to use the classic canonical polyadic (CP) decomposition (Hitchcock, 1927), which yields two different embeddings for each entity and thus low prediction performances as shown in Section 5. With unique entity embeddings, multilinear products scale well and can naturally handle both symmetry

1. Code is available at: `https://github.com/ttrouill/complex`

©2017 Théo Trouillon, Christopher R. Dance, Johannes Welbl, Sebastian Riedel, Éric Gaussier and Guillaume Bouchard.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v18/16-563.html`

and (ir)-reflexivity of relations, and when combined with an appropriate loss function, dot products can even handle transitivity (Bouchard et al., 2015). However, dealing with antisymmetric—and more generally asymmetric—relations has so far almost always implied superlinear time and space complexity (Nickel et al., 2011; Socher et al., 2013) (see Section 2), making models prone to overfitting and not scalable. Finding the best trade-off between expressiveness, generalization and complexity is the keystone of embedding models.

In this work, we argue that the standard dot product between embeddings can be a very effective composition function, provided that one uses the right *representation*: instead of using embeddings containing real numbers, we discuss and demonstrate the capabilities of complex embeddings. When using complex vectors, that is vectors with entries in $\mathbb{C}$, the dot product is often called the *Hermitian* (or sesquilinear) dot product, as it involves the conjugate-transpose of one of the two vectors. As a consequence, the dot product is not symmetric any more, and facts about one relation can receive different scores depending on the ordering of the entities involved in the fact. In summary, complex embeddings naturally represent arbitrary relations while retaining the efficiency of a dot product, that is linearity in both space and time complexity.

This paper extends a previously published article (Trouillon et al., 2016). This extended version adds proofs of existence of the proposed model in both single and multi-relational settings, as well as proofs of the non-uniqueness of the complex embeddings for a given relation. Bounds on the rank of the proposed decomposition are also demonstrated and discussed. The learning algorithm is provided in more details, and more experiments are provided, especially regarding the training time of the models.

The remainder of the paper is organized as follows. We first provide justification and intuition for using complex embeddings in the square matrix case (Section 2), where there is only a single type of relation between entities, and show the existence of the proposed decomposition for all possible relations. The formulation is then extended to a stacked set of square matrices in a third-order tensor to represent multiple relations (Section 3). The stochastic gradient descent algorithm used to learn the model is detailed in Section 4, where we present an equivalent reformulation of the proposed model that involves only real embeddings. This should help practitioners when implementing our method, without requiring the use of complex numbers in their software implementation. We then describe experiments on large-scale public benchmark knowledge graphs in which we empirically show that this representation leads not only to simpler and faster algorithms, but also gives a systematic accuracy improvement over current state-of-the-art alternatives (Section 5). Related work is discussed in Section 6.

2. Relations as the Real Parts of Low-Rank Normal Matrices

We consider in this section a simplified link prediction task with a single relation, and introduce complex embeddings for low-rank matrix factorization.

We will first discuss the desired properties of embedding models, show how this problem relates to the spectral theorems, and discuss the classes of matrices these theorems encompass in the real and in the complex case. We then propose a new matrix decomposition—to the best of our knowledge—and a proof of its existence for all real square matrices. Finally we discuss the rank of the proposed decomposition.

2.1 Modeling Relations

Let $\mathcal{E}$ be a set of entities, with $|\mathcal{E}| = n$. The truth of the single relation holding between two entities is represented by a sign value $y_{so} \in \{-1, 1\}$, where 1 represents true facts and -1 false facts, $s \in \mathcal{E}$ is the subject entity and $o \in \mathcal{E}$ is the object entity. The probability for the relation holding true is given by

$$P(y_{so} = 1) = \sigma(x_{so}) \tag{1}$$

where $X \in \mathbb{R}^{n \times n}$ is a latent matrix of scores indexed by the subject (rows) and object entities (columns), Y is a partially-observed sign matrix indexed in identical fashion, and σ is a suitable sigmoid function. Throughout this paper we used the logistic inverse link function $\sigma(x) = \frac{1}{1+e^{-x}}$.

2.1.1 Handling Both Asymmetry and Unique Entity Embeddings

In this work we pursue three objectives: finding a generic structure for X that leads to (i) a computationally efficient model, (ii) an expressive enough approximation of common relations in real world knowledge graphs, and (iii) good generalization performances in practice. Standard matrix factorization approximates X by a matrix product $UV^\top$, where U and V are two functionally-independent $n \times K$ matrices, K being the rank of the matrix. Within this formulation it is assumed that entities appearing as subjects are different from entities appearing as objects. In the Netflix challenge (Koren et al., 2009) for example, each row u_i corresponds to the user i and each column v_j corresponds to the movie j. This extensively studied type of model is closely related to the singular value decomposition (SVD) and fits well with the case where the matrix X is rectangular.

However, in many knowledge graph completion problems, the same entity i can appear as both subject *or* object and will have two different embedding vectors, u_i and v_i, depending on whether it appears as subject or object of a relation. It seems natural to learn unique embeddings of entities, as initially proposed by Nickel et al. (2011) and Bordes et al. (2011) and since then used systematically in other prominent approaches (Bordes et al., 2013b; Yang et al., 2015; Socher et al., 2013). In the factorization setting, using the same embeddings for left- and right-side factors boils down to a specific case of eigenvalue decomposition: *orthogonal diagonalization*.

Definition 1 *A real square matrix $X \in \mathbb{R}^{n \times n}$ is orthogonally diagonalizable if it can be written as $X = EWE^\top$, where $E, W \in \mathbb{R}^{n \times n}$, W is diagonal, and E orthogonal so that $EE^\top = E^\top E = I$ where I is the identity matrix.*

The spectral theorem for symmetric matrices tells us that a matrix is orthogonally diagonalizable if and only if it is symmetric (Cauchy, 1829). It is therefore often used to approximate covariance matrices, kernel functions and distance or similarity matrices.

However as previously stated, this paper is explicitly interested in problems where matrices—and thus the relation patterns they represent—can also be antisymmetric, or even not have any particular symmetry pattern at all (asymmetry). In order to both use a unique embedding for entities and extend the expressiveness to asymmetric relations, researchers have generalised the notion of dot products to *scoring functions*, also known as

Model	Scoring Function ϕ	Relation Parameters	$\mathcal{O}_{time}$	$\mathcal{O}_{space}$
CP (Hitchcock, 1927)	$\langle w_r, u_s, v_o \rangle$	$w_r \in \mathbb{R}^K$	$\mathcal{O}(K)$	$\mathcal{O}(K)$
RESCAL (Nickel et al., 2011)	$e_s^T W_r e_o$	$W_r \in \mathbb{R}^{K^2}$	$\mathcal{O}(K^2)$	$\mathcal{O}(K^2)$
TransE (Bordes et al., 2013b)	$-\|\|(e_s + w_r) - e_o\|\|_p$	$w_r \in \mathbb{R}^K$	$\mathcal{O}(K)$	$\mathcal{O}(K)$
NTN (Socher et al., 2013)	$u_r^\top f(e_s W_r^{[1..D]} e_o + V_r \begin{bmatrix} e_s \\ e_o \end{bmatrix} + b_r)$	$W_r \in \mathbb{R}^{K^2 D}, b_r \in \mathbb{R}^K$ $V_r \in \mathbb{R}^{2KD}, u_r \in \mathbb{R}^K$	$\mathcal{O}(K^2 D)$	$\mathcal{O}(K^2 D)$
DistMult (Yang et al., 2015)	$\langle w_r, e_s, e_o \rangle$	$w_r \in \mathbb{R}^K$	$\mathcal{O}(K)$	$\mathcal{O}(K)$
HolE (Nickel et al., 2016b)	$w_r^T(\mathcal{F}^{-1}[\overline{\mathcal{F}[e_s]} \odot \mathcal{F}[e_o]]))$	$w_r \in \mathbb{R}^K$	$\mathcal{O}(K \log K)$	$\mathcal{O}(K)$
ComplEx (this paper)	$\text{Re}(\langle w_r, e_s, \bar{e}_o \rangle)$	$w_r \in \mathbb{C}^K$	$\mathcal{O}(K)$	$\mathcal{O}(K)$

Table 1: Scoring functions of state-of-the-art latent factor models for a given fact $r(s, o)$, along with the representation of their relation parameters, and time and space (memory) complexity. K is the dimensionality of the embeddings. The entity embeddings e_s and e_o of subject s and object o are in $\mathbb{R}^K$ for each model, except for ComplEx, where $e_s, e_o \in \mathbb{C}^K$. $\bar{x}$ is the complex conjugate, and D is an additional latent dimension of the NTN model. $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote respectively the Fourier transform and its inverse, $\odot$ is the element-wise product between two vectors, Re(.) denotes the real part of a complex vector, and $\langle \cdot, \cdot, \cdot \rangle$ denotes the trilinear product.

composition functions, that allow more general combinations of embeddings. We briefly recall several examples of scoring functions in Table 1, as well as the extension proposed in this paper.

These models propose different trade-offs between the three essential points:

- Expressiveness, which is the ability to represent symmetric, antisymmetric and more generally asymmetric relations.
- Scalability, which means keeping linear time and space complexity scoring function.
- Generalization, for which having unique entity embeddings is critical.

RESCAL (Nickel et al., 2011) and NTN (Socher et al., 2013) are very expressive, but their scoring functions have quadratic complexity in the rank of the factorization. More recently the HolE model (Nickel et al., 2016b) proposes a solution that has quasi-linear complexity in time and linear space complexity. DistMult (Yang et al., 2015) can be seen as a joint orthogonal diagonalization with real embeddings, hence handling only symmetric relations. Conversely, TransE (Bordes et al., 2013b) handles symmetric relations to the price of strong constraints on its embeddings. The canonical-polyadic decomposition (CP) (Hitchcock, 1927) generalizes poorly with its different embeddings for entities as subject and as object.

We reconcile expressiveness, scalability and generalization by going back to the realm of well-studied matrix factorizations, and making use of complex linear algebra, a scarcely used tool in the machine learning community.

2.1.2 Decomposition in the Complex Domain

We introduce a new decomposition of real square matrices using unitary diagonalization, the generalization of orthogonal diagonalization to complex matrices. This allows decomposition of *arbitrary* real square matrices with unique representations of rows and columns.

Let us first recall some notions of complex linear algebra as well as specific cases of diagonalization of real square matrices, before building our proposition upon these results.

A complex-valued vector $x \in \mathbb{C}^K$, with $x = \text{Re}(x) + i\text{Im}(x)$ is composed of a real part $\text{Re}(x) \in \mathbb{R}^K$ and an imaginary part $\text{Im}(x) \in \mathbb{R}^K$, where i denotes the square root of -1. The conjugate $\overline{x}$ of a complex vector inverts the sign of its imaginary part: $\overline{x} = \text{Re}(x) - i\text{Im}(x)$.

Conjugation appears in the usual dot product for complex numbers, called the *Hermitian* product, or *sesquilinear* form, which is defined as:

$$\begin{aligned} \langle u, v \rangle &:= \bar{u}^\top v \\ &= \text{Re}(u)^\top \text{Re}(v) + \text{Im}(u)^\top \text{Im}(v) \\ &\quad + i(\text{Re}(u)^\top \text{Im}(v) - \text{Im}(u)^\top \text{Re}(v)) \,. \end{aligned}$$

A simple way to justify the Hermitian product for composing complex vectors is that it provides a valid topological norm in the induced vector space. For example, $\bar{x}^\top x = 0$ implies $x = 0$ while this is not the case for the bilinear form $x^\top x$ as there are many complex vectors x for which $x^\top x = 0$.

This yields an interesting property of the Hermitian product concerning the order of the involved vectors: $\langle u, v \rangle = \overline{\langle v, u \rangle}$, meaning that the real part of the product is symmetric, while the imaginary part is antisymmetric.

For matrices, we shall write $X^* \in \mathbb{C}^{n \times m}$ for the conjugate-transpose $X^* = (\overline{X})^\top = \overline{X^\top}$. The conjugate transpose is also often written $X^\dagger$ or X^{H}.

Definition 2 *A complex square matrix $X \in \mathbb{C}^{n \times n}$ is unitarily diagonalizable if it can be written as $X = EWE^*$, where $E, W \in \mathbb{C}^{n \times n}$, W is diagonal, and E is unitary such that $EE^* = E^*E = I$.*

Definition 3 *A complex square matrix X is normal if it commutes with its conjugate-transpose so that $XX^* = X^*X$.*

We can now state the spectral theorem for normal matrices.

Theorem 1 (Spectral theorem for normal matrices, von Neumann (1929)) *Let X be a complex square matrix. Then X is unitarily diagonalizable if and only if X is normal.*

It is easy to check that all real symmetric matrices are normal, and have pure real eigenvectors and eigenvalues. But the set of purely real normal matrices also includes all real antisymmetric matrices (useful to model hierarchical relations such as `IsOlder`), as well as all real orthogonal matrices (including permutation matrices), and many other matrices that are useful to represent binary relations, such as assignment matrices which represent bipartite graphs. However, far from all matrices expressed as $X = EWE^*$ are purely real, and Equation (1) requires the scores X to be purely real.

As we only focus on *real* square matrices in this work, let us summarize all the cases where X is real square and $X = EWE^*$ if X is unitarily diagonalizable, where $E, W \in \mathbb{C}^{n\times n}$, W is diagonal and E is unitary:

- X is symmetric if and only if X is orthogonally diagonalizable and E and W are purely real.
- X is normal and non-symmetric if and only if X is unitarily diagonalizable and E and W are *not* both purely real.
- X is not normal if and only if X is not unitarily diagonalizable.

We generalize all three cases by showing that, for any $X \in \mathbb{R}^{n\times n}$, there exists a unitary diagonalization in the complex domain, of which the real part equals X:

$$X = \mathrm{Re}(EWE^*)\,. \tag{2}$$

In other words, the unitary diagonalization is projected onto the real subspace.

Theorem 2 *Suppose $X \in \mathbb{R}^{n\times n}$ is a real square matrix. Then there exists a normal matrix $Z \in \mathbb{C}^{n\times n}$ such that $\mathrm{Re}(Z) = X$.*

Proof Let $Z := X + iX^\top$. Then

$$Z^* = X^\top - iX = -i(iX^\top + X) = -iZ\,,$$

so that

$$ZZ^* = Z(-iZ) = (-iZ)Z = Z^*Z\,.$$

Therefore Z is normal. ■

Note that there also exists a normal matrix $Z = X^\top + iX$ such that $\mathrm{Im}(Z) = X$.

Following Theorem 1 and Theorem 2, any real square matrix can be written as the real part of a complex diagonal matrix through a unitary change of basis.

Corollary 1 *Suppose $X \in \mathbb{R}^{n\times n}$ is a real square matrix. Then there exist $E, W \in \mathbb{C}^{n\times n}$, where E is unitary, and W is diagonal, such that $X = \mathrm{Re}(EWE^*)$.*

Proof From Theorem 2, we can write $X = \mathrm{Re}(Z)$, where Z is a normal matrix, and from Theorem 1, Z is unitarily diagonalizable. ■

Applied to the knowledge graph completion setting, the rows of E here are vectorial representations of the entities corresponding to rows and columns of the relation score matrix X. The score for the relation holding true between entities s and o is hence

$$x_{so} = \mathrm{Re}(e_s^\top W \bar{e}_o) \tag{3}$$

where $e_s, e_o \in \mathbb{C}^n$ and $W \in \mathbb{C}^{n\times n}$ is diagonal. For a given entity, its subject embedding vector is the complex conjugate of its object embedding vector.

To illustrate this difference of expressiveness with respect to real-valued embeddings, let us consider two complex embeddings $e_s, e_o \in \mathbb{C}$ of dimension 1, with arbitrary values: $e_s = 1 - 2i$, and $e_o = -3 + i$; as well as their real-valued, twice-bigger counterparts: $e'_s = \binom{1}{-2} \in \mathbb{R}^2$ and $e'_o = \binom{-3}{1} \in \mathbb{R}^2$. In the real-valued case, that corresponds to the DistMult model (Yang et al., 2015), the score is $x_{so} = e'^\top_s W' e'_o$. Figure 1 represents the heatmaps of the scores x_{so} and x_{os}, as a function of $W \in \mathbb{C}$ in the complex-valued case, and as a function of $W' \in \mathbb{R}^2$ diagonal in the real-valued case. In the real-valued case, that is symmetric in the subject and object entities, the scores x_{so} and x_{os} are equal for any value of $W' \in \mathbb{R}^2$ diagonal. Whereas in the complex-valued case, the variation of $W \in \mathbb{C}$ allows to score x_{so} and x_{os} with any desired pair of values.

This decomposition however is non-unique, a simple example of this non-uniqueness is obtained by adding a purely imaginary constant to the eigenvalues. Let $X \in \mathbb{R}^{n\times n}$, and $X = \mathrm{Re}(EWE^*)$ where E is unitary, W is diagonal. Then for any real constant $c \in \mathbb{R}$ we have:

$$\begin{aligned} X &= \mathrm{Re}(E(W + icI)E^*) \\ &= \mathrm{Re}(EWE^* + icEIE^*) \\ &= \mathrm{Re}(EWE^* + icI) \\ &= \mathrm{Re}(EWE^*)\,. \end{aligned}$$

In general, there are many other possible couples of matrices E and W that preserve the real part of the decomposition. In practice however this is no synonym of low generalization abilities, as many effective matrix and tensor decomposition methods used in machine learning lead to non-unique solutions (Paatero and Tapper, 1994; Nickel et al., 2011). In this case also, the learned representations prove useful as shown in the experimental section.

2.2 Low-Rank Decomposition

Addressing knowledge graph completion with data-driven approaches assumes that there is a sufficient regularity in the observed data to generalize to unobserved facts. When formulated as a matrix completion problem, as it is the case in this section, one way of implementing this hypothesis is to make the assumption that the matrix has a low rank or approximately low rank. We first discuss the rank of the proposed decomposition, and then introduce the sign-rank and extend the bound developed on the rank to the sign-rank.

2.2.1 Rank Upper Bound

First, we recall one definition of the rank of a matrix (Horn and Johnson, 2012).

Definition 4 *The rank of an m-by-n complex matrix $\mathrm{rank}(X) = \mathrm{rank}(X^\top) = k$, if X has exactly k linearly independent columns.*

Also note that if X is diagonalizable so that $X = EWE^{-1}$ with $\mathrm{rank}(X) = k$, then W has k non-zero diagonal entries for some diagonal W and some invertible matrix E. From this it is easy to derive a known additive property of the rank:

$$\mathrm{rank}(B + C) \leq \mathrm{rank}(B) + \mathrm{rank}(C) \tag{4}$$

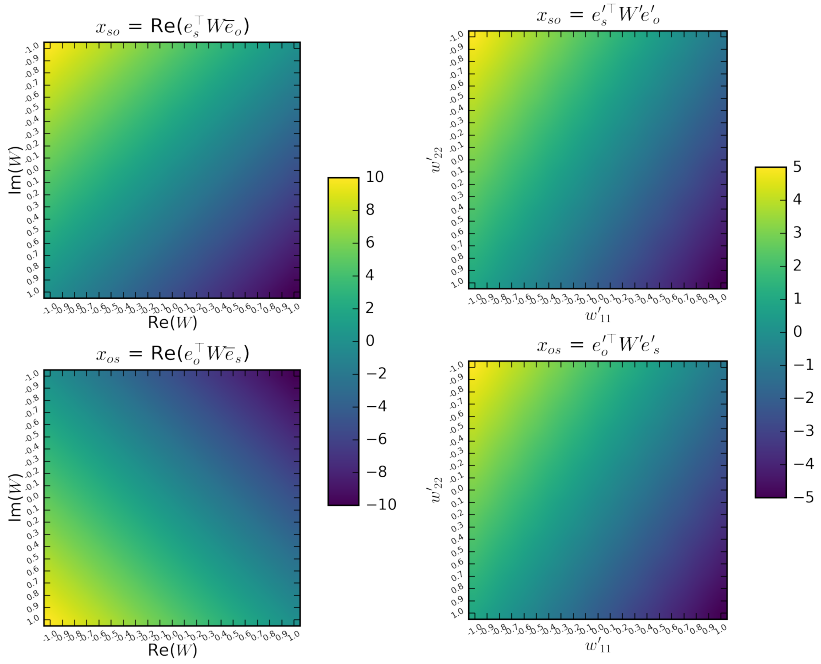


Figure 1: Left: Scores $x_{so} = \mathrm{Re}(e_s^\top W \overline{e}_o)$ (top) and $x_{os} = \mathrm{Re}(e_o^\top W \overline{e}_s)$ (bottom) for the proposed complex-valued decomposition, plotted as a function of $W \in \mathbb{C}$, for fixed entity embeddings $e_s = 1-2i$, and $e_o = -3+i$. Right: Scores $x_{so} = e_s'^\top W' e_o'$ (top) and $x_{os} = e_o'^\top W' e_s'$ (bottom) for the corresponding real-valued decomposition with the same number of free real-valued parameters (*i.e.* in twice the dimension), plotted as a function of $W' \in \mathbb{R}^2$ diagonal, for fixed entity embeddings $e_s' = \binom{1}{-2}$ and $e_o' = \binom{-3}{1}$. By varying $W \in \mathbb{C}$, the proposed complex-valued decomposition can attribute any pair of scores to x_{so} and x_{os}, whereas $x_{so} = x_{os}$ for all $W' \in \mathbb{R}^2$ with the real-valued decomposition.

where $B, C \in \mathbb{C}^{m \times n}$.

We now show that any rank k real square matrix can be reconstructed from a $2k$-dimensional unitary diagonalization.

Corollary 2 *Suppose $X \in \mathbb{R}^{n \times n}$ and rank$(X) = k$. Then there exist $E \in \mathbb{C}^{n \times 2k}$ such that the columns of E form an orthonormal basis of $\mathbb{C}^{2k}$, $W \in \mathbb{C}^{2k \times 2k}$ is diagonal, and $X = \mathrm{Re}(EWE^*)$.*

Proof Consider the complex square matrix $Z := X + iX^\top$. We have $\mathrm{rank}(iX^\top) = \mathrm{rank}(X^\top) = \mathrm{rank}(X) = k$.

From Equation (4), $\mathrm{rank}(Z) \leq \mathrm{rank}(X) + \mathrm{rank}(iX^\top) = 2k$.

The proof of Theorem 2 shows that Z is normal. Thus $Z = EWE^*$ with $E \in \mathbb{C}^{n \times 2k}$, $W \in \mathbb{C}^{2k \times 2k}$ where the columns of E form an orthonormal basis of $\mathbb{C}^{2k}$, and W is diagonal. ■

Since E is not necessarily square, we replace the unitary requirement of Corollary 1 by the requirement that its columns form an orthonormal basis of its smallest dimension, $2k$.

Also, given that such decomposition always exists in dimension n (Theorem 2), this upper bound is not relevant when $\mathrm{rank}(X) \geq \frac{n}{2}$.

2.2.2 Sign-Rank Upper Bound

Since we encode the truth values of each fact with ± 1, we deal with square *sign matrices*: $Y \in \{-1, 1\}^{n \times n}$. Sign matrices have an alternative rank definition, the *sign-rank*.

Definition 5 *The sign-rank* $\mathrm{rank}_\pm(Y)$ *of an m-by-n sign matrix Y, is the rank of the m-by-n real matrix of least rank that has the same sign-pattern as Y, so that*

$$\mathrm{rank}_\pm(Y) := \min_{X \in \mathbb{R}^{m \times n}} \{\mathrm{rank}(X) \,|\, \mathrm{sign}(X) = Y\},$$

where $\mathrm{sign}(X)_{ij} = \mathrm{sign}(x_{ij})$.

We define the *sign function* of $c \in \mathbb{R}$ as

$$\mathrm{sign}(c) = \begin{cases} 1 & \text{if } c \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

where the value $c = 0$ is here arbitrarily assigned to 1 to allow zero entries in X, conversely to the stricter usual definition of the sign-rank.

To make generalization possible, we hypothesize that the true matrix Y has a low sign-rank, and thus can be reconstructed by the sign of a low-rank score matrix X. The low sign-rank assumption is theoretically justified by the fact that the sign-rank is a natural complexity measure of sign matrices (Linial et al., 2007) and is linked to learnability (Alon et al., 2016) and empirically confirmed by the wide success of factorization models (Nickel et al., 2016a).

Using Corollary 2, we can now show that any square sign matrix of sign-rank k can be reconstructed from a rank $2k$ unitary diagonalization.

Corollary 3 *Suppose $Y \in \{-1, 1\}^{n \times n}$, $\mathrm{rank}_\pm(Y) = k$. Then there exists $E \in \mathbb{C}^{n \times 2k}$, $W \in \mathbb{C}^{2k \times 2k}$ where the columns of E form an orthonormal basis of $\mathbb{C}^{2k}$, and W is diagonal, such that $Y = \mathrm{sign}(\mathrm{Re}(EWE^*))$.*

Proof By definition, if $\mathrm{rank}_\pm(Y) = k$, there exists a real square matrix X such that $\mathrm{rank}(X) = k$ and $\mathrm{sign}(X) = Y$. From Corollary 2, $X = \mathrm{Re}(EWE^*)$ where $E \in \mathbb{C}^{n \times 2k}$, $W \in \mathbb{C}^{2k \times 2k}$ where the columns of E form an orthonormal basis of $\mathbb{C}^{2k}$, and W is diagonal. ■

Previous attempts to approximate the sign-rank in relational learning did not use complex numbers. They showed the existence of compact factorizations under conditions on the sign matrix (Nickel et al., 2014), or only in specific cases (Bouchard et al., 2015). In contrast, our results show that if a square sign matrix has sign-rank k, then it can be exactly decomposed through a $2k$-dimensional unitary diagonalization.

Although we can only show the existence of a complex decomposition of rank $2k$ for a matrix with sign-rank k, the sign rank of Y is often *much* lower than the rank of Y, as we do not know any matrix $Y \in \{-1,1\}^{n\times n}$ for which $\text{rank}_{\pm}(Y) > \sqrt{n}$ (Alon et al., 2016). For example, the $n \times n$ identity matrix has rank n, but its sign-rank is only 3! By swapping the columns $2j$ and $2j-1$ for j in $1,\ldots,\frac{n}{2}$, the identity matrix corresponds to the relation `marriedTo`, a relation known to be hard to factorize over the reals (Nickel et al., 2014), since the rank is invariant by row/column permutations. Yet our model can express it at most in rank 6, for any n.

Hence, by enforcing a low-rank $K \ll n$ on EWE^*, individual relation scores $x_{so} = \text{Re}(e_s^\top W \bar{e}_o)$ between entities s and o can be efficiently predicted, as $e_s, e_o \in \mathbb{C}^K$ and $W \in \mathbb{C}^{K\times K}$ is diagonal.

Finding the K that matches the sign-rank of Y corresponds to finding the smallest K that brings the 0–1 loss on X to 0, as link prediction can be seen as binary classification of the facts. In practice, and as classically done in machine learning to avoid this NP-hard problem, we use a continuous surrogate of the 0–1 loss, in this case the logistic loss as described in Section 4, and validate models on different values of K, as described in Section 5.

2.2.3 Rank Bound Discussion

Corollaries 2 and 3 use the aforementioned subadditive property of the rank to derive the $2k$ upper bound. Let us give an example for which this bound is strictly greater than k.

Consider the following 2-by-2 sign matrix:

$$Y = \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}.$$

Not only is this matrix not normal, but one can also easily check that there is no *real* normal 2-by-2 matrix that has the same sign-pattern as Y. Clearly, Y is a rank 1 matrix since its columns are linearly dependent, hence its sign-rank is also 1. From Corollary 3, we know that there is a normal matrix whose real part has the same sign-pattern as Y, and whose rank is at most 2.

However, there is no rank 1 unitary diagonalization of which the real part equals Y. Otherwise we could find a 2-by-2 complex matrix Z such that $\text{Re}(z_{11}) < 0$ and $\text{Re}(z_{22}) > 0$, where $z_{11} = e_1 w \bar{e}_1 = w|e_1|^2$, $z_{22} = e_2 w \bar{e}_2 = w|e_2|^2$, $e \in \mathbb{C}^2, w \in \mathbb{C}$. This is obviously unsatisfiable. This example generalizes to any n-by-n square sign matrix that only has -1 on its first row and is hence rank 1, the same argument holds considering $\text{Re}(z_{11}) < 0$ and $\text{Re}(z_{nn}) > 0$.

This example shows that the upper bound on the rank of the unitary diagonalization showed in Corollaries 2 and 3 can be strictly greater than k, the rank or sign-rank, of the decomposed matrix. However, there might be other examples for which the addition of

an imaginary part could—additionally to making the matrix normal—create some linear dependence between the rows/columns and thus decrease the rank of the matrix, up to a factor of 2.

We summarize this section in three points:

1. The proposed factorization encompasses all possible score matrices X for a single binary relation.
2. By construction, the factorization is well suited to represent both symmetric and antisymmetric relations.
3. Relation patterns can be efficiently approximated with a low-rank factorization using complex-valued embeddings.

3. Extension to Multi-Relational Data

Let us now extend the previous discussion to models with multiple relations. Let $\mathcal{R}$ be the set of relations, with $|\mathcal{R}| = m$. We shall now write $\mathbf{X} \in \mathbb{R}^{m\times n\times n}$ for the score tensor, $X_r \in \mathbb{R}^{n\times n}$ for the score matrix of the relation $r \in \mathcal{R}$, and $\mathbf{Y} \in \{-1,1\}^{m\times n\times n}$ for the partially-observed sign tensor.

Given one relation $r \in \mathcal{R}$ and two entities $s, o \in \mathcal{E}$, the probability that the fact *r(s,o)* is true given by:

$$P(y_{rso} = 1) = \sigma(x_{rso}) = \sigma(\phi(r, s, o; \Theta)) \tag{5}$$

where ϕ is the scoring function of the model considered and Θ denotes the model parameters. We denote the set of all possible facts (or triples) for a knowledge graph by $\mathcal{T} = \mathcal{R} \times \mathcal{E} \times \mathcal{E}$. While the tensor $\mathbf{X}$ as a whole is unknown, we assume that we observe a set of true and false triples $\Omega = \{((r,s,o), y_{rso}) \,|\, (r,s,o) \in \mathcal{T}_\Omega\}$ where $y_{rso} \in \{-1,1\}$ and $\mathcal{T}_\Omega \subseteq \mathcal{T}$ is the set of observed triples. The goal is to find the probabilities of entries $y_{r's'o'}$ for a set of targeted unobserved triples $\{(r',s',o') \in \mathcal{T} \setminus \mathcal{T}_\Omega\}$.

Depending on the scoring function $\phi(r,s,o;\Theta)$ used to model the score tensor $\mathbf{X}$, we obtain different models. Examples of scoring functions are given in Table 1.

3.1 Complex Factorization Extension to Tensors

The single-relation model is extended by jointly factorizing all the square matrices of scores into a 3rd-order tensor $\mathbf{X} \in \mathbb{R}^{m\times n\times n}$, with a different diagonal matrix $W_r \in \mathbb{C}^{K\times K}$ for each relation r, and by sharing the entity embeddings $E \in \mathbb{C}^{n\times K}$ across all relations:

$$\begin{aligned} \phi(r,s,o;\Theta) &= \text{Re}(e_s^\top W_r \bar{e}_o) \\ &= \text{Re}\Big(\sum_{k=1}^{K} w_{rk} e_{sk} \bar{e}_{ok}\Big) \\ &= \text{Re}(\langle w_r, e_s, \bar{e}_o \rangle) \end{aligned} \tag{6}$$

where K is the rank hyperparameter, $e_s, e_o \in \mathbb{C}^K$ are the rows in E corresponding to the entities s and o, $w_r = \text{diag}(W_r) \in \mathbb{C}^K$ is a complex vector, and $\langle a,b,c \rangle := \sum_k a_k b_k c_k$ is the

component-wise multilinear dot product[2]. For this scoring function, the set of parameters Θ is $\{e_i, w_r \in \mathbb{C}^K, i \in \mathcal{E}, r \in \mathcal{R}\}$. This resembles the real part of a complex matrix decomposition as in the single-relation case discussed above. However, we now have a different vector of eigenvalues for every relation. Expanding the real part of this product gives:

$$\begin{aligned}\mathrm{Re}(\langle w_r, e_s, \bar{e}_o\rangle) \;=\; & \langle \mathrm{Re}(w_r), \mathrm{Re}(e_s), \mathrm{Re}(e_o)\rangle \\ & + \langle \mathrm{Re}(w_r), \mathrm{Im}(e_s), \mathrm{Im}(e_o)\rangle \\ & + \langle \mathrm{Im}(w_r), \mathrm{Re}(e_s), \mathrm{Im}(e_o)\rangle \\ & - \langle \mathrm{Im}(w_r), \mathrm{Im}(e_s), \mathrm{Re}(e_o)\rangle \;. \end{aligned} \tag{7}$$

These equations provide two interesting views of the model:

- *Changing the representation*: Equation (6) would correspond to DistMult with real embeddings (see Table 1), but handles asymmetry thanks to the complex conjugate of the object-entity embedding.
- *Changing the scoring function*: Equation (7) only involves real vectors corresponding to the real and imaginary parts of the embeddings and relations.

By separating the real and imaginary parts of the relation embedding w_r as shown in Equation (7), it is apparent that these parts naturally act as weights on each latent dimension: $\mathrm{Re}(w_r)$ over the real part of $\langle e_o, e_s\rangle$ which is symmetric, and $\mathrm{Im}(w)$ over the imaginary part of $\langle e_o, e_s\rangle$ which is antisymmetric.

Indeed, the decomposition of each score matrix X_r for each $r \in \mathcal{R}$ can be written as the sum of a symmetric matrix and an antisymmetric matrix. To see this, let us rewrite the decomposition of each score matrix X_r in matrix notation. We write the real part of matrices with primes $E' = \mathrm{Re}(E)$ and imaginary parts with double primes $E'' = \mathrm{Im}(E)$:

$$\begin{aligned} X_r &= \mathrm{Re}(E W_r E^*) \\ &= \mathrm{Re}((E' + iE'')(W'_r + iW''_r)(E' - iE'')^\top) \\ &= (E'W'_rE'^\top + E''W'_rE''^\top) + (E'W''_rE''^\top - E''W''_rE'^\top)\,. \end{aligned} \tag{8}$$

It is trivial to check that the matrix $E'W'_rE'^\top + E''W'_rE''^\top$ is symmetric and that the matrix $E'W''_rE''^\top - E''W''_rE'^\top$ is antisymmetric. Hence this model is well suited to model jointly symmetric and antisymmetric relations between pairs of entities, while still using the same entity representations for subjects and objects. When learning, it simply needs to collapse $W''_r = \mathrm{Im}(W_r)$ to zero for symmetric relations $r \in \mathcal{R}$, and $W'_r = \mathrm{Re}(W_r)$ to zero for antisymmetric relations $r \in \mathcal{R}$, as X_r is indeed symmetric when W_r is purely real, and antisymmetric when W_r is purely imaginary.

From a geometrical point of view, each relation embedding w_r is an anisotropic scaling of the basis defined by the entity embeddings E, followed by a projection onto the real subspace.

2. This is not the Hermitian extension of the multilinear dot product as there appears to be no standard definition of the Hermitian multilinear product in the linear algebra literature.

3.2 Existence of the Tensor Factorization

Let us first discuss the existence of the multi-relational model where the rank of the decomposition $K \leq n$, which relates to simultaneous unitary decomposition.

Definition 6 *A family of matrices $X_1, \ldots, X_m \in \mathbb{C}^{n\times n}$ is simultaneously unitarily diagonalizable, if there is a single unitary matrix $E \in \mathbb{C}^{n\times n}$, such that $X_i = EW_iE^*$ for all i in $1, \ldots, m$, where $W_i \in \mathbb{C}^{n\times n}$ are diagonal.*

Definition 7 *A family of normal matrices $X_1, \ldots, X_m \in \mathbb{C}^{n\times n}$ is a commuting family of normal matrices, if $X_iX_j^* = X_i^*X_j$, for all i, j in $1, \ldots, m$.*

Theorem 3 (see Horn and Johnson (2012)) *Suppose $\mathcal{F}$ is the family of matrices X_1, $\ldots$, $X_m \in \mathbb{C}^{n\times n}$. Then $\mathcal{F}$ is a commuting family of normal matrices if and only if $\mathcal{F}$ is simultaneously unitarily diagonalizable.*

To apply Theorem 3 to the proposed factorization, we would have to make the hypothesis that the relation score matrices X_r are a commuting family, which is too strong a hypothesis. Actually, the model is slightly different since we take only the real part of the tensor factorization. In the single-relation case, taking only the real part of the decomposition rids us of the normality requirement of Theorem 1 for the decomposition to exist, as shown in Theorem 2.

In the multiple-relation case, it is an open question whether taking the real part of the simultaneous unitary diagonalization will enable us to decompose families of arbitrary real square matrices—that is with a single unitary matrix E that has *at most* n columns. Though it seems unlikely, we could not find a counter-example yet.

However, by letting the rank of the tensor factorization K to be greater than n, we can show that the proposed tensor decomposition exists for families of arbitrary real square matrices, by simply concatenating the decomposition of Theorem 2 of each real square matrix X_i.

Theorem 4 *Suppose $X_1, \ldots, X_m \in \mathbb{R}^{n\times n}$. Then there exists $E \in \mathbb{C}^{n\times nm}$ and $W_i \in \mathbb{C}^{nm\times nm}$ are diagonal, such that $X_i = \mathrm{Re}(EW_iE^*)$ for all i in $1, \ldots, m$.*

Proof From Theorem 2 we have $X_i = \mathrm{Re}(E_iW_iE_i^*)$, where $W_i \in \mathbb{C}^{n\times n}$ is diagonal, and each $E_i \in \mathbb{C}^{n\times n}$ is unitary for all i in $1, \ldots, m$.

Let $E = [E_1 \ldots E_m]$, and

$$\Lambda_i = \begin{bmatrix} \mathbf{0}^{((i-1)n)\times((i-1)n)} & & \\ & W_i & \\ & & \mathbf{0}^{((m-i)n)\times((m-i)n)} \end{bmatrix}$$

where $\mathbf{0}^{l\times l}$ the zero $l \times l$ matrix. Therefore $X_i = \mathrm{Re}(E\Lambda_iE^*)$ for all i in $1, \ldots, m$. ■

By construction, the rank of the decomposition is at most nm. When $m \leq n$, this bound actually matches the general upper bound on the rank of the canonical polyadic (CP) decomposition (Hitchcock, 1927; Kruskal, 1989). Since m corresponds to the number

of relations and n to the number of entities, m is always smaller than n in real world knowledge graphs, hence the bound holds in practice.

Though when it comes to relational learning, we might expect the actual rank to be much lower than nm for two reasons. The first one, as discussed above, is that we are dealing with sign tensors, hence the rank of the matrices X_r need only match the sign-rank of the partially-observed matrices Y_r. The second one is that the matrices are related to each other, as they all represent the same entities in different relations, and thus benefit from sharing latent dimensions. As opposed to the construction exposed in the proof of Theorem 4, where other relations dimensions are canceled out. In practice, the rank needed to generalize well is indeed much lower than nm as we show experimentally in Figure 5.

Also, note that with the construction of the proof of Theorem 4, the matrix $E = [E_1 \dots E_m]$ is not unitary any more. However the unitary constraints in the matrix case serve only the proof of existence, which is just one solution among the infinite ones of same rank. In practice, imposing orthonormality is essentially a numerical commodity for the decomposition of dense matrices, through iterative methods for example (Saad, 1992). When it comes to matrix and tensor completion, and thus generalisation, imposing such constraints is more of a numerical hassle than anything else, especially for gradient methods. As there is no apparent link between orthonormality and generalisation properties, we did not impose these constraints when learning this model in the following experiments.

4. Algorithm

Algorithm 1 describes stochastic gradient descent (SGD) to learn the proposed multi-relational model with the AdaGrad learning-rate updates (Duchi et al., 2011). We refer to the proposed model as ComplEx, for Complex Embeddings. We expose a version of the algorithm that uses only real-valued vectors, in order to facilitate its implementation. To do so, we use separate real-valued representations of the real and imaginary parts of the embeddings.

These real and imaginary part vectors are initialized with vectors having a zero-mean normal distribution with unit variance. If the training set Ω contains only positive triples, negatives are generated for each batch using the *local closed-world assumption* as in Bordes et al. (2013b). That is, for each triple, we randomly change either the subject or the object, to form a negative example. In this case the parameter $\eta > 0$ sets the number of negative triples to generate for each positive triple. Collision with positive triples in Ω is not checked, as it occurs rarely in real world knowledge graphs as they are largely sparse, and may also be computationally expensive.

Squared gradients are accumulated to compute AdaGrad learning rates, then gradients are updated. Every s iterations, the parameters Θ are evaluated over the evaluation set Ω_v (*evaluate_AP_or_MRR*$(\Omega_v; \Theta)$ function in Algorithm 1). If the data set contains both positive and negative examples, average precision (AP) is used to evaluate the model. If the data set contains only positives, then mean reciprocal rank (MRR) is used as average precision cannot be computed without true negatives. The optimization process is stopped when the measure considered decreases compared to the last evaluation (early stopping).

Bern(p) is the Bernoulli distribution, the *one_random_sample*$(\mathcal{E})$ function sample uniformly one entity in the set of all entities $\mathcal{E}$, and the *sample_batch_of_size_b*(Ω, b) function sample b true and false triples uniformly at random from the training set Ω.

For a given embedding size K, let us rewrite Equation (7), by denoting the real part of embeddings with primes and the imaginary part with double primes: $e'_i = \mathrm{Re}(e_i)$, $e''_i = \mathrm{Im}(e_i)$, $w'_r = \mathrm{Re}(w_r)$, $w''_r = \mathrm{Im}(w_r)$. The set of parameters is $\Theta = \{e'_i, e''_i, w'_r, w''_r \in \mathbb{R}^K, i \in \mathcal{E}, r \in \mathcal{R}\}$, and the scoring function involves only real vectors:

$$\begin{aligned}\phi(r, s, o; \Theta) = \quad & \langle w'_r, e'_s, e'_o \rangle + \langle w'_r, e''_s, e''_o \rangle \\ & + \langle w''_r, e'_s, e''_o \rangle - \langle w''_r, e''_s, e'_o \rangle \end{aligned} \tag{9}$$

where each entity and each relation has two real embeddings.

Gradients are now easy to write:

$$\begin{aligned}
\nabla_{e'_s} \phi(r, s, o; \Theta) &= (w'_r \odot e'_o) + (w''_r \odot e''_o), \\
\nabla_{e''_s} \phi(r, s, o; \Theta) &= (w'_r \odot e''_o) - (w''_r \odot e'_o), \\
\nabla_{e'_o} \phi(r, s, o; \Theta) &= (w'_r \odot e'_s) - (w''_r \odot e''_s), \\
\nabla_{e''_o} \phi(r, s, o; \Theta) &= (w'_r \odot e''_s) + (w''_r \odot e'_s), \\
\nabla_{w'_r} \phi(r, s, o; \Theta) &= (e'_s \odot e'_o) + (e''_s \odot e''_o), \\
\nabla_{w''_r} \phi(r, s, o; \Theta) &= (e'_s \odot e''_o) - (e''_s \odot e'_o),
\end{aligned}$$

where $\odot$ is the element-wise (Hadamard) product.

We optimized the negative log-likelihood of the logistic model described in Equation (5) with L^2 regularization on the parameters Θ:

$$\gamma(\Omega; \Theta) = \sum_{((r,s,o),y) \in \Omega} \log(1 + \exp(-y\phi(r, s, o; \Theta))) + \lambda ||\Theta||_2^2 \tag{10}$$

where $\lambda \in \mathbb{R}_+$ is the regularization parameter.

To handle regularization, note that using separate representations for the real and imaginary parts does not change anything as the squared L^2-norm of a complex vector $v = v' + iv''$ is the sum of the squared modulus of each entry:

$$\begin{aligned}
||v||_2^2 &= \sum_j \sqrt{v'^2_j + v''^2_j}^2 \\
&= \sum_j v'^2_j + \sum_j v''^2_j \\
&= ||v'||_2^2 + ||v''||_2^2 ,
\end{aligned}$$

which is actually the sum of the L^2-norms of the vectors of the real and imaginary parts.

We can finally write the gradient of γ with respect to a *real* embedding v for one triple (r, s, o) and its truth value y:

$$\nabla_v \gamma(\{((r, s, o), y)\}; \Theta) = -y\sigma(-y\phi(r, s, o; \Theta)) \nabla_v \phi(r, s, o; \Theta) + 2\lambda v \,. \tag{11}$$

Algorithm 1 Stochastic gradient descent with AdaGrad for the ComplEx model

Input Training set Ω, validation set Ω_v, learning rate $\alpha \in \mathbb{R}_{++}$, rank $K \in \mathbb{Z}_{++}$, L^2 regularization factor $\lambda \in \mathbb{R}_{+}$, negative ratio $\eta \in \mathbb{Z}_{++}$, batch size $b \in \mathbb{Z}_{++}$, maximum iteration $m \in \mathbb{Z}_{++}$, validate every $s \in \mathbb{Z}_{++}$ iterations, AdaGrad regularizer $\epsilon = 10^{-8}$.

Output Embeddings e', e'', w', w''.

$e'_i \sim \mathcal{N}(\mathbf{0}^k, I^{k\times k})$, $e''_i \sim \mathcal{N}(\mathbf{0}^k, I^{k\times k})$ for each $i \in \mathcal{E}$
$w'_i \sim \mathcal{N}(\mathbf{0}^k, I^{k\times k})$, $w''_i \sim \mathcal{N}(\mathbf{0}^k, I^{k\times k})$ for each $i \in \mathcal{R}$
$g_{e'_i} \leftarrow \mathbf{0}^k$, $g_{e''_i} \leftarrow \mathbf{0}^k$ for each $i \in \mathcal{E}$
$g_{w'_i} \leftarrow \mathbf{0}^k$, $g_{w''_i} \leftarrow \mathbf{0}^k$ for each $i \in \mathcal{R}$
previous_score $\leftarrow 0$
for $i = 1, \ldots, m$ **do**
  for $j = 1, \ldots, |\Omega|/b$ **do**
    $\Omega_b \leftarrow$ *sample_batch_of_size_b*(Ω, b)
    // Negative sampling:
    $\Omega_n \leftarrow \{\emptyset\}$
    for $((r, s, o), y)$ in Ω_b **do**
      for $l = 1, \ldots, \eta$ **do**
        $e \leftarrow$ *one_random_sample*$(\mathcal{E})$
        if Bern$(0.5) > 0.5$ **then**
          $\Omega_n \leftarrow \Omega_n \cup \{((r, e, o), -1)\}$
        else
          $\Omega_n \leftarrow \Omega_n \cup \{((r, s, e), -1)\}$
        end if
      end for
    end for
    $\Omega_b \leftarrow \Omega_b \cup \Omega_n$
    for $((r, s, o), y)$ in Ω_b **do**
      for v in Θ **do**
        // AdaGrad updates:
        $g_v \leftarrow g_v + (\nabla_v \gamma(\{((r, s, o), y)\}; \Theta))^2$
        // Gradient updates:
        $v \leftarrow v - \frac{\alpha}{g_v + \epsilon} \nabla_v \gamma(\{((r, s, o), y)\}; \Theta)$
      end for
    end for
  end for
  // Early stopping
  if $i \mod s = 0$ **then**
    current_score $\leftarrow$ *evaluate_AP_or_MRR*$(\Omega_v; \Theta)$
    if *current_score* $\leq$ *previous_score* **then**
      break
    end if
    previous_score $\leftarrow$ *current_score*
  end if
end for
return Θ

5. Experiments

We evaluated the method proposed in this paper on both synthetic and real data sets. The synthetic data set contains both symmetric and antisymmetric relations, whereas the real data sets are standard link prediction benchmarks based on real knowledge graphs.

We compared ComplEx to state-of-the-art models, namely TransE (Bordes et al., 2013b), DistMult (Yang et al., 2015), RESCAL (Nickel et al., 2011) and also to the canonical polyadic decomposition (CP) (Hitchcock, 1927), to emphasize empirically the importance of learning unique embeddings for entities. For experimental fairness, we reimplemented these models within the same framework as the ComplEx model, using a Theano-based SGD implementation[3] (Bergstra et al., 2010).

For the TransE model, results were obtained with its original max-margin loss, as it turned out to yield better results for this model only. To use this max-margin loss on data sets with observed negatives (Sections 5.1 and 5.2), positive triples were replicated when necessary to match the number of negative triples, as described in Garcia-Duran et al. (2016). All other models are trained with the negative log-likelihood of the logistic model (Equation (10)). In all the following experiments we used a maximum number of iterations $m = 1000$, a batch size $b = \frac{|\Omega|}{100}$, and validated the models for early stopping every $s = 50$ iterations.

5.1 Synthetic Task

To assess our claim that ComplEx can accurately model jointly symmetry and antisymmetry, we randomly generated a knowledge graph of two relations and 30 entities. One relation is entirely symmetric, while the other is completely antisymmetric. This data set corresponds to a $2 \times 30 \times 30$ tensor. Figure 2 shows a part of this randomly generated tensor, with a symmetric slice and an antisymmetric slice, decomposed into training, validation and test sets. To ensure that all test values are predictable, the upper triangular parts of the matrices are always kept in the training set, and the diagonals are unobserved. We conducted a 5-fold cross-validation on the lower-triangular matrices, using the upper-triangular parts plus 3 folds for training, one fold for validation and one fold for testing. Each training set contains 1392 observed triples, whereas validation and test sets contain 174 triples each.

Figure 3 shows the best cross-validated average precision (area under the precision-recall curve) for different factorization models of ranks ranging up to 50. The regularization parameter λ is validated in {0.1, 0.03, 0.01, 0.003,0.001, 0.0003, 0.00001, 0.0} and the learning rate α was initialized to 0.5.

As expected, DistMult (Yang et al., 2015) is not able to model antisymmetry and only predicts the symmetric relations correctly. Although TransE (Bordes et al., 2013b) is not a symmetric model, it performs poorly in practice, particularly on the antisymmetric relation. RESCAL (Nickel et al., 2011), with its large number of parameters, quickly overfits as the rank grows. Canonical Polyadic (CP) decomposition (Hitchcock, 1927) fails on both relations as it has to push symmetric and antisymmetric patterns through the entity embeddings. Surprisingly, only ComplEx succeeds even on such simple data.

3. https://github.com/lmjohns3/downhill

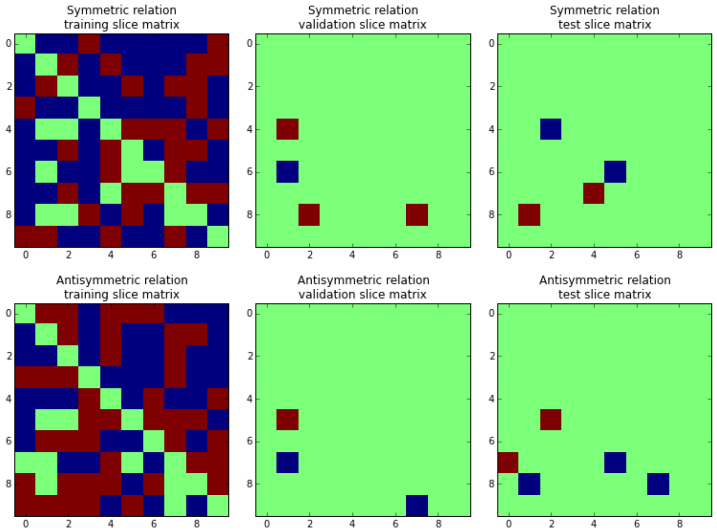


Figure 2: Parts of the training, validation and test sets of the generated experiment with one symmetric and one antisymmetric relation. Red pixels are positive triples, blue are negatives, and green missing ones. Top: Plots of the symmetric slice (relation) for the 10 first entities. Bottom: Plots of the antisymmetric slice for the 10 first entities.

5.2 Real Fully-Observed Data Sets: Kinships and UMLS

We then compare all models on two fully observed data sets, that contain both positive and negative triples, also called the *closed-world assumption*. The Kinships data set (Denham, 1973) describes the 26 different kinship relations of the Alyawarra tribe in Australia, among 104 individuals. The unified medical language system (UMLS) data set (McCray, 2003) represents 135 medical concepts and diseases, linked by 49 relations describing their interactions. Metadata for the two data sets is summarized in Table 2.

Data set	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	Total number of triples
Kinships	104	26	281,216
UMLS	135	49	893,025

Table 2: Number of entities $|\mathcal{E}|$, relations $|\mathcal{R}|$, and observed triples (all are observed) for the Kinships and UMLS data sets.

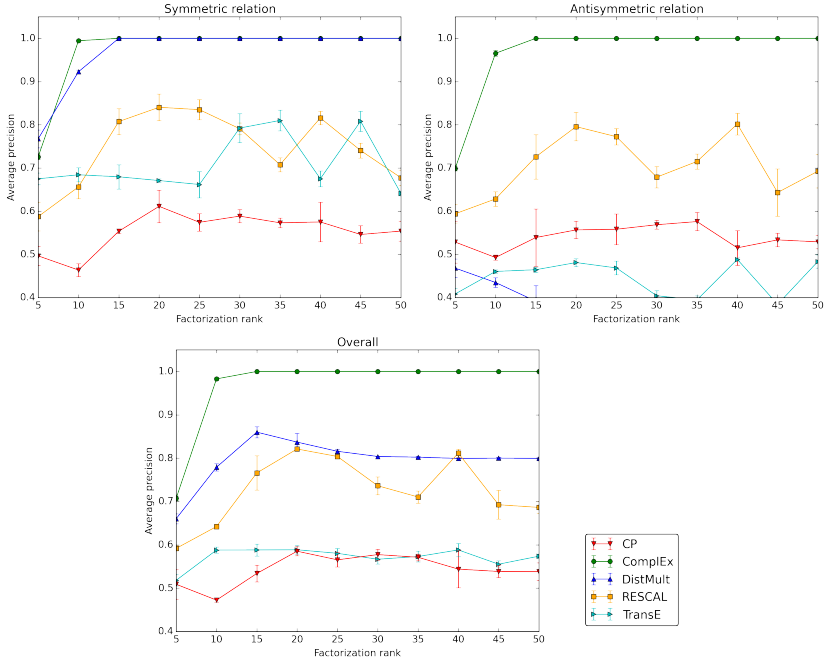


Figure 3: Average precision (AP) for each factorization rank ranging from 5 to 50 for different state-of-the-art models on the synthetic task. Learning is performed jointly on the symmetric relation and on the antisymmetric relation. Top-left: AP over the symmetric relation only. Top-right: AP over the antisymmetric relation only. Bottom: Overall AP.

We performed a 10-fold cross-validation, keeping 8 for training, one for validation and one for testing. Figure 4 shows the best cross-validated average precision for ranks ranging up to 50, and error bars show the standard deviation over the 10 runs. The regularization parameter λ is validated in $\{0.1, 0.03, 0.01, 0.003, 0.001, 0.0003, 0.00001, 0.0\}$ and the learning rate α was initialized to 0.5.

On both data sets ComplEx, RESCAL and CP are very close, with a slight advantage for ComplEx on Kinships, and for RESCAL on UMLS. DistMult performs poorly here as many relations are antisymmetric both in UMLS (causal links, anatomical hierarchies) and Kinships (being father, uncle or grand-father).

The fact that CP, RESCAL and ComplEx work so well on these data sets illustrates the importance of having an expressive enough model, as DistMult fails because

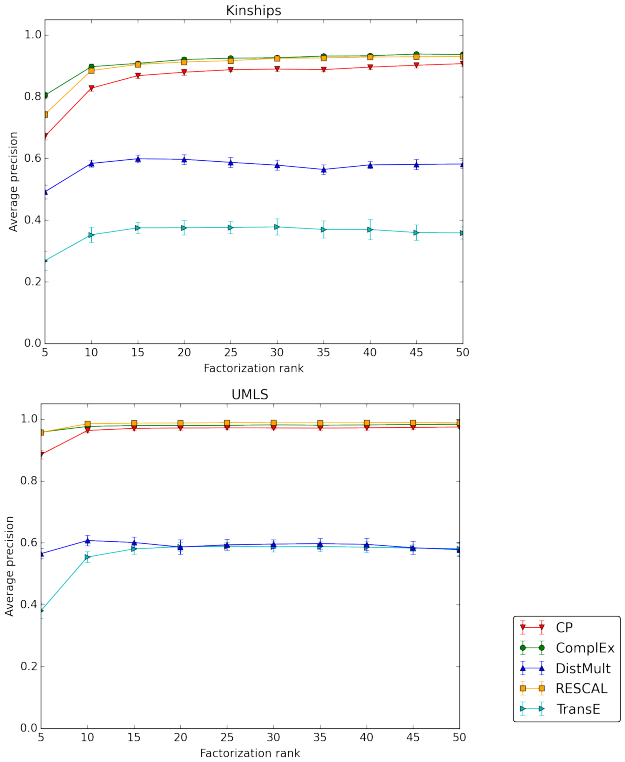


Figure 4: Average precision (AP) for each factorization rank ranging from 5 to 50 for different state-of-the-art models on the Kinships data set (top) and on the UMLS data set (bottom).

of antisymmetry; the power of the multilinear product—that is the tensor factorization approach—as TransE can be seen as a sum of bilinear products (Garcia-Duran et al., 2016); but not yet the importance of having unique entity embeddings, as CP works well. We believe having separate subject and object-entity embeddings works well under the closed-world assumption, because of the amount of training data compared to the number of embeddings to learn. Though when only a fractions of the positive training examples are observed (as it is most often the case), we will see in the next experiments that enforcing unique entity embeddings is key to good generalization.

			Number of triples in sets:		
Data set	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$	Training	Validation	Test
WN18	40,943	18	141,442	5,000	5,000
FB15K	14,951	1,345	483,142	50,000	59,071

Table 3: Number of entities $|\mathcal{E}|$, relations $|\mathcal{R}|$, and observed triples in each split for the FB15K and WN18 data sets.

5.3 Real Sparse Data Sets: FB15K and WN18

Finally, we evaluated ComplEx on the FB15K and WN18 data sets, as they are well established benchmarks for the link prediction task. FB15K is a subset of Freebase (Bollacker et al., 2008), a curated knowledge graph of general facts, whereas WN18 is a subset of WordNet (Fellbaum, 1998), a database featuring lexical relations between words. We used the same training, validation and test set splits as in Bordes et al. (2013b). Table 3 summarizes the metadata of the two data sets.

5.3.1 Experimental Setup

As both data sets contain only positive triples, we generated negative samples using the local closed-world assumption, as described in Section 4. For evaluation, we measure the quality of the ranking of each test triple among all possible subject and object substitutions : $r(s', o)$ and $r(s, o')$, for each s', o' in $\mathcal{E}$, as used in previous studies (Bordes et al., 2013b; Nickel et al., 2016b). Mean Reciprocal Rank (MRR) and Hits at N are standard evaluation measures for these data sets and come in two flavours: raw and filtered. The filtered metrics are computed *after* removing all the other positive observed triples that appear in either training, validation or test set from the ranking, whereas the raw metrics do not remove these.

Since ranking measures are used, previous studies generally preferred a max-margin ranking loss for the task (Bordes et al., 2013b; Nickel et al., 2016b). We chose to use the negative log-likelihood of the logistic model—as described in the previous section—as it is a continuous surrogate of the sign-rank, and has been shown to learn compact representations for several important relations, especially for transitive relations (Bouchard et al., 2015). As previously stated, we tried both losses in preliminary work, and indeed training the models with the log-likelihood yielded better results than with the max-margin ranking loss, especially on FB15K—except with TransE.

We report both filtered and raw MRR, and filtered Hits at 1, 3 and 10 in Table 4 for the evaluated models. The HolE model has recently been shown to be equivalent to ComplEx (Hayashi and Shimbo, 2017), we record the original results for HolE as reported in Nickel et al. (2016b) and briefly discuss the discrepancy of results obtained with ComplEx.

Reported results are given for the best set of hyper-parameters evaluated on the validation set for each model, after a distributed grid-search on the following values: $K \in \{10, 20, 50, 100, 150, 200\}$, $\lambda \in \{0.1, 0.03, 0.01, 0.003, 0.001, 0.0003, 0.0\}$, $\alpha \in \{1.0, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01\}$, $\eta \in \{1, 2, 5, 10\}$ with λ the L^2 regularization parameter, α the initial learning rate, and η the number of negatives generated per positive training triple. We also

	WN18					FB15K				
	MRR		Hits at			MRR		Hits at		
Model	Filtered	Raw	1	3	10	Filtered	Raw	1	3	10
CP	0.075	0.058	0.049	0.080	0.125	0.326	0.152	0.219	0.376	0.532
TransE	0.454	0.335	0.089	0.823	0.934	0.380	0.221	0.231	0.472	0.641
RESCAL	0.894	0.583	0.867	0.918	0.935	0.461	0.226	0.324	0.536	0.720
DistMult	0.822	0.532	0.728	0.914	0.936	0.654	**0.242**	0.546	0.733	0.824
HolE*	0.938	**0.616**	0.930	**0.945**	**0.949**	0.524	0.232	0.402	0.613	0.739
ComplEx	**0.941**	0.587	**0.936**	**0.945**	0.947	**0.692**	**0.242**	**0.599**	**0.759**	**0.840**

Table 4: Filtered and raw mean reciprocal rank (MRR) for the models tested on the FB15K and WN18 data sets. Hits@N metrics are filtered. *Results reported from Nickel et al. (2016b) for the HolE model, that has been shown to be equivalent to ComplEx (Hayashi and Shimbo, 2017), score divergence on FB15K is only due to the loss function used (Trouillon and Nickel, 2017).

tried varying the batch size but this had no impact and we settled with 100 batches per epoch. With the best hyper-parameters, training the ComplEx model on a single GPU (NVIDIA Tesla P40) takes 45 minutes on WN18 ($K = 150, \eta = 1$), and three hours on FB15K ($K = 200, \eta = 10$).

5.3.2 Results

WN18 describes lexical and semantic hierarchies between concepts and contains many antisymmetric relations such as hypernymy, hyponymy, and being part of. Indeed, the DistMult and TransE models are outperformed here by ComplEx and HolE, which are on a par with respective filtered MRR scores of 0.941 and 0.938, which is expected as both models are equivalent.

Table 5 shows the filtered MRR for the reimplemented models and each relation of WN18, confirming the advantage of ComplEx on antisymmetric relations while losing nothing on the others. 2D projections of the relation embeddings (Figures 8 & 9) visually corroborate the results.

On FB15K, the gap is much more pronounced and the ComplEx model largely outperforms HolE, with a filtered MRR of 0.692 and 59.9% of Hits at 1, compared to 0.524 and 40.2% for HolE. This difference of scores between the two models, though they have been proved to be equivalent (Hayashi and Shimbo, 2017), is due to the use of the aforementioned max-margin loss in the original HolE publication (Nickel et al., 2016b) that performs worse than the log-likelihood on this dataset, and to the generation of more than one negative sample per positive in these experiments. This has been confirmed and discussed in details by Trouillon and Nickel (2017). The fact that DistMult yields fairly high scores (0.654 filtered MRR) is also due to the task itself and the evaluation measures used. As the dataset only involves true facts, the test set never includes the opposite facts $r(o, s)$ of each test fact $r(s, o)$ for *antisymmetric* relations—as the opposite fact is always false. Thus highly scoring the opposite fact barely impacts the rankings for antisymmetric relations. This is not the case in the fully observed experiments (Section 5.2), as the opposite fact is known

Relation name	ComplEx	RESCAL	DistMult	TransE	CP
hypernym	**0.953**	0.935	0.791	0.446	0.109
hyponym	**0.946**	0.932	0.710	0.361	0.009
member_meronym	**0.921**	0.851	0.704	0.418	0.019
member_holonym	**0.946**	0.861	0.740	0.465	0.134
instance_hypernym	**0.965**	0.833	0.943	0.961	0.233
instance_hyponym	**0.945**	0.849	0.940	0.745	0.040
has_part	**0.933**	0.879	0.753	0.426	0.035
part_of	**0.940**	0.888	0.867	0.455	0.094
member_of_domain_topic	**0.924**	0.865	0.914	0.861	0.007
synset_domain_topic_of	**0.930**	0.855	0.919	0.917	0.153
member_of_domain_usage	**0.917**	0.629	**0.917**	0.875	0.001
synset_domain_usage_of	**1.000**	0.541	**1.000**	**1.000**	0.134
member_of_domain_region	**0.865**	0.632	0.635	**0.865**	0.001
synset_domain_region_of	0.919	0.655	0.888	**0.986**	0.149
derivationally_related_form	**0.946**	0.928	0.940	0.384	0.100
similar_to	**1.000**	0.001	**1.000**	0.244	0.000
verb_group	**0.936**	0.857	0.897	0.323	0.035
also_see	0.603	0.302	**0.607**	0.279	0.020

Table 5: Filtered Mean Reciprocal Rank (MRR) for the models tested on each relation of the WordNet data set (WN18).

to be false—for antisymmetric relations—and largely impacts the average precision of the DistMult model (Figure 4).

RESCAL, that represents each relation with a $K \times K$ matrix, performs well on WN18 as there are few relations and hence not so many parameters. On FB15K though, it probably overfits due to the large number of relations and thus the large number of parameters to learn, and performs worse than a less expressive model like DistMult. On both data sets, TransE and CP are largely left behind. This illustrates again the power of the multilinear product in the first case, and the importance of learning unique entity embeddings in the second. CP performs especially poorly on WN18 due to the small number of relations, which magnifies this subject/object difference.

Figure 5 shows that the filtered MRR of the ComplEx model quickly converges on both data sets, showing that the low-rank hypothesis is reasonable in practice. The little gain of performances for ranks comprised between 50 and 200 also shows that ComplEx does not perform better because it has twice as many parameters for the same rank—the real and imaginary parts—compared to other linear space complexity models but indeed thanks to its better expressiveness.

Best ranks were generally 150 or 200, in both cases scores were always very close for all models, suggesting there was no need to grid-search on higher ranks. The number of negative samples per positive sample also had a large influence on the filtered MRR on FB15K (up to +0.08 improvement from 1 to 10 negatives), but not much on WN18. On both data sets regularization was important (up to +0.05 on filtered MRR between $\lambda = 0$ and optimal one). We found the initial learning rate to be very important on FB15K, while not so

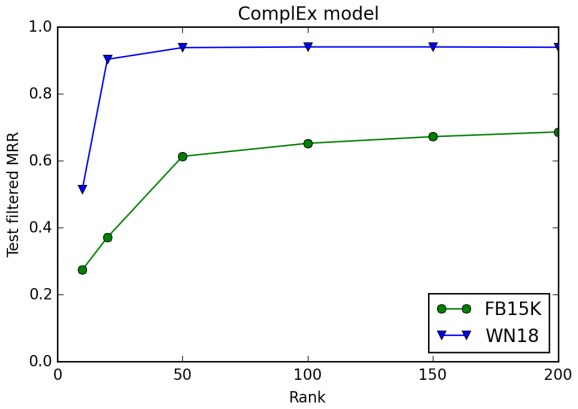


Figure 5: Best filtered MRR for ComplEx on the FB15K and WN18 data sets for different ranks. Increasing the rank gives little performance gain for ranks of $50 - 200$.

much on WN18. We think this may also explain the large gap of improvement ComplEx provides on this data set compared to previously published results—as DistMult results are also better than those previously reported (Yang et al., 2015)—along with the use of the log-likelihood objective. It seems that in general AdaGrad is relatively insensitive to the initial learning rate, perhaps causing some overconfidence in its ability to tune the step size online and consequently leading to less efforts when selecting the initial step size.

5.4 Training time

As defended in Section 2, having a linear time and space complexity becomes critical when the dataset grows. To illustrate this, we report in Figure 6 the evolution of the filtered MRR on the validation set as a function of time, for the best set of validated hyper-parameters for each model. The convergence criterion used is the decrease of the validation filtered MRR—computed every 50 iterations—with a maximum number of iterations of 1000 (see Algorithm 1). All models have a linear complexity except for RESCAL that has a quadratic one in the rank of the decomposition, as it learns one matrix embedding for each relation $r \in \mathcal{R}$. Timings are measured on a single NVIDIA Tesla P40 GPU.

On WN18, all models reach convergence in a reasonable time, between 15 minutes and 1 hour and 20 minutes. The difference between RESCAL and the other models is not sharp there, first because its optimal embedding size ($K = 50$) is lower compared to the other models. Secondly, there are only $|\mathcal{R}| = 18$ relations in WN18, hence the memory footprint of RESCAL is pretty similar to the other models—because it represents only relations with matrices and not entities. On FB15K, the difference is much more pronounced, as RESCAL optimal rank is similar to the other models; and with $|\mathcal{R}| = 1345$ relations, RESCAL has

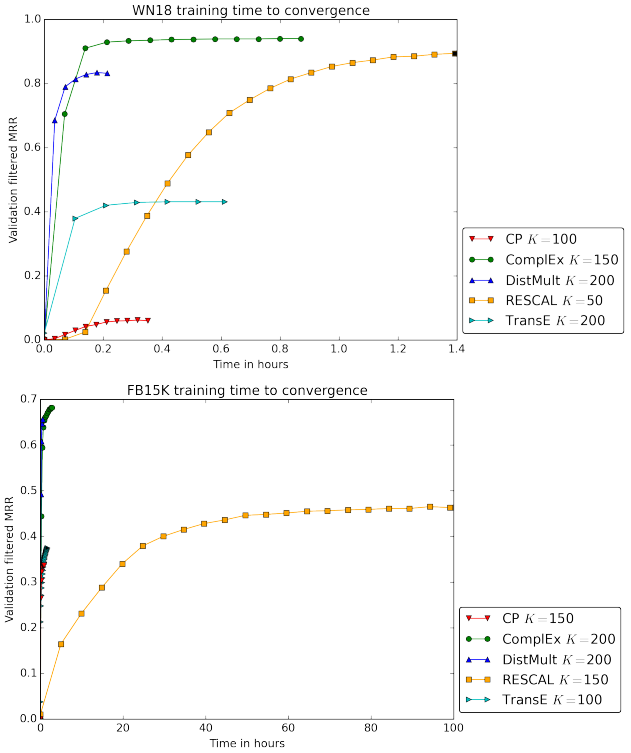


Figure 6: Evolution of the filtered MRR on the validation set as a function of time, on WN18 (top) and FB15K (bottom) for each model for its best set of hyper-parameters. The best rank K is reported in legend. Final black marker indicates that the maximum number of iterations (1000) has been reached (RESCAL on WN18, TransE on FB15K).

a much higher memory footprint, which implies more processor cache misses due to the uniformly-random nature of the SGD sampling.

RESCAL took more than four days to train on FB15K, whereas other models took between 40 minutes and 3 hours. While a few days might seem manageable, this could not be the case on larger data sets, as FB15K is but a small subset of Freebase that contains $|\mathcal{R}| = 35000$ relations (Bollacker et al., 2008). This experimentally supports our claim that linear complexity is required for scalability.

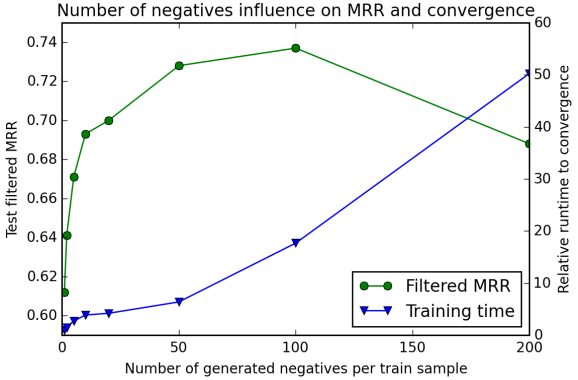


Figure 7: Influence of the number of negative triples generated per positive training example on the filtered test MRR and on training time to convergence on FB15K for the ComplEx model with $K = 200$, $\lambda = 0.01$ and $\alpha = 0.5$. Times are given relative to the training time with one negative triple generated per positive training sample ($= 1$ on time scale).

5.4.1 Influence of Negative Samples

We further investigated the influence of the number of negatives generated per positive training sample. In the previous experiment, due to computational limitations, the number of negatives per training sample, η, was validated over the set $\{1, 2, 5, 10\}$. On WN18 it proved to be of no help to have more than one generated negative per positive. Here we explore in which proportions increasing the number of generated negatives leads to better results on FB15K. To do so, we fixed the best validated λ, K, α obtained from the previous experiment. We then let η vary in $\{1, 2, 5, 10, 20, 50, 100, 200\}$.

Figure 7 shows the influence of the number of generated negatives per positive training triple on the performance of ComplEx on FB15K. Generating more negatives clearly improves the results up to 100 negative triples, with a filtered MRR of 0.737 and 64.8% of Hits@1, before decreasing again with 200 negatives, probably due to the too large class imbalance. The model also converges with fewer epochs, which compensates partially for the additional training time per epoch, up to 50 negatives. It then grows linearly as the number of negatives increases.

5.4.2 WN18 Embeddings Visualization

We used principal component analysis (PCA) to visualize embeddings of the relations of the WordNet data set (WN18). We plotted the four first components of the best DistMult and ComplEx model's embeddings in Figures 8 & 9. For the ComplEx model, we simply concatenated the real and imaginary parts of each embedding.

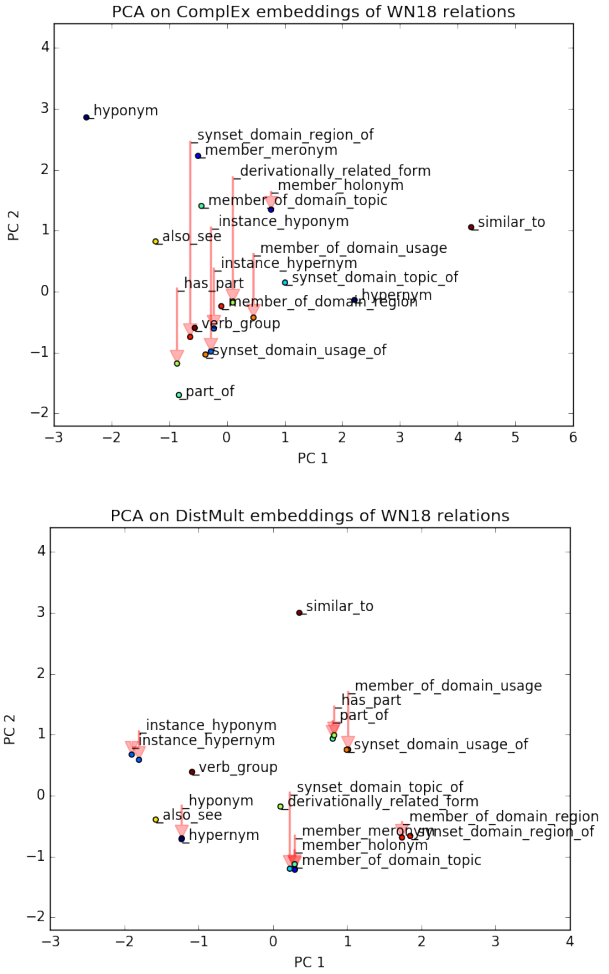


Figure 8: Plots of the first and second components of the WN18 relations embeddings using principal component analysis. Red arrows link the labels to their point. Top: ComplEx embeddings. Bottom: DistMult embeddings. Opposite relations are clustered together by DistMult while correctly separated by ComplEx.

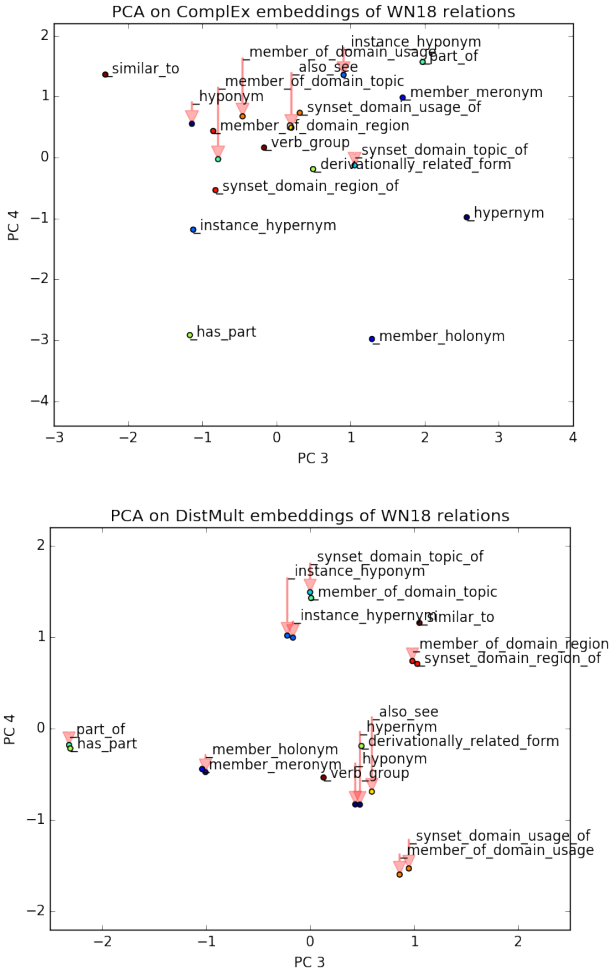


Figure 9: Plots of the third and fourth components of the WN18 relations embeddings using principal component analysis. Red arrows link the labels to their point. Top: ComplEx embeddings. Bottom: DistMult embeddings. Opposite relations are clustered together by DistMult while correctly separated by ComplEx.

Most of WN18 relations describe hierarchies, and are thus antisymmetric. Each of these hierarchic relations has its inverse relation in the data set. For example: `hypernym` / `hyponym`, `part_of` / `has_part`, `synset_domain_topic_of` / `member_of_domain_topic`. Since DistMult is unable to model antisymmetry, it will correctly represent the nature of each pair of opposite relations, but not the direction of the relations. Loosely speaking, in the `hypernym` / `hyponym` pair the nature is sharing semantics, and the direction is that one entity generalizes the semantics of the other. This makes DistMult representing the opposite relations with very close embeddings. It is especially striking for the third and fourth principal component (Figure 9). Conversely, ComplEx manages to oppose spatially the opposite relations.

6. Related Work

We first discuss related work about complex-valued matrix and tensor decompositions, and then review other approaches for knowledge graph completion.

6.1 Complex Numbers

When factorization methods are applied, the representation of the decomposition is generally chosen in accordance with the data, despite the fact that most real square matrices only have eigenvalues in the complex domain. Indeed in the machine learning community, the data is usually real-valued, and thus eigendecomposition is used for symmetric matrices, or other decompositions such as (real-valued) singular value decomposition (Beltrami, 1873), non-negative matrix factorization (Paatero and Tapper, 1994), or canonical polyadic decomposition when it comes to tensors (Hitchcock, 1927).

Conversely, in signal processing, data is often complex-valued (Stoica and Moses, 2005) and the complex-valued counterparts of these decompositions are then used. Joint diagonalization is also a much more common tool than in machine learning for decomposing sets of (complex) dense square matrices (Belouchrani et al., 1997; De Lathauwer et al., 2001).

Some works on recommender systems use complex numbers as an encoding facility, to merge two real-valued relations, similarity and liking, into one single complex-valued matrix which is then decomposed with complex embeddings (Kunegis et al., 2012; Xie et al., 2015). Still, unlike our work, it is not real data that is decomposed in the complex domain.

In deep learning, Danihelka et al. (2016) proposed an LSTM extended with an associative memory based on complex-valued vectors for memorization tasks, and Hu et al. (2016) a complex-valued neural network for speech synthesis. In both cases again, the data is first encoded in complex vectors that are then fed into the network.

Conversely to these contributions, this work suggests that processing real-valued data with complex-valued representation, through a projection onto the real-valued subspace, can be a very simple way of increasing the expressiveness of the model considered.

6.2 Knowledge Graph Completion

Many knowledge graphs have recently arisen, pushed by the W3C recommendation to use the resource description framework (RDF) (Cyganiak et al., 2014) for data representation. Examples of such knowledge graphs include DBPedia (Auer et al., 2007), Freebase (Bollacker

et al., 2008) and the Google Knowledge Vault (Dong et al., 2014). Motivating applications of knowledge graph completion include question answering (Bordes et al., 2014b) and more generally probabilistic querying of knowledge bases (Huang and Liu, 2009; Krompaß et al., 2014).

First approaches to relational learning relied upon probabilistic graphical models (Getoor and Taskar, 2007), such as bayesian networks (Friedman et al., 1999) and markov logic networks (Richardson and Domingos, 2006; Raedt et al., 2016).

With the first embedding models, asymmetry of relations was quickly seen as a problem and asymmetric extensions of tensors were studied, mostly by either considering independent embeddings (Franz et al., 2009) or considering relations as matrices instead of vectors in the RESCAL model (Nickel et al., 2011), or both (Sutskever, 2009). Direct extensions were based on uni-,bi- and trigram latent factors for triple data (Garcia-Duran et al., 2016), as well as a low-rank relation matrix (Jenatton et al., 2012). Bordes et al. (2014a) propose a two-layer model where subject and object embeddings are first separately combined with the relation embedding, then each intermediate representation is combined into the final score.

Pairwise interaction models were also considered to improve prediction performances. For example, the Universal Schema approach (Riedel et al., 2013) factorizes a 2D unfolding of the tensor (a matrix of entity pairs vs. relations) while Welbl et al. (2016) extend this also to other pairs. Riedel et al. (2013) also consider augmenting the knowledge graph facts by exctracting them from textual data, as does Toutanova et al. (2015). Injecting prior knowledge in the form of Horn clauses in the objective loss of the Universal Schema model has also been considered (Rocktaschel et al., 2015). Chang et al. (2014) enhance the RESCAL model to take into account information about the entity types. For recommender systems (thus with different subject/object sets of entities), Baruch (2014) proposed a non-commutative extension of the CP decomposition model. More recently, Gaifman models that learn neighborhood embeddings of local structures in the knowledge graph showed competitive performances (Niepert, 2016).

In the Neural Tensor Network (NTN) model, Socher et al. (2013) combine linear transformations and multiple bilinear forms of subject and object embeddings to jointly feed them into a nonlinear neural layer. Its non-linearity and multiple ways of including interactions between embeddings gives it an advantage in expressiveness over models with simpler scoring function like DistMult or RESCAL. As a downside, its very large number of parameters can make the NTN model harder to train and overfit more easily.

The original multilinear DistMult model is symmetric in subject and object for every relation (Yang et al., 2015) and achieves good performance on FB15K and WN18 data sets. However it is likely due to the absence of true negatives in these data sets, as discussed in Section 5.3.2.

The TransE model from Bordes et al. (2013b) also embeds entities and relations in the same space and imposes a geometrical structural bias into the model: the subject entity vector should be close to the object entity vector once translated by the relation vector.

A recent novel way to handle antisymmetry is via the Holographic Embeddings (HolE) model by Nickel et al. (2016b). In HolE the circular correlation is used for combining entity embeddings, measuring the covariance between embeddings at different dimension

shifts. This model has been shown to be equivalent to the ComplEx model (Hayashi and Shimbo, 2017; Trouillon and Nickel, 2017).

7. Discussion and Future Work

Though the decomposition proposed in this paper is clearly not unique, it is able to learn meaningful representations. Still, characterizing all possible unitary diagonalizations that preserve the real part is an interesting open question. Especially in an approximation setting with a constrained rank, in order to characterize the decompositions that minimize a given reconstruction error. That might allow the creation of an iterative algorithm similar to eigendecomposition iterative methods (Saad, 1992) for computing such a decomposition for any given real square matrix.

The proposed decomposition could also find applications in many other asymmetric square matrices decompositions applications, such as spectral graph theory for directed graphs (Cvetković et al., 1997), but also factorization of asymmetric measures matrices such as asymmetric distance matrices (Mao and Saul, 2004) and asymmetric similarity matrices (Pirasteh et al., 2015).

From an optimization point of view, the objective function (Equation (10)) is clearly non-convex, and we could indeed not be reaching a globally optimal decomposition using stochastic gradient descent. Recent results show that there are no spurious local minima in the completion problem of positive semi-definite matrix (Ge et al., 2016; Bhojanapalli et al., 2016). Studying the extensibility of these results to our decomposition is another possible line of future work. The first step would be generalizing these results to symmetric real-valued matrix completion, then generalization to normal matrices should be straightforward. The two last steps would be extending to matrices that are expressed as real part of normal matrices, and finally to the joint decomposition of such matrices as a tensor. We indeed noticed a remarkable stability of the scores across different random initialization of ComplEx for the same hyper-parameters, which suggests the possibility of such theoretical property.

Practically, an obvious extension is to merge our approach with known extensions to tensor factorization models in order to further improve predictive performance. For example, the use of pairwise embeddings (Riedel et al., 2013; Welbl et al., 2016) together with complex numbers might lead to improved results in many situations that involve non-compositionality. Adding bigram embeddings to the objective could also improve the results as shown on other models (Garcia-Duran et al., 2016). Another direction would be to develop a more intelligent negative sampling procedure, to generate more informative negatives with respect to the positive sample from which they have been sampled. This would reduce the number of negatives required to reach good performance, thus accelerating training time. Extension to relations between more than two entities, n-tuples, is not straightforward, as ComplEx's expressiveness comes from the complex conjugation of the object-entity, that breaks the symmetry between the subject and object embeddings in the scoring function. This stems from the Hermitian product, which seems to have no standard multilinear extension in the linear algebra literature, this question hence remains largely open.

8. Conclusion

We described a new matrix and tensor decomposition with complex-valued latent factors called ComplEx. The decomposition exists for all real square matrices, expressed as the real part of normal matrices. The result extends to sets of real square matrices—tensors—and answers to the requirements of the knowledge graph completion task : handling a large variety of different relations including antisymmetric and asymmetric ones, while being scalable. Experiments confirm its theoretical versatility, as it substantially improves over the state-of-the-art on real knowledge graphs. It shows that real world relations can be efficiently approximated as the real part of low-rank normal matrices. The generality of the theoretical results and the effectiveness of the experimental ones motivate for the application to other real square matrices factorization problems. More generally, we hope that this paper will stimulate the use of complex linear algebra in the machine learning community, even and especially for processing real-valued data.

Acknowledgments

This work was supported in part by the Association Nationale de la Recherche et de la Technologie through the CIFRE grant 2014/0121, in part by the Paul Allen Foundation through an Allen Distinguished Investigator grant, and in part by a Google Focused Research Award. We would like to thank Ariadna Quattoni, Stéphane Clinchant, Jean-Marc Andréoli, Sofia Michel, Alejandro Blumentals, Léo Hubert and Pierre Comon for their helpful comments and feedback.

References

Noga Alon, Shay Moran, and Amir Yehudayoff. Sign rank versus vc dimension. In *Conference on Learning Theory*, pages 47–80, 2016.

Sren Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, and Zachary Ives. DBpedia: A nucleus for a web of open data. In *International Semantic Web Conference, Busan, Korea*, pages 11–15. Springer, 2007.

Guy Baruch. A ternary non-commutative latent factor model for scalable three-way real tensor completion. *arXiv preprint arXiv:1410.7383*, 2014.

Adel Belouchrani, Karim Abed-Meraim, J-F Cardoso, and Eric Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, 1997.

Eugenio Beltrami. Sulle funzioni bilineari. *Giornale di Matematiche ad Uso degli Studenti Delle Universita*, 11(2):98–106, 1873.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Python for Scientific Computing Conference (SciPy)*, June 2010.

Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Global optimality of local search for low rank matrix recovery. *arXiv preprint arXiv:1605.07221*, 2016.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, 2008.

Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *AAAI Conference on Artificial Intelligence*, 2011.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Irreflexive and hierarchical relations as translations. *Computing Research Repository*, abs/1304.7158, 2013a.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013b.

Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. A semantic matching energy function for learning with multi-relational data. *Machine Learning*, 94(2):233–259, 2014a.

Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 165–180. Springer, 2014b.

Guillaume Bouchard, Sameer Singh, and Théo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. *AAAI Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches*, 2015.

Augustin-Louis Cauchy. Sur l'équation à l'aide de laquelle on détermine les inégalités séculaires des mouvements des planètes. *Œuvres complètes, IIe série*, 9:174–195, 1829.

K. W. Chang, W. T. Yih, B. Yang, and C. Meek. Typed tensor decomposition of knowledge bases for relation extraction. In *Conference on Empirical Methods on Natural Language Processing*, 2014.

Dragoš M. Cvetković, Peter Rowlinson, and Slobodan Simic. *Eigenspaces of graphs*. Number 66. Cambridge University Press, 1997.

Richard Cyganiak, David Wood, and Markus Lanthaler. Rdf 1.1 concepts and abstract syntax. *W3C Recommendation*, 2014.

Ivo Danihelka, Greg Wayne, Benigno Uria, Nal Kalchbrenner, and Alex Graves. Associative long short-term memory. *arXiv preprint arXiv:1602.03032*, 2016.

Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. Independent component analysis and (simultaneous) third-order tensor diagonalization. *IEEE Transactions on Signal Processing*, 49(10):2262–2271, 2001.

Woodrow W Denham. *The detection of patterns in Alyawara nonverbal behavior*. PhD thesis, University of Washington, Seattle., 1973.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 601–610, 2014.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

Thomas Franz, Antje Schultz, Sergej Sizov, and Steffen Staab. Triplerank: Ranking semantic web data by tensor decomposition. In *International Semantic Web Conference*, pages 213–228, 2009.

Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning Probabilistic Relational Models. In *International Joint Conference on Artificial Intelligence*, number August, pages 1300–1309, 1999. ISBN 3540422897. doi: 10.1.1.101.3165.

Alberto Garcia-Duran, Antoine Bordes, Nicolas Usunier, and Yves Grandvalet. Combining two and three-way embedding models for link prediction in knowledge bases. *Journal of Artificial Intelligence Research*, 55:715–742, 2016.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. *arXiv preprint arXiv:1605.07272*, 2016.

Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning*. The MIT Press, 2007. ISBN 0262072882.

Katsuhiko Hayashi and Masashi Shimbo. On the equivalence of holographic and complex embeddings for link prediction. *arXiv preprint arXiv:1702.05563*, 2017.

F. L. Hitchcock. The expression of a tensor or a polyadic as a sum of products. *J. Math. Phys*, 6(1):164–189, 1927.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge University Press, 2012.

Qiong Hu, Junichi Yamagishi, Korin Richmond, Kartick Subramanian, and Yannis Stylianou. Initial investigation of speech synthesis based on complex-valued neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5630–5634, 2016.

Hai Huang and Chengfei Liu. Query evaluation on probabilistic rdf databases. In *International Conference on Web Information Systems Engineering*, pages 307–320. Springer, 2009.

Rodolphe Jenatton, Antoine Bordes, Nicolas Le Roux, and Guillaume Obozinski. A Latent Factor Model for Highly Multi-relational Data. In *Advances in Neural Information Processing Systems 25*, pages 3167–3175, 2012.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Denis Krompaß, Maximilian Nickel, and Volker Tresp. Querying factorized probabilistic triple databases. In *International Semantic Web Conference*, pages 114–129, 2014.

Joseph B Kruskal. Rank, decomposition, and uniqueness for 3-way and n-way arrays. In *Multiway data analysis*, pages 7–18. North-Holland Publishing Co., 1989.

Jérôme Kunegis, Gerd Gröner, and Thomas Gottron. Online dating recommender systems: The split-complex number approach. In *ACM RecSys Workshop on Recommender Systems and the Social Web*, pages 37–44. ACM, 2012.

Nati Linial, Shahar Mendelson, Gideon Schechtman, and Adi Shraibman. Complexity measures of sign matrices. *Combinatorica*, 27(4):439–463, 2007.

Yun Mao and Lawrence K Saul. Modeling distances in large-scale networks by matrix factorization. In *ACM SIGCOMM conference on Internet Measurement*, pages 278–287, 2004.

Alexa T McCray. An upper-level ontology for the biomedical domain. *Comparative and Functional Genomics*, 4(1):80–84, 2003.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning*, pages 809–816, 2011.

Maximilian Nickel, Xueyan Jiang, and Volker Tresp. Reducing the rank in relational factorization models by including observable patterns. In *Advances in Neural Information Processing Systems*, pages 1179–1187, 2014.

Maximilian Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016a.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio. Holographic embeddings of knowledge graphs. In *AAAI Conference on Artificial Intelligence*, pages 1955–1961, 2016b.

Mathias Niepert. Discriminative gaifman models. In *Advances in Neural Information Processing Systems*, pages 3405–3413, 2016.

Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2): 111–126, 1994.

Parivash Pirasteh, Dosam Hwang, and Jason J Jung. Exploiting matrix factorization to asymmetric user similarities in recommendation systems. *Knowledge-Based Systems*, 83: 51–57, 2015.

Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(2):1–189, 2016.

Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62 (1-2):107–136, 2006.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. Relation extraction with matrix factorization and universal schemas. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, pages 74–84, 2013.

T Rocktaschel, S Singh, and S Riedel. Injecting Logical Background Knowledge into Embeddings for Relation Extraction. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1119–1129, 2015.

Youcef Saad. *Numerical methods for large eigenvalue problems*, volume 158. SIAM, 1992.

Satya S Sahoo, Wolfgang Halb, Sebastian Hellmann, Kingsley Idehen, Ted Thibodeau Jr, Sören Auer, Juan Sequeda, and Ahmed Ezzat. A survey of current approaches for mapping of relational databases to rdf. *W3C RDB2RDF Incubator Group Report*, pages 113–130, 2009.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

Petre Stoica and Randolph L Moses. *Spectral analysis of signals*, volume 452. Pearson Prentice Hall Upper Saddle River, NJ, 2005.

Ilya Sutskever. Modelling relational data using bayesian clustered tensor factorization. In *Advances in Neural Information Processing Systems*, pages 1–8, 2009.

Kristina Toutanova, Patrick Pantel, and Michael Gamon. Representing Text for Joint Embedding of Text and Knowledge Bases. In *Conference on Empirical Methods on Natural Language Processing*, 2015.

Théo Trouillon and Maximilian Nickel. Complex and holographic embeddings of knowledge graphs: a comparison. *International Workshop on Statistical Relational AI*, 2017.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, volume 48, pages 2071–2080, 2016.

John von Neumann. Zur algebra der funktionaloperationen und der theorie der normalen operatoren. *Mathematische Annalen*, 102:370–427, 1929.

Johannes Welbl, Guillaume Bouchard, and Sebastian Riedel. A factorization machine framework for testing bigram embeddings in knowledge base completion. In *Workshop on Automated Knowledge Base Construction AKBC@NAACL-HLT*, pages 103–107, 2016.

Feng Xie, Zhen Chen, Jiaxing Shang, Xiaoping Feng, and Jun Li. A link prediction approach for item recommendation with complex numbers. *Knowledge-Based Systems*, 81:148–158, 2015.

Bishan Yang, Wen-Tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *International Conference on Learning Representations*, 2015.

Journal of Machine Learning Research 18 (2017) 1-36 Submitted 4/16; Revised 8/17; Published 12/17

Stabilized Sparse Online Learning for Sparse Data

Yuting Ma YMA@STAT.COLUMBIA.EDU
Department of Statistics
Columbia University
New York, NY, USA

Tian Zheng TZHENG@STAT.COLUMBIA.EDU
Department of Statistics
Columbia University
New York, NY, USA

Editor: Sathiya Keerthi

Abstract

Stochastic gradient descent (SGD) is commonly used for optimization in large-scale machine learning problems. Langford et al. (2009) introduce a sparse online learning method to induce sparsity via truncated gradient. With high-dimensional sparse data, however, this method suffers from slow convergence and high variance due to heterogeneity in feature sparsity. To mitigate this issue, we introduce a stabilized truncated stochastic gradient descent algorithm. We employ a soft-thresholding scheme on the weight vector where the imposed shrinkage is adaptive to the amount of information available in each feature. The variability in the resulted sparse weight vector is further controlled by stability selection integrated with the informative truncation. To facilitate better convergence, we adopt an annealing strategy on the truncation rate, which leads to a balanced trade-off between exploration and exploitation in learning a sparse weight vector. Numerical experiments show that our algorithm compares favorably with the original truncated gradient SGD in terms of prediction accuracy, achieving both better sparsity and stability.

Keywords: sparse online learning, sparse features, truncated gradient, stability selection, adaptive shrinkage

1. Introduction

Modern data sets pose many challenges for existing learning algorithms due to their unprecedented large scales in both sample sizes and input dimensions. It demands both efficient processing of massive data and effective extraction of crucial information from an enormous pool of heterogeneous features. In response to these challenges, a promising approach is to exploit online learning methodologies that performs incremental learning over the training samples in a sequential manner. In an online learning algorithm, one sample instance is processed at a time to obtain a simple update, and the process is repeated via multiple passes over the entire training set. In comparison with batch learning algorithms in which all sample points are scrutinized at every single step, online learning algorithms have been shown to be more efficient and scalable for data of large size that cannot fit into the limited memory of a single computer. As a result, online learning algorithms have been widely adopted for solving large-scale machine learning tasks (e.g., Bottou, 1998).

©2017 Yuting Ma and Tian Zheng.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-190.html.

In this paper, we focus on first-order subgradient-based online learning algorithms, which have been studied extensively in the literature for dense data.[1] Among these algorithms, popular methods include the Stochastic Gradient Descent (SGD) algorithm in (Zhang, 2004) and (Bottou, 2010), the mirror descent algorithm introduced by Beck and Teboulle (2003) and the dual averaging algorithm by Nesterov (2009). Since these methods only require the computation of a (sub)gradient for each incoming sample, they can be scaled efficiently to high-dimensional inputs by taking advantage of the finiteness of the training sample. In particular, the stochastic gradient descent algorithm is the most commonly used algorithm in the literature of subgradient-based online learning. It enjoys an exceptionally low computational complexity while attaining steady convergence under mild conditions as discussed in (Bottou, 1998), even for cases where the loss function is not everywhere differentiable.

Despite of their computational efficiency, online learning algorithms without further constraint on the parameter space suffers the "curse of dimensionality" to the same extent as their non-online counterparts. Embedded in a dense high-dimensional parameter space, not only does the resulted model lack interpretability, its variance is also inflated. As a solution, *sparse online learning* was introduced to induce sparsity in the parameter space under the online learning framework (e.g., Langford et al., 2009). It aims at learning a linear classifier with a sparse weight vector, which has been an active topic in this area. For most efforts in the literature, sparsity is introduced by applying L_1 regularization on a loss function, such as in (Shalev-Shwartz and Tewari, 2011), in the same fashion as in the classific LASSO method by Tibshirani (1996). For example, Duchi and Singer (2009) extend the framework of Forward-Backward splitting method introduced by Lions and Mercier (1979) which alternates between an unconstrained truncation step on the sample gradient and an optimization step on the loss function with a penalty on the distance from the truncated weight vector.Langford et al. (2009) and Carpenter (2008) both explore the idea of imposing a *soft-threshold* on the weight vector $\mathbf{w} \in \mathbb{R}^p$ updated by the stochastic gradient descent algorithm:

$$w_j = \text{sign}(w_j)\max(|w_j| - \lambda, 0), \quad j = 1, \ldots, p.$$

This class of methods is known as the *truncated gradient algorithm*. For every K standard SGD updates, the weight vector is shrunk by a fixed amount to induce sparsity. In (Duchi et al., 2010), the same strategy has also been combined with a variant of the mirror descent algorithm in (Beck and Teboulle, 2003). Wang et al. (2015) further extends the truncated gradient framework to adjust for cost-effectiveness. This simple yet efficient method of truncated gradients particularly motivates the algorithm proposed in this paper. Strategies different from the truncation-based algorithm have also been proposed. For example, Xiao (2009) proposed the Regularized Dual-Averaging (RDA) algorithm which builds upon the primal-dual subgradient method by Nesterov (2009). The RDA algorithm learns a sparse weight vector by solving an optimization problem using the running average over all preceding gradients, instead of a single gradient at each iteration.

Closely related to sparse online learning is another area of active research, *online feature selection*. Instead of enforcing just a shrinkage on the weight vectors via L_1 regularization, online feature selection algorithms explicitly invoke feature selection by imposing a hard L_0 constraint on the weight vector, (e.g., Wang et al., 2014; Wu et al., 2014). In other words, online feature selection algorithms focus on generating a resulted weight vector that has a high sparsity level by directly

1. *Dense data* is defined as a data set in which the number of nonzero entries in all columns of its design matrix are in the order of $O(n)$ while the ones of *sparse data* are in the order of $O(\log(n))$ or less.

shrinking a large proportion of the weights directly to zero (also referred to as a *hard thresholding*). In practice, L_0 regularization is computationally expensive to solve due to its non-differentiability. The set of selected features also suffers from high variability as the decisions of hard-thresholding are based on single random samples in an online learning setting. Therefore, important features can be discarded simply owing to random perturbations.

Most recent subgradient-based online learning algorithms do not consider potential structures or heterogeneity in the input features. As pointed out by Duchi et al. (2011), current methods largely follow a predetermined procedural scheme that is oblivious to the characteristics of data being used at each iteration. In large-scale applications, a common and important structure is heterogeneity in sparsity levels of the input features, i.e., the variability in the number of nonzero entries among features. For instance, consider the bag-of-word features in text mining applications.[2] For a learning task, the importance of a feature is not necessarily associated with the frequencies of its values. In genetics, for example in (Morris and Zeggini, 2010), rare variants ($\leq 1\%$ in the population) have been found to be associated with disease risks. Both dense and sparse features may contain important information for the learning task. However, in the presence of heterogeneity in sparsity levels, using a simple L_1 regularization in an online setting will predispose rare features to be truncated more than necessary. The resulted sparse weight vectors usually exhibit high variance in terms of both weight values and the membership in the set of features with nonzero weights. As a result, the convergence of the standard truncation-based framework may also be hampered by this high variability. When the amount of information is scarce due to sparsity at each iteration, the convergence of the weight vector would understandably take a large number of iterations to approach the optimum. In two recent papers, Oiwa et al. (2011) and Oiwa et al. (2012) tackle this problem via L_1 penalty weighted by the accumulated norm of subgradients for extending several basic frameworks in sparse online learning. Their results suggest that, by acknowledging the sparsity structure in the features, both prediction accuracy and sparsity are improved over the original algorithms while maintaining the same convergence rate. However, their resulted weight vectors are unstable as the imposed subgradient-based regularization are excessively noisy due to the randomness of incoming samples in online learning. The membership in the set of selected features with nonzero weights is also very sensitive to the orderings of the training samples.

In this paper, we propose a *stabilized truncated stochastic gradient descent* algorithm for high-dimensional sparse data. The learning framework is motivated by that of the Truncated Gradient algorithm proposed by Langford et al. (2009). To deal with the aforementioned issues with sparse online learning methods applied to high-dimensional sparse data, we introduce three innovative components to reduce variability in the learned weight vector and stabilize the selected features. First, when applying the soft-thresholding, instead of a uniform truncation on all features, we perform only *informative truncations*, based on actual information from individual features during the preceding computation window of K updates. By doing so, we reduce the heterogeneous truncation bias associated with feature sparsity. The key idea here is to ensure that each truncation for each feature is based on sufficient information, and the amount of shrinkage is adjusted for the information available on each feature. Second, beyond the soft-thresholding corresponding to the ordinary L_1 regularization, the resulted weight vector is *stabilized* by staged purges of irrelevant features permanently from the active set of features. Here, *irrelevant features* are defined as features whose weights have been repeatedly truncated. Motivated by *stability selection* introduced

2. Here, by *sparse features*, we refer to features for which most samples assume a constant value (e.g., 0) and a few samples take on other values. Without loss of generality, we assume the majority constant is 0 throughout this paper.

by Meinshausen and Bühlmann (2010), these permanent purges prevent irrelevant features from oscillating between the active and non-active set of features, The "*purging*" process also resembles hard-thresholding in online feature selection and results in a stabler sparse solution than other sparse online learning algorithms. Results on the theoretical regret bound (See Section 4) show that this stabilization step helps improve over the original truncated gradient algorithm, especially when the target weight vector is notably sparse. To attune the proposed learning algorithm to the sparsity of the remaining active features, the third component of our algorithm is adjusting the amount of shrinkage progressively instead of fixing it at a predetermined value across all stages of the learning process. A novel hyperparameter, *rejection rate*, is introduced to balance between *exploration* of different sparse combinations of features at the beginning and the *exploitation* of the selected features to construct accurate estimate at a later stage. Our method gradually anneal the rejection rate to acquire the necessary amount of shrinkage on the fly for achieving the desired balance.

The rest of paper is organized as follows. Section 2 reviews the Truncated Gradient algorithm based on Stochastic Gradient Descent (SGD) framework for sparse learning proposed by Langford et al. (2009). In Section 3, we introduce, in details, the three novel components of our proposed algorithm. Theoretical analysis of the expected online regret bound is given in Section 4, along with the computational complexity. Section 5 gives practical remarks for efficient implementation. In Section 6, we evaluate the performance of the proposed algorithm on several real-world high-dimensional data sets with varying sparsity levels. We illustrate that the proposed method leads to improved stability and prediction performance for both sparse and dense data, with the most improvement observed in data with the highest average sparsity level. Section 7 concludes with further discussion on the proposed algorithm.

2. Truncated Stochastic Gradient Descent for Sparse Learning

Assume that we have a set of training data $\mathcal{D} = \{z_i = (\mathbf{x}_i, y_i), i = 1, \ldots, n\}$, where the feature vector $\mathbf{x}_i \in \mathbb{R}^p$ and the scalar output $y_i \in \mathbb{R}$. In the following, we use $\mathbf{x}_i$ to represent the vector of the ith sample of length p and $\mathbf{x}_{\cdot j}$ for the jth feature vector of all samples of length n. In this paper, we are interested in the case that both p and n are large and the feature vectors $\mathbf{x}_{\cdot j}$'s, $j = 1, \ldots, p$, are sparse. We consider a loss function $l(\hat{y}, y)$ that measures the cost of predicting $\hat{y}$ when the truth is y. The prediction $\hat{y}$ is given by function $f_{\mathbf{w}}(\mathbf{x})$ from a family $\mathcal{F}$ parametrized by a weight vector $\mathbf{w}$. Denote $L(\mathbf{w}, \mathbf{z}) \stackrel{\text{def}}{=\!=} l(f_{\mathbf{w}}(\mathbf{x}), y)$. The learning goal is to obtain an optimal weight vector $\hat{\mathbf{w}}$ that minimize the loss function $\sum_{i=1}^{n} L(\mathbf{w}, z_i)$ over the training data, with sparsity in the weight vector induced by a regularization term $\Psi(\mathbf{w})$. We can then formulate the learning task as a regularized minimization problem:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathbb{R}^p} \sum_{i=1}^{n} L(\mathbf{w}, z_i) + \Psi(\mathbf{w}). \tag{1}$$

The above optimization problem is often solved using some version of *gradient descent*. When both p and n are large, the computation becomes very demanding. To address this computational complexity, the Stochastic Gradient Descent (SGD) algorithm was proposed as a stochastic approximation of the full gradient algorithm. Instead of computing the gradient over the entire training set as under the batch setting, the stochastic gradient descent algorithm uses approximate gradients

based on subsets of the training data. This is particularly attractive to large scale problems as it leads to a substantial reduction in computing complexity and potentially distributed implementation.

For applications with large data sets or streaming data feeds, SGD has also been used as a subgradient-based *online learning* method. Cesa-Bianchi et al. (2004) argue that online learning and stochastic optimization are closely related and interchangeable most of the time. For simplicity, in the following, we focus our discussion and algorithmic description under the online learning framework with regret bound models. Nonetheless, our results can be readily generalized to stochastic optimization as well.

In online learning, the algorithm receives a training sample $z_t = (\mathbf{x}_t, y_t)$ at a time from a continuous feed. Without sparsity regularization, at time t, the weight vector is updated in an online fashion with a single training sample $z_t \in \mathcal{D}$ drawn randomly

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta L'(\mathbf{w}_{t-1}, z_t), \tag{2}$$

where $\eta > 0$ is the learning rate and $L'(\mathbf{w}_t, z_t) \in \partial_{\mathbf{w}_t} L(\mathbf{w}_t, z_t)$ is a subgradient of the loss function $L(\mathbf{w}_t, z_t)$ with respect to $\mathbf{w}_t$. The set of subgradients of f at the point x is called the subdifferential of f at x, and is denoted $\partial f(x)$. A function f is called subdifferentiable if it is subdifferentiable at all $x \in \mathbf{dom}\ f$. When $L(\mathbf{w}, \cdot)$ is differentiable at $\mathbf{w}$, $\partial_{\mathbf{w}} L(\mathbf{w}, \cdot) = \{\nabla_{\mathbf{w}} L(\mathbf{w}, \cdot)\}$. At the same time, a sequence of decisions $\mathbf{w}_t$ is generated at $t = 1, 2, \ldots,$ that encounters a loss $L(\mathbf{w}_t, z_t)$ respectively.

The goal of online learning algorithm with sparsity regularization is to achieve *low* regret with respect to a fixed optimal weight vector $\mathbf{w}^* \in \mathcal{W} \subset \mathbb{R}^p$. Here, $\mathcal{W}$ is the parameter space for sparse weights vectors (see Assumption 3 on page 15 for more details.) The regret is defined as:

$$R_T(\mathbf{w}^*) \triangleq \sum_{t=1}^{T} \left(L(\mathbf{w}_t, z_t) + \Psi(\mathbf{w}_t)\right) - \sum_{t=1}^{T} \left(L(\mathbf{w}^*, z_t) + \Psi(\mathbf{w}^*)\right). \tag{3}$$

In this paper, we focus on the L_1 regularization where $\Psi(\mathbf{w}) = g||\mathbf{w}||_1$ and g is the regularizing parameter. When adopted in an online learning framework, standard SGD algorithm does not work well in addressing (1) with L_1 penalty. Firstly, a simple online update requires the projection of the weight vector $\mathbf{w}$ onto a L_1-ball at each step, which is computationally expensive with a large number of features. Secondly, with noisy approximate subgradient computed using a single sample, the weights can easily deviate from zero due to the random fluctuations in z_t's. Such a scheme is therefore inefficient to maintain a sufficiently sparse weight vector.

To address this issue, Langford et al. (2009) induced sparsity in $\mathbf{w}$ by subjecting the stochastic gradient descent algorithm to soft-thresholding. For every $K \in \mathbb{N}^+$ iterations at step t, each of which is as defined in (2), the weight vector is shrunk by a soft-threshold operator T with a *gravity parameter* $\mathbf{g} \in \mathcal{G} \subset \mathbb{R}^p$ with $g_j \geq 0$ for $j = 1, \ldots, p$. For a vector $\mathbf{w} = [w_1, \ldots, w_p] \in \mathbb{R}^p$,

$$\hat{\mathbf{w}}_t = T(\mathbf{w}_t, \mathbf{g}), \tag{4}$$

where $T(\mathbf{w}, \mathbf{g}) = [T(w_1, g_1), \ldots, T(w_p, g_p)]$ with the operator T defined by

$$T(w_j, g_j) \triangleq \begin{cases} \max(w_j - g_j, 0), & \text{if } w_j > 0; \\ \min(w_j + g_j, 0), & \text{if } w_j \leq 0. \end{cases} \tag{5}$$

Algorithm 1 $B_0(\mathbf{w}_0, g_0)$: A burst of K updates with truncation using a universal gravity as in truncated SGD by Langford et al. (2009).

Input: $\mathbf{w}_0$ at initialization and the base gravity g_0.
Parameters: K, η.
for $t = 1$ **to** K **do**
  Draw $z_t \in \mathcal{D}$ uniformly at random.
  $\mathbf{w}_t = \mathbf{w}_{t-1} - \eta L'(\mathbf{w}_{t-1}, z_t)$, where $L'(\mathbf{w}_{t-1}, z_t) \in \partial_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}, z_t)$.
end for
$\hat{\mathbf{w}} = T(\mathbf{w}_K, g_0 K \mathbf{1}_p)$.
Return: $\hat{\mathbf{w}}$.

As one can see, the sequence of K SGD updates can be treated as a unit computational block, which will be referred to as a *burst* hereafter. Here the word *burst* indicates that it is a sequence of repetitive actions, e.g., the standard SGD updates as defined in (2), without interruption. Each burst is followed by a *soft-thresholding truncation* defined in (4), which puts a shrinkage on the learned weight vector.

A burst can be viewed as a base feature selection realized on a set of random samples with L_1 regularization as in the classical LASSO by Tibshirani (1996). Within a burst, let $\mathcal{X}_K$ be the set of K random samples on which the weight vector $\hat{\mathbf{w}}$ is stochastically learned. We define the set of features with nonzero weights in $\hat{\mathbf{w}}$ as its *active (feature) set*:

$$\hat{\mathcal{S}}^g(\hat{\mathbf{w}}; \mathcal{X}_K) = \{j : |\hat{w}_j| > 0\}, \tag{6}$$

with a corresponding gravity $\mathbf{g}$. The steps within a truncated burst are summarized in Algorithm 1.

In the truncated gradient algorithm, the gravity parameter is a constant across all dimensions as $\mathbf{g} = g_0 K \mathbf{1}_p$, where $g_0 \in \mathbb{R} \geq 0$ is a *base gravity* for each update in a burst and $\mathbf{1}_p \triangleq (1, \ldots, 1) \in \mathbb{R}^p$. In general, with greater parameter g_0 and smaller burst size K, more sparsity is attained. When $g_0 = 0$, the update in (4) becomes identical to the standard stochastic gradient descent update in (2). Langford et al. (2009) showed that this updating process can be regarded as an online counterpart of L_1 regularization in the sense that it approximately solves (1) in the limit as $K \to \infty$ and $\eta \to 0$.

3. Stabilized Truncated SGD for Sparse Learning

Truncated SGD works well for dense data. When it comes to high-dimensional *sparse* inputs, however, it suffers from a number of issues. Shalev-Shwartz and Tewari (2011) observe that the truncated gradient algorithm is incapable of maintaining sparsity of the weight vector as it iterates. Recall that, under the online learning setting, the weight vector $\mathbf{w}$ is updated with a noisy approximation of the true expected gradient using one sample at a time, from a random ordering of the data. With sparse inputs, it is highly probable that an important feature does not have a nonzero entry for many consequent samples, and is meaningfully updated for only a few times out of the K updates in a burst. As a result, it would be truncated after a few iterations and brought back to nonzero after another few updates. At the same time, sparsity in inputs will also give rise to sporadic large nonzero updates for irrelevant features, which cannot be fully resolved by the soft-threshold operator. The derived weight vector $\mathbf{w}$'s are of high variance, inadequate sparsity and poor generalizability. As an example, the number of nonzero variables in the weight vector during the last

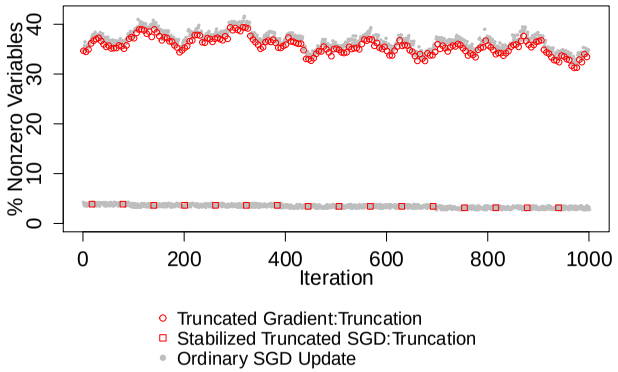


Figure 1: An example of the truncated SGD algorithm by Langford et al. (2009) and the proposed stabilized truncated SGD algorithm applied to a high-dimensional sparse data set. Here we compare the percentage of nonzero variables in the resulted weight vector at each iteration during the last 1000 iterations of both algorithms. The underlying data set is the text mining data set, *Dexter*, with $10,000$ features and 0.48% of sparsity, which is described in details in Section 6.

1000 stochastic updates from the truncated gradient algorithm implemented on a high-dimensional sparse data set are shown in Figure 1[3]. It can be seen that the numbers of nonzero features in the weight vectors learned by the truncated SGD algorithm ($K = 5$) remain large and highly unstable throughout these 1000 iterations, oscillating within 10% of the total number of features. As a comparison, also in Figure 1, we plot the results from our proposed stabilized truncated SGD applied to the same data. During these last 1000 updates, the proposed algorithm is using a less frequent truncation schedule due to our *annealed reject rate*. It attains both high sparsity in the weight vector and high stability with high-dimensional sparse data.

In this section, we introduce the *stabilized truncated Stochastic Gradient Descent (SGD)* algorithm. It attains a truly sparse weight vector that is stable and gives generalizable performance. Our proposed method attunes to the sparsity of each feature and adopts *informative truncation*. The algorithm keeps track of whether individual features have had enough information to be confidently subject to soft-thresholding. Based on the truncation results, we systematically reduce the active feature set by permanently discarding features that are truncated to zero with high probability via *stability selection*. We further improve the efficiency of our algorithm by adapting gravity to the sparsity of the current active feature set as the algorithm proceeds.

3. The results are based on the Dexter text mining data set. Please see Section 6 for details.

3.1 Informative Truncation

In the truncated SGD algorithm, Langford et al. (2009) suggest a general guideline for determining gravity in the batch mode by scaling a base gravity g_0 by K, the number of updates, for a single truncation after a burst. A direct online adaptation of a L_1 regularization would shrinks the weight vector at every iteration. The above batch mode operation is to delay the shrinkage for K iterations so that the truncation is executed based on information collected from K random samples instead of from a single instance. This guideline implicitly assumes that the K SGD updates in a burst are equally informative, which is in general true for dense features. For sparse features, however, under the online learning setting, not every update is informative about every feature due to the scarcity of nonzero entries. The original uniform formula, $\mathbf{g} = Kg_0\mathbf{1}_p$, for gravity would then create an undesirable differential treatment for features with different levels of sparsity. With a relatively small K, it is very likely that a substantial proportion of features would have no non-zero values on a size-K subsample used in a particular burst. The weights for these features remain unchanged after K updates. Consequently, the set of sparse features run the risk of being truncated to zero based on very few informative updates. The truncation decision is therefore mostly determined by a feature's sparsity level, rather than its relevance to the class boundary.

To make the learning be informed of the heterogeneity in sparsity level among features, we introduce the *informative truncation* step, extended from the idea of base gradient used in Algorithm 1. Instead of applying a universal gravity proportional to K to all features, the amount of shrinkage is set proportional to the number of times that a feature is actually updated with nonzero values in the size-K subsample, i.e., the number of *informative updates*. Specifically, within each burst, the algorithm keeps a vector of *counters*, $\tilde{\mathbf{k}} \in \mathbb{R}^p$, of the numbers of informative updates for the features $\mathbf{x}_{\cdot,j}$, $j = 1, \ldots, p$. Let $g_0 \in \mathbb{R} \geq 0$ be the base gravity parameter that serves as the unit amount of shrinkage for each informative update on each feature. At the end of each burst, we shrink feature $\mathbf{x}_{\cdot,j}$ by $g_0\tilde{k}_j$. In other words, here we set $\mathbf{g} = g_0\tilde{\mathbf{k}}$. The computational steps for a burst with informative truncation in summarized in Algorithm 2.

Algorithm 2 $B_1(w_0, g_0)$: A burst of K updates with informative truncation.

Input: $\mathbf{w}_0$ at initialization and the base gravity g_0.
Initialization: $\tilde{\mathbf{k}} = \mathbf{0}_p \in \mathbb{R}^p$.
Parameters: K, η.
for $t = 1$ **to** K **do**
  Draw $z_t = (\mathbf{x}_t, y_t) \in \mathcal{D}$ uniformly at random.
  $\mathbf{w}_t = \mathbf{w}_{t-1} - \eta L'(\mathbf{w}_{t-1}, z_t)$, where $L'(\mathbf{w}_{t-1}, z_t) \in \partial_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}, z_t)$.
  $\tilde{\mathbf{k}} \leftarrow \tilde{\mathbf{k}} + I(|\mathbf{x}_t| > 0)$.
end for
$\hat{\mathbf{w}} = T(\mathbf{w}_K, g_0\tilde{\mathbf{k}})$.
Return: $\hat{\mathbf{w}}$, $\tilde{\mathbf{k}}$.

The proposed informative truncation scheme ensures that the decision of truncation is made based on sufficient and equivalent amount of evidence for evaluating each feature. A theoretical justification of how informative truncation helps improving truncation bias can be found in Lemma 3 (Section 4). This feature-specific gravity attunes to the sparsity structure incurred at each burst without *ad-hoc* adjustment. It also avoids data pre-processing for locating sparse entries, which can be computationally expensive and compromises the advantage of online computation. In comparison

to the truncated gradient algorithm that quickly shrinks many features to zero indiscriminately, informative truncation keeps sparse features until enough evaluation is conducted. In doing so, sparse yet important features will be retained. The proposed approach also reduce the variability in the resulted sparse weight vector during the training process. Duchi et al. (2011) use a similar strategy that allows the learning algorithm to adaptively adjust its learning rates for different features based on cumulative update history. They use the L_2 norm of accumulated gradients to regulate the learning rate. By adapting the gravity with the counter $\tilde{\mathbf{k}}$ within each burst, our proposed strategy here can be viewed as applying the L_0 norm to the accumulated gradients that is refreshed every K steps.

3.2 Stability Selection

Despite of its scalability, subgradient-based online learning algorithms commonly suffer from instability. It has been shown both theoretically and empirically that stochastic gradient descent algorithms are sensitive to random perturbations in training data as well as specifications of learning rate (e.g., Toulis et al., 2015; Hardt et al., 2015). This instability is particularly pronounced in sparse online learning with sparse data, as discussed in Section 1. Under an online learning setting, using random ordering of the training sample as inputs, the algorithm would produce distinct weight vectors and unstable memberships of the final active feature set. Moreover, there has been a lot of discussion on the link between the instability of an learning algorithm and its deteriorated generalizability in the literature (e.g., Bousquet and Elisseeff, 2002; Kutin and Niyogi, 2002; Rakhlin et al., 2005; Shalev-Shwartz et al., 2010).

To tackle this instability issue, in the proposed algorithm, we exploit the method of *stability selection* to improve its robustness to random perturbation in the training data. Stability selection by Meinshausen and Bühlmann (2010) does not launch a new feature selection method. Rather, its aim is to enhance and improve a sparse learning method via subsampling. The key idea of stability selection is similar to the generic bootstrap. It feeds the base feature selection procedure with multiple random subsamples to derive an empirical selection probability. Based on aggregated results from subsamples, a subset of features is selected with low variability across different subsamples. With proven consistency in variable selection, stability selection helps remove noisy irrelevant features and thus reduce the variability in learning a sparse weight vector.

Incorporating stability selection into our proposed framework, each truncated burst with gravity parameter $\mathbf{g}$ is treated as an individual sparse learning engine. It takes K random samples and carries out a feature selection to obtain a sparse weight vector. In the following, we define first the notion of *selection probability* for the stability selection step in our proposed algorithm.

Definition 1 (selection probability) *Let $\mathcal{X}_K$ be a random subsample of $\{1,\ldots,n\}$ of size K, drawn without placement. Parametrized by the gravity parameter $\mathbf{g}$, the probability of the feature $\mathbf{x}_{\cdot,j}$ being in the active set of a truncated burst that returns $\hat{\mathbf{w}}$ is*

$$\Pi_j^{\mathbf{g}} = P^*\left(j \in \hat{S}^{\mathbf{g}}(\hat{\mathbf{w}}; \mathcal{X}_K)\right) = \mathbb{E}_{\mathcal{D}}\left[I(|\hat{w}_j| > 0)\right],$$

where the probability P^* is with respect to the random subsampling of $\mathcal{X}_K$. Let $\Pi^{\mathbf{g}} = [\Pi_1^{\mathbf{g}}, \ldots, \Pi_p^{\mathbf{g}}]$.

For simplicity, we drop the superscript $\mathbf{g}$ of $\Pi^{\mathbf{g}}$ in later discussions. For the rest of the paper, the selection probability Π always refers to $\Pi^{\mathbf{g}}$ that corresponds to weight vector $\hat{\mathbf{w}}$ with gravity parameter $\mathbf{g}$.

Under unknown data distribution, the selection probabilities cannot be computed explicitly. Instead, they are estimated empirically. Since each truncation burst performs a screening on all features, the frequency of each feature being selected by a sequence of bursts can be used to derive an estimator of the selection probability. We denote a sequence of $n_K > 0$ truncated bursts as a *stage*. A preliminary empirical estimate of the selection probability is given by

$$\hat{\Pi}_j = \begin{cases} \frac{\sum_{\tau:\tilde{k}_{j,\tau}>0} I(|\hat{\mathbf{w}}_{j,\tau}|>0)}{\sum_{\tau=1}^{n_K} I(\tilde{k}_{j,\tau}>0)}, & \text{for } j \text{ s.t. } \sum_{\tau=1}^{n_K} \tilde{k}_{j,\tau} > 0; \\ 1, & \text{otherwise,} \end{cases} \tag{7}$$

where $\tilde{\mathbf{k}}_\tau$ are the counters of informative updates for burst τ, $\tau = 1,\ldots,n_K$.

Different from the conventional stability selection setting, $\hat{\mathbf{w}}_\tau$'s are obtained sequentially and thus are dependent with each other. When n_K is small, different subsamples produce selection probability estimates using (7) exhibit high variability, even when initialized with the same weight vector at $\tau = 1$. On the other hand, a large value of n_K requires a prohibitively large number of iterations for convergence. To resolve the issues of estimating selection probability using a single sequence of SGD updates, we introduce a multi-thread framework of updating paths. Multiple threads of sequential SGD updates are executed in a distributed fashion, which readily utilizes modern multi-core computer architecture. With M processors, we initialize the algorithm on each path of SGD updates with a random permutation of the training data, $\mathcal{D}$, denoted as $\mathcal{D}^{(1)},\ldots,\mathcal{D}^{(m)}$. Then independently, M stages of bursts run in parallel along M paths, which return with $\hat{\mathbf{w}}_\tau^{(m)}$, $\tau = 1,\ldots,n_K$, $m = 1,\ldots,M$. The joint estimate of selection probability with gravity $\mathbf{g}$ is obtained as

$$\hat{\Pi}_j = \begin{cases} \frac{\sum_{m=1}^{M}\sum_{\tau:\tilde{k}_{j,\tau}^{(m)}>0} I(|\hat{\mathbf{w}}_{j,\tau}^{(m)}|>0)}{\sum_{m=1}^{M}\sum_{\tau=1}^{n_K} I(\tilde{k}_{j,\tau}^{(m)}>0)}, & \text{for } j \text{ s.t. } \sum_{m=1}^{M}\sum_{\tau=1}^{n_K} \tilde{k}_{j,\tau}^{(m)} > 0; \\ 1, & \text{otherwise.} \end{cases} \tag{8}$$

When more processors are available, a smaller n_K is required for the algorithm to obtain a stable estimate of selection probability. The dependence among $\hat{\mathbf{w}}_\tau$'s is also attenuated when M random subsets of samples are used for the estimation. This strategy falls under parallelized stochastic gradient descent methods, which is discussed in detail by Zinkevich et al. (2010).

Under the framework of stability selection, each stage on every path uses a random subsample. The estimated selection probability quantifies the chance that a feature is found to have high relevance to class differences given a random subsample. At the end of each stage, *stable features* are identified as those that belong to a large fraction of active sets incurred during this stage of bursts.

Definition 2 (Stable Features) *For a purging threshold $\pi_0 \in [0,1]$, the set of stable features with gravity parameter $\mathbf{g}$ is defined as*

$$\hat{\Omega}^{\mathbf{g}} = \{j : \Pi_j^{\mathbf{g}} \geq \pi_0\}. \tag{9}$$

For simplicity, we write the stable set $\hat{\Omega}^{\mathbf{g}}$ as $\hat{\Omega}$ when there is no ambiguity.

Stability selection retains features that have high selection probabilities and discard those with low selection probabilities. At the end of a stage of M paths, we *purge* the features that are not in

Algorithm 3 $B_2(\mathbf{w}_0, g_0, \hat{\Omega}, \mathcal{D}^{(m)})$: Informative truncated burst with stability selection in thread m.

Input: $\mathbf{w}_0$, g_0, the input data $\mathcal{D}^{(m)}$ and the current set of stable features $\hat{\Omega}$, which is the output of equation (9) using (8) with predetermined threshold π_0.
Parameters: K, η.
Initialize $\mathbf{v}_0 = (w_{0,j})_{j\in\hat{\Omega}}$, $\tilde{\mathbf{k}} = \mathbf{0}_{|\hat{\Omega}|}$.
for $t = 1$ **to** K **do**
  Draw $z_t = (\mathbf{x}_t, y_t)$ sequentially from $\mathcal{D}^{(m)}$.
  $\tilde{z}_t = (z_{t,j})_{j\in\hat{\Omega}}$.
  $\mathbf{v}_t = \mathbf{v}_{t-1} - \eta L'(\mathbf{v}_{t-1}, \tilde{z}_t)$, , where $L'(\mathbf{v}_{t-1}, z_t) \in \partial_{\mathbf{v}_{t-1}} L(\mathbf{v}_{t-1}, z_t)$.
  $\tilde{\mathbf{k}} \leftarrow \tilde{\mathbf{k}} + I(|\mathbf{x}_t| > 0)$.
end for
$\hat{\mathbf{u}} = T(\mathbf{v}_K, g_0\tilde{\mathbf{k}})$.
$\hat{w}_j = \begin{cases} \hat{u}_{j'}, & \text{if } j \in \hat{\Omega} \text{ and } \hat{\Omega}_{j'} = j; \\ 0, & \text{if } j \notin \hat{\Omega}. \end{cases}$
Return: $\hat{\mathbf{w}}$, $\tilde{\mathbf{k}}$.

the set of stable features by permanently setting their corresponding weights to zero, and remove them from subsequent updates. We define the *stabilized* weight vector as

$$\tilde{\mathbf{w}} = \hat{\mathbf{w}} \cdot I_{\hat{\Omega}}. \tag{10}$$

As discussed above, due to the nature of online learning with sparse data, there are two undesirable learning setbacks in a single truncated burst. The first occurs when an important feature has its weight stuck at zero due to inadequate information in the subsample used, while the second case is when a noise feature's weight gets sporadic large updates by chance. Using informative bursts, we can avert the first type of setbacks and using selection probability based on multiple bursts, we can spot noisy features more easily. In the presence of a large number of noisy features, the learned weights for important features suffer from high variance. Via stability selection, we systematically remove noisy features permanently from the feature pool. Furthermore, the choice of a proper regularization parameter is crucial yet known to be difficult for sparse learning, especially due to the unknown noise level. Applying stability selection renders the algorithm less sensitive to choice of the base gravity parameter g_0 in learning a sparse weight vector via truncated gradient. As we will show using results from our numerical experiments, this purging by stability selection leads to a notable reduction in the estimation variance of the weight vector. Here, π_0 is a tuning parameter in practice. We have found that the learning results in the numerical experiments are not sensitive to different values of π_0 within a reasonable range. Under mild assumptions discussed in Section 4, we derive a lower bound of the expected improvement in convergence by employing stability selection in the learning process in Lemma 2.

3.3 Adaptive Gravity with Annealed Rejection Rate

The truncated SGD algorithm adopts a universal and fixed base gravity parameter at all truncations. As pointed out by Langford et al. (2009) , a large value of the base gravity g_0 achieves more sparsity but the accuracy is compromised, while a small value of g_0 leads to less sparse weight vector yet attaining better performance. In other words, different extents of shrinkage serve different

purposes of a learning algorithm. The needs for shrinkage also changes as the weight vector and the stable set evolves. Intuitively, the truncation is expected to be greedy at the beginning so that the number of nonzero feature can be quickly reduced for better computational efficiency and learning performance. As the algorithm proceeds, fewer features remain in the stable set. We should then be careful not to shrink important features with a truncation that is too harsh.

A large base gravity g_0 is effective in inducing sparsity at the beginning of the algorithm when the weight vector $\hat{\mathbf{w}}$ is dense. As the algorithm proceeds, the same value of gravity is likely to impose too much shrinkage when the learned weight vector $\hat{\mathbf{w}}$ becomes very sparse, exposing some truly important features at the risk of being purged. On the other hand, a small fixed gravity is over-conservative so that the algorithm will not shrink irrelevant features effectively, leading to slow convergence and a dense weight vector overridden by noise. Tuning a reasonable fixed base gravity parameter for a particular data set does not only creates additional computational burden, but also inadequate in addressing different learning needs during different stages of the algorithm.

As the role of *gravity* in a learning algorithm is to induce sparse estimates, in this paper, we propose an *adaptive gravity* scheme that delivers the right amount of shrinkage at each stage of the algorithm towards a desirable level of sparsity for the learned weight vector. We propose to control sparsity by a target *rejection rate* β, that is, the proportion of updates that are expected to be truncated. Guided by this target rejection rate, we derive the necessary shrinkage amount and the corresponding gravity. As we discussed in Section 3.1, a base gravity g_0 is used in our learning algorithms to create gravity values for individual features that are attuned to their data sparsity levels. Therefore our adaptive gravity scheme is carried out by adjusting g_0. At the beginning of a particular stage, we examine the truncations carried out during the previous stage. The base gravity g_0 is then adjusted to project the target rejection rate during the current stage. Specifically, at stage s, we look at the pooled set of non-truncated weight vectors and informative truncation counters $\left\{\mathbf{w}_\tau^{(m)}, \tilde{\mathbf{k}}_\tau^{(m)}, \tau = 1, \ldots, n_K, m = 1, \ldots, M\right\}$ from all the bursts conducted in the previous stages on multiple threads. The adaptive base gravity g_0 for a target rejection rate $\beta_s \in [0, 1]$ is then obtained as

$$g_{0,s}(\beta_s) \triangleq \sup\{g_0 \geq 0 : \hat{p}_s(g_0) \leq \beta_s\}. \tag{11}$$

Here $\hat{p}_s(g_0)$ is the empirical probability, i.e.,

$$\hat{p}_s(g_0) \triangleq \frac{\sum_{m=1}^{M}\sum_{\tau=1}^{n_k}\sum_{\{j:j\in\hat{\Omega}_s,\tilde{k}_{j,\tau}^{(m)}>0\}} I\left(\left|\frac{\Delta w_{j,\tau}^{(m)}}{\tilde{k}_{j,\tau}^{(m)}}\right| > g_0\right)}{\sum_{m=1}^{M}\sum_{\tau=1}^{n_k}\sum_{j\in\hat{\Omega}_s} I\left(\tilde{k}_{j,\tau}^{(m)} > 0\right)},$$

where $\Delta w_{j,\tau}^{(m)} \triangleq w_{j,\tau}^{(m)} - w_{j,\tau-1}^{(m)}$ is the amount of updates on feature $\mathbf{x}_{\cdot,j}$ during the τth burst.

We initialize the algorithm with a high rejection rate so that a large proportion of the weight vector can be reduced to zero at the end of each burst during the early stage of the algorithm. It allows the algorithm to explore as many sparse combination of features as possible at the early stage of the learning process. Along with the stability selection, the set of stable features can be quickly reduced to a manageable size by removing the majority of noises. When the weight vector becomes sparse, we decrease the rejection rate proportionally. With a lower rejection rate, and consequently a lower gravity, the algorithm can better exploit the subsequent standard SGD updates for a more accurate estimate of the true weight vector. As the rejection rate decreases to 0, the algorithm converges to the standard stochastic gradient descent algorithm on a small subset of stable features.

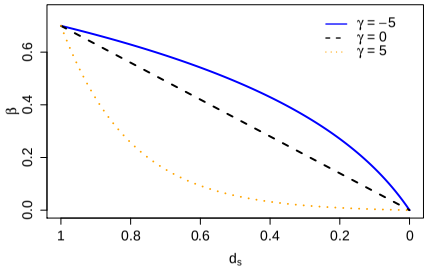


Figure 2: Examples of the rejection rate annealing function with different values of γ as defined in (12). Here $\gamma = -5, 0$, and 5 respectively. A positive annealing rate would reduce the rejection rate quickly as the proportion of non-zeros weight values, d_s, decreases, whereas a negative annealing rate would maintain it at a relatively high level.

To achieve the balance between exploration and exploitation, we construct an annealing function for the rejection rate that decreases monotonically as the level of sparsity decreases. Let $\beta_0 \in [0, 1]$ be the *maximum rejection rate* at initialization and let γ be the *annealing rate*. The annealing function ϕ for the rejection rate at stage $s + 1$ is given by

$$\begin{aligned} \beta_{s+1} &= \phi(d_s; \beta_0, \gamma) \\ &= \begin{cases} \beta_0 \left[\exp\left(-\gamma d_s\right) - d_s e^{-\gamma}\right] & \gamma \geq 0; \\ \beta_0 \frac{\log(1-\gamma(1-d_s))}{\log(1-\gamma)}, & \gamma < 0, \end{cases} \end{aligned} \tag{12}$$

where $d_s = \frac{|\hat{\Omega}_s|}{p}$ is the level of weight vector sparsity at the end of stage s. The greater the value γ is, the faster the rejection rate is annealed to zero as the number of stable features decreases.

A positive, zero and negative value of γ corresponds to exponential decay, linear decay and logarithmic decay of the rejection rate, respectively. Figure 2 presents examples of the rejection rate anneal function with $\gamma = -5, 0$, and 5 respectively.

By using adaptive gravity (11) with annealed rejection rate (12), the amount of shrinkage is adjusted to the current level of sparsity of the weight vector quantified by the size of the stable set $|\hat{\Omega}_s|$ or the L_0 norm of the purged $\tilde{\mathbf{w}}$. Instead of tuning a fixed gravity parameter as in (Langford et al., 2009), for our proposed algorithm, we tune the annealing rate γ and the maximum rejection rate β_0. Here γ balances the trade-off between exploration and exploitation and β_0 determines the initial intensity of truncation. It enables the tuning process to be tailored to the data at hand as well as being comparable across different data sets. In Section 6, the tuning results instantiate that a negative annealing rate is preferred for highly sparse data, such as the RCV1 data set, since the a high rejection rate needs to be maintained longer allowing sufficient information of sparse features to be evaluated by the learning process. On the other hand, a positive annealing rate is chosen for relatively dense data, such as the Arcene data set, where the high frequency of nonzero values

Algorithm 4 $\Psi(\mathcal{D})$: Stabilized truncated stochastic gradient descent for sparse learning.

Input: the training data $\mathcal{D}$.
Parameters: β_0, γ, π_0, K, n_K, M, η.
Initialization: For each $m \in \{1, \ldots, M\}$, initialize $\tilde{\mathbf{w}}_0^{(m)} = \mathbf{0}_p$, $\tilde{\mathbf{k}} = \mathbf{0}_p$ and $\mathcal{D}^{(m)}$ is a random permutation of $\mathcal{D}$; $\hat{\Omega}_0 = \{1, \ldots, p\}$.
For $s = 1, 2, \ldots,$
Set $\hat{\mathbf{w}}_0^{(m)} = \tilde{\mathbf{w}}_{s-1}^{(m)}$
repeat
  Obtain $g_{0,s}(\beta_s)$ as in (11).
  for all $m \in \{1, \ldots, M\}$ **parallel do**
    for $\tau = 1$ **to** n_K **do**
      $\hat{\mathbf{w}}_\tau^{(m)} = B_2\left(\hat{\mathbf{w}}_{\tau-1}^{(m)}, g_{0,s}(\beta_s), \hat{\Omega}_{s-1}, \mathcal{D}^{(m)}\right)$.
    end for
  end for
  Compute $\hat{\Pi}_s$ as in (8) and update the set of stable features $\hat{\Omega}_s = \left\{j : \hat{\Pi}_{s,j} \geq \pi_0\right\}$.
  $\tilde{\mathbf{w}}_s^{(m)} = \hat{\mathbf{w}}_{n_K}^{(m)} I_{\hat{\Omega}_s}$, $m = 1, \ldots, M$.
  $\beta_{s+1} = \phi(d_s; \beta_0, \gamma)$ where $d_s = \frac{|\hat{\Omega}_s|}{p}$.
  Aggregation: $\underline{\mathbf{w}} = \frac{1}{M} \sum_{m=1}^{M} \tilde{\mathbf{w}}_s^{(m)}$.
until $\underline{\mathbf{w}}$ converges.
Return: $\underline{\mathbf{w}}$.

permit fast reduction of the active set. The complete algorithm of the stabilized truncated stochastic gradient descent algorithm is summarized in Algorithm 4. .

4. Properties of the Stabilized Truncated Stochastic Gradient Descent Algorithm

The learning goal of sparse online learning is to achieve a low regret as defined in (3). In this section, we analyze the online regret bound of the proposed stabilized truncated SGD algorithm in Algorithm 4 with convex loss. For simplicity, the effect of adaptive gravity with annealed rejection rate is not considered here. To achieve viable result, we make the following assumptions.

Assumption 1 *The absolute values of the weight vector* $\mathbf{w}$ *are bounded above, that is,* $|w_j| \leq C$ *for some* $C \in [0, \infty)$, $j = 1, \ldots, p$.

Assumption 2 *The loss function* $L(\mathbf{w}, z)$ *is convex in* $\mathbf{w}$*, and there exist non-negative constants* A *and* B *such that, for all* $\mathbf{w} \in \mathbb{R}^p$ *and* $z \in \mathbb{R}^{p+1}$, $||\nabla_{\mathbf{w}} L(\mathbf{w}, z)||^2 \leq AL(\mathbf{w}, z) + B$.

For linear prediction problems, the class of loss function that satisfies Assumption 2 includes common loss functions used in machine learning problems, such as the L_2 loss, the hinge loss and the logistic loss, with the condition that $\sup_{\mathbf{x}} ||x|| \leq C_x$ for some constant $C_x > 0$.

Assumption 3 *Assume that the set* $\mathcal{W}$ *for identifying* $\mathbf{w}^*$ *has the following properties:*

1. *$\mathcal{W}$ is the parameter space for weight vectors $\mathbf{w}$ that is subject to a sparsity constraint*

$$||\mathbf{w}||_0 = d^*.$$

For the optimal weight vector $\mathbf{w}^*$, we denote $\Omega^* = \{j : |w_j^*| > 0\}$ and $d^* = |\Omega^*|$. We further assume that d^* is sufficiently small and the gravity parameter g associated with $\mathbf{w}^*$ is reasonably large so that, for $\tau = 1, 2, \ldots$, the average number of active selected features from each truncated bursts, $q^{\mathbf{g}}$, given a gravity $\mathbf{g}$, is greater than or equal to d^*.

2. *Assume features $\mathbf{x}_1, \ldots, \mathbf{x}_p$ has various sparsity distribution that $\Pr(|x_{i,j}| > 0) = \lambda_j$, where $\lambda_j \in [0, 1]$, for $i = 1, \ldots, n$, and $\mathbf{w}_j^* = 0$ if $\lambda_j = 0$, for $j = 1, \ldots, p$.*

Assumption 3 posits that $\mathbf{w}$'s parameter space of interest is substantially sparse, which is the main focus of sparse learning and of this paper. The theoretical analysis in the following concerns for an fixed optimal weight vector $\mathbf{w}^*$, if exists, under such a constraint. Nevertheless, this condition does not confine the applicable scenarios of the proposed method to a fixed subclass of problems. It suggests a balance between the model sparsity and the value of the gravity parameter that is implicitly embedded within the parameter tuning process.

Assumption 4 *Let $\Omega = \{1, \ldots, p\}$ and $N = \Omega \setminus \Omega^*$. Let $\hat{\Omega}^{\mathcal{G}} = \bigcup_{\mathbf{g} \in \mathcal{G}} \hat{\Omega}^{\mathbf{g} \in \mathcal{G}}$. Assume that the classifier $f_{\mathbf{w}}$ is not worse than random guessing. That is,*

$$\frac{\mathbb{E}(|\Omega^* \bigcap \hat{\Omega}^{\mathcal{G}}|)}{\mathbb{E}(|N \bigcap \hat{\Omega}^{\mathcal{G}}|)} \geq \frac{|\Omega^*|}{|N|}.$$

Lemma 1 (Error Control in Stability Selection (Meinshausen and Bühlmann, 2010)) *Let V be the set of falsely selected variables. Given that the distribution of $\left\{ I\left(j \in \hat{S}^g(\hat{\mathbf{w}}; \mathcal{X}_K \right),\ j \in 1, \ldots, p \right\}$ is exchangeable for all $g \in \mathcal{G}$ and if Assumption 4 holds, the expected cardinality of $|V|$ is then bounded for $\pi_0 \in \left(\frac{1}{2}, 1\right)$ by*

$$\mathbb{E}(|V|) \leq \frac{1}{2\pi_0 - 1} \frac{(q^{\mathbf{g}})^2}{p}. \tag{13}$$

Lemma 2 *Assume all assumptions above holds. Let $\hat{\mathbf{w}}$ be a non-stabilized dense weight vector, i.e., an output weight vector from Algorithm 3. Let $\tilde{\mathbf{w}}$ be the stabilized weight vector derived from $\hat{\mathbf{w}}$, which is purged by the stability selection* (10) *with a set of stable features $\hat{\Omega}$. Let $q^{\mathbf{g}}$ be the average number of nonzero entries in $\hat{\mathbf{w}}$'s of the previous truncated bursts, i.e.,the number of selected features from n_k truncated bursts, with gravity $\mathbf{g}$. Then, there exists an $\varepsilon \in \left(0, \frac{\pi_0 C(p - |\hat{\Omega}|)}{|\hat{\Omega}|}\right)$ with $\hat{S}_\varepsilon = \{j : \mathbb{E}(|\hat{w}_j|) > \varepsilon\}$ such that the bound on the expected difference between the distance from the non-stabilized weight vector to $\mathbf{w}^*$ and the distance from the stabilized weight vector to $\mathbf{w}^*$ is given by*

$$\mathbb{E}\left(||\hat{\mathbf{w}} - \mathbf{w}^*||^2 - ||\tilde{\mathbf{w}} - \mathbf{w}^*||^2\right) \geq \varepsilon^2(|\hat{S}_\varepsilon| - |\hat{\Omega}|) + 2\pi_0 C^2 \left[\left(1 - \frac{q^{\mathbf{g}}}{2\pi_0 p - p}\right) q^{\mathbf{g}} - d^* \right]. \tag{14}$$

When the purging threshold is sufficiently high such that $\pi_0 \in \left(\frac{1}{2} + \frac{(q^{\mathbf{g}})^2}{2p(q^{\mathbf{g}} - d^*)}, 1\right)$,

$$\mathbb{E}\left(||\hat{\mathbf{w}} - \mathbf{w}^*||^2 - ||\tilde{\mathbf{w}} - \mathbf{w}^*||^2\right) \geq 0.$$

Proof : See Appendix A. ■

Lemma 2 quantifies the gain of using stabilization when $\mathbf{w}^*$ is highly sparse, where stability selection efficiently shrinks high variable estimates to zero. When the purging threshold π_0 is sufficiently high such that $\pi_0 \in \left(\frac{1}{2} + \frac{(q^{\mathbf{g}})^2}{2p(q^{\mathbf{g}} - d^*)}, 1\right)$, the lower bound achieved by (14) is guaranteed to be positive. Furthermore, this result also indicates that the expected difference between distances from the non-stabilized and stabilized weight vector to the sparse $\mathbf{w}^*$ depends on the differences between the sizes of the temporary nonzero set of features before purging, $|\hat{S}_\varepsilon|$, and the size of the stable features after purging. In expectation, the stabilized weight vector is closer to the target sparse weight vector as the operation of purging efficiently reduces the size of stable features. This suggests a much faster convergence with stabilization. Lemma 2 also provides an insight on the benefit from using adaptive gravity with annealed rejection rate. At the beginning of the algorithm, the gap between the size of $|\hat{S}_\varepsilon|$ and the size of the set of stable features $|\hat{\Omega}|$ is large when aiming for extensive exploration of different sparse combination of features. Hence, the improvement brought by stabilization is more substantial during the early state of learning period. As the algorithm proceeds and the set of stable features becomes smaller and stabler, it dwindles the leeway that allows the aforementioned two sets to be different. Consequently, the proposed algorithm is gradually tuned toward the standard stochastic gradient descent algorithm to facilitate better convergence at the later period of the learning process.

Lemma 3 *Let $\mathbf{w}_0$ be the weight vector at initialization. After the first burst, let $\bar{\mathbf{w}}_1$ be the truncated weight vector using universal gravity $g_0 K$ as in Algorithm 1 and let $\hat{\mathbf{w}}_1$ be the truncated weight vector with informative truncation as in Algorithm 2. Then, under Assumption 3,*

$$\mathbb{E}\left(||\bar{\mathbf{w}}_1 - \mathbf{w}^*||^2 - ||\hat{\mathbf{w}}_1 - \mathbf{w}^*||^2\right) \geq 2g_0 \sum_{t=1}^{K} \|\zeta_t \mathbf{w}^*\|_1 \geq 0, \tag{15}$$

where $\zeta_{t,j} = I(|x_{t,j}| = 0)$ for $j = 1, \ldots, p$ and $\zeta_t = (\zeta_{t,1}, \ldots, \zeta_{t,p})^T$.

Proof : See Appendix B. ■

In Lemma 3, we compare the distances towards $\mathbf{w}^*$ from *1)* the weight vector with uniform gravity and *2)* the weight vector with informative truncation that depends on the number of zero entries occurred in a burst. Such a gap suggests the effectiveness of informative truncation on sparse data in which feature sparsity is highly heterogeneous. In the scenarios where very few nonzero entries appear in a burst, the informative truncation imposes gravity that is proportional to the information presented in a burst. It is a fairer treatment than uniform truncation and leads to a large improvement in expectation. When features are all considerably dense in a burst, the informative truncation is equivalent to the uniform truncation.

In short, Lemma 2 demonstrates the improvement in expected squared error due to stabilization on the weight vector. Lemma 3, on the other hand, quantifies the improvements in reduce truncation bias when implementing informative truncation on sparse features with heterogeneous sparsity levels.

Given Lemma 2 and Lemma 3, we have the expected regret bound of the proposed Algorithm 4 in Theorem 1.

Theorem 1 *Assume all assumptions above holds. Consider the updating rules for the weight vector in Algorithm 3. On an arbitrary path, with* $\underline{\mathbf{w}}_0 = 0$ *and* $\eta > 0$*, let* $\{\underline{\mathbf{w}}_t\}_{t=1}^T$ *be the resulted weight vector and* $\left\{\underline{g}_t\right\}_{t=1}^T$ *be the gravity values applied to the weight vectors generated by Algorithm 4, along with the base gravity parameters* $\left\{\underline{g}_{0,t}\right\}_{t=1}^T$*. Set the purging threshold* π_0 *to be sufficiently large such that* $\pi_0 \in \left(\frac{1}{2} + \frac{(q^{\mathbf{g}})^2}{2p(q^{\mathbf{g}}-d^*)}, 1\right)$*. There exists a sequence of* $\varepsilon_t \in \left(0, \frac{\pi_0 C(p-|\hat{\Omega}_t|)}{|\hat{\Omega}_t|}\right)$ *at each stability selection with the set of stable features* $\hat{\Omega}_t$ *such that the expectation of the regret defined in* (3) *is bounded above by*

$$\begin{aligned}
&\mathbb{E}\left(\sum_{t=1}^T \left[L(\underline{\mathbf{w}}_t, z_t) + K\underline{g}_t||\underline{\mathbf{w}}_t||_1\right] - \sum_{t=1}^T \left[L(\mathbf{w}^*, z_t) + K\underline{g}_t||\mathbf{w}^*||_1\right]\right) \\
\leq &\frac{\eta A}{2-\eta A}\left(\mathbb{E}\left[\sum_{t=1}^T L(\mathbf{w}^*, z_t) + K\underline{g}_{0,t}(||\mathbf{w}^*||_1 - ||\underline{\mathbf{w}}_t||_1)\right]\right) + \frac{1}{2-\eta A}\left(\eta TB + \frac{1}{\eta}||\mathbf{w}^*||^2\right) \\
&- \frac{1}{2\eta - \eta^2 A}\sum_{t=1}^T \varepsilon_t^2 I\left(\frac{t}{Kn_K} \in \mathbb{Z}\right)(|\hat{S}_{\varepsilon_t,t}| - |\hat{\Omega}_t|), \qquad (16)
\end{aligned}$$

where $\hat{S}_{\varepsilon,t} = \{j : \mathbb{E}(|\hat{w}_{t,j}|) > \varepsilon_t\}$ *and* $\hat{\mathbf{w}}_t$ *is the weight vector at time* t *before stabilization.*

Proof : See Appendix C. ■

In the result of Theorem 1, the first two parts of the right-hand-side of the expected regret bound (16) is similar to the bound obtained in (Langford et al., 2009). It implies the trade-off between attained sparsity in the resulted weight vector and the regret performance. When the applied gravity is small under the joint effect of the base gravity g_0 and the size of each burst K, the sparsity is less but the expected regret bound is lower. On the other hand, when the applied gravity is large, the resulted weight vector is more sparse but at the risk of higher regret. Based on Lemma 2, the proposed algorithm is guaranteed to achieve lower regret bound in expectation when the target sparse weight vector is highly sparse. As quantified in the third term of the right-hand-side of (16), the improvement comes from the reduction of the active set at each purging. By its virtue, noisy features are removed from the set of stable features and thus are absent in later SGD updates and truncations.

Theorem 1 is stated with a constant learning rate η. It is possible to obtain a lower regret bound in expectation with adaptive learning rate η_t decaying with t, such as $\eta_t = \frac{1}{\sqrt{t}}$, which is commonly used in the literature of online learning and stochastic optimization. However, the discussion of using an varying learning rate is not a main focus of this paper and adds extra complexity of the analysis. Without knowing T in advance, this may lead to a no-regret bound as suggested in (Langford et al., 2009). Instead, in Corollary 1, we show that the convergence rate of the proposed algorithm is $O(\sqrt{T})$ with $\eta = \frac{1}{\sqrt{T}}$.

Corollary 1 *Assume that all conditions of Theorem 1 are satisfied. Let the learning rate* η *be* $\frac{1}{\sqrt{T}}$*. The upper bound of the expected regret is*

$$\mathbb{E}\left(\sum_{t=1}^T \left[L(\underline{\mathbf{w}}_t, z_t) + \underline{g}_t||\underline{\mathbf{w}}_t||_1\right] - \sum_{t=1}^T \left[L(\mathbf{w}^*, z_t) + \underline{g}_t||\mathbf{w}^*||_1\right]\right) \leq O(\sqrt{T}),$$

where $\underline{g}_t = K\underline{g}_{0,t}$*.*

Proof By plugging in $\eta = \frac{1}{\sqrt{T}}$ to the result from Theorem 1, we get

$$\begin{aligned}
&\mathbb{E}\left(\sum_{t=1}^T \left[L(\underline{\mathbf{w}}_t, z_t) + \underline{g}_t||\underline{\mathbf{w}}_t||_1\right] - \sum_{t=1}^T \left[L(\mathbf{w}^*, z_t) + \underline{g}_t||\mathbf{w}^*||_1\right]\right) \\
\leq &\frac{A}{2\sqrt{T} - A}\left(\mathbb{E}\left[\sum_{t=1}^T L(\mathbf{w}^*, z_t) + \underline{g}_t(||\mathbf{w}^*||_1 - ||\underline{\mathbf{w}}_t||_1\right]\right) \\
&+ \frac{T}{2\sqrt{T} - A}\left(\eta TB + \frac{1}{\eta}||\mathbf{w}^*||^2\right) - \frac{T}{2\sqrt{T} - A}\sum_{t=1}^T \varepsilon_t^2 I(\frac{t}{Kn_K} \in \mathbb{Z})(|\hat{S}_{\varepsilon,t}| - |\hat{\Omega}_t|).
\end{aligned}$$

The result is then straightforward. ■

Assume that the input features have d nonzero entries on average. With linear prediction model $f_{\mathbf{w}}(x) = \mathbf{w}^T x$, the computational complexity at each iteration is $O(d)$. Leveraging the sparse structure, the informative truncation only requires an additional $O(Kd)$ space for recording the counters. The purging process of stability selection consumes $O(\delta)$, $\delta = Kn_K Md$, space for storing the generated intermediate weight vectors and $O(\delta \log(\delta))$ computational complexity. Both storage and computational cost decrease when the set of stable features diminishes as the algorithm proceeds. Since the parameters K, n_K, and M is normally set to be small values, the complexity mostly depends on $O(d \log(d))$. In summary, the proposed algorithm scales with the number of nonzero entries instead of the total dimensions, making it appealing to high-dimensional applications.

5. Practical Remarks

When implementing Algorithm 4 in practice, the performance can be further improved in terms of both accuracy and computational efficiency by employing a couple of practical techniques. It includes applying *informative purging* and attenuating the truncation frequency to achieve more accurate sparse learning and steadier convergence.

The first improvement can be implemented by better addressing the issue of scarcity of incoming samples. For computing selection probabilities, instead of using only information from the current stage, we can inherit information from previous stages for features that are too scarce to accumulate enough updates during one stage. Specifically, we introduce an *accumulated counter* κ_s at stage s as the total number of times that a feature is updated within a burst during this stage:

$$\kappa_{j,s} = \sum_{m=1}^M \sum_{\tau=1}^{n_K} \tilde{k}_{j,\tau}^{(m)}, \quad j = 1, \ldots, p,$$

which is essentially the denominator of the selection probability in (8). Similarly, we define an *accumulated truncation indicator* $\mathbf{b}_s$ at stage s as the total number of times that a feature is truncated to zero given valid update(s):

$$\mathbf{b}_{j,s} = \sum_{m=1}^{M} \sum_{\tau:\tilde{k}_{j,\tau}^{(m)}>0} I(|\hat{\mathbf{w}}_{j,\tau}^{(m)}| > 0), \quad j = 1, \ldots, p.$$

A feature is then evaluated in the stability selection *only* if there are enough updates from the present stage and from any unused information carried over from previous stages. Given a threshold $\delta_K \geq 0$, let $\tilde{\kappa}_{j,s} \triangleq \tilde{\kappa}_{j,s-1} I(\tilde{\kappa}_{j,s-1} < \delta_K) + \kappa_{j,s}$ and $\tilde{\mathbf{b}}_{j,s} \triangleq \tilde{\mathbf{b}}_{j,s-1} I(\tilde{\kappa}_{j,s-1} < \delta_K) + \mathbf{b}_{j,s}$. The selection probability is modified as

$$\hat{\Pi}_{j,s} = \begin{cases} \frac{\tilde{\mathbf{b}}_{j,s}}{\tilde{\kappa}_s}, & \text{for } j \text{ s.t.} \tilde{\kappa}_s > \delta_K; \\ 1, & \text{otherwise,} \end{cases} \qquad \text{for } j = 1, \ldots, p. \tag{17}$$

This strategy extends the key idea in Section 3.1 that, with sparse data, each decision need to be based on sufficient evidence. Using the "carried-over" information allows the algorithm to utilize information available in a sequence of SGD updates while attuned to the needs of features with different levels of sparsity. In practice, this modification facilitates faster convergence especially for ultra-sparse data.

The second practical strategy is that the size of each burst, K, can be adaptively adjusted in a similar fashion as the rejection rate β in (12). At the end of each stage, the burst size K_s is updated as

$$K_s = \left\lceil K_0 \log\left(\frac{1}{\alpha d_{s-1}}\right)\right\rceil,$$

where $K_0 > 0$ is the initial burst size and, as in (12), $d_s = \frac{|\hat{\Omega}_s|}{p}$. The tuning parameter $\alpha > 0$ adjusts the annealing rate of the truncation frequency. Although the result in Theorem 1 is based on a fixed K, it can be easily shown that the same upper bound can also be attained with an increasing K_s. By increasing K in the later stage of the algorithm, when the majority of irrelevant features have been removed from the stable set, the chance of erroneous truncation is reduced. Such scheme further steers the algorithm from the mode of exploring potential sparse combination of features in the early stage toward the fine tuning of the weight vector by exploiting information from more samples in a sequence. It also facilitates faster convergence as the size of the stable set approaches to a sufficiently small number, as the algorithm converges to the standard stochastic gradient descent approximately.

6. Results

In this section, we present experimental results evaluating the performance of the proposed stabilized truncated SGD algorithm in high-dimensional classification problems with sparsity regularization. In this paper, we focus on linear prediction model for binary classification where $f_{\mathbf{w}}(x) = \mathbf{w}^T x$ and $\hat{y} = \text{sign}(f_w(x))$ with the observed class label $y \in \{-1, 1\}$. We consider two commonly used convex loss functions in machine learning tasks that both satisfy Assumption 1:

- Hinge loss: $l(f, y) = \max(1 - fy, 0)$;

- Logistic loss: $l(f, y) = \log(1 + \exp(-fy))$.

Using five data sets from different domains, the performance of our algorithm and other algorithms for comparison are evaluated on classification performance and feature selection stability and sparsity. We first define measure of feature stability in Section 6.1.

6.1 Feature Selection Stability

The goal of sparse learning is to select a subset of truly informative features with stabilized estimation variance as well as increased classification accuracy and model interpretability. Subgradient-based online learning methods depend heavily by the random ordering of samples on which they are fed to the algorithm. Such dependence leads to much deteriorated performance when it comes to high-dimensional sparse inputs. For a particular feature, the positions of its nonzero occurrences in a random ordering of samples greatly affect its learning outcome, in terms of learnt weight and membership in the set of selected features. Therefore, in addition to attaining a low generalization error, a desirable sparse online learning method should also produce an informative feature subset that is stable and robust to random permutations of input data. To evaluate feature selection stability of subgradient-based sparse learning methods, we define in the following a numerical measure of similarity between selected feature subsets resulted from different random permutations of data. Given an output weight vector $\underline{\mathbf{w}}$ from a subgradient-based algorithm with input data $\mathcal{D}$, similarly as in (6), we denote the selected feature subset as

$$\mathcal{S}(\underline{\mathbf{w}}; \mathcal{D}) = \{j : |\underline{w}_j| > 0, \underline{\mathbf{w}} = \Psi(\mathcal{D}\}.$$

Given two random permutations of the training data $\mathcal{D}$, $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$, the similarity between the two sets of selected feature subsets $\mathcal{S}_1 = \mathcal{S}(\underline{\mathbf{w}}_1; \mathcal{D}^{(1)})$ and $\mathcal{S}_2 = \mathcal{S}(\underline{\mathbf{w}}_2; \mathcal{D}^{(2)})$ is measured by the Cohen's kappa coefficient introduced by Cohen (1960),

$$\kappa(\mathcal{S}_1, \mathcal{S}_2) = \frac{q_o - q_e}{1 - q_e},$$

where q_o is the *relative observed agreement* between $\mathcal{S}_1$ and $\mathcal{S}_2$:

$$q_o = \frac{p_{11} + p_{22}}{p},$$

and q_e is the *hypothetical probability of change agreement*: $\mathcal{S}_1$ and $\mathcal{S}_2$

$$q_e = \frac{(p_{11} + p_{12})(p_{11} + p_{21})}{p^2} + \frac{(p_{12} + p_{22})(p_{21} + p_{22})}{p^2},$$

with $p_{11} = |\mathcal{S}_1 \cap \mathcal{S}_2|$, $p_{12} = |\mathcal{S}_1 \cap \mathcal{S}_2^C|$, $p_{21} = |\mathcal{S}_1^C \cap \mathcal{S}_2|$, $p_{22} = |\mathcal{S}_1^C \cap \mathcal{S}_2^C|$, and $p = p_{11} + p_{12} + p_{21} + p_{22}$ is the size of variable pool.

Note that $\kappa(\mathcal{S}_1, \mathcal{S}_2) \in [-1, 1]$, where $\kappa(\mathcal{S}_1, \mathcal{S}_2) = 1$ if $\mathcal{S}_1$ and $\mathcal{S}_2$ completely overlap with each other and $\kappa(\mathcal{S}_1, \mathcal{S}_2) = -1$ when $\mathcal{S}_1$ and $\mathcal{S}_2$ are in complete disagreement with $\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset$ and $\mathcal{S}_1^C \cap \mathcal{S}_2^C = \emptyset$.

Based on Cohen's kappa coefficient, we define the measure of feature selection stability of $\mathcal{S}(\underline{\mathbf{w}}; \cdot)$ returned by a procedure Ψ using randomly ordered data $\mathcal{D}^{(1)}$ and $\mathcal{D}^{(2)}$ as

$$s(\Psi) = \mathbb{E}_{\mathcal{D}^{(1)}, \mathcal{D}^{(2)}} \left[\kappa(\mathcal{S}(\underline{\mathbf{w}}_1; \mathcal{D}^{(1)}), \mathcal{S}(\underline{\mathbf{w}}_2; \mathcal{D}^{(2)})\right],$$

which is motivated by (Sun et al., 2013).

In practice, we use the empirical average $\hat{s}(\Psi)$ over B random permutations of the training data to measure the stability of the a subgradient-based online learning algorithm $\Psi(\cdot)$:

$$\hat{s}(\Psi) = \frac{1}{B(B-1)} \sum_{i=1}^{D} \sum_{j \neq i} \left[\kappa(\mathcal{S}_{\underline{\mathbf{w}}_i}; \mathcal{D}^{(i)}), \mathcal{S}(\underline{\mathbf{w}}_j; \mathcal{D}^{(j)}) \right]. \tag{18}$$

6.2 Experiment Setup

We evaluate the performance of our algorithm on several real-world classification data sets with up to $100,000$ features. These data sets have different levels of sparsity with various sample sizes. The information of experiment data sets are summarized in Table 1. The first four data sets were constructed for NIPS 2003 Feature Selection Challenge[4] (see Guyon et al., 2004, for details), which were preprocessed with added "probes" as random features distributed similarly to the real features. Thus, a good performance does not only lie in low generalization error rates, but also in sparse weight vectors that identify the truly important features. Reuters CV1 (RCV1) is a popular text classification data set with a bag-of-words representation. We use the binary version from the LIBSVM data set collection[5] introduced by Cai and He (2012). We create the training and validation set using a 70-30 random splits. All datasets are normalized such that each feature has variance 1.

Data Set	Domain	Dimensions	Data Density (Sparsity)	Training Size	Validation Size
RCV1	Text Mining	47,236	0.16%	14,169	6,073
Dexter	Text Mining	20,000	0.48%	300	300
Dorothea	Drug Discovery	100,000	0.91%	800	350
Gisette	Digits Recognition	5,000	13.00%	6,000	1,000
Arcene	Mass-Spectrometry	10,000	50.00%	100	100

Table 1: Datasets used in the numerical experiments. Sparsity is defined as the average feature sparsity levels, which is the column-wise average percentage of nonzero entries, in the training data.

We compare the proposed algorithm with the standard stochastic gradient descent algorithm in (Bottou, 1998) and another three other sparsity-inducing stochastic methods, including the truncated gradient algorithm in (Langford et al., 2009), the Regularized Dual Averaging (RDA) algorithm in (Xiao, 2009), and the forward backward splitting (FOBOS) algorithm in (Duchi and Singer, 2009). The RDA algorithm updates the weight vector at each step based on a running average $\bar{\mathbf{g}}_t$ of all subgradients $\{\mathbf{g}_\tau = L'(\mathbf{w}_\tau, z_\tau) \in \partial_{\mathbf{w}_\tau} L(\mathbf{w}_\tau, z_\tau), \tau = 1, \ldots, t\}$ in previous iterations as

$$\bar{\mathbf{g}}_t = \frac{t-1}{t} \mathbf{g}_{t-1} + \frac{1}{t} \mathbf{g}_t.$$

4. The data source can be found on `https://archive.ics.uci.edu/ml/index.html`.
5. The data source can be found on `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

Given the average subgradient , the next weight vector is computed by solving the minimization problem

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left\{ \frac{1}{t} \sum_{\tau}^{t} \langle L'(\mathbf{w}_t, z_t), \mathbf{w} \rangle + \Psi(\mathbf{w}) + \frac{\beta_t}{t} h(\mathbf{w}) \right\}, \tag{19}$$

where $\Psi(\mathbf{w})$ is the regularizer, $h(w)$ is an auxiliary strongly convex function, and $\{\beta_t\}_{t \geq 1}$ is a nonnegative and nondecreasing input sequence, which determines the convergence properties of the algorithm.

In the context of L_1 regularization, the RDA algorithm is derived by setting $\Psi(\mathbf{w}) = \lambda ||\mathbf{w}||_1$, $\beta_t = \gamma \sqrt{t}$, and replacing $h(w)$ with a parametrized version:

$$h_\rho = \frac{1}{2} ||\mathbf{w}||_2^2 + \rho ||\mathbf{w}||_1,$$

where $\rho \geq 0$ is a sparsity-enhancing parameter. Hence, the minimization problem in (19) has a explicit solution as, for $j = 1, \ldots, p$,

$$\mathbf{w}_{t+1}^{(j)} = \begin{cases} 0, & \text{if } |\bar{\mathbf{g}}_t^{(j)}| \leq \lambda_t^{\text{RDA}}; \\ -\frac{\sqrt{t}}{\gamma} \left(\bar{\mathbf{g}}_t^{(j)} - \lambda_t^{\text{RDA}} \text{sgn}(\bar{\mathbf{g}}_t^{(j)}) \right), & \text{otherwise,} \end{cases}$$

which is equivalent to

$$\mathbf{w}_{t+1} = T(\bar{\mathbf{g}}_t, \lambda_t^{\text{RDA}}),$$

where $\lambda_t^{\text{RDA}} = \lambda + \gamma \rho / \sqrt{t}$.

The FOBOS algorithm alternates between two phases. On each iteration, it first perform an unconstrained gradient descent step as in the standard SGD algorithm, whose output is denoted as $\mathbf{w}_{t+\frac{1}{2}}$. Then it cast and solve an instantaneous optimization problems that trades off between minimization of a regularization term and a close proximity to the result of the first phase:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left\{ \frac{1}{2} ||\mathbf{w} - \mathbf{w}_{t+\frac{1}{2}}||^2 + \eta_{t+\frac{1}{2}} \Psi(\mathbf{w}) \right\}, \tag{20}$$

where the regularization function is scaled by an interim step size $\eta_{t+\frac{1}{2}}$.

With L_1 regularization where $\Psi(w) = \lambda ||\mathbf{w}||_1$, the second-phase update can be computed as

$$\mathbf{w}_{t+1} = T(\mathbf{w}_{t+\frac{1}{2}}, \eta_{t+\frac{1}{2}} \lambda).$$

To evaluate the stability of resulted weight vectors, we randomly permute the indices of the training samples for $B = 50$ times to produce stochastic samples that are fed to the algorithms. Such randomization helps identify the instability in learning results in terms of both error rate and the selected features.

For implementing all five stochastic methods, we allow the algorithms to run on the training data for $\{5, 10, 20, 30, 40, 50, 60\}$ passes. In the standard stochastic gradient descent algorithm, we choose the optimal learning rate between 0.1 to 0.5. The base gravity parameter in the truncated gradient algorithm is chosen from the range $[0.001, 0.01]$. For RDA, we follow the suggestions in (Xiao, 2009) which sets $\gamma = 5000$ and $\rho = 0.005$ (effectively $\gamma\rho = 25$). We tune the parameter

λ from the set of values $\{5 \times 10^{-5}, 1 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}, 0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$ for both RDA and FOBOS. As instructed by Duchi and Singer (2009), η_t is set to be $\frac{1}{\sqrt{t}}$ in FOBOS. For the proposed stabilized truncated SGD algorithm, the size of each burst (the truncation frequency) is fixed at $K = 5$ and $n_K = 5$ for all data sets. In the proposed algorithm, we initiate the rejection rate at $\beta_0 = 0.7$ with annealing rate chosen from $\{-7, -5, -3, -1, 0, 1, 3\}$. The stability selection threshold π_0 is tuned within the range $[0.5, 0.6, \ldots, 0.9]$. For multi-thread implementation in Algorithm 4, $M = 16$ is used running in parallel on a high-performance computing cluster.

6.3 Results

6.3.1 CLASSIFICATION PERFORMANCE

As shown in Table 2, the proposed algorithm shows improvement over the truncated gradient algorithm over all data sets and has better performances than RDA and FOBOS in most of the experiments. As data density increases, the truncated gradient algorithm performs better with hinge loss than with logistic loss. With hinge loss, the algorithm updates the weight vector only for samples within a small margin from the boundary. With logistic loss, the algorithm updates with continuous increments for all incoming samples and favors dense features in sparse learning. With highly sparse samples sequentially feed to the algorithm, the truncation in every K iterations is conducted with insufficient information about the true gradient due to the lack of nonzero entries in sparse features. Truncation with logistic loss leads to over selection of dense features and overfitting. The other two sparsity-inducing methods for comparison, RDA and FOBOS, also have large fluctuations across data sets with different levels of sparsity over these two loss functions, especially when data is highly sparse. In comparison, the performance of the proposed algorithm is consistent for various dimensions and sparsity levels. Our algorithm is also shown to be robust to different choices of loss function. The comparatively lower test errors, especially for highly sparse data, mainly owes to the proposed algorithm's fairer treatments of features with heterogeneous sparsity.

From Table 2 we also observe substantially lower variances in test errors under random permutations of samples. The proposed algorithm has the lowest standard deviations of test errors across all data sets and under both loss functions. Such an improvement over truncated gradient algorithm and other comparing methods comes from both informative truncation with adaptive gravity and stability selection. Based on these selection probabilities we carry out feature purging. Our selection probability is computed based on multiple bursts whose truncation is guided by fair amounts of shrinkage. Hence, the removal decisions of features from the set of stable variables are grounded in reliable information on feature importance, which is more likely to be shared across different permutations of the data. The accumulations of sufficient information for all features help the proposed algorithm to be robust to random fluctuations in online setting.

6.3.2 FEATURE SELECTION AND SPARSITY

As far as sparse learning for feature selection goes, the proposed algorithm, as shown in Table 3, is the only method that delivers sparse weight vectors across five example data sets with various sparsity levels. It is shown in Table 3 that the resulted weight vectors are far more sparse than the truncated gradient algorithm in all data sets with both loss functions. When comparing these two algorithms, the distinction in attained sparsity levels is particularly striking with logistic loss

	Data Set	Standard SGD	Truncated Gradient	RDA	FOBOS	Stabilized Truncated SGD
Hinge Loss	RCV1	**2.86**	6.59	4.62	8.44	3.01
		.08	*.09*	*.10*	*.15*	*.04*
	Dexter	7.77	8.37	11.42	10.91	**6.58**
		.96	*.91*	*.59*	*1.21*	*.57*
	Dorothea	6.11	6.83	6.83	8.51	**5.14**
		.60	*1.23*	*.18*	*1.96*	*0.43*
	Gisette	2.71	6.32	**2.40**	5.50	3.05
		.43	*3.02*	*.45*	*1.20*	*.30*
	Arcene	19.26	22.06	22.10	18.72	**17.60**
		3.67	*8.08*	*4.00*	*3.26*	*2.62*
Logistic Loss	RCV1	3.68	18.55	5.69	14.18	**3.19**
		.07	*.78*	*.08*	*.39*	*.05*
	Dexter	7.91	10.25	9.41	10.45	**6.41**
		.36	*.58*	*.68*	*.81*	*.43*
	Dorothea	6.26	7.29	6.87	7.15	**5.47**
		.62	*1.28*	*.31*	*1.35*	*.35*
	Gisette	2.52	6.35	2.21	5.98	**2.11**
		.29	*4.08*	*.20*	*.85*	*.26*
	Arcene	17.38	19.48	21.62	18.44	**13.38**
		2.93	*5.01*	*2.05*	*3.11*	*1.26*

Table 2: Mean test errors (%) and the corresponding standard deviations with hinge loss and logistic loss over 50 random ordering of the training samples.

function and when the sparsity level of the input data is low, such as the case of the Arcene data set. These results also indicate that the truncate gradient algorithm fails to extract significant features and to obtain sparse solutions, which motivates the development of techniques discussed in this paper. On the other hand, the variances in the percentage of nonzero features of the proposed algorithm are reduced by approximately a magnitude of 10. The contribution of stabilization is demonstrated again in terms of feature selection. Although RDA achieves very low sparsity in Dorothea, such a behavior is not observed in other data sets. It is shown to result in particularly denser weight vector with highly sparse data, such as RCV1, indicating its weakness in identifying truly informative features when information is scarce. FOBOS demonstrates overall poorer performance in terms of inducing sparse weight vector as compared to RDA and the proposed algorithm. Similar to its performance in terms of test error, the proposed algorithm delivers highly stable results in feature selection regardless of the choice of loss function and of random permutations of the data. We also achieve sufficient sparse result owing to stability selection which prevent noisy features from adding back to the stable set of variables in the online setting. On the other hand, the high sparsity in weight vector does not overshadow the generalizability performance as the informative truncation with unbiased shrinkage underlies a better estimation of the selection probability for the construction of the set of stable variables.

We further compare the data sparsity levels of the features selected by both sparse online learning algorithms. In Figure 3, using the Dorothea data set as an example, we show the fraction

	Data Set	Standard SGD	Truncated Gradient	RDA	FOBOS	Stabilized Truncated SGD
Hinge Loss	RCV1	63.68	6.95	11.38	5.01	0.86
		.34	*.42*	*.10*	*.28*	*.10*
	Dexter	29.46	14.74	1.61	37.69	1.98
		.35	*.82*	*.06*	*2.43*	*.15*
	Dorothea	39.84	18.62	0.55	14.89	6.68
		.62	*5.94*	*.02*	*3.37*	*.09*
	Gisette	90.88	63.8	17.45	61.81	3.84
		.21	*3.72*	*.43*	*3.94*	*.34*
	Arcene	98.22	61.08	12.15	54.5	3.67
		.07	*25.55*	*.90*	*13.37*	*.40*
Logistic Loss	RCV1	82.31	2.75	6.98	2.89	1.44
		.00	*.35*	*.05*	*.29*	*.12*
	Dexter	38.76	13.91	4.84	30.21	1.32
		.00	*.71*	*.14*	*1.87*	*.13*
	Dorothea	68.63	19.31	0.43	20.4	1.67
		.00	*3.00*	*.01*	*3.51*	*.47*
	Gisette	93.96	66.31	14.99	58.83	3.22
		.00	*3.31*	*.34*	*3.44*	*.49*
	Arcene	98.36	88.34	12.71	45.96	5.35
		.00	*4.38*	*.27*	*9.76*	*.24*

Table 3: The average percentages (%) of nonzero features selected in the resulted weight vector and the corresponding standard deviations with hinge loss and logistic loss over 50 random permutations of the training samples.

of selected features at different data density levels. It is shown that truncated gradient algorithm is more likely to select dense features over sparse ones with both loss functions. The majority of the sparse features are truncated to zero regardless of their importance. Moreover, the fraction of selected feature exhibits a linear relationship with the level of feature density in truncated gradient algorithm. This pattern suggests that the amount of shrinkage applied to features should be approximately proportional to their data density, on the probability of having informative updates. This provides an independent justification of the informative truncation introduced in Section 3.1. In contrast, features selected by the proposed algorithm have approximately uniform distribution of sparsity levels. This improvement over the truncated gradient algorithm mostly owes to the use of informative truncation. With the crucial treatment on the heterogeneity in feature sparsity, the proposed algorithm is effective in keeping rare but important features from premature truncations.

Based on the measure of feature selection stability defined in (18), the performance of the original truncated gradient algorithm and the proposed algorithm are summarized in Table 4. The proposed algorithm demonstrates its excellent stability as compared to the truncated gradient algorithm. It shows greater overlap in selected features among different permutations of the training data. This result also echos with the lower standard deviations in the test error (Table 2) and in the number of selected features (Table 3). As discussed in Section 3, the insensitivity to data perturbation is mainly due to the fair shrinkage applied on each feature by using informative truncation and to the

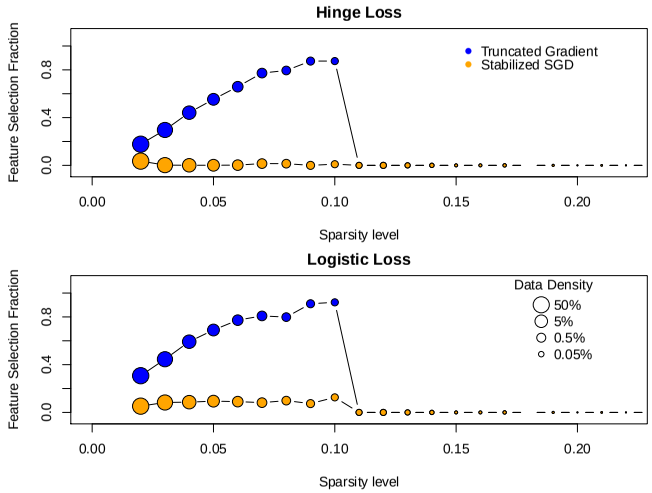


Figure 3: The fraction of features selection at different sparsity levels by the truncated gradient algorithm and the proposed algorithm respectively with the Dorothea data set as an example. The selected features are extracted as the features with nonzero entries in the resulted weight vector from the algorithms with a single fixed ordering of the training data. The x-axis represents the sparsity level, which is the proportion of nonzero entries in a feature in the training sample. The y-axis indicates the fraction of selected features among features with a discretized sparsity level. The size of each point scales with the proportion of features that fall in level of sparsity. It can be seen that the truncated gradient algorithm over-selects denser features. The proposed algorithm does not have a pre-exposed preference towards any density level.

sufficient accumulations of information in selection probabilities before applying stability selection. These measures suggest that the proposed algorithm is particularly favorable for high-dimensional sparse data.

7. Conclusion

In this paper, we address the problem of inducing sparsity in subgradient-based online learning algorithm when applied to high-dimensional sparse data. Based on the truncated gradient framework, we reduce truncation bias due to heterogeneity in feature sparsity via informative truncation. Integrated with soft-thresholding truncation, stability selection helps eliminate irrelevant features along the learning process in order to achieve stable and sufficiently sparse results. The adaptive gravity method dynamically adjusts the shrinkage at different stages of the learning process to balance between the exploration of sparse combination of nonzero weights and the exploitation of finite

	Hinge Loss		Logistic Loss	
Data Set	Truncated Gradient	Stabilized Truncated SGD	Truncated Gradient	Stabilized Truncated SGD
RCV1	0.38	0.44	0.29	0.60
Dexter	0.46	0.61	0.23	0.58
Dorothea	0.19	0.52	0.19	0.33
Gisette	0.48	0.60	0.54	0.54
Arcene	0.26	0.55	0.35	0.69

Table 4: Summary of feature selection stability performance of the resulted weight vector measured by (18) based on Cohen's kappa coefficient.

updates for better convergence. This strategy also consolidates the tuning needs of the proposed algorithm into two major parameters, the annealing rate γ and the purging threshold π_0. We present a theoretical analysis of the expected regret bound, which offered a quantification of the potential improvement attained by the proposed algorithm. At last, we use extensive experimental studies to evaluate the performance of the proposed algorithm. The results demonstrate that our algorithm achieve favorable results in terms of both prediction accuracy and feature selection, compared to the original truncated gradient algorithm.

Acknowledgments

We acknowledge computing resources from Columbia University's Shared Research Computing Facility project, which is supported by NIH Research Facility Improvement Grant 1G20RR030893-01, and associated funds from the New York State Empire State Development, Division of Science Technology and Innovation (NYSTAR) Contract C090171, both awarded April 15, 2010.

Appendices

A. Proof of Lemma 2

Proof Note that $\mathbf{w}^*$ can be written as $\mathbf{w}^* I_{\Omega^*}$. Based on the stability selection in (10), we have $\tilde{\mathbf{w}} = \hat{\mathbf{w}} I_{\hat{\Omega}}$. Let $\Omega = \{1, \ldots, p\}$. The full set Ω can be further divided into four disjoint sets: $S_1 = \Omega \setminus (\Omega^* \cup \hat{\Omega})$, $S_2 = \Omega^* \setminus \hat{\Omega}$, $S_3 = \hat{\Omega} \setminus \Omega^*$, and $S_4 = \Omega^* \cap \hat{\Omega}$, respectively.

Firstly, we consider the L_2 distance from the non-stabilized weight vector $\hat{\mathbf{w}}$ to $\mathbf{w}^*$ under Assumption 3 and the L_2 distance from the stabilized weight vector $\tilde{\mathbf{w}}$ to $\mathbf{w}^*$.

$$\begin{aligned}||\hat{\mathbf{w}} - \mathbf{w}^*||^2 &= ||\hat{\mathbf{w}} - \mathbf{w}^* I_{\Omega^*}||^2 \\ &= \sum_{j \in \Omega^*} (\hat{w}_j - w_j^*)^2 + \sum_{j \notin \Omega^*} \hat{w}_j^2, \\ &= \sum_{j \in S_2 \cup S_4} (\hat{w}_j - w_j^*)^2 + \sum_{j \in S_1 \cup S_3} \hat{w}_j^2.\end{aligned}$$

Since

$$(\hat{\mathbf{w}} I_{\hat{\Omega}} - \mathbf{w}^* I_{\Omega^*})_j^2 = \begin{cases} 0, & \text{if } j \notin \Omega^* \text{ and } j \notin \hat{\Omega}; \\ (w_j^*)^2, & \text{if } j \in \Omega^* \text{ and } j \notin \hat{\Omega}; \\ (\hat{w}_j)^2, & \text{if } j \notin \Omega^* \text{ and } j \in \hat{\Omega}; \\ (\hat{w}_j - w_j^*)^2, & \text{if } j \in \Omega^* \text{ and } j \in \hat{\Omega}, \end{cases}$$

$$\begin{aligned}||\tilde{\mathbf{w}} - \mathbf{w}^*||^2 &= \|\hat{\mathbf{w}} I_{\hat{\Omega}} - \mathbf{w}^* I_{\Omega^*}\|^2 \\ &= \sum_{j \in S_2} (w_j^*)^2 + \sum_{j \in S_3} (\hat{w}_j)^2 + \sum_{j \in S_4} (\hat{w}_j - w_j^*)^2.\end{aligned}$$

Then

$$\begin{aligned}||\hat{\mathbf{w}} - \mathbf{w}^*||^2 - ||\tilde{\mathbf{w}} - \mathbf{w}^*||^2 &= \sum_{j \in S_2} (\hat{w}_j - w_j^*)^2 + \sum_{j \in S_1} \hat{w}_j^2 - \sum_{j \in S_2} (w_j^*)^2 \\ &= \sum_{j \in S_2} \hat{w}_j^2 + \sum_{j \in S_2} (w_j^*)^2 - 2 \sum_{j \in S_2} w_j^* \hat{w}_j - \sum_{j \in S_2} (w_j^*)^2 + \sum_{j \in S_1} \hat{w}_j^2 \\ &= \sum_{j \in S_1 \cup S_2} \hat{w}_j^2 - 2 \sum_{j \in S_2} w_j^* \hat{w}_j \\ &= ||\hat{\mathbf{w}}(1 - I_{\hat{\Omega}})||^2 - 2 \sum_{j \in S_2} w_j^* \hat{w}_j. \end{aligned} \tag{21}$$

For the first part in (21), there exists a small $\varepsilon \in \left(0, \frac{\pi_0 C(p - |\hat{\Omega}|)}{|\hat{\Omega}|}\right)$ with $\hat{S}_\varepsilon = \{j : \mathbb{E}(|\hat{w}_j|) > \varepsilon\}$ such that

$$\begin{aligned}\mathbb{E}\left(||\hat{\mathbf{w}}(1 - I_{\hat{\Omega}})||^2\right) =& \mathbb{E}\left(\sum_{j=1}^p \hat{w}_j^2 I(\hat{p}_j \leq \pi_0)\right) \\ =& \mathbb{E}\left(\sum_{j=1}^p \hat{w}_j^2 I(\hat{p}_j \leq \pi_0) I(|\hat{w}_j| \leq \varepsilon)\right) + \mathbb{E}\left(\sum_{j=1}^p \hat{w}_j^2 I(\hat{p}_j \leq \pi_0) I(|\hat{w}_j| > \varepsilon)\right) \\ \geq& 0 + \varepsilon^2 \mathbb{E}\left(\sum_{j=1}^p I(\hat{p}_j \leq \pi_0) I(|\hat{w}_j| \geq \varepsilon)\right) \\ \geq& \varepsilon^2 (|\hat{S}_\varepsilon| - |\hat{\Omega}|) > 0, \end{aligned} \tag{22}$$

where the last inequality is due to $|\hat{S}_\varepsilon| > |\hat{\Omega}|$ with the specified range of ε.

For the second part in (21), recall that $S_2 = \Omega^* \setminus \hat{\Omega} = \Omega^* \setminus (\hat{\Omega} \setminus V)$. Using (13) from Lemma 1 in (Meinshausen and Bühlmann, 2010), we have

$$\begin{aligned}\mathbb{E}(|S_2|) &= \mathbb{E}(\Omega^* - (\hat{\Omega} - V)) \\ &= d^* - E(\hat{\Omega}) - \mathbb{E}(V)) \\ &= d^* - q^{\mathbf{g}} + \mathbb{E}(V) \\ &\leq d^* - q^{\mathbf{g}} + \frac{1}{2\pi_0 - 1}\frac{(q^{\mathbf{g}})^2}{p} \\ &\leq \left(\frac{q^g}{2\pi_0 p - p} - 1\right) q^g + d^*.\end{aligned} \tag{23}$$

And, since, for any $j \in \Omega \setminus \hat{\Omega}$, $\Pr(|\hat{w}_j| > 0) \leq \pi_0$,

$$\mathbb{E}(|\hat{w}_j|) = \int_0^C \Pr(|\hat{w}_j| \geq u)du \leq \Pr(|\hat{w}_j| > 0)C \leq \pi_0 C.$$

Then,

$$\mathbb{E}\left(\sum_{j \in S_2} w_j^* \hat{w}_j\right) \leq \pi_0 C^2 \mathbb{E}(|S_2|) \leq \pi_0 C^2 \mathbb{E}(|S_2|)\left[\left(\frac{q^g}{2\pi_0 p - p} - 1\right) q^g + d^*\right].$$

Thus we have

$$\mathbb{E}\left(||\hat{\mathbf{w}} - \mathbf{w}^*||^2 - ||\tilde{\mathbf{w}} - \mathbf{w}^*||^2\right) \geq \varepsilon^2(|\hat{S}_\varepsilon| - |\hat{\Omega}|) + 2\pi_0 C^2\left[\left(1 - \frac{q^g}{2\pi_0 p - p}\right) q^g - d^*\right]. \tag{24}$$

The first half of (24) is proved to be non-negative by the inequality in (22). With a sufficiently large $\pi_0 \in \left(\frac{1}{2} + \frac{(q^{\mathbf{g}})^2}{2p(q^{\mathbf{g}} - d^*)}, 1\right)$, (24) is non-negative by taking π_0 into the second half of it with $C \geq 0$ under Assumption 1.

■

B. Proof of Lemma 3

Proof Based on Algorithm 1 and Algorithm 2, we have the following relationships:

$$\bar{\mathbf{w}}_1 = T(\mathbf{h}_1, Kg_0) = \text{sign}(\mathbf{h}_1)\max(|\mathbf{h}_1| - g_0 \mathbf{1}_p K, 0), \tag{25}$$

$$\hat{\mathbf{w}}_1 = T(\mathbf{h}_1, \tilde{\mathbf{k}} g_0) = \text{sign}(\mathbf{h}_1)\max(|\mathbf{h}_t| - g_0 \tilde{\mathbf{k}}_1, 0), \tag{26}$$

where

$$\mathbf{h}_1 = \mathbf{w}_0 - \sum_{t=1}^{K} \eta L'\left(\mathbf{w}_{t-1}, z_t\right).$$

Without loss of generality, we consider the difference in the squared errors to the optimal weight vector $\mathbf{w}^*$ between$\bar{\mathbf{w}}_1$ and $\hat{\mathbf{w}}_1$:

$$\begin{aligned}&||\bar{\mathbf{w}}_1 - \mathbf{w}^*||^2 - ||\hat{\mathbf{w}}_1 - \mathbf{w}^*||^2 \\ =&||(\bar{\mathbf{w}}_1 - \mathbf{h}_1) - (\mathbf{w}^* - \mathbf{h}_1)||^2 - ||(\hat{\mathbf{w}}_1 - \mathbf{h}_1) - (\mathbf{w}^* - \mathbf{h}_1)||^2 \\ =&\left(||\bar{\mathbf{w}}_1 - \mathbf{h}_1||^2 - ||\hat{\mathbf{w}}_1 - \mathbf{h}_1||^2\right) - 2\left[(\bar{\mathbf{w}}_1 - \mathbf{h}_1)^T(\mathbf{w}^* - \mathbf{h}_1) - (\hat{\mathbf{w}}_1 - \mathbf{h}_1)^T(\mathbf{w}^* - \mathbf{h}_1)\right].\end{aligned} \tag{27}$$

In the first part of (27), based on (25) and (26), we have

$$\begin{aligned}||\bar{\mathbf{w}}_1 - \mathbf{h}_1||^2 &= pK^2 g_0^2, \\ ||\hat{\mathbf{w}}_1 - \mathbf{h}_1||^2 &= ||\tilde{\mathbf{k}}_1||^2 g_0^2,\end{aligned}$$

where $\tilde{\mathbf{k}}_1$ is the counter of informative updates in the first burst with $\tilde{k}_{1,j} \leq K$ for $j = 1, \ldots, p$. Hence,

$$||\bar{\mathbf{w}}_1 - \mathbf{h}_1||^2 - ||\hat{\mathbf{w}}_1 - \mathbf{h}_1||^2 \geq 0. \tag{28}$$

Based on (25) and (26),

$$\begin{aligned}(\bar{\mathbf{w}}_1 - \mathbf{h}_t)^T \mathbf{h}_1 &= -g_0 \sum_{j=1}^{p} \text{sign}(h_{1,j}) K \mathbf{h}_{1,j} = -g_0 K ||\mathbf{h}_t||_1, \\ (\hat{\mathbf{w}}_1 - \mathbf{h}_t)^T \mathbf{h}_1 &= -g_0 \sum_{j=1}^{p} \text{sign}(h_{1,j}) \tilde{k}_{1,j} \mathbf{h}_{1,j} = -g_0 ||\tilde{\mathbf{k}}_1 \mathbf{h}_t||_1.\end{aligned}$$

Thus in the second part of (27), we have

$$\begin{aligned}&-2\left[(\bar{\mathbf{w}}_1 - \mathbf{h}_1)^T(\mathbf{w}^* - \mathbf{h}_1) - (\hat{\mathbf{w}}_1 - \mathbf{h}_1)^T(\mathbf{w}^* - \mathbf{h}_1)\right] \\ =&\left[(\bar{\mathbf{w}}_1 - \mathbf{h}_1)^T \mathbf{h}_1 - (\hat{\mathbf{w}}_1 - \mathbf{h}_1)^T \mathbf{h}_1\right] - \left[(\bar{\mathbf{w}}_1 - \mathbf{h}_1)^T \mathbf{w}^* - (\hat{\mathbf{w}}_1 - \mathbf{h}_1)^T \mathbf{w}^*\right] \\ =&2g_0\left[\left(K\|\mathbf{h}_1\|_1 - \|\tilde{\mathbf{k}}_1 \mathbf{h}_1\|_1\right) - (\bar{\mathbf{w}}_1 - \hat{\mathbf{w}}_1)^T \mathbf{w}^*\right].\end{aligned} \tag{29}$$

Since $K \geq \tilde{k}_{1,j}$ for $j = 1, \ldots, p$, the first part of (29) is guaranteed to be nonnegative. In the second part of (29) , let $\hat{w}_{1,j} = \hat{v}_{1,j} I(|h_{1,j}| > \tilde{k}_{1,j} g_0)$ and $\bar{w}_{1,j} = \bar{v}_{1,j} I(|h_{1,j}| > K g_0)$. Denote $\delta_t = L(w_{t-1}, z_t)$ to be the gradient of the loss function at a certain iteration. Since in this paper we consider linear prediction model,

$$\delta_t = L'(\mathbf{w}_{t-1}, z_t) = G(f_{\mathbf{w_{t-1}}}(\mathbf{x}_t))\mathbf{x}_t.$$

Hence, $\delta_{t,j} = \delta_{t,j} I(|x_{t,j}| > 0)$.

Again based on relationships in (25) and (26), we consider the following two scenarios:

- When $h_{1,j} > 0$:

$$\hat{v}_{1,j} = \sum_{t=1}^{K} \left[I(|x_{t,j}| > 0)(\delta_{t,j} - g_0) \right],$$
$$\bar{v}_{1,j} = \sum_{t=1}^{K} \left[I(|x_{t,j}| > 0)\delta_{t,j} - g_0 \right].$$

Thus,

$$\begin{aligned}
\bar{v}_{1,j} - \hat{v}_{1,j} =& - \sum_{t=1}^{K} I(|x_{t,j}| = 0) g_0, \\
\bar{w}_{1,j} - \hat{w}_{1,j} =& - \sum_{t=1}^{K} I(|x_{t,j}| = 0) g_0 I(h_{1,j} > K g_0) - \sum_{t=1}^{K} (h_{1,j} \\
& - \tilde{k}_{1,j} g_0) I(\tilde{k}_{1,j} g_0 < h_{1,j} < K g_0) \\
\leq & - \sum_{t=1}^{K} I(|x_{t,j}| = 0) g_0 I(h_{1,j} > K g_0). \qquad (30)
\end{aligned}$$

- When $h_{1,j} < 0$, similarly, we get

$$\begin{aligned}
\bar{v}_{1,j} - \hat{v}_{1,j} =& \sum_{t=1}^{K} I(|x_{t,j}| = 0) g_0, \\
\bar{w}_{1,j} - \hat{w}_{1,j} =& \sum_{t=1}^{K} I(|x_{t,j}| = 0) g_0 I(h_{1,j} < -K g_0) \\
& + \sum_{t=1}^{K} (h_{1,j} + \tilde{k}_{1,j} g_0) I(-K g_0 < h_{1,j} < -\tilde{k}_{1,j} g_0) \\
\geq & \sum_{t=1}^{K} I(|x_{t,j}| = 0) g_0 I(h_{1,j} < -K g_0). \qquad (31)
\end{aligned}$$

Hence, based on jointly we have

$$\begin{aligned}
& \mathbb{E}\left((\bar{\mathbf{w}}_1 - \hat{\mathbf{w}}_1)^T \mathbf{w}^* \right) \\
=& \sum_{j=1}^{p} \mathbb{E}((\bar{w}_{1,j} - \hat{w}_{1,j}) w_j^* | h_{1,j} > 0) \Pr(h_{1,j} > 0) + \mathbb{E}((\bar{w}_{1,j} - \hat{w}_{1,j}) w_j^* | h_{1,j} < 0) \Pr(h_{1,j} < 0) \\
\leq & - g_0 \sum_{t=1}^{K} \sum_{j=1}^{p} I(|x_{t,j}| = 0) |w_j^*| \leq 0.
\end{aligned}$$

The second inequality is derived from the condition that $\lambda_j = 0$ if $\mathbf{w}_j^* = 0$ and the following fact. Since $\mathbf{h}_1$ is the sum of K stochastic gradients descent steps and each one of the stochastic gradient

which is an unbiased estimator of the true gradient (see Bottou, 1998, for analysis), we have

$$\mathbb{E}(I(h_{1,j} < 0) w_j^*) < 0,$$
$$\mathbb{E}(I(h_{1,j} > 0) w_j^*) > 0.$$

Hence, together with (28) and (29), we have

$$||\bar{\mathbf{w}}_1 - \mathbf{w}^*||^2 - ||\hat{\mathbf{w}}_1 - \mathbf{w}^*||^2 \geq 0.$$

■

C. Proof of Theorem 1

Proof At a given time t when truncation is performed, let $\mathbf{h}_t = \mathbf{w}_{t-1} - \eta \nabla_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}, z)$ and let $\hat{\mathbf{w}}_t$ be the truncated but non-stabilized weight vector based on the truncation in (4) that $\hat{\mathbf{w}}_t = T(\mathbf{h}_t, g_{0,t}\tilde{\mathbf{k}}_t)$ with the base gravity $g_{0,t}$. Let $\hat{\Omega}$ be the current set of stable features and the stabilized weight vector is obtained as $\bar{\mathbf{w}}_t = \hat{\mathbf{w}}_t I_{\hat{\Omega}}$.

Firstly, we have

$$\begin{aligned}
||\hat{\mathbf{w}}_t - \mathbf{w}^*||^2 \leq & ||\mathbf{w}^* - \hat{\mathbf{w}}_t||^2 + ||\hat{\mathbf{w}}_t - \mathbf{h}_t||^2 \qquad (32) \\
= & ||\mathbf{w}^* - \mathbf{h}_t||^2 - 2(\mathbf{w}^* - \hat{\mathbf{w}}_t)^T (\hat{\mathbf{w}}_t - \mathbf{h}_t).
\end{aligned}$$

In the first part of (32),

$$\begin{aligned}
||\mathbf{w}^* - \mathbf{h}_t||^2 =& ||\mathbf{w}^* - \mathbf{w}_{t-1} + \mathbf{w}_{t-1} - \mathbf{h}_t||^2 \\
=& ||\mathbf{w}^* - \mathbf{w}_{t-1}||^2 + ||\mathbf{w}_{t-1} - \mathbf{h}_t||^2 + 2(\mathbf{w}^* - \mathbf{w}_{t-1})^T (\mathbf{w}_{t-1} - \mathbf{h}_t) \\
=& ||\mathbf{w}^* - \mathbf{w}_{t-1}||^2 + ||\mathbf{w}_{t-1} - \eta \nabla_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}, z) - \mathbf{w}_{t-1}||^2 \\
& + 2(\mathbf{w}^* - \mathbf{w}_{t-1})^T (\mathbf{w}_{t-1} - \mathbf{w}_{t-1} + \eta \nabla_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}, z)) \\
=& ||\mathbf{w}^* - \mathbf{w}_{t-1}||^2 + \eta^2 ||\nabla_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}, z)||^2 + 2\eta (\mathbf{w}^* - \mathbf{w}_{t-1})^T \nabla_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}, z). \qquad (33)
\end{aligned}$$

Since $L(w, z)$ is convex,

$$(\mathbf{w}^* - \mathbf{w}_{t-1})^T \nabla_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}, z) \leq L(\mathbf{w}^*, z) - L(\mathbf{w}_{t-1}, z).$$

And based on Assumption 1,

$$||\nabla_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}, z)||^2 \leq A L(\mathbf{w}_{t-1}, z) + B.$$

Thus (33) has the upper bound

$$||\mathbf{w}^* - \mathbf{h}_t||^2 \leq ||\mathbf{w}^* - \mathbf{w}_{t-1}||^2 + \eta^2 (A L(\mathbf{w}_{t-1}, z) + B) + 2\eta \left[L(\mathbf{w}^*, z) - L(\mathbf{w}_{t-1}, z) \right].$$

In the second part of (32), since $\hat{\mathbf{w}}_t = T(\mathbf{h}_t, g_t)$, $\hat{\mathbf{w}}_t = \text{sign}(\mathbf{h}_t) \max(|\mathbf{h}_t| - g_{0,t}\tilde{\mathbf{k}}_t, 0)$ and thus

$$(\hat{\mathbf{w}}_t - \mathbf{h}_t)^T \hat{\mathbf{w}}_t = -g_{0,t} \sum_{j=1}^{p} \text{sign}(h_{t,j}) \tilde{k}_{t,j} \hat{\mathbf{w}}_{t,j} = -g_{0,t} ||\tilde{\mathbf{k}}_t \hat{\mathbf{w}}_t||_1.$$

Then

$$\begin{aligned}-(\mathbf{w}^* - \hat{\mathbf{w}}_t)^T(\hat{\mathbf{w}}_t - \mathbf{h}_t) &= -(\mathbf{w}^*)^T(\hat{\mathbf{w}}_t - \mathbf{h}_t) + \hat{\mathbf{w}}_t^T(\hat{\mathbf{w}}_t - \mathbf{h}_t) \\ &\leq -\sum_{j=1}^{p} |w_j^*||\hat{w}_{t,j} - h_{t,j}| - g_{0,t}||\tilde{\mathbf{k}}_t\hat{\mathbf{w}}_t||_1 \\ &\leq -g_{0,t}\sum_{j=1}^{p} |\tilde{k}_{t,j}w_j^*| - g_{0,t}||\tilde{\mathbf{k}}_t\hat{\mathbf{w}}_t||_1 \\ &\leq -Kg_{0,t}\sum_{j=1}^{p} |w_j^*| - Kg_{0,t}||\hat{\mathbf{w}}_t||_1 \\ &= -Kg_{0,t}\left(||\mathbf{w}^*||_1 - ||\hat{\mathbf{w}}_t||_1\right).\end{aligned}$$

The second last inequality is due to $\tilde{k}_{j,t} \leq K$ for all $j = 1, \ldots, p$.

Thus, in total, equation (32) has the upper bound

$$\begin{aligned}||\hat{\mathbf{w}}_t - \mathbf{w}^*||^2 \leq ||\mathbf{w}^* - \mathbf{w}_{t-1}||^2 + \eta^2(AL(\mathbf{w}_{t-1}, z) + B) \\ + 2\eta\left[L(\mathbf{w}^*, z) - L(\mathbf{w}_{t-1}, z)\right] + 2Kg_{0,t}\left(||\mathbf{w}^*||_1 - ||\hat{\mathbf{w}}_t||_1\right).\end{aligned} \tag{34}$$

On the other hand, from Lemma 2 we have

$$\mathbb{E}\left(||\hat{\mathbf{w}}_t - \mathbf{w}^*||^2 - ||\tilde{\mathbf{w}}_t - \mathbf{w}^*||^2\right) \geq \varepsilon^2(|\hat{S}_\varepsilon| - |\hat{\Omega}|) + 2\pi_0 C^2\left[\left(1 - \frac{q^{\mathbf{g_t}}}{2\pi_0 p - p}\right)q^{\mathbf{g}_t} - d^*\right], \tag{35}$$

where $\mathbf{g}_t = g_{0,t}\tilde{\mathbf{k}}_t$.

Since $\pi_0 \in \left(\frac{1}{2} + \frac{(q^{\mathbf{g}})^2}{2p(q^{\mathbf{g}} - d^*)}, 1\right)$ and, by Assumption 3, $q^{\mathbf{g}} > d^*$ for $t = 1, \ldots, T$, we have

$$2\pi_0 C^2\left[\left(1 - \frac{q^{\mathbf{g}_t}}{2\pi_0 p - p}\right)q^{\mathbf{g}_t} - d^*\right] \geq 0, \quad \text{for } t = 1, \ldots, T.$$

Thus, we can further write (35) as

$$\mathbb{E}\left(||\tilde{\mathbf{w}}_t - \mathbf{w}^*||^2\right) \leq \mathbb{E}\left(||\hat{\mathbf{w}}_t - \mathbf{w}^*||^2\right) - \varepsilon^2(|\hat{S}_\varepsilon| - |\hat{\Omega}|). \tag{36}$$

By rearranging (36) and combining it with (34), we get

$$\begin{aligned}&\mathbb{E}\left(\left(1 - \frac{1}{2}\eta A\right)L(\mathbf{w}_{t-1}, z_t) + \frac{Kg_{0,t}}{\eta}||\mathbf{w}_{t-1}||_1\right) \\ \leq&\mathbb{E}\left(L(\mathbf{w}^*, z_t) + \frac{Kg_{0,t}}{\eta}||\mathbf{w}^*||_1\right) + \frac{1}{2\eta}\mathbb{E}\left(||\mathbf{w}^* - \mathbf{w}_{t-1}||^2 - ||\mathbf{w}^* - \tilde{\mathbf{w}}_t||^2\right) + \frac{\eta B}{2} \\ &- \frac{\varepsilon^2}{2\eta}(|\hat{S}_{\varepsilon t}| - |\hat{\Omega}|).\end{aligned} \tag{37}$$

Based on Algorithm 3, let $\underline{\mathbf{w}}_t$ be the output weight vector at the end of time t and let $\underline{g}_t$ be the applied gravity parameter at time t, where

$$\underline{g}_{0,t} = \begin{cases} \frac{g_{0,t}}{\eta}, & \text{if } t/K \text{ is an integer;} \\ 0, & \text{otherwise.}\end{cases}$$

$$\underline{g}_t = \begin{cases} \tilde{\mathbf{k}}\frac{g_{0,t}}{\eta}, & \text{if } t/K \text{ is an integer;} \\ 0, & \text{otherwise.}\end{cases}$$

and,

$$\underline{\mathbf{w}}_t = \begin{cases} \tilde{\mathbf{w}}_t, & \text{if } t/(Kn_K) \text{ is an integer;} \\ \hat{\mathbf{w}}_t, & \text{otherwise.}\end{cases}$$

By initializing the weight vector as a vector of zeros of length p, we sum up (37) over $t = 1, \ldots, T$ with telescoping to obtain the following result:

$$\begin{aligned}&\left(1 - \frac{1}{2}\eta A\right)\mathbb{E}\left(\sum_{t=1}^{T}\left[L(\underline{\mathbf{w}}_t, z_t) + \frac{K\underline{g}_{0,t}}{1 - 1/2\eta A}||\underline{\mathbf{w}}_t||_1\right]\right) \\ \leq&\mathbb{E}\left(\sum_{t=1}^{T} L(\mathbf{w}^*, z_t) + K\underline{g}_{0,t}||\mathbf{w}^*||_1\right) + \frac{1}{2\eta}\mathbb{E}\left(||\mathbf{w}^* - \underline{\mathbf{w}}_1||^2 - ||\mathbf{w}^* - \underline{\mathbf{w}}_T||^2\right) + \frac{\eta BT}{2} \\ &- \frac{1}{2\eta}\sum_{t=1}^{T}\varepsilon_t^2 I(\frac{t}{Kn_K} \in \mathbb{Z})(|\hat{S}_{\varepsilon,t}| - |\hat{\Omega}_t|) \\ \leq&\mathbb{E}\left(\sum_{t=1}^{T} L(\mathbf{w}^*, z_t) + K\underline{g}_{0,t}||\mathbf{w}^*||_1\right) + \frac{||\mathbf{w}^*||^2}{2\eta} + \frac{\eta BT}{2} - \frac{1}{2\eta}\sum_{t=1}^{T}\varepsilon_t^2 I(\frac{t}{Kn_K} \in \mathbb{Z})(|\hat{S}_{\varepsilon,t}| - |\hat{\Omega}_t|).\end{aligned} \tag{38}$$

The theorem follows by rearrange terms in (38) and that $\tilde{k}_{t,j} \leq K$ for $j = 1, \ldots, p$ and $t = 1, \ldots, T$.

■

References

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Léon Bottou. Online learning and stochastic approximations. *Online Learning in Neural Networks*, 17(9):142, 1998.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Deng Cai and Xiaofei He. Manifold adaptive experimental design for text categorization. *Knowledge and Data Engineering, IEEE Transactions on*, 24(4):707–719, 2012.

Bob Carpenter. Lazy sparse stochastic gradient descent for regularized multinomial logistic regression. *Alias-i, Inc., Tech. Rep*, pages 1–20, 2008.

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of online learning algorithms. *Information Theory, IEEE Transactions on*, 50(9):2050–2057, 2004.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

John Duchi and Yoram Singer. Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, 10:2899–2934, 2009.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

John C Duchi, Shai Shalev-Shwartz, Yoram Singer, and Ambuj Tewari. Composite objective mirror descent. In *COLT*, pages 14–26. Citeseer, 2010.

Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Result analysis of the nips 2003 feature selection challenge. In *Advances in Neural Information Processing Systems*, pages 545–552, 2004.

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv preprint arXiv:1509.01240*, 2015.

Samuel Kutin and Partha Niyogi. Almost-everywhere algorithmic stability and generalization error. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 275–282. Morgan Kaufmann Publishers Inc., 2002.

John Langford, Lihong Li, and Tong Zhang. Sparse online learning via truncated gradient. In *Advances in Neural Information Processing Systems*, pages 905–912, 2009.

Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

Andrew P Morris and Eleftheria Zeggini. An evaluation of statistical approaches to rare variant analysis in genetic association studies. *Genetic Epidemiology*, 34(2):188–193, 2010.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, 2009.

Hidekazu Oiwa, Shin Matsushima, and Hiroshi Nakagawa. Frequency-aware truncated methods for sparse online learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 533–548. Springer, 2011.

Hidekazu Oiwa, Satoru Matsushima, and Hirotoshi Nakagawa. Healing truncation bias: self-weighted truncation framework for dual averaging. In *2012 IEEE Twelfth International Conference on Data Mining (ICDM)*, pages 575–584. IEEE, 2012.

Alexander Rakhlin, Sayan Mukherjee, and Tomaso Poggio. Stability results in learning theory. *Analysis and Applications*, 3(04):397–417, 2005.

Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for l 1-regularized loss minimization. *The Journal of Machine Learning Research*, 12:1865–1892, 2011.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, 2010.

Wei Sun, Junhui Wang, and Yixin Fang. Consistent selection of tuning parameters via variable selection stability. *The Journal of Machine Learning Research*, 14(1):3419–3440, 2013.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Panos Toulis, Dustin Tran, and Edoardo M Airoldi. Stability and optimality in stochastic gradient descent. *arXiv preprint arXiv:1505.02417*, 2015.

Dayong Wang, Pengcheng Wu, Peilin Zhao, and Steven CH Hoi. A framework of sparse online learning and its applications. *arXiv preprint arXiv:1507.07146*, 2015.

Jialei Wang, Peilin Zhao, Steven CH Hoi, and Rong Jin. Online feature selection and its applications. *Knowledge and Data Engineering, IEEE Transactions on*, 26(3):698–710, 2014.

Yue Wu, Steven CH Hoi, Tao Mei, and Nenghai Yu. Large-scale online feature selection for ultra-high dimensional sparse data. *arXiv preprint arXiv:1409.7794*, 2014.

Lin Xiao. Dual averaging method for regularized stochastic learning and online optimization. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2009.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 116. ACM, 2004.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex J Smola. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2010.

Journal of Machine Learning Research 18 (2017) 1-31 Submitted 2/17; Revised 7/17; Published 12/17

Active-set Methods for Submodular Minimization Problems

K. S. Sesh Kumar S.KARRI@IMPERIAL.AC.UK
Imperial College London, Department of Computing
180 Queen's Gate, London SW7 2AZ

Francis Bach FRANCIS.BACH@ENS.FR
INRIA - Sierra Project-team
Département d'Informatique de l'Ecole Normale Supérieure (UMR CNRS/ENS/INRIA)
2, rue Simone Iff
75012 Paris, France

Editor: Sebastian Nowozin

Abstract

We consider the submodular function minimization (SFM) and the quadratic minimization problems regularized by the Lovász extension of the submodular function. These optimization problems are intimately related; for example, min-cut problems and total variation denoising problems, where the cut function is submodular and its Lovász extension is given by the associated total variation. When a quadratic loss is regularized by the total variation of a cut function, it thus becomes a total variation denoising problem and we use the same terminology in this paper for "general" submodular functions. We propose a new active-set algorithm for total variation denoising with the assumption of an oracle that solves the corresponding SFM problem. This can be seen as local descent algorithm over ordered partitions with explicit convergence guarantees. It is more flexible than the existing algorithms with the ability for warm-restarts using the solution of a closely related problem. Further, we also consider the case when a submodular function can be decomposed into the sum of two submodular functions F_1 and F_2 and assume SFM oracles for these two functions. We propose a new active-set algorithm for total variation denoising (and hence SFM by thresholding the solution at zero). This algorithm also performs local descent over ordered partitions and its ability to warm start considerably improves the performance of the algorithm. In the experiments, we compare the performance of the proposed algorithms with state-of-the-art algorithms, showing that it reduces the calls to SFM oracles.

Keywords: discrete optimization, submodular function minimization, convex optimization, cut functions, total variation denoising.

1. Introduction

Submodular optimization problems such as total variation denoising and submodular function minimization are convex optimization problems which are common in computer vision, signal processing and machine learning (Bach, 2013), with notable applications to graph cut-based image segmentation (Boykov et al., 2001), sensor placement (Krause and Guestrin, 2011), or document summarization (Lin and Bilmes, 2011).

Let F be a normalized submodular function defined on $V = \{1, \dots, n\}$, i.e., $F : 2^V \to \mathbb{R}$ such that $F(\varnothing) = 0$ and an n-dimensional real vector u, i.e., $u \in \mathbb{R}^n$. In this paper, we consider the submodular function minimization (SFM) problem,

$$\min_{A \subset V} F(A) - u(A), \tag{1}$$

where we use the convention $u(A) = u^\top 1_A$ and $1_A \in \{0,1\}^n$ is the indicator vector of the set A. Note that general submodular functions can always be decomposed into a normalized submodular function, F, i.e., $F(\varnothing) = 0$ and a modular function u (see Bach, 2013).

Let f be the Lovász extension of the submodular function F. Let us consider the following continuous optimization problem

$$\min_{w \in [0,1]^n} f(w) - u^\top w. \tag{2}$$

As a consequence of submodularity, the discrete and continuous optimization problems in Eq. 1 and Eq. 2 respectively have the same optimal solutions (Lovász, 1982). Let us consider another related continuous optimization problem

$$\min_{w \in \mathbb{R}^n} f(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2. \tag{3}$$

If F is a cut function in a weighted undirected graph, then f is its associated total variation, hence the denomination of *total variation denoising* (TV) problem for Eq. 3, which we use in this paper—since it is equivalent to minimizing $\frac{1}{2}\|u - w\|_2^2 + f(w)$. The unique solution of the total variation denoising in Eq. 3 can be used to obtain the solution of the SFM problem in Eq. 1 by thresholding at 0. Conversely, we may obtain the optimal solution of the total variation denoising in Eq. 3 by solving a series of SFM problems using divide-and-conquer strategy.

Relationship with existing work. Generic algorithms to optimize SFM in Eq. 1 or TV in Eq. 3 problems which only access F through function values, e.g., subgradient descent or min-norm-point algorithm (Fujishige, 1984), are too slow without any assumptions (Bach, 2013), as for signal processing applications, high precision is typically required (and often the exact solution).

For decomposable problems, i.e., when $F = F_1 + \cdots + F_r$, where each F_j is "simple", some algorithms use more powerful oracles than function evaluations, improving the running times. These powerful oracles include SFM oracles that can solve the SFM problem of simple submodular function, F_j given by

$$\min_{A \subset V} F_j(A) - u_j(A), \tag{4}$$

where $u_j \in \mathbb{R}^n$. The other set of powerful oracles are total variation or TV oracles, that solve TV problems of the form

$$\min_{w \in \mathbb{R}^n} f_j(w) - u_j^\top w + \tfrac{1}{2}\|w\|_2^2, \tag{5}$$

where $u_j \in \mathbb{R}^n$. Note that, in general, the exact total variation oracles are at most $O(n)$ times more expensive than their respective SFM oracle as they solve all SFM problems

$$\min_{A \subset V} F_j(A) - u_j(A) + \lambda|A|, \tag{6}$$

for all $\lambda \in \mathbb{R}$, which have at most $O(n)$ unique solutions. For more details refer to Fujishige (1980) and Bach (2013). Here, $|A|$ denotes the cardinality of the set A. There does exist a subclass of submodular functions (cut functions and other submodular functions that can be written in form of

©2017 K. S. Sesh Kumar and Francis Bach.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-101.html.

cuts) whose total variation oracles are only $O(1)$ times more expensive than the corresponding SFM oracles but are still too expensive in practice.

Stobbe and Krause (2010) used SFM oracles instead of function value oracles but their algorithm remains slow in practice. However, when total variation oracles for each F_j are used, they become competitive (Komodakis et al., 2011; Kumar et al., 2015; Jegelka et al., 2013). Therefore, our goal is to design fast optimization strategies using only efficient SFM oracles for each function F_j rather than their expensive TV oracles (Kumar et al., 2015; Jegelka et al., 2013) to solve the SFM and TV denoising problems of F given by Eq. 1 and Eq. 3 respectively. An algorithm was proposed by Landrieu and Obozinski (2016) to search over partition space for solving Eq. 3 with the unary terms $(-u^\top w)$ replaced by a convex differentiable function but it applies only to functions F, which are cut functions.

In this paper, we exploit the polytope structure of these non-smooth optimization problems with exponentially many constraints, i.e., 2^n, where each face of the constraint set is indexed by an ordered partition of the underlying ground set $V = \{1, \dots, n\}$. The main insight of this paper is that given the main polytope associated with a submodular function (namely the base polytope described in Section 2) and an ordered partition, we may uniquely define a tangent cone of the polytope. Further, orthogonal projections onto the tangent cone may be done efficiently by isotonic regressions (Best and Chakravarti, 1990). The time needed is linear in the number of elements of the ordered partition used to define the tangent cone. We need SFM oracles only to check the optimality of the ordered partition. Given the orthogonal projection s onto the tangent cone, if the minimum of $F(A) - s(A)$ with respect to $A \subseteq V$ is positive then it is optimal. If it is not optimal, it gives us the violating constraints in the form of active-sets that enable us to generate a new ordered partition among the exponentially many ordered partitions.

Contributions. We make two main contributions:

- Given a submodular function F with an SFM oracle, we propose a new active-set algorithm for total variation denoising in Section 3, which is more efficient and flexible than existing ones. This algorithm may be seen as a local descent algorithm over ordered partitions. It has the additional advantage of allowing warm restarts, which will be beneficial when we have to solve a large number of total variation denoising problems as shown in Section 5.
- Given a decomposition of $F = F_1 + F_2$, with available SFM oracles for each F_j, we propose an active-set algorithm for total variation denoising in Section 4 (and hence for SFM by thresholding the solution at zero). These algorithms optimize over ordered partitions (one per function F_j). Following Jegelka et al. (2013) and Kumar et al. (2015), they are also naturally parallelizable. Given that only SFM oracles are needed, it is much more flexible than the algorithms requiring a TV oracle, and allow more applications as shown in Section 5.

2. Review of Submodular Analysis

A set-function $F : 2^V \to \mathbb{R}$ is submodular if $F(A) + F(B) \geqslant F(A \cup B) + F(A \cap B)$ for any subsets A, B of V. Our main motivating examples in this paper are cuts in a weighted undirected graph with weight function $a : V \times V \to \mathbb{R}_+$, which can be defined as

$$F(A) = \sum_{i<j} a(i,j) |1_{i \in A} - 1_{j \notin A}|, \tag{7}$$

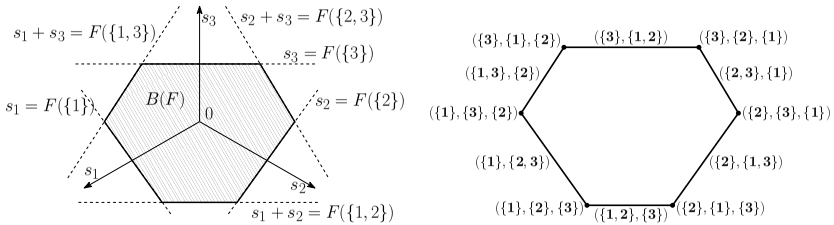


Figure 1: Base polytope for $n=3$. (a) definition from its supporting hyperplanes $\{s(A) = F(A)\}$. (b) each face (point or segment) of $B(F)$ is associated with an ordered partition.

where $i, j \in V$. Note that there are other submodular functions on which our algorithm works, e.g., concave function on the cardinality set, which *cannot* be represented in the form of Eq. 7 (Ladicky et al., 2010; Kolmogorov, 2012). However, we use cut functions as a running example to better explain our algorithm as they are most widely studied and understood among submodular functions. We now review the relevant concepts from submodular analysis (for more details, see Bach, 2013; Fujishige, 2005).

Lovász extension and convexity. The power set 2^V is naturally identified with the vertices $\{0,1\}^n$ of the hypercube in n dimensions (going from $A \subseteq V$ to $1_A \in \{0,1\}^n$). Thus, any set-function may be seen as a function f on $\{0,1\}^n$. It turns out that f may be extended to the full hypercube $[0,1]^n$ by piecewise-linear interpolation, and then to the whole vector space $\mathbb{R}^n$.

Given a vector $w \in \mathbb{R}^n$, and given its *unique* level-set representation as $w = \sum_{i=1}^m v_i 1_{A_i}$, with $(A_1, \dots, A_m)$ a partition of V and $v_1 > \cdots > v_m$, $f(w)$ is equal to $f(w) = \sum_{i=1}^m v_i \big[F(B_i) - F(B_{i-1})\big]$, where $B_i = (A_1 \cup \cdots \cup A_i)$. For cut functions, the Lovász extension happens to be equal to the *total variation*, $f(w) = \sum_{i<j} a(i,j)|w_i - w_j|$, hence our denomination total variation denoising for the problem in Eq. 3.

This extension is piecewise linear for any set-function F. It turns out that it is convex if and only if F is submodular (Lovász, 1982). Any piecewise linear convex function may be represented as the support function of a certain polytope K, i.e., as $f(w) = \max_{s \in K} w^\top s$ (Rockafellar, 1997). For the Lovász extension of a submodular function, there is an explicit description of K, which we now review.

Base polytope. We define the *base polytope* as

$$B(F) = \big\{ s \in \mathbb{R}^n,\ s(V) = F(V),\ \forall A \subset V, s(A) \leqslant F(A) \big\}.$$

Given that it is included in the affine hyperplane $\{s(V) = F(V)\}$, it is traditionally represented by the projection on that hyperplane (see Figure 1 (a)). A key result in submodular analysis is that the Lovász extension is the support function of $B(F)$, that is, for any $w \in \mathbb{R}^n$,

$$f(w) = \sup_{s \in B(F)} w^\top s. \tag{8}$$

The maximizers above may be computed in closed form from an ordered level-set representation of w using a greedy algorithm, which (a) first sorts the elements of w in decreasing order such

that $w_{\sigma(1)} \geq \ldots \geq w_{\sigma(n)}$ where σ represents the order of the elements in V; and (b) computes $s_{\sigma(k)} = F(\{\sigma(1), \ldots, \sigma(k)\}) - F(\{\sigma(1), \ldots, \sigma(k-1)\})$.

SFM as a convex optimization problem. Another key result of submodular analysis is that minimizing a submodular function F (i.e., minimizing the Lovász extension f on $\{0,1\}^n$), is equivalent to minimizing the Lovász extension f on the full hypercube $[0,1]^n$ (a convex optimization problem). Moreover, with convex duality we have

$$\begin{aligned}\min_{A \subseteq V} F(A) - u(A) &= \min_{w \in \{0,1\}^n} f(w) - u^\top w = \min_{w \in [0,1]^n} f(w) - u^\top w \\ &= \min_{w \in [0,1]^n} \max_{s \in B(F)} s^\top w - u^\top w \\ &= \max_{s \in B(F)} \min_{w \in [0,1]^n} s^\top w - u^\top w = \max_{s \in B(F)} \sum_{i=1}^n \min\{s_i - u_i, 0\}.\end{aligned}$$

This dual problem allows to obtain certificates of optimality for the primal-dual pairs $w \in [0,1]^n$ and $s \in B(F)$ using the quantity

$$\text{gap}(w,s) := f(w) - u^\top w - \sum_{i=1}^n \min\{s_i - u_i, 0\},$$

which is always non-negative. It is equal to zero only at optimality and the corresponding (w,s) form an optimal primal-dual pair.

Total variation denoising as projection onto the base polytope. A consequence of the representation of f as a support function leads to the following primal/dual pair (Bach, 2013, Sec. 8):

$$\begin{aligned}\min_{w \in \mathbb{R}^n} f(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2 &= \min_{w \in \mathbb{R}^n} \max_{s \in B(F)} s^\top w - u^\top w + \tfrac{1}{2}\|w\|_2^2 \text{ using Eq. 8,} \\ &= \max_{s \in B(F)} \min_{w \in \mathbb{R}^n} s^\top w - u^\top w + \tfrac{1}{2}\|w\|_2^2, \\ &= \max_{s \in B(F)} -\tfrac{1}{2}\|s - u\|_2^2, \end{aligned} \quad (9)$$

with $w = u - s$ at optimality. Thus the TV problem is equivalent to the orthogonal projection of u onto $B(F)$.

From TV denoising to SFM. The SFM problem in Eq. 1 and the TV problem in Eq. 3 are tightly connected. Indeed, given the unique solution w of the TV problem, we obtain a solution of $\min_{A \subseteq V} F(A) - u(A)$ by thresholding w at 0, i.e., by taking $A = \{i \in V, w_i \geqslant 0\}$ (Fujishige, 1980).

Conversely, one may solve the TV problem by an appropriate sequence of SFM problems. The original divide-and-conquer algorithm may involve $O(n)$ SFM problems (Groenevelt, 1991). The extended algorithm of Jegelka et al. (2013) can reach a precision ε in $O(\log \frac{1}{\varepsilon})$ iterations but can only get the exact solution in $O(n)$ oracles. Fast efficient algorithms are proposed to solve TV problems with $O(1)$ oracles (Chambolle and Darbon, 2009; Goldfarb and Yin, 2009) but are specific to cut functions on simple graphs (chains and trees) as they exploit the weight representation given by Eq. 7. Our algorithm in Section 3 is a generalization of the divide-and-conquer strategy for solving the TV problem with general submodular functions.

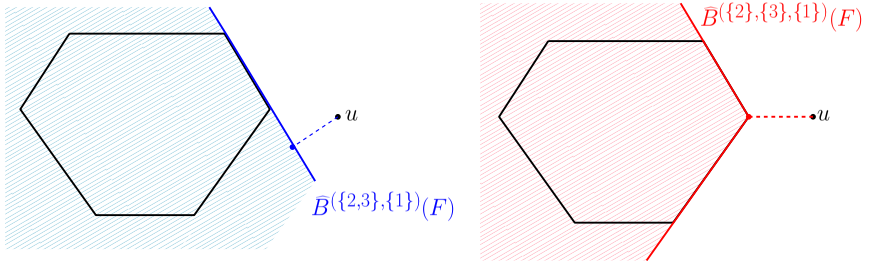


Figure 2: Projection algorithm for a single polytope: first projecting on the outer approximation $\widehat{B}^{(\{2,3\},\{1\})}(F)$, with a projected element which is not in $B(F)$ (blue), then on $\widehat{B}^{(\{2\},\{3\},\{1\})}(F)$, with a projected element being the projection of s onto $B(F)$ (red).

3. Ordered Partitions and Isotonic Regression

The main insight of this paper is (a) to consider the detailed face structure of the base polytope $B(F)$ and (b) to notice that for the outer approximation of $B(F)$ based on the tangent cone to a certain face, the orthogonal projection problem (which is equivalent to constrained TV denoising) may be solved efficiently using a simple algorithm, originally proposed to solve isotonic regression in linear time. This allows an explicit efficient local search over ordered partitions.

3.1 Outer Approximations of $B(F)$

Supporting hyperplanes. The base polytope is defined as the intersection of half-spaces $\{s(A) \leqslant F(A)\}$, for $A \subseteq V$. Therefore, faces of $B(F)$ are indexed by subsets of the power set. As a consequence of submodularity (Bach, 2013; Fujishige, 2005), the faces of the base polytope $B(F)$ are characterized by "ordered partitions" $\mathcal{A} = (A_1, \ldots, A_m)$ with $V = A_1 \cup \cdots \cup A_m$. Then, a face of $B(F)$ is such that $s(B_i) = F(B_i)$ for all $B_i = A_1 \cup \cdots \cup A_i$, $i = 1, \ldots, m$. See Figure 1 (b) for the enumeration of faces for $n = 3$ based on an enumeration of all ordered partitions. Such ordered partitions are associated to vectors $w = \sum_{i=1}^m v_i 1_{A_i}$ with $v_1 > \cdots > v_m$ with all solutions of $\max_{s \in B(F)} w^\top s$ being on the corresponding face.

From a face of $B(F)$ defined by the ordered partition $\mathcal{A}$, we may define its *tangent cone* $\widehat{B}^{\mathcal{A}}(F)$ at this face as the set

$$\widehat{B}^{\mathcal{A}}(F) = \Big\{ s \in \mathbb{R}^n, s(V) = F(V), \forall i \in \{1, \ldots, m-1\}, s(B_i) \leqslant F(B_i) \Big\}. \quad (10)$$

Since we have relaxed all the constraints unrelated to $\mathcal{A}$, these are outer approximations of $B(F)$ as illustrated in Figure 2 for two ordered partitions.

Support function. We may compute the support function of $\widehat{B}^{\mathcal{A}}(F)$, which is an upper bound on $f(w)$ since this set is an outer approximation of $B(F)$ as follows:

$$\sup_{s \in \widehat{B}^{\mathcal{A}}(F)} w^\top s = \sup_{s \in \mathbb{R}^n} \inf_{\lambda \in \mathbb{R}_+^{m-1} \times \mathbb{R}} w^\top s - \sum_{i=1}^m \lambda_i (s(B_i) - F(B_i)) \text{ , using Lagrangian duality,}$$

$$
\begin{aligned}
&= \inf_{\lambda \in \mathbb{R}_+^{m-1} \times \mathbb{R}} \sup_{s \in \mathbb{R}^n} s^\top \Big(w - \sum_{i=1}^m (\lambda_i + \cdots + \lambda_m) 1_{A_i} \Big) \\
&\qquad\qquad + \sum_{i=1}^m (\lambda_i + \cdots + \lambda_m) \big[F(B_i) - F(B_{i-1}) \big], \\
&= \inf_{\lambda \in \mathbb{R}_+^{m-1} \times \mathbb{R}} \sum_{i=1}^m (\lambda_i + \cdots + \lambda_m) \big[F(B_i) - F(B_{i-1}) \big] \\
&\qquad\qquad \text{such that } w = \sum_{i=1}^m (\lambda_i + \cdots + \lambda_m) 1_{A_i}.
\end{aligned}
$$

Thus, by defining $v_i = \lambda_i + \cdots + \lambda_m$, which are decreasing, the support function is finite for w having ordered level sets corresponding to the ordered partition $\mathcal{A}$ (we then say that w is *compatible* with $\mathcal{A}$). In other words, if $w = \sum_{i=1}^m v_i 1_{A_i}$, the support functions is equal to the Lovász extension $f(w)$. Otherwise, when w is not compatible with $\mathcal{A}$, the support function is infinite.

Let us now denote $\mathcal{W}^{\mathcal{A}}$ as a set of all weight vectors w that are compatible with the ordered partition $\mathcal{A}$. This can be defined as

$$
\mathcal{W}^{\mathcal{A}} = \Big\{ w \in \mathbb{R}^n \mid \exists v \in \mathbb{R}^m, w = \sum_{i=1}^m v_i 1_{A_i}, v_1 \geq \ldots \geq v_m \Big\}.
$$

Therefore,

$$
\sup_{s \in \widehat{B}^{\mathcal{A}}(F)} w^\top s = \begin{cases} f(w) & \text{if } w \in \mathcal{W}^{\mathcal{A}}, \\ \infty & \text{otherwise.} \end{cases} \tag{11}
$$

3.2 Isotonic Regression for Restricted Problems

Given an ordered partition $\mathcal{A} = (A_1, \ldots, A_m)$ of V, we consider the original TV problem restricted to w in $\mathcal{W}^{\mathcal{A}}$. Since on this constraint set $f(w) = \sum_{i=1}^m v_i \big[F(B_i) - F(B_{i-1}) \big]$ is a linear function, this is equivalent to

$$
\min_{v \in \mathbb{R}^m} \sum_{i=1}^m v_i \big[F(B_i) - F(B_{i-1}) - u(A_i) \big] + \tfrac{1}{2} \textstyle\sum_{i=1}^m |A_i| v_i^2 \text{ such that } v_1 \geqslant \cdots \geqslant v_m. \tag{12}
$$

This may be done by isotonic regression in complexity $O(m)$ by the weighted pool-adjacent-violator algorithm (Best and Chakravarti, 1990). Typically the solution v will have some values that are equal to each other, which corresponds to merging some sets A_i. If these merges are made, we now obtain a *basic ordered partition*[1] such that our optimal w has *strictly decreasing* values. Because none of the constraints are tight, primal stationarity leads to explicit values of v given by $v_i = u(A_i)/|A_i| - (F(B_i) - F(B_{i-1}))/|A_i|$, i.e., given $\mathcal{A}$, the exact solution of the TV problem may be obtained in closed form.

Dual interpretation. Eq. 12 is a constrained TV denoising problem that minimizes the cost function in Eq. 3 but with the constraint that weights are compatible with the ordered partition $\mathcal{A}$,

1. Given a submodular function F and an ordered partition $\mathcal{A}$, when the unique solution problem in Eq. 12 is such that $v_1 > \cdots > v_m$, we say that we $\mathcal{A}$ is a *basic ordered partition* for $F - u$. Given any ordered partition, isotonic regression allows to compute a coarser partition (obtained by partially merging some sets) which is basic.

i.e., $\min_{w \in \mathcal{W}^{\mathcal{A}}} f(w) - u^\top w + \frac{1}{2}\|w\|_2^2$. The dual of the problem can be derived in exactly the same way as shown in Eq. 9 in the previous section, using the definition of the support function defined by Eq. 11. The corresponding dual is given by $\max_{s \in \widehat{B}^{\mathcal{A}}(F)} -\frac{1}{2}\|s - u\|_2^2$, with the relationship $w = u - s$ at optimality. Thus, this corresponds to projecting u onto the outer approximation of the base polytope, $\widehat{B}^{\mathcal{A}}(F)$, which only has m constraints instead of the $2^n - 1$ constraints defining $B(F)$. See an illustration in Figure 2.

3.3 Checking Optimality of a Basic Ordered Partition

Given a basic ordered partition $\mathcal{A}$, the associated $w \in \mathbb{R}^n$ obtained from Eq. 12 is optimal for the TV problem in Eq. 3 if and only if $s = u - w \in B(F)$ due to optimality conditions in Eq. 9, which can be checked by minimizing the submodular function $F - s$. For a basic partition, a more efficient algorithm is available.

By repeated application of submodularity, we have for all sets $C \subseteq V$, if $C_i = C \cap A_i$:

$$
\begin{aligned}
F(C) - s(C) &= F(V \cap C) - \sum_{i=1}^m s(C_i) \text{ (as } s \text{ is a modular function)}, \\
&= F(B_m \cap C) - \sum_{i=1}^m s(C_i) + \sum_{i=1}^{m-1} F(B_i \cap C) - F(B_i \cap C) \text{ (as } B_m = V), \\
&= \sum_{i=1}^m F(B_i \cap C) - F(B_{i-1} \cap C) - s(C_i) \text{ (let } B_0 = \varnothing \text{ and as } F(\varnothing) = 0), \\
&= \sum_{i=1}^m F((B_{i-1} \cup A_i) \cap C) - F(B_{i-1} \cap C) - s(C_i) \text{ (since } B_i = B_{i-1} \cup A_i), \\
&= \sum_{i=1}^m F((B_{i-1} \cap C) \cup (A_i \cap C)) - F(B_{i-1} \cap C) - s(C_i), \\
&= \sum_{i=1}^m F((B_{i-1} \cap C) \cup C_i) - F(B_{i-1} \cap C) - s(C_i), \\
&\geqslant \sum_{i=1}^m \big[F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i) \big] \\
&\qquad\qquad \text{(as } (B_{i-1} \cap C) \subseteq B_{i-1} \text{ and due to submodularity of } F).
\end{aligned}
$$

Moreover, we have $s(A_i) = F(B_i) - F(B_{i-1})$, which implies $s(B_i) = F(B_i)$ for all $i \in \{1, \ldots, m\}$, and thus all subproblems $\min_{C_i \subseteq A_i} F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i)$ have non-positive values. This implies that we may check optimality by solving these m subproblems: s is optimal if and only if all of them have zero values. This leads to smaller subproblems whose overall complexity is less than a single SFM oracle call. Moreover, for cut functions, it may be solved by a single oracle call on a graph where some edges have been removed (Tarjan et al., 2006).

Given all sets C_i, we may then define a new ordered partition by splitting all A_i for which $F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i) < 0$. If no split is possible, the pair (w, s) is optimal for Eq. 3. Otherwise, this new strictly finer partition may not be basic, the value of the optimization problem

in Eq. 12 is strictly lower as shown in Section 3.5 (and leads to another basic ordered partition), which ensures the finite convergence of the algorithm.

3.4 Active-set Algorithm

This leads to the novel active-set algorithm below.

Data: Submodular function F with SFM oracle, $u \in \mathbb{R}^n$, ordered partition $\mathcal{A}$
Result: primal optimal: $w \in \mathbb{R}^n$ and dual optimal: $s \in B(F)$
while `True` **do**
  Solve Eq. 12 using isotonic regression and update $\mathcal{A}$ with the basic ordered partition ;
  Check optimality by solving $\min_{C_i \subseteq A_i} F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i)$ for $i \in \{1, \dots, m\}$;
  if *s is optimal* **then**
    `break` ;
  else
    for $i \in \{1, \dots, m\}$, split the set A_i into C_i and $A_i \setminus C_i$ in that order to get an updated ordered partition $\mathcal{A}$;
  end
end

Relationship with divide-and-conquer algorithm. When starting from the trivial ordered partition $\mathcal{A} = (V)$, then we exactly obtain a parallel version of the divide-and-conquer algorithm (Groenevelt, 1991), that is, the isotonic regression problem is always solved without using the constraints of monotonicity, i.e., there are no merges, it is not necessary to re-solve the problems where nothing has changed. This shows that the number of iterations is then less than n.

The key added benefits in our formulation is the possibility of warm-starting, which can be very useful for building paths of solutions with different weights on the total variation. This is also useful for decomposable functions where many TV oracles are needed with close-by inputs. See experiments in Section 5.

3.5 Proof of Convergence

In order to prove the convergence of the algorithm in Section 3.4, we only need to show that if the optimality check fails in step (4), then step (7) introduces splits in the partition, which ensures that the isotonic regression in step (2) of the next iteration has a strictly lower value. Let us recall the isotonic regression problem solved in step (2):

$$\min_{v \in \mathbb{R}^m} \sum_{i=1}^{m} \Big(v_i \big[F(B_i) - F(B_{i-1}) - u(A_i) \big] + \tfrac{1}{2} |A_i| v_i^2 \Big) \tag{13}$$

$$\text{such that } v_1 \geqslant \cdots \geqslant v_m. \tag{14}$$

Steps (2) ensures that the ordered partition $\mathcal{A}$ is a basic ordered partition warranting that the inequality constraints are strict, i.e., no two partitions have the same value v_i and the values v_i for each element of the partition $i = \{1, \dots, m\}$ is given through

$$v_i |A_i| = u(A_i) - (F(B_i) - F(B_{i-1})), \tag{15}$$

which can be calculated in closed form.

The optimality check in step (4) decouples into checking the optimality in each subproblem as shown in Section 3.3. If the optimality test fails, then there is a subset of C_i of A_i for some of elements of the partition $\mathcal{A}$ such that

$$F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i) < 0. \tag{16}$$

We will show that the splits introduced by step (7) strictly reduces the function value of isotonic regression in Eq. 13, while maintaining the feasibility of the problem. The splits modify the cost function of the isotonic regression as follows, as the objective function in Eq. 13 is equal to

$$\sum_{i=1}^{m} \Big(v_i \big[F(B_{i-1} \cup C_i) - F(B_{i-1}) - u(C_i) \big] + v_i \big[F(B_i) - F(B_{i-1} \cup C_i) - u(A_i \setminus C_i) \big] + \tfrac{1}{2} v_i^2 |C_i| + \tfrac{1}{2} v_i^2 |A_i \setminus C_i| \Big). \tag{17}$$

Let us assume a positive $t \in \mathbb{R}$, which is small enough. The direction that the isotonic regression moves is $v_i + t$ for the partition corresponding to C_i and $v_i - t$ for the partition corresponding to $A_i \setminus C_i$ maintaining the feasibility of the isotonic regression problem, i.e., $v_1 \geqslant \cdots \geqslant v_i + t > v_i - t \geqslant \cdots \geqslant v_m$. The function value is given by

$$\begin{aligned}
&\sum_{i=1}^{m} \Big((v_i + t) \big[F(B_{i-1} \cup C_i) - F(B_{i-1}) - u(C_i) \big] \\
&\quad + (v_i - t) \big[F(B_i) - F(B_{i-1} \cup C_i) - u(A_i \setminus C_i) \big] + \tfrac{1}{2} (v_i + t)^2 |C_i| + \tfrac{1}{2} (v_i - t)^2 |A_i \setminus C_i| \Big) \\
= \; &\sum_{i=1}^{m} \Big(\big(v_i \big[F(B_{i-1} \cup C_i) - F(B_{i-1}) - u(C_i) \big] + v_i \big[F(B_i) - F(B_{i-1} \cup C_i) - u(A_i \setminus C_i) \big] \\
&\quad + \tfrac{1}{2} v_i^2 |C_i| + \tfrac{1}{2} v_i^2 |A_i \setminus C_i| \big) + \tfrac{1}{2} t^2 |A_i| \\
&\quad + t \big(2F(B_{i-1} \cup C_i) - F(B_{i-1}) - F(B_i) - u(C_i) + u(A_i \setminus C_i) + v_i |C_i| - v_i |A_i \setminus C_i| \big) \Big).
\end{aligned}$$

From this we can compute the directional derivative of the function at $t = 0$, which is given by

$$\begin{aligned}
& 2F(B_{i-1} \cup C_i) - F(B_{i-1}) - F(B_i) - u(C_i) + u(A_i \setminus C_i) + |C_i| v_i - |A_i \setminus C_i| v_i \\
= \; & 2F(B_{i-1} \cup C_i) - F(B_{i-1}) - F(B_i) - 2u(C_i) + u(A_i) + 2|C_i| v_i - |A_i| v_i \\
= \; & 2\big(F(B_{i-1} \cup C_i) - F(B_{i-1}) - u(C_i) + v_i |C_i| \big) \text{ (substituting Eq. 15)} \\
= \; & 2\big(F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i) \big) < 0 \text{ (as } s = u - w \text{ and Eq. 16).}
\end{aligned}$$

This shows that the function strictly decreases with the splits introduced in step (7).

3.6 Discussion

Certificates of optimality. The algorithm has dual-infeasible iterates s (they only belong to $B(F)$ at convergence). Suppose that after step (3) we have $F(C) - s(C) \geqslant -\varepsilon$ for all $C \subset V$, where ε

shrinks as we run more iterations of the outer loop. This implies that $s \in B(F + \varepsilon 1_{\mathrm{Card} \in (1,n)})$, i.e., $s \in B(F_\varepsilon)$ with $F_\varepsilon = F + \varepsilon 1_{\mathrm{Card} \in (1,n)}$. Since by construction $w = u - s$, we have:

$$\begin{aligned} f_\varepsilon(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2 + \tfrac{1}{2}\|s-u\|_2^2 &= \varepsilon \big| \max_{j \in V} w_j - \min_{j \in V} w_j \big| + f(w) - u^\top w + \|w\|^2 \\ &= \varepsilon \big| \max_{j \in V} w_j - \min_{j \in V} w_j \big| \\ &\quad + \sum_{i=1}^m v_i \big[F(B_i) - F(B_{i-1}) - u(A_i) \big] + \sum_{i=1}^m |A_i| v_i^2 \\ &= \varepsilon \big| \max_{j \in V} w_j - \min_{j \in V} w_j \big| \text{ (using Eq. 15)} \\ &= \varepsilon \, \mathrm{range}(w), \end{aligned}$$

where $\mathrm{range}(w) = \max_{k \in V} w_k - \min_{k \in V} w_k$. This means that w is approximately optimal for $f(w) - u^\top w + \frac{1}{2}\|w\|_2^2$ with *certified gap* less than $\varepsilon \, \mathrm{range}(w) + \varepsilon \, \mathrm{range}(w^*)$.

Maximal range of an active-set solution. For any ordered partition $\mathcal{A}$, and the optimal value of w (which we know in closed form), we have $\mathrm{range}(w) \leqslant \mathrm{range}(u) + \max_{i \in V} \big\{ F(\{i\}) + F(V \backslash \{i\}) - F(V) \big\}$. Indeed, for the u part of the expression, this is because values of w are averages of values of u; for the F part of the expression, we always have by submodularity:

$$\begin{aligned} F(B_i) - F(B_{i-1}) &\leqslant \sum_{k \in A_i} F(\{k\}) \text{ and} \\ F(B_i) - F(B_{i-1}) &\geqslant - \sum_{k \in A_i} F(V) - F(V \backslash \{k\}). \end{aligned}$$

This means that the certificate can be used in practice by replacing $\mathrm{range}(w^*)$ by its upperbound. See experimental evaluation for a 2D total variation denoising in Appendix A.

Exact solution. If the submodular function only takes integer values and we have an approximate solution of the TV problem with gap $\varepsilon \leqslant \frac{1}{4n}$, then we have the optimal solution (Chakrabarty et al., 2014).

Relationship with traditional active-set algorithm. Given an ordered partition $\mathcal{A}$, an active-set method solves the unconstrained optimization problem in Eq. 12 to obtain a value of v using the primary stationary conditions. The corresponding primal value $w = \sum_{i=1}^m v_i 1_{A_i}$ and dual value $s = u - w$ are optimal, if and only if,

$$\text{Primal feasibility : } w \in \mathcal{W}^{\mathcal{A}}, \tag{18}$$

$$\text{Dual feasibility : } s \in B(F). \tag{19}$$

If Eq. 18 is not satisfied, a move towards the optimal w is performed to ensure primal feasibility by performing line search, i.e., two consecutive sets A_i and A_{i+1} with increasing values $v_i < v_{i+1}$ are merged and a potential w is computed until primal feasibility is met. Then dual feasibility is checked and potential splits are proposed.

In our approach, we consider a different strategy which is more direct and does many merges simultaneously by using *isotonic regression*. Our method explicitly moves from ordered partitions to ordered partitions and computes an optimal vector w, which is always feasible.

4. Decomposable Problems

Many interesting problems in signal processing and computer vision naturally involve submodular functions F that decompose into $F = F_1 + \cdots + F_r$, with r "simple" submodular functions (Bach, 2013). For example, a cut function in a 2D grid decomposes into a function F_1 composed of cuts along vertical lines and a function F_2 composed of cuts along horizontal lines. For both of these functions, SFM oracles may be solved in $O(n)$ by message passing. For simplicity, in this paper, we consider the case $r = 2$ functions, but following Komodakis et al. (2011) and Jegelka et al. (2013), our framework easily extends to $r > 2$.

4.1 Reformulation as the Distance Between Two Polytopes

Following Jegelka et al. (2013), we have the primal/dual problems :

$$\begin{aligned} \min_{w \in \mathbb{R}^n} f_1(w) + f_2(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2 &= \min_{w \in \mathbb{R}^n} \max_{s_1 \in B(F_1),\, s_2 \in B(F_2)} w^\top (s_1 + s_2) - u^\top w + \tfrac{1}{2}\|w\|_2^2 \\ &= \max_{s_1 \in B(F_1),\, s_2 \in B(F_2)} \min_{w \in \mathbb{R}^n} (s_1 + s_2 - u)^\top w + \tfrac{1}{2}\|w\|_2^2 \\ &= \max_{s_1 \in B(F_1),\, s_2 \in B(F_2)} -\tfrac{1}{2}\|s_1 + s_2 - u\|_2^2, \end{aligned} \tag{20}$$

with $w = u - s_1 - s_2$ at optimality.

This is the projection of u on the sum of the base polytopes $B(F_1) + B(F_2) = B(F)$. Further, this may be interpreted as finding the distance between two polytopes $B(F_1) - u/2$ and $u/2 - B(F_2)$. Note that these two polytopes typically do not intersect (they will if and only if $w = 0$ is the optimal solution of the TV problem, which is an uninteresting situation). We now review Alternating projections (AP) (Jegelka et al., 2013), Averaged alternating reflections (AAR) (Jegelka et al., 2013) and Dykstra's alternating projections (DAP) (Chambolle and Pock, 2015) to show that a large number of total variation denoising problems need to be solved to obtain an optimal solution of Eq. 20. The ability to warm start and solve these total variation denoising using our algorithm in Section 3.4 can greatly improve the performance of each of these algorithms.

Alternating projections (AP). The alternating projection algorithm (Bauschke et al., 1997) was proposed to solve the convex feasibility problem, i.e., to obtain a feasible point in the intersection of two polytopes. It is equivalent to performing block coordinate descent on the dual derived in Eq. 20. Let us denote the projection onto a polytope K as Π_K, i.e., $\Pi_K(y) = \mathrm{argmax}_{x \in K} -\|x - y\|_2^2$. Therefore, alternating projections lead to the following updates for our problem.

$$z_t = \Pi_{u/2 - B(F_2)} \Pi_{B(F_1) - u/2}(z_{t-1}),$$

where z_0 is an arbitrary starting point. Thus each of these steps require TV oracles for F_1 and F_2 since projection onto the base polytope is equivalent to performing TV denoising as shown in Eq. 9.

Averaged alternating reflections (AAR). The averaged alternating reflection algorithm (Bauschke et al., 2004), which is also known as Douglas-Rachford splitting can be used to solve convex feasibility problems. It is observed to converge quicker than alternating projection (Jegelka et al., 2013; Kumar et al., 2015) in practice. We now introduce a reflection operator for the polytope K as R_K, i.e., $R_K = 2\Pi_K - I$, where I is the identity operator. Therefore, reflection of t on a polytope K is

given by $R_K(t) = 2\Pi_K(t) - t$. The updates of each iteration of the averaged alternating reflections, which starts with an auxiliary sequence z_0 initialized to 0 vector, are given by

$$z_t = \tfrac{1}{2}(I + R_{u/2-B(F_2)} R_{B(F_1)-u/2})(z_{t-1}).$$

In the feasible case, i.e., intersecting polytopes, the sequence z_t weakly converges to a point in the intersection of the polytopes. However, in our case, we have non intersecting polytopes which leads to a converging sequence of z_t with AP but a diverging sequence of z_t with AAR. However, when we project z_t by using the projection operation, $s_{1,t} = \Pi_{B(F_1)-u/2}(z_t)$; $s_{2,t} = \Pi_{u/2-B(F_2)}(s_{1,t})$ the sequences $s_{1,t}$ and $s_{2,t}$ converge to the nearest points on the polytopes, $B(F_1) - u/2$ and $u/2 - B(F_2)$ (Bauschke et al., 2004) respectively.

Dykstra's alternating projections (DAP). Dykstra's alternating projection algorithm (Bauschke and Borwein, 1994) retrieves a convex feasible point closest to an arbitrary point, which we assume to be 0. It can also be used and has a form of primal descent interpretation, i.e., as coordinate descent for a well-formulated primal problem (Gaffke and Mathar, 1989). Let us denote ι_K as the indicator function of a convex set K. In our case we consider finding the nearest points on the polytopes $B(F_1) - u/2$ and $u/2 - B(F_2)$ closest to 0, which can be formally written as:

$$\begin{aligned}
&\min_{\substack{s\in B(F_1)-\frac{u}{2}\\ s\in \frac{u}{2}-B(F_2)}} \frac{1}{2}\|s\|_2^2 \\
= \;& \min_{s\in\mathbb{R}^n} \frac{1}{2}\|s\|_2^2 + \iota_{B(F_1)-\frac{u}{2}}(s) + \iota_{\frac{u}{2}-B(F_2)}(s) \\
= \;& \min_{s\in\mathbb{R}^n} \frac{1}{2}\|s\|_2^2 + \iota_{B(F_1)-\frac{u}{2}}(s) + \iota_{B(F_2)-\frac{u}{2}}(-s) \\
= \;& \min_{s\in\mathbb{R}^n} \left(\frac{1}{2}\|s\|_2^2 + \max_{w_1\in\mathbb{R}^n} w_1^\top s - f_1(w_1) + \frac{w_1^\top u}{2} + \max_{w_2\in\mathbb{R}^n} -w_2^\top s - f_2(w_2) + \frac{w_2^\top u}{2}\right) \\
= \;& \max_{\substack{w_1\in\mathbb{R}^n\\ w_2\in\mathbb{R}^n}} \left(- f_1(w_1) - f_2(w_2) + \frac{(w_1+w_2)^\top u}{2} + \min_{s\in\mathbb{R}^n}\left(\frac{1}{2}\|s\|_2^2 + (w_1-w_2)^\top s\right)\right) \\
= \;& \max_{\substack{w_1\in\mathbb{R}^n\\ w_2\in\mathbb{R}^n}} -f_1(w_1) - f_2(w_2) + \frac{(w_1+w_2)^\top u}{2} - \frac{1}{2}\|w_1-w_2\|_2^2 \\
= \;& \min_{\substack{w_1\in\mathbb{R}^n\\ w_2\in\mathbb{R}^n}} f_1(w_1) + f_2(w_2) - \frac{(w_1+w_2)^\top u}{2} + \frac{1}{2}\|w_1-w_2\|_2^2,
\end{aligned}$$

where $s = w_2 - w_1$ at optimality. The block coordinate descent algorithm then gives

$$\begin{aligned}
w_{1,t} &= \text{prox}_{f_1-u/2}(w_{2,t-1}) = (I - \Pi_{B(F_1)-u/2})(w_{2,t-1}), \\
s_{1,t} &= w_{2,t-1} - w_{1,t}, \\
w_{2,t} &= \text{prox}_{f_2-u/2}(w_{1,t}) = (I - \Pi_{B(F_2)-u/2})(w_{1,t}), \\
s_{2,t} &= w_{1,t} - w_{2,t},
\end{aligned}$$

where I is the identity matrix. This is exactly the same as Dykstra's alternating projection steps.

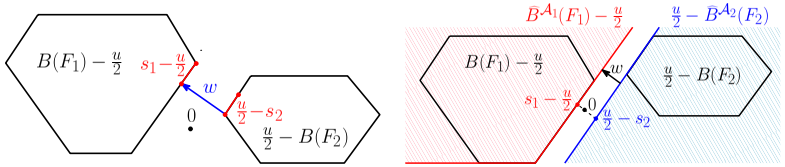


Figure 3: Closest point between two polytopes. (a) Output of Dykstra's alternating projection algorithm for the TV problem, the pair (s_1, s_2) may not be unique while $w = s_1 + s_2 - u$ is. (b) Dykstra's alternating projection output for outer approximations.

We have implemented it, and it behaves similar to alternating projections, but it still requires TV oracles for projection (see experiments in Section 5). There is however a key difference: while alternating projections and alternating reflections always converge to a pair of closest points, Dykstra's alternating projection algorithm converges to a *specific* pair of points, namely the pair closest to the initialization of the algorithm (Bauschke and Borwein, 1994); see an illustration in Figure 3-(a). This insight will be key in our algorithm to avoid cycling.

Assuming TV oracles are available for F_1 and F_2, Jegelka et al. (2013) and Kumar et al. (2015) use alternating projection (Bauschke et al., 1997) and alternating reflection (Bauschke et al., 2004) algorithms to solve dual optimization problem in Eq. 20 . Nishihara et al. (2014) gave a extensive theoretical analysis of alternating projection and showed that it converges linearly. However, these algorithms are equivalent to block *dual* coordinate descent and cannot be cast explicitly as descent algorithms for the primal TV problem. On the other hand, Dykstra's alternating projection is a descent algorithm on the primal, which enables local search over partitions. Complex TV oracles are often implemented by using SFM oracles recursively with the divide-and-conquer strategy on the individual functions. Using our algorithm in Section 3.4, they can be made more efficient using warm-starts (see experiments in Section 5).

4.2 Local Search over Partitions using Active-set Method

Given our algorithm for a single function, it is natural to perform a local search over two partitions $\mathcal{A}_1$ and $\mathcal{A}_2$, one for each function F_1 and F_2, and consider in the primal formulation a weight vector w compatible with both $\mathcal{A}_1$ and $\mathcal{A}_2$; or, equivalently, in the dual formulation, two outer approximations $\widehat{B}^{\mathcal{A}_1}(F_1)$ and $\widehat{B}^{\mathcal{A}_2}(F_2)$. That is, given the ordered partitions $\mathcal{A}_1$ and $\mathcal{A}_2$, using a similar derivation as in Eq. 20, we obtain the primal/dual pairs of optimization problems

$$\max_{\substack{s_1\in\widehat{B}^{\mathcal{A}_1}(F_1)\\ s_2\in\widehat{B}^{\mathcal{A}_2}(F_2)}} -\tfrac{1}{2}\|u - s_1 - s_2\|_2^2 = \min_{\substack{w\in\mathcal{W}^{\mathcal{A}_1}\\ w\in\mathcal{W}^{\mathcal{A}_2}}} f_1(w) + f_2(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2,$$

with $w = u - s_1 - s_2$ at optimality.

Primal solution by isotonic regression. The primal solution w is unique by strong convexity. Moreover, it has to be compatible with both $\mathcal{A}_1$ and $\mathcal{A}_2$, which is equivalent to being compatible with the *coalesced* ordered partition $\mathcal{A} = \text{coalesce}(\mathcal{A}_1, \mathcal{A}_2)$ defined as the coarsest ordered partition compatible by both. As shown in Appendix B, $\mathcal{A}$ may be found in time $O(\min(m_1, m_2)n)$.

Given $\mathcal{A}$, the primal solution w of the subproblem may be found by isotonic regression like in Section 3.2 in time $O(m)$ where m is the number of sets in $\mathcal{A}$. However, finding the optimal dual variables s_1 and s_2 turns out to be more problematic. We know that $s_1 + s_2 = u - w$ and that $s_1 + s_2 \in \widehat{B}^{\mathcal{A}}(F)$, but the split of $s_1 + s_2$ into (s_1, s_2) is unknown.

Obtaining dual solutions. Given ordered partitions $\mathcal{A}_1$ and $\mathcal{A}_2$, a unique well-defined pair (s_1, s_2) could be obtained by using convex feasibility algorithms such as alternating projections (Bauschke et al., 1997) or alternating reflections (Bauschke et al., 2004). However, the result would depend in non understood ways on the initialization, and we have observed cycling of the active-set algorithm. Using Dykstra's alternating projection algorithm allows us to converge to a unique well-defined pair (s_1, s_2) that will lead to a provably non-cycling algorithm.

When running the Dykstra's alternating projection algorithm starting from 0 on the polytopes $\widehat{B}^{\mathcal{A}_1}(F_1) - u/2$ and $u/2 - \widehat{B}^{\mathcal{A}_2}(F_2)$, if w is the unique distance vector between the two polytopes, then the iterates converge to the projection of 0 onto the convex sets of elements in the two polytopes that achieve the minimum distance (Bauschke and Borwein, 1994). See Figure 3-(b) for an illustration. This algorithm is however slow to converge when the polytopes do not intersect. Note that $w \neq 0$ in most of our situations and convergence is hard to monitor because primal iterates of the Dykstra's alternating projection diverge (Bauschke and Borwein, 1994).

Translated intersecting polytopes. In our situation, we have more to work with than just the ordered partitions: *we also know the vector* w (as mentioned earlier, it is obtained cheaply from isotonic regression). Indeed, from Lemma 2.2 and Theorem 3.8 from Bauschke and Borwein (1994), given this vector w, we may translate the two polytopes and now obtain a formulation where the two polytopes do intersect; that is we aim at projecting 0 on the (non-empty) intersection of $\widehat{B}^{\mathcal{A}_1}(F_1) - u/2 + w/2$ and $u/2 - w/2 - \widehat{B}^{\mathcal{A}_2}(F_2)$. See Figure 4. We also refer to this as the *translated Dykstra problem*[2] in the rest of the paper. This is equivalent to solving the following optimization problem

$$
\begin{aligned}
&\min_{\substack{s \in \widehat{B}^{\mathcal{A}_1}(F_1) - \frac{u-w}{2} \\ s \in \frac{u-w}{2} - \widehat{B}^{\mathcal{A}_2}(F_2)}} \tfrac{1}{2}\|s\|_2^2 \qquad (21)\\
&= \min_{s \in \mathbb{R}^n} \tfrac{1}{2}\|s\|_2^2 + \iota_{\widehat{B}^{\mathcal{A}_1}(F_1) - \frac{u-w}{2}}(s) + \iota_{\frac{u-w}{2} - \widehat{B}^{\mathcal{A}_2}(F_2)}(s),\\
&= \min_{s \in \mathbb{R}^n} \tfrac{1}{2}\|s\|_2^2 + \iota_{\widehat{B}^{\mathcal{A}_1}(F_1) - \frac{u-w}{2}}(s) + \iota_{\widehat{B}^{\mathcal{A}_2}(F_2) - \frac{u-w}{2}}(-s),\\
&= \min_{s \in \mathbb{R}^n} \Big(\tfrac{1}{2}\|s\|_2^2 + \max_{w_1 \in \mathcal{W}^{\mathcal{A}_1}} w_1^\top s - f_1(w_1) + \tfrac{w_1^\top (u-w)}{2} \\
&\qquad\qquad + \max_{w_2 \in \mathcal{W}^{\mathcal{A}_2}} -w_2^\top s - f_2(w_2) + \frac{w_2^\top (u-w)}{2} \Big),\\
&= \max_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} \Big(-f_1(w_1) - f_2(w_2) + \frac{(w_1+w_2)^\top (u-w)}{2} + \min_{s \in \mathbb{R}^n} \frac{1}{2}\|s\|_2^2 + (w_1 - w_2)^\top s \Big),\\
&= \max_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} \Big(-f_1(w_1) - f_2(w_2) + \frac{(w_1+w_2)^\top (u-w)}{2} - \frac{1}{2}\|w_1 - w_2\|_2^2 \Big),
\end{aligned}
$$

2. We refer to finding a Dykstra solution for translated intersecting polytopes as translated Dykstra problem.

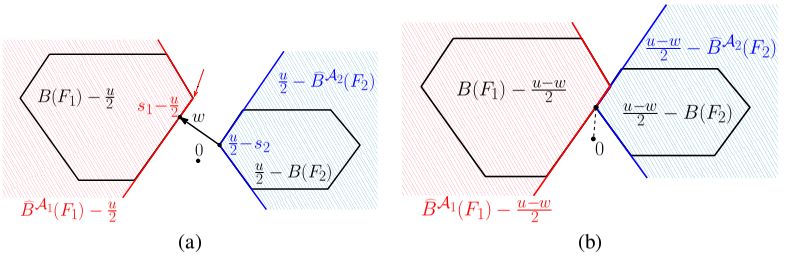


Figure 4: Translated intersecting polytopes. (a) output of our algorithm before translation. (b) Translated formulation.

$$
= \min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} \Big(f_1(w_1) + f_2(w_2) - \frac{(w_1+w_2)^\top (u-w)}{2} + \frac{1}{2}\|w_1 - w_2\|_2^2 \Big), \qquad (22)
$$

with $s = w_2 - w_1$ at optimality.

In Section 4.4 we propose algorithms to solve the above optimization problems. Assuming that we are able to solve this step efficiently, we now present our active-set algorithm for decomposable functions below.

4.3 Active-set Algorithm for Decomposable Functions

Data: Submodular functions F_1 and F_2 with SFM oracles, $u \in \mathbb{R}^n$, ordered partitions $\mathcal{A}_1, \mathcal{A}_2$
Result: primal optimal: $w \in \mathbb{R}^n$ and dual optimal: $s_1 \in B(F_1), s_2 \in B(F_2)$

while `True` **do**
  $\mathcal{A} = \text{coalesce}(\mathcal{A}_1, \mathcal{A}_2)$;
  Estimate w by solving $\min_{w \in \mathcal{W}^{\mathcal{A}}} f(w) - u^\top w + \frac{1}{2}\|w\|_2^2$ using isotonic regression ;
  Projection step: Estimate $s_1 \in \widehat{B}^{\mathcal{A}_1}(F_1)$ and $s_2 \in \widehat{B}^{\mathcal{A}_2}(F_2)$ by projecting 0 onto the intersection of $\widehat{B}^{\mathcal{A}_1}(F_1) - u/2 + w/2$ and $u/2 - w/2 - \widehat{B}^{\mathcal{A}_2}(F_2)$ using any of the algorithms described in Section 4.4 ;
  Merge the sets in $\mathcal{A}_j$ which are tight for $s_j, j \in \{1, 2\}$;
  Check optimality of s_1 and s_2 as described in Section 3.3;
  if *s_1 and s_2 are optimal* **then**
    `break` ;
  else
    for $j \in \{1, 2\}$ and $i_j \in \{1, \dots, m_j\}$, split the set A_{j,i_j} into C_{j,i_j} and $A_{j,i_j} \setminus C_{j,i_j}$ in that order to get an updated ordered partition $\mathcal{A}_j$;
  end
end

Given two ordered partitions $\mathcal{A}_1$ and $\mathcal{A}_2$, we obtain $s_1 \in \hat{B}^{\mathcal{A}_1}(F_1)$ and $s_2 \in \hat{B}^{\mathcal{A}_2}(F_2)$ as described in the following section. The solution $w = u - s_1 - s_2$ is optimal if and only if both

$s_1 \in B(F_1)$ and $s_2 \in B(F_2)$. When checking the optimality described in Section 3.3, we split the partition. As shown in Appendix C, either (a) $\|w\|_2^2$ strictly increases at each iteration, or (b) $\|w\|_2^2$ remains constant but $\|s_1 - s_2\|_2^2$ strictly increases. This implies that the algorithm is finitely convergent.

4.4 Optimizing the "Translated Dykstra Problem"

In this section, we describe algorithms for the "projection step" of the active-set algorithm proposed in Section 4.3 that optimizes the translated Dykstra problem in Eq. 22, i.e.,

$$\min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} f_1(w_1) + f_2(w_2) - \tfrac{(w_1+w_2)^\top (u-w)}{2} + \tfrac{1}{2}\|w_1 - w_2\|^2. \tag{23}$$

The corresponding dual optimization problem is given by

$$\min_{\substack{s \in \widehat{B}^{\mathcal{A}_1}(F_1) - \frac{u-w}{2} \\ s \in \frac{u-w}{2} - \widehat{B}^{\mathcal{A}_2}(F_2)}} \tfrac{1}{2}\|s\|_2^2, \tag{24}$$

with the optimality condition $s = w_2 - w_1$. Note that the only link to submodularity is that f_1 and f_2 are linear functions on $\mathcal{W}^{\mathcal{A}_1}$ and $\mathcal{W}^{\mathcal{A}_2}$, respectively. The rest of this section primarily deals with optimizing a quadratic program and we present two algorithms in Section 4.4.1 and Section 4.4.2.

4.4.1 ACCELERATED DYKSTRA'S ALGORITHM

In this section, we find the projection of the origin onto the intersection of the translated base polytopes obtained by solving the optimization problem in Eq. 22 given by

$$\min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} f_1(w_1) + f_2(w_2) - \tfrac{(w_1+w_2)^\top (u-w)}{2} + \tfrac{1}{2}\|w_1 - w_2\|^2,$$

using Dykstra's alternating projection. It can be solved using the following Dykstra's iterations:

$$\begin{aligned}
s_{1,t} &= \Pi_{\widehat{B}^{\mathcal{A}_1}(F_1)}(u/2 - w/2 + w_{2,t-1}), \\
w_{1,t} &= u/2 - w/2 + w_{2,t-1} - s_{1,t}, \\
s_{2,t} &= \Pi_{\widehat{B}^{\mathcal{A}_2}(F_2)}(u/2 - w/2 + w_{1,t}), \\
w_{2,t} &= u/2 - w/2 + w_{1,t} - s_{2,t},
\end{aligned}$$

with Π_C denoting the orthogonal projection onto the sets C, solved here by isotonic regression. Note that the value of the auxiliary variable w_2 can be warm-started. The algorithm converges linearly for polyhedral sets (Shusheng, 2000).

In our simulations, we have used the recent accelerated version of Chambolle and Pock (2015), which led to faster convergence. In order to monitor convergence of the algorithm, we compute the value of $\|u - w - s_{1,t} - s_{2,t}\|_1$, which is equal to zero at convergence. Note that the algorithm is not finitely convergent and gives only *approximate* solutions. Therefore, we introduce the approximation parameters ε_1 and ε_2 such that s_1 lies in the ϵ_1-neighborhood of the base polytope of F_1, i.e., $\min_{A \subseteq V} F_1(A) - s_1(A) \geq -\varepsilon_1$ and s_2 lies in the ε_2-neighborhood of the base polytope of F_2,

i.e., $\min_{A \subseteq V} F_2(A) - s_2(A) \geq -\varepsilon_2$, respectively. See Appendix E for more details on the approximation. The optimization problem can also be decoupled into smaller optimization problems by using the knowledge of the face of the base polytopes on which s_1 and s_2 lie. This is still slow to converge in practice and therefore we present an active-set method in the next section.

4.4.2 PRIMAL ACTIVE-SET METHOD

In this section, we find the projection of the origin onto the intersection of the translated base polytopes given by Eq. 22 using the standard active-set method (Nocedal and Wright, 2006) by solving a set of linear equations. For this purpose, we derive the equivalent optimization problems using equality constraints.

The ordered partition, $\mathcal{A}_j$ is given by $(A_{j,1}, \dots, A_{j,m_j})$, where m_j is the number of elements in the ordered partitions. Let B_{j,i_j} be defined as $(A_{j,1} \cup \cdots \cup A_{j,i_j})$. Therefore,

$$f_j(w_j) = \sum_{i_j=1}^{m_j} v_{j,i_j} \Big(F_j(B_{j,i_j}) - F_j(B_{j,i_j-1}) \Big) \tag{25}$$

$$w_j = \sum_{i_j=1}^{m_j} v_{j,i_j} 1_{A_{j,i_j}} \tag{26}$$

$$\text{with the constraints, } v_{j,1} \geqslant \cdots \geqslant v_{j,m_j}. \tag{27}$$

On substituting Eq. 25, Eq. 26 and Eq. 27 in Eq. 22, we have an equivalent optimization problem:

$$\begin{aligned}
\min_{\substack{v_{1,1} \geq \ldots \geq v_{1,m_1} \\ v_{2,1} \geq \ldots \geq v_{2,m_2}}} & \sum_{i_1=1}^{m_1} \Big(F_1(B_{1,i_1}) - F_1(B_{1,i_1-1}) - \frac{u(A_{1,i_1}) - w(A_{1,i_1})}{2} \Big) v_{1,i_1} \\
+ & \sum_{i_2=1}^{m_2} \Big(F_2(B_{2,i_2}) - F_2(B_{2,i_2-1}) - \frac{u(A_{2,i_2}) - w(A_{2,i_2})}{2} \Big) v_{2,i_2} \\
+ & \sum_{i_1=1}^{m_1} \frac{1}{2}|A_{1,i_1}| v_{1,i_1}^2 + \sum_{i_2=1}^{m_2} \frac{1}{2}|A_{2,i_2}| v_{2,i_2}^2 \sum_{i_1=1}^{m_1} \sum_{i_2=1}^{m_2} v_{1,i_1} v_{2,i_2} 1_{A_{1,i_1}}^\top 1_{A_{2,i_2}}.
\end{aligned}$$

This can be written as a quadratic program in $x = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ with inequality constraints in the following form

$$\min_{\substack{x \in \mathbb{R}^{m_1+m_2} \\ D(\mathcal{A}_1, \mathcal{A}_2) x \gtrsim 0}} \frac{1}{2} x^\top Q(\mathcal{A}_1, \mathcal{A}_2) x + c(\mathcal{A}_1, \mathcal{A}_2)^\top x. \tag{28}$$

Here, $D(\mathcal{A}_1, \mathcal{A}_2)$ is a sparse matrix of size $(m_1 + m_2 - 2) \times (m_1 + m_2)$, which is a block diagonal matrix containing the difference or first order derivative matrices of sizes $m_1 - 1 \times m_1$ and $m_2 - 1 \times m_2$ as the blocks and $c(\mathcal{A}_1, \mathcal{A}_2)$ is a linear vector that can be computed using the function evaluations of F_1 and F_2. Note that these evaluations need to be done only once.

Computing $Q(\mathcal{A}_1, \mathcal{A}_2)$. Let us consider a bipartite graph, $G = (\mathcal{A}_1, \mathcal{A}_2, E)$, with $m_1 + m_2$ nodes representing the ordered partitions of $\mathcal{A}_1$ and $\mathcal{A}_2$ respectively. The weight of the edge between each element of ordered partitions of $\mathcal{A}_1$, represented by A_{1,i_1} and each element of ordered partitions of $\mathcal{A}_2$, represented by A_{2,i_2} is the number of elements of the ground set V that lie in both these partitions and can be written as $e(A_{1,i_1}, A_{2,i_2}) = 1_{A_{1,i_1}}^\top 1_{A_{2,i_2}}$ for all $e \in E$. The matrix $Q(\mathcal{A}_1, \mathcal{A}_2)$ represents the Laplacian matrix of the graph G. Figure 5 shows a sample bipartite graph with $m_1 = 6$ and $m_2 = 5$.

Initializing with primal feasible point. Primal active-set methods start with a primal feasible point and continue to maintain primal feasible iterates. In our case, the starting point may be obtained using the weight vector w that is estimated using isotonic regression. The vector w is compatible both with $\mathcal{A}_1$ and $\mathcal{A}_2$, i.e., $w \in \mathcal{W}^{\mathcal{A}_1}$ and $w \in \mathcal{W}^{\mathcal{A}_2}$. Therefore, we may obtain the vectors v_1 and v_2 from w and initialize the primal feasible starting point using v_1 and v_2.

Optimizing the quadratic program in Eq. 28 by using active-set methods is equivalent to finding the face of the constraint set on which the optimal solution lies. For this purpose, we need to be able to solve the quadratic program in Eq. 28 with equality constraints.

Equality constrained QP. Let us now consider the following quadratic program with equality constraints

$$\min_{\substack{p \in \mathbb{R}^{m_1+m_2} \\ D'p=0}} \frac{1}{2} p^\top Q(\mathcal{A}_1, \mathcal{A}_2) p + \big(Q(\mathcal{A}_1, \mathcal{A}_2) x_k + c(\mathcal{A}_1, \mathcal{A}_2)\big)^\top p, \tag{29}$$

where D' is the subset of the constraints in $D(\mathcal{A}_1, \mathcal{A}_2)$, i.e., indices of constraints that are tight and x_k is a primal-feasible point. We refer the indices of the tight constraints as the *working set* and represent them by the set W in the algorithm. Therefore, the set of constraints in D' is the restriction of the constraint set $D(\mathcal{A}_1, \mathcal{A}_2)$ to the working set constraints denoted by W. The vector p gives the direction of strict descent of the cost function in Eq. 28 from feasible point x_k (Nocedal and Wright, 2006).

Without loss of generality, let us assume that the equality constraints are $v_{j,k_j} = v_{j,k_j+1}$ for any k_j in $[0, m_j)$. Let $\mathcal{A}'_j$ be the new ordered partition formed by merging A_{j,k_j} and A_{j,k_j+1} as $v_{j,k_j} = v_{j,k_j+1}$. Similarly, x'_t can be computed from x_t by merging the weights v_{j,k_j} and v_{j,k_j+1} into a single weight for the merged element of the ordered partition. Finding the optimal vector p' using the quadratic program in Eq. 29 with respect to the ordered partition $\mathcal{A}'_j$ is equivalent to solving the following unconstrained quadratic problem,

$$\mathcal{Q}(\mathcal{A}'_1, \mathcal{A}'_2, x'_t) = \min_{p' \in \mathbb{R}^{m'_1+m'_2}} \left(\frac{1}{2} p'^\top Q(\mathcal{A}'_1, \mathcal{A}'_2) p' + \big(Q(\mathcal{A}'_1, \mathcal{A}'_2) x'_t + c(\mathcal{A}'_1, \mathcal{A}'_2)\big)^\top p' \right), \tag{30}$$

where m'_j is the number of elements of the ordered partition $\mathcal{A}'_j$. This can be estimated by solving a linear system using conjugate gradient descent. The complexity of each iteration of the conjugate gradient is given by $O((m'_1 + m'_2)k)$ where k is the number of non-zero elements in the sparse matrix, $Q(\mathcal{A}'_1, \mathcal{A}'_2)$ (Vishnoi, 2013). We can build p from p' by repeating the values for the elements of the partition that were merged.

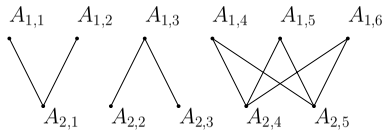


Figure 5: Bipartite graph to compute $Q(\mathcal{A}_1, \mathcal{A}_2)$ with $\mathcal{A}_1$ having $m_1 = 6$ components and $\mathcal{A}_2$ having $m_2 = 5$.

Primal active-set algorithm. We now can describe the standard primal active-set method.

Data: Laplacian matrix, $Q(\mathcal{A}_1, \mathcal{A}_2)$ and vector, $c(\mathcal{A}_1, \mathcal{A}_2)$, ordered partitions $\mathcal{A}_1, \mathcal{A}_2$
Result: $x^* \in \mathbb{R}^{m_1+m_2}$

Initialize: $t = 0$;
Primal feasible x_0 using the solution of isotonic regression w;
W_0 with indices of rows of $D(\mathcal{A}_1, \mathcal{A}_2)$ that are equal to 0;
while `True` **do**
  Estimate primal: Solve equality constrained QP in Eq. 29 with equality constraints indexed by working set W_t to find optimal p ;
  if *p == 0* **then**
    Estimate dual: $\lambda = D(\mathcal{A}_1, \mathcal{A}_2)^{-\top}\big(Q(\mathcal{A}_1, \mathcal{A}_2) x_t + c(\mathcal{A}_1, \mathcal{A}_2)\big)$;
    if $\lambda_i \geq 0$ *for all* $i \in W_t$ **then**
      `break` ;
    else
      update $j = \operatorname{argmin}_{j \in W_t} \lambda_j$;
      update $W_{t+1} = W_t \setminus \{j\}$;
      update $x_{t+1} = x_t$;
    end
  else
    Line search: Find least β that retains feasibility of $x_{t+1} = x_t + \beta p$ and find the blocking constraints B_t ;
    $W_{t+1} = W_t \cup B_t$;
  end
  $t = t + 1$
end
return $x^* = x_t$

We can estimate w_1 and w_2 from x^*, which will enable us to estimate s feasible in Eq. 24. Therefore we can estimate the dual variable $s_1 \in B^{\mathcal{A}_1}(F_1)$ and $s_2 \in B^{\mathcal{A}_2}(F_2)$ using s.

4.5 Decoupled Problem

In our context, the quadratic program in Eq. 28 can be decoupled into smaller optimization problems. Let us consider the bipartite graph $G = (\mathcal{A}_1, \mathcal{A}_2, E)$ of which Q is the Laplacian matrix. The number of connected components of the graph, G, is equal to the number of level-sets of w.

Let m be the total number of connected components in G. These connected components define a partition on the ground set V and a total order on elements of the partition can be obtained using the levels sets of w. Let k denote the index of each bipartite subgraph of G represented by $G_k = (\mathcal{A}_{1,k}, \mathcal{A}_{2,k}, E_k)$, where $k = 1, 2, \ldots, m$. Let J_k denote the indices of the nodes of G_k in G.

$$x^*_{J_k} = \operatorname*{argmin}_{\substack{x \in \mathbb{R}^{m_{1,k}+m_{2,k}} \\ D(\mathcal{A}_1,\mathcal{A}_2)_k x \succcurlyeq 0}} \frac{1}{2} x^\top Q(\mathcal{A}_1, \mathcal{A}_2)_{J_k J_k} x + c(\mathcal{A}_1, \mathcal{A}_2)^\top_{J_k} x, \tag{31}$$

where $m_{j,k}$ is size of $\mathcal{A}_{j,k}$. Therefore, $m_{1,k} + m_{2,k}$ is the total number of nodes in the subgraph G_k. Note that this is exactly equivalent to decomposition of the base polytope of F_j into base polytopes of submodular functions formed by contracting F_j on each individual component representing the connected component k. See Appendix D for more details.

5. Experiments

In this section, we show the results of the algorithms proposed on various problems. We first consider the problem of solving total variation denoising for a non decomposable function using active-set methods in Section 5.1, specifically cut functions. Here, our experiments mainly focus on the time comparisons with state-of-art methods and also show an important setting where we show the gain due to the ability to warm-start our algorithm. In Section 5.2, we consider cut functions on a 3D grid decomposed into a function of the 2D grid and a function of chains. We then consider a 2D grid and a concave function on cardinality, which is not a cut function. Our algorithm leads to marginal gains for the usual non decomposable functions. However, in the non decomposable case there are many total variation problems to be solved. The ability to warm-start lends to huge improvements when compared to the usage of standard total variation oracles in this setting.

5.1 Non-decomposable Total Variation Denoising

Our experiments consider images, which are 2-dimensional grids with vertex neighborhood of size 4. The data set comprises of 6 different images of varying sizes. We consider a large image of size 5616×3744 and recursively scale into a smaller image of half the width and half the height maintaining the aspect ratio. Therefore, the size of each image is four times smaller than the size of the previous image. We restrict to anisotropic uniform-weighted total variation to compare with Chambolle and Darbon (2009) but our algorithms work as well with weighted total variation, which is standard in computer vision, and on any graph with SFM oracles. Therefore, the unweighted total variation is

$$f(w) = \lambda \sum_{i \sim j} |w_i - w_j|,$$

where λ is a regularizing constant for solving the total variation problem in Eq. 3.

Maxflow (Boykov and Kolmogorov, 2004) is used as the SFM oracle for checking the optimality of the ordered partitions. Figure 6(a) shows the number of SFM oracle calls required to solve the TV problem for images of various sizes. Note that for the algorithm of Chambolle and Darbon (2009) each SFM oracle call optimizes smaller problems sequentially, while each SFM oracle call in our method optimizes several independent smaller problems in parallel. Therefore, our method has lesser number of oracle calls than Chambolle and Darbon (2009). However, oracle complexity of

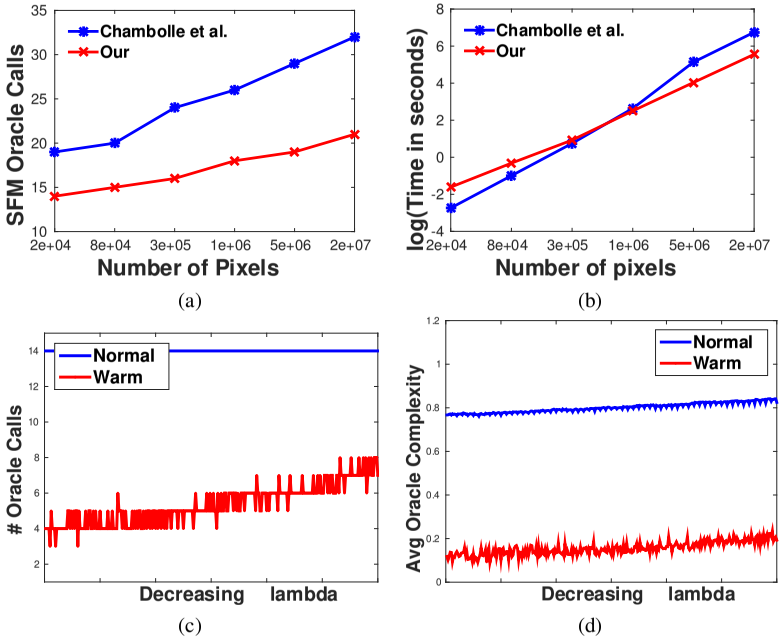


Figure 6: (a) Number of SFM oracle calls for images of various sizes, (b) Time taken for images of various sizes, (c) Number of iterations with and without warm start, (d) Average complexity of the oracle with and without warm start.

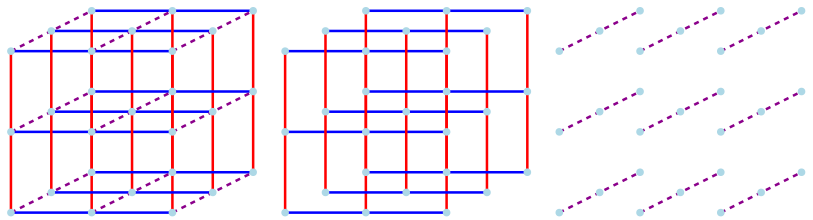
each call is higher for the our method when compared to Chambolle and Darbon (2009). Figure 6(b) shows the time required for each of the methods to solve the TV problem to convergence. We have an optimized code and only use the oracle as plugin which takes about 80-85 percent of the running time. This is primarily the reason our approach takes more time than (Chambolle and Darbon, 2009) despite having fewer oracle calls for small images.

Figure 6(c) also shows the ability to warm start by using the output of a related problem, i.e., when computing the solution for several values of λ (which is typical in practice). In this case, we use optimal ordered partitions of the problem with larger λ to warm start the problem with smaller λ. It can be observed that warm start of the algorithm requires lesser number of oracle calls to converge than using the initialization with trivial ordered partition. Warm start also largely helps in reducing the burden on the SFM oracle. With warm starts the number of ordered partitions does not change much over iterations. Hence, it suffices to query only ordered partitions that have changed. To analyze this we define *oracle complexity* as the fraction of pixels in the elements of the partitions that need to be queried. Oracle complexity is averaged over iterations to understand the average burden on the oracle per iteration. With warm starts this reduces drastically, which can be observed in Figure 6(d).

5.2 Decomposable Total Variation Denoising and SFM

Cut functions. In the decomposable case, we consider the SFM and TV problems on a cut function defined on a 3D-grid. The 3D-grid consists of lines parallel lines in each dimension as shown in Figure 7. It can be decomposed into two functions F_1 and F_2, where F_1 is composed on parallel 2D-grids and F_2 is composed of parallel chains. From Figure 7, the function F_1 represents all the solid edges(red and blue) whereas the function F_2 represents the dashed edges(magenta). For brevity, we refer to each 2D-grid of the function F_1 as a *frame*. The SFM oracle for the function F_1 is the maxflow-mincut (Boykov and Kolmogorov, 2004) algorithm, which may run in parallel for all frames. Similarly the SFM oracle for the function F_2 is the message passing algorithm, which may run in parallel for all chains. The corresponding TV oracles, i.e., *projection algorithm* for F_1 and F_2 may be solved using the algorithm described in Section 3.4 due to availability of the respective SFM oracles. We consider averaged alternating reflection (AAR) (Bauschke et al., 2004) by solving each projection without warm-start and counting the total number of SFM oracle calls of F_1 to solve the SFM and TV on the 3D-grid as our baseline. (SGD-P) denotes the dual subgradient based method (Komodakis et al., 2011) modified with Polyak's rule (Poljak, 1987) to solve SFM on the 3D-grid. We show the performance of alternating projection (AP-WS), averaged alternating reflection (AAR-WS) (Bauschke et al., 2004) and Dykstra's alternating projection (DAP-WS) (Bauschke and Borwein, 1994) using warm start of each projection with the ordered partitions. WS denotes warm start variant of each of the algorithm. The performance of the active-set algorithm proposed in Section 4.3 with inner loop solved using the primal active-set method proposed in Section 4.4.2 is represented by (ACTIVE).

In our experiments, we consider the 3D volumetric data set of the Stanford bunny (max) of size $102 \times 100 \times 79$. The function F_1 represents 102 frames while F_2 represents the 7900 chains. The dimension of each frame in F_1 is 100×79, while the length of each chain in F_2 is 102. Figure 8 (a) and (b) show that (AP-WS), (AAR-WS), (DAP-WS) and (ACTIVE) require relatively less number of oracle calls when compared to when compared to AAR or SGD-P. Note that we count 2D SFM oracle calls as they are more expensive than the SFM oracles on chains.



(a) (b) (c)

Figure 7: (a) Cut function F defined on a 3D-grid may be decomposed into: (b) F_1 represented by solid edges (red and blue) and (c) F_2 represented by dashed lines (magenta)

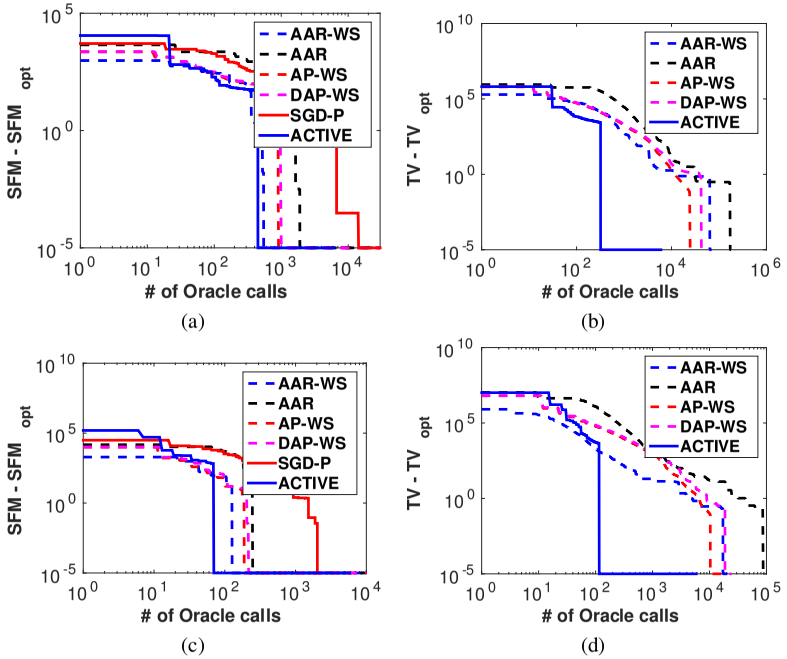


Figure 8: (a) Number of 2D SFM calls to obtain 3D SFM, (b) Number of 2D SFM calls to obtain 3D TV, (c) Number of 2D SFM calls to obtain SFM of 2D + concave function, (d) Number of 2D SFM calls to obtain TV of 2D + concave function.

Time comparisons. We also performed time comparisons between the iterative methods and the combinatorial methods on standard data sets. The standard mincut-maxflow (Boykov and Kolmogorov, 2004) on the 3D volumetric data set of the Standard bunny (max) of size $102 \times 100 \times 79$ takes 0.11 seconds while averaged alternating reflections (AAR) without warm start takes 0.38 seconds. The averaged alternating reflections with warm start (AAR-WS) takes 0.21 seconds and the active-set method (ACTIVE) takes 0.38 seconds. The main bottleneck in the active-set method is the inversion of the Laplacian matrix and it could considerably improve by using methods suggested by Vishnoi (2013). Note that the projection on the base polytopes of F_1 and F_2 can be parallelized by projecting onto each of the 2D frame of F_1 and each line of F_2 respectively (Kumar et al., 2015). The times for (AAR), (AAR-WS) and (ACTIVE) use parallel multi-core architectures[3] while the combinatorial algorithm only uses a single core. Note that cut functions on grid structures are only a small subclass of submodular functions with such efficient combinatorial algorithms. In contrast, our algorithm works on more general class of sum of submodular functions than just with cut functions.

Concave functions on cardinality. In this experiment we consider our SFM problem of sum of a 2D cut on a graph of size 5616×3744 and a super pixel based concave function on cardinality (Stobbe and Krause, 2010; Jegelka et al., 2013). The unary potentials of each pixel is calculated using the Gaussian mixture model of the color features. The edge weight $a(i, j) = \exp(-\|y_i - y_j\|^2)$, where y_i denotes the RGB values of the pixel i. In order to evaluate the concave function, regions R_j are extracted via superpixels and, for each R_j, defining the function $F_2(S) = |S||R_j \setminus S|$. We use 200 and 500 regions. Figure 8 (c) and (d) shows that (AP-WS), (AAR-WS), (DAP-WS) and (ACTIVE) algorithms converge for solving TV quickly by using only SFM oracles and relatively less number of oracle calls. Note that we count 2D SFM oracle calls.

6. Conclusion

In this paper, we present an efficient active-set algorithm for optimizing quadratic losses regularized by Lovász extension of a submodular function using the SFM oracle of the function. We also present an active-set algorithms to minimize sum of "simple" submodular functions using SFM oracles of the individual "simple" functions. We also show that these algorithms are competitive to the existing state-of-art algorithms to minimize submodular functions.

Acknowledgments

We acknowledge support from the European Research Council grant SIERRA (project 239993). K. S. Sesh Kumar also acknowledges the support from the European Research Council grant of Prof. Vladimir Kolmogorov DOiCV(project 616160) at IST Austria. The comments of the reviewers have helped us improve the presentation significantly.

3. 20 core, Intel(R) Xeon(R) CPU E5-2670 v2 @ 2.50GHz with 100Gigabytes of memory. We only use up to 16 cores of the machine to ensure accurate timings.

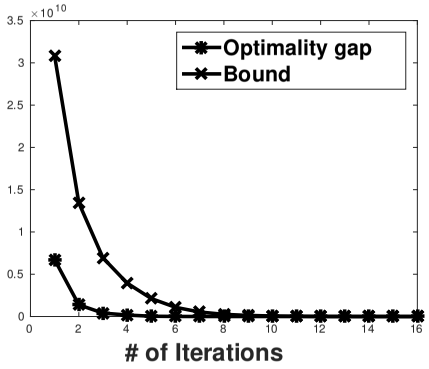


Figure 9: Certified gap vs. bound for TV denoising on a 2D image using the algorithm in Section 3.4.

Appendix A. Certificates of Optimality

We consider a total variation denoising problem on an 2D image of dimensions 384×288 using the algorithm proposed in Section 3.4 for non-decomposable functions, where we assume a 2D SFM oracle. Here, we plot the optimality gap given by $(f(w) - u^\top w + \frac{1}{2}\|w\|_2^2) - (f(w^*) - u^\top w^* + \frac{1}{2}\|w^*\|_2^2)$ and the bound, $\varepsilon \operatorname{range}(w) + \varepsilon \operatorname{range}(w^*)$, proposed in Section 3.6. Here, w is the solution at the end of each iteration of the algorithm and w^* is the optimal solution.

Appendix B. Algorithms for Coalescing Partitions

The basic interpretation in coalescing two ordered partitions is as follows. Given an ordered partition $\mathcal{A}_1$ and $\mathcal{A}_2$ with m_1 and m_2 elements in the partitions respectively, we define for each $j = 1, 2$, $\forall i_j = (1, \ldots, m_j)$, $B_{j,i_j} = (A_{j,1} \cup \ldots \cup A_{j,i_j})$. The inequalities defining the outer approximation of the base polytopes are given by hyperplanes defined by

$$\forall i_j = (1, \ldots, m_j), s_j(B_{j,i_j}) \leq F_j(B_{j,i_j}).$$

The hyperplanes defined by common sets of both these partitions, defines the coalesced ordered partitions. The following algorithm performs coalescing between these partitions.

- **Input**: Ordered partitions $\mathcal{A}_1$ and $\mathcal{A}_2$.
- **Initialize**: $x = 1, y = 1, z = 1$ and $C = \varnothing$.
- **Algorithm**: Iterate until $x = m_1$ and $y = m_2$
  1. If $|B_{1,x}| > |B_{2,y}|$ then $y := y + 1$.
  2. If $|B_{1,x}| < |B_{2,y}|$ then $x := x + 1$.

3. If $B_{1,x} == B_{2,y}$ then
   - $A_z = (B_{1,x} \setminus C)$,
   - $C = B_{1,x}$, and
   - $z := z + 1$.

– **Output**: $m = z$, ordered partitions $\mathcal{A} = (A_1, \ldots, A_m)$.

Running time. The algorithm terminates in $\min(m_1, m_2)$ iterations and the checking condition for step (3) takes n iterations. Therefore, the algorithm overall takes a time of $O(\min(m_1, m_2)n)$.

Appendix C. Optimality of Algorithm for Decomposable Problems

In step (10) of the algorithms, when we split partitions, the value of the primal/dual pair of optimization algorithms

$$\max_{\substack{s_1 \in \widehat{B}^{\mathcal{A}_1}(F_1) \\ s_2 \in \widehat{B}^{\mathcal{A}_2}(F_2)}} -\tfrac{1}{2}\|u - s_1 - s_2\|_2^2 = \min_{\substack{w \in \mathcal{W}^{\mathcal{A}_1} \\ w \in \mathcal{W}^{\mathcal{A}_2}}} f_1(w) + f_2(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2,$$

cannot increase. This is because, when splitting, the constraint set for the minimization problem only gets bigger. Since at optimality, we have $w = u - s_1 - s_2$, $\|w\|_2$ cannot decrease, which shows the first statement.

Now, if $\|w\|_2$ remains constant after an iteration, then it has to be the same (and not only have the same norm), because the optimal s_1 and s_2 can only move in the direction orthogonal to w.

In step (4) of the algorithm, we project 0 on the (non-empty) intersection of $\widehat{B}^{\mathcal{A}_1}(F_1) - u/2 + w/2$ and $u/2 - w/2 - \widehat{B}^{\mathcal{A}_2}(F_2)$. This corresponds to minimizing $\frac{1}{2}\|s_1 - u/2 + w/2\|^2$ such that $s_1 \in \widehat{B}^{\mathcal{A}_1}(F_1)$ and $s_2 = u - w - s_1 \in \widehat{B}^{\mathcal{A}_2}(F_2)$. This is equivalent to minimizing $\frac{1}{8}\|s_1 - s_2\|^2$. We have:

$$\begin{aligned}
\max_{\substack{s_1 \in \widehat{B}^{\mathcal{A}_1}(F_1) \\ s_2 \in \widehat{B}^{\mathcal{A}_2}(F_2) \\ s_1 + s_2 = u - w}} -\frac{1}{8}\|s_1 - s_2\|_2^2 &= \min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} \max_{\substack{s_1 \in \mathbb{R}^n, s_2 \in \mathbb{R}^n \\ s_1 + s_2 = u - w}} \Big(-\frac{1}{8}\|s_1 - s_2\|_2^2 + f_1(w_1) + f_2(w_2) \\
&\qquad\qquad -w_1^\top s_1 - w_2^\top s_2 \Big) \\
&= \min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} \max_{s_2 \in \mathbb{R}^n} \Big(-\frac{1}{8}\|u - w - 2s_2\|_2^2 + f_1(w_1) + f_2(w_2) \\
&\qquad\qquad -w_1^\top (u - w - s_2) - w_2^\top s_2 \Big) \\
&= \min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} \max_{s_2 \in \mathbb{R}^n} \Big(-\frac{1}{8}\|u - w\|_2^2 - \frac{1}{2}\|s_2\|_2^2 + \frac{1}{2} s_2^\top (u - w) \\
&\qquad\qquad + f_1(w_1) + f_2(w_2) - w_1^\top (u - w - s_2) - w_2^\top s_2 \Big) \\
&= \min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} -w_1^\top (u - w) + f_1(w_1) + f_2(w_2) - \frac{1}{8}\|u - w\|_2^2 \\
&\qquad\qquad + \max_{s_2 \in \mathbb{R}^n} -\frac{1}{2}\|s_2\|_2^2 + s_2^\top \big(\tfrac{u-w}{2} + w_1 - w_2\big) \\
&= \min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} \Big(-w_1^\top (u - w) + f_1(w_1) + f_2(w_2) - \frac{1}{8}\|u - w\|_2^2 \\
&\qquad\qquad + \frac{1}{2}\|\tfrac{u-w}{2} + w_1 - w_2\|_2^2 \Big) \\
&= \min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} \Big(-w_1^\top (u - w) + f_1(w_1) + f_2(w_2) + \frac{1}{2}\|w_1 - w_2\|_2^2 \\
&\qquad\qquad + \frac{1}{2}(u - w)^\top (w_1 - w_2) \Big) \\
&= \min_{\substack{w_1 \in \mathcal{W}^{\mathcal{A}_1} \\ w_2 \in \mathcal{W}^{\mathcal{A}_2}}} \Big(f_1(w_1) + f_2(w_2) - \frac{1}{2}(u - w)^\top (w_1 + w_2) \\
&\qquad\qquad + \frac{1}{2}\|w_1 - w_2\|_2^2 \Big),
\end{aligned}$$

Thus s_1 and s_2 are dual to certain vectors w_1 and w_2, which minimize a decoupled formulation in f_1 and f_2. To check optimality, like in the single function case, it decouples over the constant sets of w_1 and w_2, which is exactly what step (5) is performing.

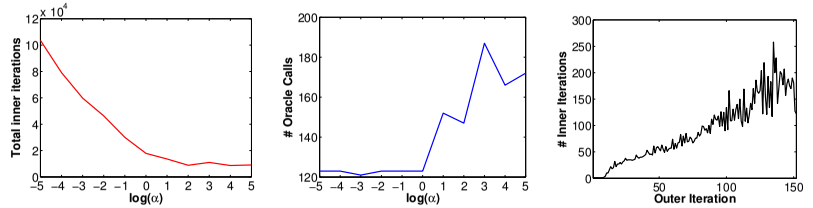
If the check is satisfied, it means that w_1 and w_2 are in fact optimal for the problem above without the restriction in compatibilities, which implies that they are the Dykstra solutions for the TV problem.

If the check is not satisfied, then the same reasoning as for the one function case, leads directions of descent for the new primal problem above. Hence it decreases; since its value is equal to $-\frac{1}{8}\|s_1 - s_2\|_2^2$, the value of $\|s_1 - s_2\|_2^2$ must increase, hence the second statement.

Appendix D. Decoupled Problems.

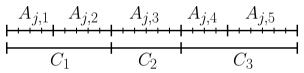
Given that we deal with polytopes, knowing w implies that we know the faces on which we have to looked for. It turns outs that for base polytopes, these faces are products of base polytopes for modified functions (a similar fact holds for their outer approximations).

Given the ordered partition $\mathcal{A}'$ defined by the level sets of w (which have to be finer than $\mathcal{A}_1$ and $\mathcal{A}_2$), we know that we may restrict $\widehat{B}^{\mathcal{A}_j}(F_j)$ to elements s such that $s(B) = F(B)$ for all sup-level sets B of w (which have to be unions of contiguous elements of A_j); see an illustration below.



(a) (b) (c)

Figure 10: (a) Total number of inner iterations for varying α. (b) Total number of outer iterations for varying α. and (c) Number of inner iterations per each outer iteration for the $\alpha = 10^1$.



More precisely, if $C_1, \dots, C_{m'}$ are constant sets of w ordered with decreasing values. Then, we may search for s_j independently for each subvector $(s_j)_{C_k} \in \mathbb{R}^{C_k}$, $k \in \{1, \dots, m'\}$ and with the constraint that

$$(s_j)_{C_k} \in \hat{B}^{\mathcal{A}_j \cap C_k}\big[(F_j)_{C_k | C_1 \cup \dots \cup C_{k-1}}\big],$$

where $\mathcal{A}_j \cap C_k$ is the ordered partition obtained from $\mathcal{A}_j$ once restricted onto C_k and the submodular function is the so-called contraction of F on C_k given $C_1 \cup \dots \cup C_{k-1}$, defined as $S \mapsto F_j(S \cup C_1 \cup \dots \cup C_{k-1}) - F(C_1 \cup \dots \cup C_{k-1})$. Thus this corresponds to solving m different smaller subproblems.

Appendix E. Approximate Dykstra Steps

The Dykstra step, i.e., step (4) of the algorithm proposed in Section 4.4.1 is not finitely convergent. Therefore, it needs to be solved approximately. For this purpose, we introduce a parameter α to approximately solve the Dykstra step such that $\|s_1 + s_2 - u + w\|_1 \leq \alpha(\epsilon_1 + \epsilon_2)$. Let ϵ be defined as $\alpha(\epsilon_1 + \epsilon_2)$. This shows that the s_1 and s_2 are ϵ-accurate. Therefore, α must be chosen in such a way that we avoid cycling in our algorithm. However, another alternative is to warm start the Dykstra step with w_1 and w_2 of the previous iteration. This ensures we don't go back to the same w_1 and w_2, which we have already encountered and avoid cycling. Figure 10 shows the performance of our algorithm for a simple problem of 100×100 2D-grid with 4-neighborhood and uniform weights on the edges with varying α. Figure 10-(a) shows the total number of inner iterations required to solve the TV problem. Figure 10-(b) gives the total number of SFM oracle calls required to solve the TV problem. In Figure 10-(c), we show the number of inner iterations in every outer iteration for the best α we have encountered.

References

Maxflow dataset online. `http://vision.csd.uwo.ca/maxflow-data`.

F. Bach. *Learning with Submodular Functions: A Convex Optimization Perspective*, volume 6 of *Foundations and Trends in Machine Learning*. NOW, 2013.

H. H. Bauschke and J. M. Borwein. Dykstra's alternating projection algorithm for two sets. *Journal of Approximation Theory*, 79(3):418–443, 1994.

H. H. Bauschke, J. M. Borwein, and A. S. Lewis. The method of cyclic projections for closed convex sets in Hilbert space. *Contemporary Mathematics*, 204:1–38, 1997.

H. H. Bauschke, P. L. Combettes, and D. Luke. Finding best approximation pairs relative to two closed convex sets in Hilbert spaces. *Journal of Approximation Theory*, 127(2):178–192, 2004.

M. J. Best and N. Chakravarti. Active set algorithms for isotonic regression: a unifying framework. *Mathematical Programming*, 47(1):425–439, 1990.

Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.

Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

D. Chakrabarty, P. Jain, and P. Kothari. Provable submodular minimization using Wolfe's algorithm. In *Advances in Neural Information Processing Systems*. 2014.

A. Chambolle and J. Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84(3):288–307, 2009.

A. Chambolle and T. Pock. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. Technical Report 01099182, HAL, 2015.

S. Fujishige. Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5(2):186–196, 1980.

S. Fujishige. Submodular systems and related topics. In *Mathematical Programming at Oberwolfach II*. 1984.

S. Fujishige. *Submodular Functions and Optimization.* Elsevier, 2005.

N. Gaffke and R. Mathar. A cyclic projection algorithm via duality. *Metrika*, 36(1):29–54, 1989.

D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. *SIAM Journal on Scientific Computing*, 31(5):3712–3743, 2009.

H. Groenevelt. Two algorithms for maximizing a separable concave function over a polymatroid feasible region. *European Journal of Operational Research*, 54(2):227–236, 1991.

S. Jegelka, F. Bach, and S. Sra. Reflection methods for user-friendly submodular optimization. In *Advances in Neural Information Processing Systems*, 2013.

V. Kolmogorov. Minimizing a sum of submodular functions. *Discrete Applied Mathematics*, 160 (15), 2012.

N. Komodakis, N. Paragios, and G. Tziritas. MRF energy minimization and beyond via dual decomposition. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 33(3):531–552, 2011.

A. Krause and C. Guestrin. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology*, 2(4), 2011.

K. S. Sesh Kumar, A. Barbero, S. Jegelka, S. Sra, and F. Bach. Convex optimization for parallel energy minimization. Technical Report 01123492, HAL, 2015.

L. Ladicky, C. Russell, P.t Kohli, and P. H. S. Torr. Graph cut based inference with co-occurrence statistics. In *Proceedings of the 11th European Conference on Computer Vision*, 2010.

L. Landrieu and G. Obozinski. Cut Pursuit: fast algorithms to learn piecewise constant functions. In *Proceedings of Artificial Intelligence and Statistics*, 2016.

H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *Proceedings of NAACL/HLT*, 2011.

L. Lovász. Submodular functions and convexity. *Mathematical programming: the state of the art, Bonn*, pages 235–257, 1982.

R. Nishihara, S. Jegelka, and M. I. Jordan. On the convergence rate of decomposable submodular function minimization. In *Advances in Neural Information Processing Systems 27*, pages 640–648, 2014.

J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, Berlin, 2006.

B. T. Poljak. *Introduction to optimization*. Optimization Software, 1987.

R. T. Rockafellar. *Convex Analysis*. Princeton U. P., 1997.

X. Shusheng. Estimation of the convergence rate of Dykstras cyclic projections algorithm in polyhedral case. *Acta Mathematicae Applicatae Sinica (English Series)*, 16(2):217–220, 2000.

P. Stobbe and A. Krause. Efficient minimization of decomposable submodular functions. In *Advances in Neural Information Processing Systems*, 2010.

R. Tarjan, J. Ward, B. Zhang, Y. Zhou, and J. Mao. Balancing applied to maximum network flow problems. In *European Symp. on Algorithms (ESA)*, pages 612–623, 2006.

N.K. Vishnoi. *Lx = B - Laplacian Solvers and Their Algorithmic Applications*. Now Publishers, 2013.

Journal of Machine Learning Research 18 (2017) 1-33 Submitted 4/17; Revised 9/17; Published 12/17

A Bayesian Mixed-Effects Model to Learn Trajectories of Changes from Repeated Manifold-Valued Observations

Jean-Baptiste Schiratti jean-baptiste.schiratti@cmap.polytechnique.fr *CMAP, Ecole Polytechnique*
Route de Saclay, 91128 *Palaiseau*

Stéphanie Allassonnière stephanie.allassonniere@polytechnique.edu
INSERM, UMRS 1138, team 22. Centre de recherche des Cordeliers, Université Paris 5 René Descartes

Olivier Colliot olivier.colliot@upmc.fr

Stanley Durrleman stanley.durrleman@inria.fr
*ARAMIS Lab, INRIA Paris, Inserm U*1127*, CNRS UMR* 7225*, Sorbonne Universités, UPMC Univ Paris* 06 *UMRS* 1127
Institut du Cerveau et de la Moelle épinière, ICM, F-75013, Paris, France

Editor: David Sontag

Abstract

We propose a generic Bayesian mixed-effects model to estimate the temporal progression of a biological phenomenon from observations obtained at multiple time points for a group of individuals. The progression is modeled by continuous trajectories in the space of measurements. Individual trajectories of progression result from spatiotemporal transformations of an average trajectory. These transformations allow for the quantification of changes in direction and pace at which the trajectories are followed. The framework of Riemannian geometry allows the model to be used with any kind of measurements with smooth constraints. A stochastic version of the Expectation-Maximization algorithm is used to produce maximum *a posteriori* estimates of the parameters. We evaluated our method using a series of neuropsychological test scores from patients with mild cognitive impairments, later diagnosed with Alzheimer's disease, and simulated evolutions of symmetric positive definite matrices. The data-driven model of impairment of cognitive functions illustrated the variability in the ordering and timing of the decline of these functions in the population. We showed that the estimated spatiotemporal transformations effectively put into correspondence significant events in the progression of individuals.

Keywords: longitudinal model, spatiotemporal analysis, Riemannian geometry, stochastic expectation-maximization algorithm

1. Introduction

The study of the temporal progression of a biological or natural phenomenon is central to several scientific fields. For instance, the study of progressive diseases plays a crucial role in the diagnosis and prognosis of patients. In computer vision, the dynamics of facial expressions in video sequences may be important in automatic detection and characterization of emotions.

For a given individual or object, the evolution of the observed phenomenon can be measured by several characteristics or features, which describe the state of the individual at a given time point. In medicine, these features may be blood markers, height, or weight, but also structured multivariate data such as medical images. The shape of a human face may be described by the position of characteristic points on the nose, mouth or brows. These features may be represented, at a given time point, by a point in a high-dimensional space. The temporal evolution of these features may be modeled therefore as a smooth parametric curve in the space of measurements, i.e. a spatiotemporal trajectory. These trajectories vary across individuals in two possible ways. Firstly, the position and direction of the trajectory differ because the measurements have intrinsically different values and different trajectory of changes for different individuals. Secondly, the pace at which the trajectory is followed (i.e. the way the curve is parametrized) varies because some individuals may follow the same progression pattern but at a different age and possibly at a different speed. We refer to the first type of variability as a *spatial* variability, and the second type as a *temporal* variability, leading together to the concept of *spatiotemporal* variability.

The goal of this paper is to automatically estimate the typical trajectory of changes and its spatiotemporal variability within a group of individuals. We aim to infer such spatiotemporal patterns from longitudinal data sets, which consist of repeated observations of the same biological phenomenon at several time points for a group of individuals. The time points and their number may vary for different individuals.

In the literature, *mixed-effects models* (Eisenhart, 1947; Laird and Ware, 1982) and (Verbeke and Molenberghs, 2009) appear as a popular method for the analysis of longitudinal data. These statistical models include *fixed* and *random effects* which provide these models with a hierarchical structure, where fixed effects described the data at the population (or group) level, and the random effects at the individual level. By fitting a mixed-effects model, one can learn an average trajectory as well as individual-specific trajectories. Moreover, mixed-effects models enforce conditions on the distribution of the random effects, thus opening up the possibility to learn a distribution of trajectories in the space of observations.

Linear Mixed Effects (LME) models are the most simple mixed-effects models introduced in Laird and Ware (1982). A particular, but yet informative case of the LME models for analysing longitudinal data is the *random slope and intercept model*. This model writes: $\mathbf{y}_{i,j} = (t_{i,j}-t_0)(\overline{\mathbf{A}}+\mathbf{A}_i)+(\overline{\mathbf{B}}+\mathbf{B}_i)+\boldsymbol{\varepsilon}_{i,j}$, where $t_0 \in \mathbb{R}$ and $(t_{i,j})_{1\leq j\leq k_i}$ denotes the time points at which the observations $\mathbf{y}_{i,j} \in \mathbb{R}^n$ of the ith individual were obtained. The population parameters (or fixed effects) of the model are the slope $\overline{\mathbf{A}}$ and the intercept $\overline{\mathbf{B}}$. The random effects are the subject-specific slopes $(\mathbf{A}_i)_{1\leq i\leq p}$ and intercepts $(\mathbf{B}_i)_{1\leq i\leq p}$, which are assumed to be normally distributed and independent of each other. This random slope and intercept model estimates an average trajectory $\overline{\mathbf{D}}(t) = (t-t_0)\overline{\mathbf{A}} + \overline{\mathbf{B}}$. The random effects of the model allows us to estimate individual trajectories $\mathbf{D}_i(t) = (t-t_0)(\overline{\mathbf{A}}+\mathbf{A}_i) + (\overline{\mathbf{B}}+\mathbf{B}_i)$, which are obtained by adjusting the slope and intercept of the average trajectory. This model is essentially built on the idea of regressing the measurements against time. The parameter t_0 can be understood as a *reference time*. If the longitudinal dataset arises from animal breeding studies, developmental studies or pharmacological studies, the reference time t_0 may be chosen to be the date of birth or the time at which a drug was administered.

©2017 J.-B. Schiratti et al..

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-197.html.

However, there are many situations in which there is no obvious reference time t_0 at which observations may be compared. In ageing, for instance, different individuals of the same age may be at different stages of ageing, or stages of disease progression. Therefore, it does not make sense to regress the measurements against age, or, in other words, to statistically compare measurements at a given age. In video sequences, there is no obvious way to find the frames corresponding to the same event in two different sequences. By contrast, we would like this temporal alignment of the trajectories to be automatically estimated from the data. Adding the reference time t_0 as a new parameter of the model is not a solution as the model becomes non-identifiable: an infinite number of triplets $(\overline{A}, \overline{B}, t_0)$ parametrize the same trajectory.

In Yang et al. (2011) and Delor et al. (2013), the authors addressed this problem by introducing *time shifts* in their statistical analysis. In Durrleman et al. (2009, 2013), time reparametrizations called *time warps* (smooth monotonic transformations of the real line) are considered to address this point in the context of longitudinal shape analysis, and parameters were estimated by optimizing an uncontrolled approximation of the likelihood. In Hong et al. (2014), the authors used parametric time warps with a regression model for shape analysis. In Lorenzi et al. (2015), the authors used Riemannian manifold techniques to estimate a model of normal brain ageing from MR images. The model was used to compute a time shift, called *morphological age shift*, which corresponds to the actual anatomical age of the subject with respect to an estimated reference age. In (Fonteijn et al., 2012; Young et al., 2015), the authors developed a statistical model called *the Event-Based Model*, which estimates an ordering of categorical variables. The model is used to estimate the progression of a series of events. However, these models do not allow for the estimation of the relative timing between two consecutive events. In Jedynak et al. (2012), the authors modeled the progression of biomarkers using a nonlinear mixed-effects model for univariate observations. This model estimates individual trajectories which are defined using individual-specific time reparametrizations of an average trajectory. However, the proposed model is not identifiable unless some conditions are imposed on the parameters of the model. Therefore, generalizing the model to multivariate observations is not straightforward. Also, the model is specific to univariate observations whereas our generic model, presented below, allows analysing any kind of observations defined by smooth constraints. This work offers pragmatic solutions to include the idea of time raparametrization in the estimation of trajectories of changes for some specific applications. Nevertheless, we are still lacking a principle and generic approach to deal with the estimation of spatiotemporal variability in longitudinal data sets.

Structured multivariate data such as images, graphs, shapes, or positive definite matrices add further difficulty as these data do not lie in Euclidean spaces. Algebraic operations such as addition or scaling are not defined or do not yield an output of the same type. The spaces in which they live are defined by smooth constraints and may be considered in general as *Riemannian manifolds*. There is no natural extension of LME models on Riemannian manifolds. In Fletcher (2011), the authors proposed an extension of linear regression for Riemannian manifolds, which was later extended for longitudinal data in a group of diffeomorphisms (Singh et al., 2013, 2014). Nevertheless, this longitudinal model strongly depends on a choice of reference time-point to define random effects, therefore making difficult its coupling with time raparametrization. In Su et al. (2014), trajectories are defined by the quotient with a time raparametrization group. This approach allowed

for the definition of statistics in the quotient space, but as a consequence did not yield any estimate of the temporal variability.

This paper proposes a Bayesian mixed-effects model, called *generic spatiotemporal model*, defined for any longitudinal observations on a Riemannian manifold. The fixed effects of the model are used to define an average trajectory and the random effects are used to define individual-specific trajectories. In order to define such individual trajectories, we introduce the notion of "exp-parallelization" of a curve on a Riemannian manifold, based on the idea of "variations of a curve". This construction allows the definition of random effects to account for the spatial variability, the distribution of which (up to an isometric transform) does not depend on a reference time-point. It allows us, therefore, to easily include random time raparametrization to account for the temporal variability in the model. All in one, the model defines distributions of spatiotemporal trajectories for data on any Riemannian manifolds. It gives a systematic way to derive specific nonlinear mixed-effects models for a large variety of observations and Riemannian manifolds.

These models need to then be fitted to given longitudinal data sets. Given their strong non-linear nature, we propose to use a stochastic version of the Expectation-Maximization (EM) (Dempster et al., 1977) algorithm, called the Monte Carlo Markov Chain Stochastic Approximation EM (MCMC-SAEM) algorithm. Theoretical results regarding the convergence of the MCMC-SAEM have been proven in Kuhn and Lavielle (2004), Allassonnière et al. (2010) and ensure that the algorithm maximizes the observed likelihood. This technique allows us to propose a generic algorithm for the estimation of the model parameters. We will instantiate this method for a set of multivariate bounded measurements and for positive definite matrices.

The paper is organized as follows: in Section 2, we give the key mathematical tools and define the generic mixed-effects model with spatiotemporal transformations for manifold-valued measurements. Particular cases of the generic model are given and discussed in Section 3. Section 4 is focused on the MCMC-SAEM which is used to estimate the parameters of the statistical model. Finally, Sections 5.2 and 5 are dedicated to empirical and experimental validations of our generic model.

2. A Bayesian Mixed-Effects Model for Longitudinal Observations on a Riemannian Manifold

This section aims at introducing a notion of Riemannian geometry called "exp-parallelization". Given a group-average trajectory on a Riemannian manifold, the notion of exp-parallelization is used to define individual trajectories. For a comprehensive review of basic concepts of Riemannian geometry, see Do Carmo Valero (1992); Petersen (2006). In this section, we assume that $\mathbb{M}$ is an open subset of $\mathbb{R}^N$ equipped with a Riemannian metric $g^{\mathbb{M}}$.

2.1 Exp-parallelization on a Riemannian Manifold

This section introduces the notion of "exp-parallelization" of a curve on a Riemannian manifold $(\mathbb{M}, g^{\mathbb{M}})$. The notion of "variation of a differentiable curve" on a manifold is defined in Do Carmo Valero (1992) (Chapter 9). It allows defining *neighbouring* curves to a given curve c. In the next section, this construction will be used to define individual trajecto-

ries. Let $(\mathbb{M}, g^{\mathbb{M}})$ denotes a *geodesically complete* Riemannian manifold equipped with its Levi-Civita connection $\nabla^{\mathbb{M}}$.

Definition 1 *Let $c : I \subset \mathbb{R} \to \mathbb{M}$ a differentiable curve on $\mathbb{M}$, $t_0 \in I$ and $\mathbf{w} \in \mathrm{T}_{c(t_0)}\mathbb{M}$ a tangent vector to $\mathbb{M}$ at $c(t_0)$. An* exp-parallelization of c in the direction of $\mathbf{w}$ *is a curve $\boldsymbol{\eta}^{\mathbf{w}}(c, \cdot) : I \to \mathbb{M}$ defined by:*

$$\forall t \in I,\ \boldsymbol{\eta}^{\mathbf{w}}(c, t) = \mathrm{Exp}^{\mathbb{M}}_{c(t)}\big(\mathrm{P}_{c,t_0,t}(\mathbf{w})\big). \tag{1}$$

This construction is illustrated in Fig. 1. Given $t \in I$, parallel transport carries the tangent vector $\mathbf{w}$ from $\mathrm{T}_{c(t_0)}\mathbb{M}$ to $\mathrm{T}_{c(t)}\mathbb{M}$ along the curve c. At the point $c(t)$, a new point on $\mathbb{M}$ is obtained by taking the Riemannian exponential of the tangent vector $\mathrm{P}_{c,t_0,t}(\mathbf{w})$. This new point is denoted by $\boldsymbol{\eta}^{\mathbf{w}}(c, t)$. As t varies, one describes a curve $\boldsymbol{\eta}^{\mathbf{w}}(c, \cdot)$ on $\mathbb{M}$, which can be understood as a "parallel" to the curve c. Note that if $\mathbb{M}$ is the Euclidean space $\mathbb{R}^N$, an exp-parallelization of a curve c, in the direction of a tangent vector $\mathbf{w}_i$, is the translation of c by the vector $\mathbf{w}_i$.

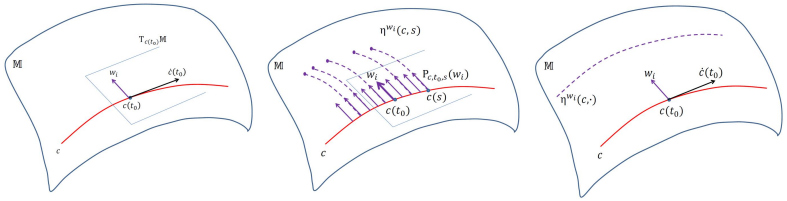


Figure 1: Exp-parallelization on a schematic manifold. Left: a non-zero vector $\mathbf{w}_i$ is chosen in $\mathrm{T}_{c(t_0)}\mathbb{M}$. Middle: the tangent vector $\mathbf{w}_i$ is transported continuously along the curve c. Then, a point $\eta^{\mathbf{w}_i}(c, s)$ is constructed at time s by use of the Riemannian exponential. Right: The curve $\eta^{\mathbf{w}_i}(c, \cdot)$ is the "parallel" resulting from the construction.

2.2 Hierarchical Structure of the Model

In this section, we consider a longitudinal dataset $(\mathbf{y}_{i,j})_{1\leq i\leq p,\, 1\leq j\leq k_i}$. The observations are obtained for a group of p individuals. For the ith individual, the observations $(\mathbf{y}_{i,j})_{1\leq j\leq k_i}$ are obtained at times $t_{i,1} < \ldots < t_{i,k_i}$. The number k_i of observations may vary from one individual to another.

The *generic spatiotemporal* model is a nonlinear mixed-effects model. As emphasized in the introduction, mixed-effects models include fixed and random effects. The *fixed-effects* are parameters which are shared by all the individuals and allow for the description of the model at the population level. *Random effects* are individual-specific random variable which describe the model at the individual level. These two types of effects provide the model with a hierarchical structure. The generic spatiotemporal model is constructed as follows. To begin with, a group-average trajectory $\boldsymbol{\gamma}_0$ is defined on the manifold $\mathbb{M}$. Given the average trajectory, subject-specific trajectories are obtained by spatiotemporal transformations,

which consist of *exp-parallelizations* of the average trajectory $\boldsymbol{\gamma}_0$ and *time reparametrization*. The data points $\mathbf{y}_{i,j}$ are seen as samples along these individual trajectories. If $\boldsymbol{\gamma}_i$ denotes the trajectory of the ith individual, the model writes: $\mathbf{y}_{i,j} = \boldsymbol{\gamma}_i(t_{i,j}) + \varepsilon_{i,j}$, where $\varepsilon_{i,j}$ is a Gaussian noise. The observation $\mathbf{y}_{i,j}$ is therefore considered as a small perturbation of a quantity which lies in a Riemannian manifold.

The group-average trajectory $\boldsymbol{\gamma}_0$ is chosen to be the unique geodesic $\boldsymbol{\gamma}_0 = \boldsymbol{\gamma}_{\mathbf{p}_0,t_0,\mathbf{v}_0}$ of $\mathbb{M}$ which goes through the point $\mathbf{p}_0 \in \mathbb{M}$ at time t_0 and with velocity $\mathbf{v}_0 \in T_{\mathbf{p}_0}\mathbb{M}$. Let $i \in \{1, \ldots, p\}$ denote the ith individual. The subject-specific trajectory $\boldsymbol{\gamma}_i$ is defined in two steps. The first step consists in constructing the curve $\eta^{\mathbf{w}_i}(\boldsymbol{\gamma}_0, \cdot)$, which is an *exp-parallelization* of the average trajectory $\boldsymbol{\gamma}_0$ in the direction of a tangent vector $\mathbf{w}_i \in T_{\mathbf{p}_0}\mathbb{M}$. This tangent vector is chosen orthogonal, for the inner product $g^{\mathbb{M}}_{\mathbf{p}_0}$, to $\dot{\boldsymbol{\gamma}}_0(t_0) = \mathbf{v}_0$. The tangent vectors $(\mathbf{w}_i)_{1\leq i\leq p}$ are random effects of the model, called *space shifts*. The orthogonality condition on the space shifts is discussed below. The second step consists of *reparametrizing in time* the exp-parallelization $\eta^{\mathbf{w}_i}(\boldsymbol{\gamma}_0, \cdot)$. We consider a subject-specific affine mapping ψ_i of the form $\psi_i(t) = \alpha_i(t - t_0 - \tau_i) + t_0$, where $\alpha_i > 0$ and $\tau_i \in \mathbb{R}$ are random effects of our model. The trajectory $\boldsymbol{\gamma}_i$ of the ith individual is $\boldsymbol{\gamma}_i(t) = \boldsymbol{\eta}^{\mathbf{w}_i}(\boldsymbol{\gamma}_0, \psi_i(t))$. The mapping ψ_i is called *time reparametrization* and the random effects α_i (respectively τ_i) are called *acceleration factor* (respectively *time shift*).

2.3 Definition of the Space Shifts

As mentioned above, the space shifts $(\mathbf{w}_i)_{1\leq i\leq p}$ are required to be orthogonal to $\dot{\boldsymbol{\gamma}}_0(t_0) = \mathbf{v}_0$ for the inner product $g^{\mathbb{M}}_{\mathbf{p}_0}$ induced by the Riemannian metric on $\mathbb{M}$. This section discusses different methods which allow to impose this orthogonality condition on the space shifts into a statistical model. The methodological challenge raised by this section consists of defining a (nonlinear) mixed-effects model with smooth constraints on some random effect of the model.

In order to ensure the interpretability of the space shifts, we consider an Independent Component Analysis (ICA) (Hyvärinen et al., 2004) decomposition of each tangent vector $\mathbf{w}_i$ as a linear combination of $N_s < N$ statistically independent tangent vectors $(\mathbf{A}_l)_{1\leq l\leq N_s}$ which are called *independent components* or *independent directions*. As a consequence, the space shifts $(\mathbf{w}_i)_{1\leq i\leq p}$ are defined as follows: $\forall i \in \{1, \ldots, p\}$, $\mathbf{w}_i = \mathbf{A}\mathbf{s}_i = \sum_{l=1}^{N_s} s_{l,i}\mathbf{A}_l$ where $\mathbf{A} = (\mathbf{A}_l)_{1\leq l\leq N_s}$ is such that each $\mathbf{A}_i$ is a vector in $\mathrm{T}_{\dot{\boldsymbol{\gamma}}_0(t_0)}\mathbb{M}$. In this definition, the weights $(s_{l,i})_{1\leq l\leq N_s}$ are random effects of the model called *sources*. By defining the space shifts this way, the generic spatiotemporal model will estimate an ICA decomposition of the space shifts. However, this definition does not ensure the orthogonality of the space shifts. A possible solution to make the vectors $\mathbf{w}_i$ orthogonal to $\mathbf{v}_0 = \dot{\boldsymbol{\gamma}}_0(t_0)$ consists of decomposing each vector $(\mathbf{A}_l)_{1\leq l\leq N_s}$ in an orthonormal basis of $\mathrm{Span}\big(\dot{\boldsymbol{\gamma}}_0(t_0)\big)^{\perp} \subset \mathrm{T}_{\mathbf{p}_0}\mathbb{M}$. Indeed, if $(\mathcal{B}_k)_{1\leq k\leq (N-1)N_s}$ is an orthonormal basis of $\mathrm{Span}\big(\dot{\boldsymbol{\gamma}}_0(t_0)\big)^{\perp}$, we assume that: $\forall l \in \{1, \ldots, N_s\}$, $A_l = \sum_{k=1}^{(N-1)N_s} \beta_{l,k}\mathcal{B}_k$. By construction, each independent component $\mathbf{A}_l$ $(1 \leq l \leq N_s)$ is orthogonal, for the inner product $g^{\mathbb{M}}_{\mathbf{p}_0}$, to $\mathbf{v}_0$. Therefore, each space shift $(\mathbf{w}_i)_{1\leq i\leq p}$ is orthogonal to $\mathbf{v}_0$ since we assumed that it writes as a linear combination of these independent components. In the following, the orthonormal basis is computed using the Gram-Schmidt algorithm or the Householder method (Coleman and Sorensen, 1984).

Moreover, it is important to note that the choice of the form of the distribution of the space-shifts does not depend on the reference time-point t_0. Indeed, the $\mathbf{w}_i = \mathbf{A}\mathbf{s}_i$ are defined in the tangent space of the curve at point $\mathbf{p}_0 = \gamma_0(t_0)$. At another point $\mathbf{p}'_0 = \gamma_0(t'_0)$, space-shifts become $\mathbf{w}'_i = \mathrm{P}_{\gamma_0,t_0,t'_0}\mathbf{w}_i$, where $\mathrm{P}_{\gamma_0,t_0,t'_0}$ is an orthogonal matrix. They are therefore distributed according to $\mathbf{w}'_i = P_{\gamma_0,t_0,t}\mathbf{A}\mathbf{s}_i$: the distribution of the sources $\mathbf{s}_i$ does not change and the independent components (i.e. the columns of $\mathbf{A}$) are adjusted to the new position on the average trajectory. In particular, the variance of the $\mathbf{w}'_i$ is invariant. This property holds for isometric invariant distributions. For instance, if $\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, then $\mathbf{w}'_i \sim \mathcal{N}(\mathbf{0}, \mathrm{P}_{\gamma_0,t_0,t'_0}\,\mathbf{\Sigma}\,\mathrm{P}^{\top}_{\gamma_0,t_0,t'_0})$.

2.4 The Statistical Model

The generic spatiotemporal model assumes that the jth observation of the ith individual derives from:

$$\mathbf{y}_{i,j} = \eta^{\mathbf{w}_i}(\boldsymbol{\gamma}_0, \psi_i(t_{i,j})) + \boldsymbol{\varepsilon}_{i,j}. \tag{2}$$

With the notations introduced above, let $\mathbf{z}_{\mathrm{pop}} = (\mathbf{p}_0, t_0, \mathbf{v}_0, (\beta_{l,k})_{l,k})$ denote the *population variables* and $(\mathbf{z}_i)_{1\leq i\leq p}$ denote the set of *individual variables* with: $\mathbf{z}_i = (\xi_i, \tau_i, (s_{l,i})_{l,i})$. Both $\mathbf{z}_{\mathrm{pop}}$ and $(\mathbf{z}_i)_{1\leq i\leq p}$ are *latent* (or random) variables assumed independent of each other and distributed as follows:

$$\begin{aligned} &\mathbf{p}_0 \sim \mathcal{N}(\overline{\mathbf{p}_0}, \sigma^2_{\mathbf{p}_0}), \quad t_0 \sim \mathcal{N}(\overline{t_0}, \sigma^2_{t_0}),\\ &\mathbf{v}_0 \sim \mathcal{N}(\overline{v_0}, \sigma^2_{v_0}), \quad \beta_{l,k} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\overline{\beta}_{l,k}, \sigma^2_{\beta}), \end{aligned} \tag{3}$$

and $\psi_i(t) = \alpha_i(t - t_0 - \tau_i) + t_0$ with $\alpha_i = \exp(\xi_i)$ and :

$$\xi_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2_{\xi}), \quad \tau_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2_{\tau}), \quad s_{l,i} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1). \tag{4}$$

where $\sigma^2_{\mathbf{p}_0}$, $\sigma^2_{t_0}$, $\sigma^2_{\mathbf{v}_0}$ and σ^2_{β} are fixed variance parameters. The noise variables $(\boldsymbol{\varepsilon}_{i,j})_{i,j}$ are assumed independent of the other random variables and identically distributed:

$$\boldsymbol{\varepsilon}_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2). \tag{5}$$

Let $\boldsymbol{\theta}_{\mathrm{var}} = (\sigma^2_{\xi}, \sigma^2_{\tau}, \sigma^2)$ denote the variance parameters which are not fixed and $\boldsymbol{\theta} = \left(\overline{\mathbf{p}_0}, \overline{t_0}, \overline{\mathbf{v}_0}, (\overline{\beta}_{l,k}), \boldsymbol{\theta}_{\mathrm{var}}\right)$ be the *parameters* of the model. The domain of $\boldsymbol{\theta}$ is denoted by Θ and defined by:

$$\begin{aligned} \Theta = \big\{\boldsymbol{\theta} = &(\overline{\mathbf{p}_0}, \overline{\mathbf{v}_0}, \overline{t_0}, (\overline{\beta}_{l,k})_{l,k}, \boldsymbol{\theta}_{\mathrm{var}}) \,/\, (\overline{\mathbf{p}_0}, \overline{\mathbf{v}_0}) \in \mathrm{T}\mathbb{M},\\ &\overline{t_0} \in \mathbb{R}, (\overline{\beta}_{l,k})_{l,k} \in \mathbb{R}^{(N-1)N_s},\, \boldsymbol{\theta}_{\mathrm{var}} \in]0, +\infty[^3\big\}. \end{aligned} \tag{6}$$

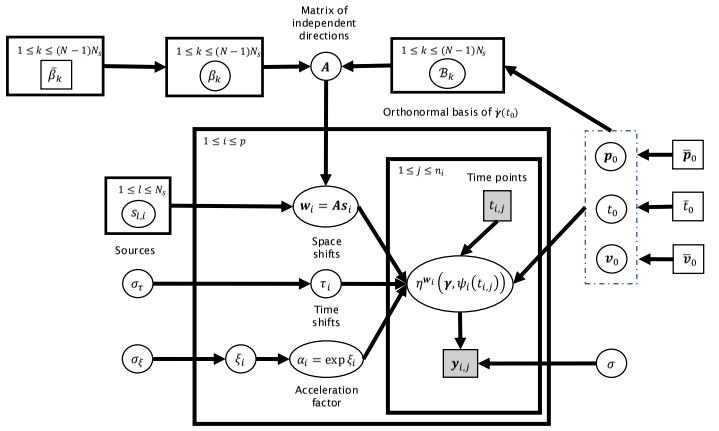


Figure 2: Graphical representation of the generic spatiotemporal model. Round shapes indicate latent variables of the model. Boxes with indexes in the upper left corner indicate a repetition. Shaded boxes indicate that the quantity is observed. This figure illustrates the dependence between the variables of the generic spatiotemporal model.

2.4.1 Discussion

The additive, or *extrinsic*, noise model in Eq. (2) makes sense because we assumed that $\mathbb{M}$ is a subset of the Euclidean space $\mathbb{R}^N$. The term $\eta^{\mathbf{w}_i}(\boldsymbol{\gamma}_0, \psi_i(t_{i,j}))$ belongs to the manifold $\mathbb{M}$ while the noise term $\boldsymbol{\varepsilon}_{i,j}$ is added in the underlying Euclidean space. However, the noise model is not *intrinsic* in the sense that the noise term $\boldsymbol{\varepsilon}_{i,j}$ is not added on the manifold. In Fletcher (2011), the author have considered an intrinsic noise model which would write: $\mathbf{y}_{i,j} = \mathrm{Exp}_{\eta^{\mathbf{w}_i}(\boldsymbol{\gamma}_0,\psi_i(t_{i,j}))}(\boldsymbol{\varepsilon}_{i,j})$. This noise model allows for it to remain on the manifold. Still, obtaining maximum *a posteriori* estimates of the parameters with this intrinsic noise model is more difficult as the model likelihood might not be available in closed-form.

We assume a centred log-normal distribution for the acceleration factors α_i. Indeed, this choice of probability distribution ensures the positiveness of the acceleration factors. With this assumption, the individual time reparametrizations do not reverse time. Other probability distributions, such as the exponential distribution, could have been considered.

3. Particular Cases of the Generic Spatiotemporal Model

The generic spatiotemporal model, introduced in the previous section, is a statistical tool which allows, given a Riemannian manifold $\mathbb{M}$ equipped with a Riemannian metric $g^{\mathbb{M}}$, to instantiate a large variety of nonlinear mixed-effects models. This section aims at describing the generic spatiotemporal model for classical Riemannian manifolds. The models for one-dimensional geodesically complete Riemannian manifolds given in Section 3.1 were introduced in Schiratti et al. (2015c). The progression models, given in Section 3.3 were introduced in Schiratti et al. (2015b).

3.1 The case of One-Dimensional Geodesically Complete Riemannian Manifolds

Let M be an open interval of $\mathbb{R}$ equipped with a Riemannian metric g^M, for which it is geodesically complete. The case of one dimensional manifolds is particular because, for all $p_0 \in M$, $\mathrm{T}_p M \simeq \mathbb{R}$ and given $v_0 \in \mathrm{T}_{p_0} M$, there is only one tangent vector w at p_0 which is orthogonal (for the inner product $g^M_{p_0}$) to v_0 : $w = 0$. As a result, if γ_0 is a geodesic of M, $t_0 \in \mathbb{R}$ and $w = 0$, then for all $s \in \mathbb{R}$, $\eta^w(\gamma_0, s) = \gamma_0(s)$. Therefore, the generic spatiotemporal model writes: $y_{i,j} = \gamma_0\big(\psi_i(t_{i,j})\big) + \varepsilon_{i,j}$, with, for all $i \in \{1, \ldots, p\}$, $\psi_i(t) = \alpha_i(t - t_0 - \tau_i) + t_0$ and $\alpha_i = \exp(\xi_i)$.

We show that, in this one-dimensional framework, a different presentation of the generic spatiotemporal model is possible. This presentation provides a different insight on the role of the latent variables $(\alpha_i, \tau_i)_{1 \leq i \leq p}$. Let $p_0 \in M$, $t_0 \in \mathbb{R}$ and $v_0 \in \mathrm{T}_{p_0} M \simeq \mathbb{R}$. Let γ_0 be the group-average trajectory defined as the geodesic which goes through the point p_0 at time t_0 and with velocity v_0. Let $1 \leq i \leq p$. The trajectory γ_i of the ith individual is defined as the geodesic γ_i which goes through the point p_0 at time $t_0 + \tau_i$ and with velocity $\alpha_i v_0$. Having defined individual trajectories of progression, the observations are seen as random samples along these trajectories: $y_{i,j} = \gamma_i(t_{i,j}) + \varepsilon_{i,j}$. In this definition, the acceleration factor α_i allows characterizing whether the ith individual is progressing faster ($\alpha_i > 1$) or slower ($\alpha_i < 1$) than the average trajectory. The time shift τ_i allows determining whether the ith individual is evolving ahead ($\tau_i < 0$) or behind ($\tau_i > 0$) the average trajectory. Moreover, it follows from a unicity property of the geodesics that, for all $i \in \{1, \ldots, p\}$, $\gamma_i(t) = \gamma_0\big(\psi_i(t)\big)$. This result legitimises the choice of affine time reparametrizations of the form $\psi_i : t \mapsto \alpha_i(t - t_0 - \tau_i) + t_0$.

3.1.1 The "Straight Lines Model"

Unbounded observations can be considered as points on the real line. The real line $M = \mathbb{R}$ equipped with its canonical metric is a geodesically complete one-dimensional Riemannian manifold. For the canonical metric, the geodesics are of the form $t \in \mathbb{R} \mapsto at + b$ with $(a, b) \in \mathbb{R}^2$. The generic spatiotemporal model writes: $y_{i,j} = p_0 + \alpha_i v_0(t_{i,j} - t_0 - \tau_i) + \varepsilon_{i,j}$. This model is referred to as the *univariate straight lines model*. Note that, even though the average and individual trajectories are straight lines, the model is *not* linear due to the multiplication between the random effects α_i and τ_i.

We propose to compare the nonlinear straight lines model to the random slope and intercept model, discussed in the introduction. This linear mixed-effects model writes: $y_{i,j} = (\overline{a} + a_i)(t_{i,j} - t_0) + (\overline{b} + b_i) + \varepsilon_{i,j}$, where $(A_i, B_i)_{1 \leq i \leq p}$ are random effects of the model which are assumed to be independent of each other and normally distributed with mean 0 and variance-covariance matrix $\mathbf{D}$. The fixed effects of this model are $(\overline{a}, \overline{b}, t_0)$. This linear model analyzes the distribution of the observations at a fixed reference time t_0. In comparison, the straight lines model analyzes the distribution of the times at which the observations reach a given value of the measurements. The two models illustrated in Fig. 3 are not equivalent because the observations generated by these two models are different. Indeed, the *random slope and intercept model* generates straight lines whose slope $\overline{a} + a_i$ and intercept $\overline{b} + b_i - t_0(\overline{a} + a_i)$ both follow Gaussian distributions. With the straight lines model, only the slope $\overline{a}a_i$ follows a Gaussian distribution. The intercept $\overline{b} - (t_0 + \tau_i)\overline{a}a_i$ actually does not follow a Gaussian distribution. It is the sum of two random variables, one following a Gaussian distribution, and the other one being the product of two independent Gaussian random variables, a_i and τ_i. The distribution of $a_i\tau_i$ is that of a (weighted) sum of two chi-square random variables (which are, in general, not independent).

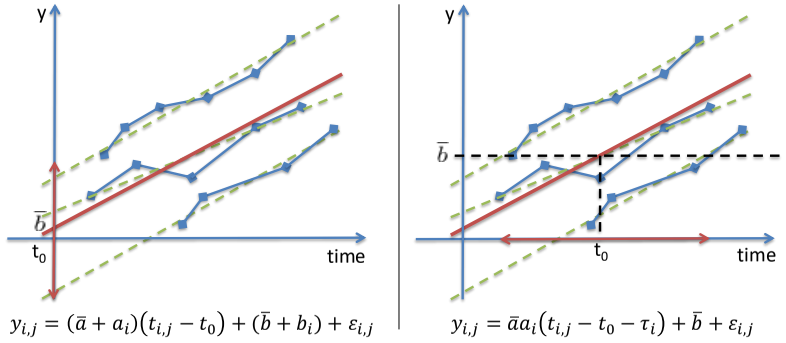


Figure 3: Schematic example of a random slope and intercept linear mixed-effects model (left) and straight lines model (right).

3.2 The "Logistic Curves Model"

If the observations are bounded, such as percentages or scores to a test, the measurements can be normalized to produce new observations in the open interval $M =]0, 1[$. We consider that this open interval of the real line is equipped with the Riemannian metric $g = (g_p)_{p \in]0,1[}$ where : $\forall p \in \mathbb{M} =]0, 1[$, $\forall (u, v) \in T_p\mathbb{M} \times T_p\mathbb{M}$, $g_p(u, v) = uM(p)v$, where $M(p) = 1/\big(p^2(1-p)^2\big)$. This Riemannian metric on $]0, 1[$ is obtained as the push-forward of the Euclidean metric on $\mathbb{R}$ by the logit transform. In Schiratti et al. (2015c), it is proven that $\mathbb{M} =]0, 1[$ is a geodesically complete Riemannian manifold and that the generic spatiotemporal model

writes:

$$y_{i,j} = \left(1 + \left(\frac{1}{p_0} - 1\right)\exp\Big(-\frac{v_0\alpha_i(t_{i,j} - t_0 - \tau_i)}{p_0(1-p_0)}\Big)\right)^{-1} + \varepsilon_{i,j}. \tag{7}$$

In this framework, the Riemannian logarithm at $p = 1/2$, which corresponds to the inflexion point of the logistics, is given by: $\forall q \in]0,1[$, $\mathrm{Log}_{1/2}(q) = (1/4)\mathrm{logit}(q)$. However, in (7), the point p_0 is not fixed to $1/2$, but is estimated as a fixed effect. The model estimates the p_0, and therefore the best tangent space, which best describes the observations. Furthermore, even if one fixes $p = 1/2$, the model lifted up on the tangent space remains nonlinear due to the multiplication between the random effects α_i and τ_i. Therefore, the logistic curves model is not equivalent to a linear model on the logit transform of the observations.

3.3 A Progression Model

The generic spatiotemporal model can be used to study the temporal progression of a family of features which characterize the evolution of a biological phenomenon. We assume that each feature is described by repeated *univariate* observations, which are random perturbations of quantities lying in a one-dimensional geodesically complete Riemannian manifold (M, g^M), open subset of $\mathbb{R}$. For each individual, at each time point, the observations $(\mathbf{y}_{i,j})_{1\leq i\leq p,\, 1\leq j\leq k_i}$ consist of a N-dimensional vector of univariate features. Hence, for this progression model, the observations $(\mathbf{y}_{i,j})_{1\leq i\leq p,\, 1\leq j\leq k_i}$ are considered as random perturbations of quantities which belong to the product manifold $\mathbb{M} = M \times \ldots \times M = M^N$. Since each Riemannian manifold (M, g^M) is geodesically complete, $\mathbb{M}$ equipped with the product metric is also geodesically complete.

On the product manifold $\mathbb{M} = M^N$, equipped with the product metric, a geodesic is of the form $t \mapsto \big(\gamma_1(t), \ldots, \gamma_N(t)\big)$, where $\gamma_1, \ldots, \gamma_N$ are geodesics of the one-dimensional Riemannian manifold M. Because we would like to model the joint temporal progression of N features, we propose to choose the group-average trajectory among a parametric family of geodesics of $\mathbb{M}$. This family is of the form:

$$\Big\{\boldsymbol{\gamma}_{0,\boldsymbol{\delta}} : t \in \mathbb{R} \mapsto \big(\gamma_0(t), \gamma_0(t+\delta_1), \ldots, \gamma_0(t+\delta_{N-1})\big)\Big\}, \tag{8}$$

with $\boldsymbol{\delta} = (0, \delta_1, \ldots, \delta_{N-1})^\top$, $\delta_i \in \mathbb{R}$ and γ_0 denotes a geodesic of the one-dimensional Riemannian manifold g^M which goes through a point $p_0 \in M$ at time t_0 with velocity v_0. The relative delay between two consecutive biomarkers is given by the parameters δ_i $(1 \leq i \leq N-1)$. The vector $\boldsymbol{\delta}$ is to be estimated as a fixed effect of the model. The first component of the vector $\boldsymbol{\delta}$ is chosen to be equal to zero in order to ensure the identifiability of the model. Note that assuming that the group average belongs to this parametric family of geodesics is equivalent to assuming that the progression of each feature is described by trajectories which have the same shape but are shifted in time.

Lemma 2 *Let $\boldsymbol{\gamma}$ be a geodesic of the product manifold $\mathbb{M} = M^N$ and let $t_0 \in \mathbb{R}$. If $\eta^{\mathbf{w}}(\boldsymbol{\gamma}, \cdot)$ denotes an exp-parallelization of the geodesic $\boldsymbol{\gamma}$ using $\mathbf{w} = (w_1, \ldots, w_N) \in \mathrm{T}_{\boldsymbol{\gamma}(t_0)}\mathbb{M}$ and with $\boldsymbol{\gamma}(t) = (\gamma_1(t), \ldots, \gamma_N(t))$, we have $\eta^{\mathbf{w}}(\boldsymbol{\gamma}, s) = \big(\gamma_1\big(\frac{w_1}{\dot{\gamma}_1(t_0)} + s\big), \ldots, \gamma_N\big(\frac{w_N}{\dot{\gamma}_N(t_0)} + s\big)\big)$, $s \in \mathbb{R}$.*

In this framework, an exp-parallelization of the group-average trajectory $\boldsymbol{\gamma}_{0,\boldsymbol{\delta}}$ can be computed using the result given in Lemma 2. Indeed, it follows from Lemma 2 that the generic spatiotemporal model 2 writes :

$$y_{i,j,k} = \gamma_0\left(\frac{w_{i,k}}{\dot{\gamma}_0(t_0 + \delta_{k-1})} + \psi_i(t_{i,j}) + \delta_{k-1}\right) + \varepsilon_{i,j,k}, \tag{9}$$

where, for all $k \in \{1, \ldots, N\}$, $(\mathbf{y}_{i,j})_k$ denotes the kth component of $\mathbf{y}_{i,j}$. In other words, $(\mathbf{y}_{i,j})_k$ is the observation associated to the kth biomarker, for the ith individual, at the jth time point. Similarly, $(\mathbf{w}_i)_k$ denotes the kth component of the space shift $\mathbf{w}_i$. For all $i \in \{1, \ldots, p\}$, $\psi_i(t) = \alpha_i(t - t_0 - \tau_i) + t_0$ is the individual specific time reparametrization. This model is referred to as the *progression model*. For this model, the latent variables are: $\mathbf{z}_{\mathrm{pop}} = \big(p_0, t_0, v_0, (\delta_k)_{1\leq k\leq N-1}, (\beta_{l,k})_{l,k}\big)$ and, for all $i \in \{1, \ldots, p\}$, $\mathbf{z}_i = \big(\xi_i, \tau_i, (s_{l,i})_{l,i}\big)$. The definition of the individual latent variables $(\mathbf{z}_i)_{1\leq i\leq p}$ remains unchanged. For the population latent variables $\mathbf{z}_{\mathrm{pop}}$, the variables $(\delta_k)_{1\leq k\leq N-1}$ are added. We assume that the latent variables $\mathbf{z}_{\mathrm{pop}}$ are distributed as follows:

$$p_0 \sim \mathcal{N}(\overline{p_0}, \sigma^2_{p_0}),\ t_0 \sim \mathcal{N}(\overline{t_0}, \sigma^2_{t_0}),\ v_0 \sim \mathcal{N}(\overline{v_0}, \sigma^2_{v_0}), \tag{10}$$

and

$$\beta_{l,k} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\overline{\beta}_{l,k}, \sigma^2_\beta),\ \delta_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\overline{\delta}_k, \sigma^2_\delta), \tag{11}$$

where $\sigma^2_{p_0}$, $\sigma^2_{t_0}$, $\sigma^2_{v_0}$ and σ^2_δ are *fixed* variance parameters. Similarly to the generic spatiotemporal model, the latent variables are assumed independent of each other and independent of the noise variables $\boldsymbol{\varepsilon}_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \sigma^2\mathrm{I}_N)$.

3.4 The Symmetric Positive Definite (SPD) Matrices Model

In this section, we describe how the generic spatiotemporal model can be used to analyze longitudinal datasets of symmetric positive definite matrices. Such datasets may arise in Diffusion Tensor Imaging (DTI) or when observing the temporal evolution of covariance matrices.

The space of 3×3 symmetric positive definite matrices is usually denoted by SDP(3), which is an open subset of the vector space of $(3,3)$ symmetric real matrices, denoted by Sym(3). By identifying Sym(3) with $\mathbb{R}^6$, $\mathbb{M} = \mathrm{SDP}(3)$ can be considered as an open submanifold of $\mathbb{R}^6$. Indeed, $\mathbb{M}$ can be equipped with a Riemannian metric. In Pennec et al. (2006), the authors defined an *affine-invariant* Riemannian metric on SDP(3). Equipped with this metric, the space of symmetric positive definite matrices is a geodesically complete Riemannian manifold, without boundaries (matrices with null eigenvalues are at infinity). The results presented below are obtained with the affine-invariant metric on SDP(3).

If $\boldsymbol{\Sigma} \in \mathrm{SDP}(3)$, $\boldsymbol{\Sigma}^{1/2}$ denotes the unique symmetric positive definite matrix $\mathbf{S}$ such that : $\mathbf{S}^2 = \boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}^{-1/2}$ denotes its inverse. Let $\mathrm{T}_{\boldsymbol{\Sigma}}\mathbb{M}$ denote the tangent space to $\mathbb{M}$ at the point $\boldsymbol{\Sigma}$. $\mathrm{T}_{\boldsymbol{\Sigma}}\mathbb{M}$ can be identified with Sym(3) and is equipped with the inner product $\langle\cdot,\cdot\rangle_{\boldsymbol{\Sigma}}$ defined by : $\forall(\mathbf{W}_1, \mathbf{W}_2) \in \mathrm{T}_{\boldsymbol{\Sigma}}\mathbb{M}$, $\langle \mathbf{W}_1, \mathbf{W}_2\rangle_{\boldsymbol{\Sigma}} = \mathrm{tr}\big(\boldsymbol{\Sigma}^{-1/2}\mathbf{W}_1^\top\boldsymbol{\Sigma}^{-1}\mathbf{W}_2\boldsymbol{\Sigma}^{-1/2}\big)$. In order to describe exp-parallelization in the Riemannian manifold $\mathrm{SPD}(n)$ equipped with

the affine invariant metric, we give a closed-form expression of the parallel transport. The result given in Lemma 3 is based in the work of Lenglet et al. (2006). For $\mathbf{P}_0 \in \mathrm{SPD}(n)$, $t_0 \in \mathbb{R}$, $\mathbf{V}_0 \in \mathrm{T}_{\mathbf{P}_0}\mathrm{SPD}(n) \simeq \mathrm{SYM}(n)$, the geodesic $\boldsymbol{\gamma}_0$, defined by $\boldsymbol{\gamma}_0(t) = \mathrm{Exp}_{\mathbf{P}_0,t_0}(\mathbf{V}_0)(t)$ for all $t \in \mathbb{R}$, is given by : $\boldsymbol{\gamma}_0(t) = \mathbf{P}_0^{1/2} \exp\big(t\mathbf{P}_0^{-1/2}\mathbf{V}_0\mathbf{P}_0^{-1/2}\big)\mathbf{P}_0^{1/2}$, where $\mathbf{P}_0^{1/2}$ (respectively $\mathbf{P}_0^{-1/2}$) denotes the unique symmetric positive definite square root of $\mathbf{P}_0$ (respectively its inverse). The proof of Lemma 3 can be adapted to obtain the expression of the geodesics. One can also obtain this expression by noting that the geodesic starting at I_n with velocity $\mathbf{V} \in \mathrm{SYM}(n)$ is given by $\exp(t\mathbf{V})$ and use the invariance of the affine-invariant metric under congruent transformations. Finally, the expression of the parallel transport along such a geodesic is given by the following lemma.

Lemma 3 *Let $\mathbf{P}_0 \in \mathrm{SPD}(n)$, $t_0 \in \mathbb{R}$ and $\mathbf{V}_0 \in \mathrm{T}_{\mathbf{P}_0}\mathrm{SPD}(n) \simeq \mathrm{SYM}(n)$. Let $\boldsymbol{\gamma}_0$ be the geodesic defined as above. If $\mathbf{W}$ is a tangent vector in $\mathrm{T}_{\mathbf{P}_0}\mathrm{SPD}(n)$, the parallel transport $\mathrm{P}_{\boldsymbol{\gamma}_0,t_0,t}(\mathbf{W})$ is given by:*

$$\forall t \in \mathbb{R},\ \mathrm{P}_{\boldsymbol{\gamma}_0,t_0,t}(\mathbf{W}) = \exp\Big(\frac{t-t_0}{2}\mathbf{V}_0\mathbf{P}_0^{-1}\Big)\mathbf{W}\exp\Big(\frac{t-t_0}{2}\mathbf{P}_0^{-1}\mathbf{V}_0\Big). \tag{12}$$

The proof of Lemma 3 is given in Appendix A.2. It follows from this lemma that the generic spatiotemporal model writes :

$$\mathbf{Y}_{i,j} = \mathbf{P}_i(t_{i,j})^{1/2}\exp\big(\mathbf{P}_i(t_{i,j})^{-1/2}\mathbf{V}_i(t_{i,j})\mathbf{P}_i(t_{i,j})^{-1/2}\big)\mathbf{P}_i(t)^{1/2} + \boldsymbol{\varepsilon}_{i,j}, \tag{13}$$

with, for all $t \in \mathbb{R}$,

$$\mathbf{P}_i(t) = \mathbf{P}_0^{1/2}\exp\big(\alpha_i(t-t_0-\tau_i)\mathbf{P}_0^{-1/2}\mathbf{V}_0\mathbf{P}_0^{-1/2}\big)\mathbf{P}_0^{1/2}, \tag{14}$$

and:

$$\mathbf{V}_i(t) = \exp\Big(\frac{\alpha_i(t-t_0-\tau_i)}{2}\mathbf{V}_0\mathbf{P}_0^{-1}\Big)\mathbf{W}_i\exp\Big(\frac{\alpha_i(t-t_0-\tau_i)}{2}\mathbf{P}_0^{-1}\mathbf{V}_0\Big). \tag{15}$$

The prior distribution for the matrices $\mathbf{P}_0$, $\mathbf{V}_0$ and $(\boldsymbol{\varepsilon}_{i,j})_{i,j}$ are defined as follows: $\mathbf{P}_0 \sim \mathcal{SN}(\overline{\mathbf{P}_0}, \sigma_{P_0}^2)$, $\mathbf{V}_0 \sim \mathcal{SN}(\overline{\mathbf{V}_0}, \sigma_{V_0}^2)$ and $\boldsymbol{\varepsilon}_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{SN}(\mathbf{0}, \sigma^2)$, where $\mathcal{SN}$ denotes the Gaussian distribution on the vector space $\mathrm{Sym}(n)$. Given $\overline{\mathbf{M}} \in \mathrm{SYM}(n)$, the probability distribution $\mathcal{SN}(\overline{\mathbf{M}}, \sigma^2)$ on $\mathrm{Sym}(n)$ is defined by the density function q such that: $q(\mathbf{M}) = \frac{1}{(2\pi)^{m/2}\sigma^m}\exp\big(-\frac{1}{2\sigma^2}\mathrm{tr}\big[(\overline{\mathbf{M}}-\mathbf{M})^2\big]\big)$ with $\mathbf{M} \in \mathrm{Sym}(n)$ and with $m = n(n+1)/2$. The "standard" distribution $\mathcal{SN}(\mathbf{0}, 1)$ is used in physics and in the theory of random matrices. It is sometimes called *Gaussian Orthogonal Ensemble*. The probability distribution of the other random effects of the model are defined as in Section 2.2. This model will be referred to as the *symmetric positive definite matrices model* or $\mathrm{SPD}(n)$ *matrices model*. In Arsigny et al. (2006), the authors considered the space $\mathrm{SPD}(n)$ equipped with the log-Euclidean metric. This metric provides the space of symmetric positive definite matrices with a structure of Riemannian manifold. Unlike with the affine-invariant metric, the space $\mathrm{SPD}(n)$ endowed with the Log-Euclidean metric is a *flat* Riemannian manifold, meaning that its sectional curvature is null everywhere. By contrast, the space $\mathrm{SPD}(n)$ equipped with the affine-invariant metric is a Riemannian manifold of non-positive curvature (Skovgaard, 1984; Moakher and

Zéraï, 2011) with no cut-locus. Within the Log-Euclidean framework, the geodesics are of the form: $\exp(\mathbf{V}_1 + t\mathbf{V}_2)$ with $\mathbf{V}_1, \mathbf{V}_2 \in \mathrm{SYM}(n)$. As expected, the geodesics are the image of a straight line in $\mathrm{SYM}(n)$ by the matrix exponential map. Future developments should include comparisons with the log-euclidean metric on the space $\mathrm{SPD}(3)$.

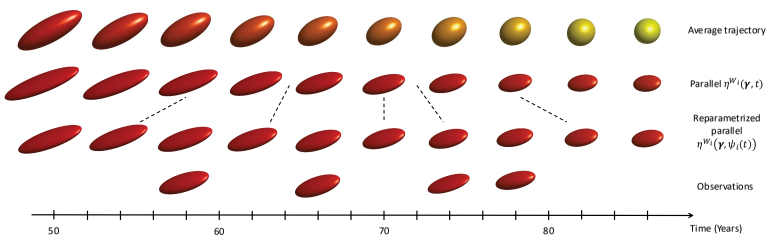


Figure 4: Simulated evolutions of diffusion tensors. First row : average trajectory from a highly anisotropic diffusion tensor to a sphere. Second row : a parallel to the average trajectory obtained using a random space shift $\mathbf{W}_i$. Third row : the reparametrization of the exp-parallelization with $\alpha_i = 0.7$ and $\tau_i = -4$ (years). Fourth row : the observations are samples from the reparametrized exp-parallelization. Each diffusion tensor is colored according to its fractional anisotropy (red for a highly anisotropic tensors, yellow for a sphere).

4. Parameters Estimation

The generic spatiotemporal model (Eq. (2)) is a nonlinear mixed-effects model for which the *observed* likelihood is not available in closed-form. Indeed, it writes as an intractable integral which could only be approximated. In order to produce maximum likelihood estimates, we could use the Expectation Maximization (EM) algorithm (Dempster et al., 1977). The first step of the EM algorithm, usually called "E-step", requires us to compute the expectation of the log-complete likelihood (the likelihood of the observations $\mathbf{y} = (\mathbf{y}_{i,j})_{1\leq i\leq p, 1\leq j\leq k_i}$ and the latent variables $\mathbf{z}$) with respect to the conditional distribution of the latent variables knowing the observations and the current values of the parameters. In the case of our model, this expectation cannot be computed in closed-form.

Therefore, we choose to estimate the parameters of the generic spatiotemporal model by using a stochastic version of the EM algorithm, in which this step is replaced by a stochastic approximation. This algorithm is the Monte Carlo Markov Chains (MCMC) Stochastic Approximation EM (MCMC-SAEM) algorithm (Allassonnière et al., 2010). The MCMC-SAEM iterates, until convergence, between three steps: simulation, stochastic approximation and maximization.

4.1 Simulation Step

Algorithm 1 The Block Metropolis-Hastings-within-Gibbs sampler

Require: Set of latent variables $\mathbf{z}^{(k-1)} = \big(\mathbf{z}_{\text{pop}}^{(k-1)}, (\mathbf{z}_i^{(k-1)})_{1\leq i\leq p}\big)$, current estimate of the parameters $\boldsymbol{\theta}^{(k-1)}$, variance-covariance matrices $\mathbf{D}_{\text{pop}}$ and $(\mathbf{D}_i)_{1\leq i\leq p}$ and $\boldsymbol{\theta}_{\text{hyper}}$

Ensure: Set of latent variables $\mathbf{z}^{(k)}$

Block $\mathbf{z}_{\text{pop}}^{(k)}$ of population latent variables:

Draw a candidate $\mathbf{z}_{\text{pop}}^* \sim \mathcal{N}\big(\mathbf{z}_{\text{pop}}^{(k-1)}, \mathbf{D}_{\text{pop}}\big)$

Compute the acceptance ratio $\alpha(\mathbf{z}_{\text{pop}}^{(k-1)}, \mathbf{z}_{\text{pop}}^*)$ defined by:

$$\alpha(\mathbf{z}_{\text{pop}}^{(k-1)}, \mathbf{z}_{\text{pop}}^*) = \frac{q(\mathbf{y} \mid \mathbf{z}_{\text{pop}}^*, (\mathbf{z}_i^{(k-1)})_{1\leq i\leq p}, \boldsymbol{\theta}^{(k-1)}) q_{\text{pop}}(\mathbf{z}_{\text{pop}}^* \mid \boldsymbol{\theta}^{(k-1)})}{q(\mathbf{y} \mid \mathbf{z}_{\text{pop}}^{(k-1)}, (\mathbf{z}_i^{(k-1)})_{1\leq i\leq p}, \boldsymbol{\theta}^{(k-1)}) q_{\text{pop}}(\mathbf{z}_{\text{pop}}^{(k-1)} \mid \boldsymbol{\theta}^{(k-1)})} \wedge 1.$$

Draw $U \sim \text{Uniform}\big([0,1]\big)$

Set: $\mathbf{z}_{\text{pop}}^{(k)} = \mathbf{z}_{\text{pop}}^*$ if $U \leq \alpha(\mathbf{z}_{\text{pop}}^{(k-1)}, \mathbf{z}_{\text{pop}}^*)$ and $\mathbf{z}_{\text{pop}}^{(k)} = \mathbf{z}_{\text{pop}}^{(k-1)}$ otherwise.

for $i = 1 \dots p$ **do**

  Blocks $(\mathbf{z}_i^{(k)})_{1\leq i\leq p}$ of individual latent variables:

  Draw a candidate $\mathbf{z}_i^* \sim \mathcal{N}\big(\mathbf{z}_i^{(k-1)}, \mathbf{D}_{\text{indiv}}\big)$

  Compute the acceptance ratio $\alpha(\mathbf{z}_i^{(k-1)}, \mathbf{z}_i^*)$ defined by:

$$\alpha(\mathbf{z}_i^{(k-1)}, \mathbf{z}_i^*) = \frac{q(\mathbf{y} \mid \mathbf{z}_{\text{pop}}^{(k)}, \mathbf{z}_{-i}^{(k-1),(k)}, \mathbf{z}_i^*, \boldsymbol{\theta}^{(k-1)}) q_i(\mathbf{z}_i^* \mid \boldsymbol{\theta}^{(k-1)})}{q(\mathbf{y} \mid \mathbf{z}_{\text{pop}}^{(k)}, \mathbf{z}_{-i}^{(k-1),(k)}, \mathbf{z}_i^{(k-1)}, \boldsymbol{\theta}^{(k-1)}) q_i(\mathbf{z}_i^{(k-1)} \mid \boldsymbol{\theta}^{(k-1)})} \wedge 1.$$

  Draw $U \sim \text{Uniform}([0,1])$

  Set: $\mathbf{z}_i^{(k)} = \mathbf{z}_i^*$ if $U \leq \alpha(\mathbf{z}_i^{(k-1)}, \mathbf{z}_i^*)$ or $\mathbf{z}_i^{(k)} = \mathbf{z}_i^{(k-1)}$ otherwise.

end for

Return: $\mathbf{z}^{(k)} = \big(\mathbf{z}_{\text{pop}}^{(k)}, (\mathbf{z}_i^{(k)})_{1\leq i\leq p}\big)$.

If $\tilde{\boldsymbol{\theta}}^{(k-1)}$ denotes the current estimate of $\tilde{\boldsymbol{\theta}}$ at the beginning of the kth iteration of the MCMC-SAEM, the simulation step consists of drawing a sample $\mathbf{z}^{(k)}$ from the transition kernel $\boldsymbol{\pi}_{\boldsymbol{\theta}^{(k-1)},\mathbf{y}}(\mathbf{z}^{(k-1),\cdot})$ of an ergodic Markov chain whose stationary distribution is the conditional distribution $q(\mathbf{z} \mid \mathbf{y}, \tilde{\boldsymbol{\theta}}^{(k-1)})$, the distribution of the latent variables $\mathbf{z}$ knowing the observations $\mathbf{y}$ and $\tilde{\boldsymbol{\theta}}^{(k-1)}$. This step is achieved using a Monte Carlo Markov Chain (MCMC) sampler. We chose to use a Block Metropolis-Hastings-within-Gibbs (Block MHwG) sampler for the sampling step of the MCMC-SAEM. Each Metropolis-Hastings step of the algorithm consists in a multivariate symmetric random walk. The Block MHwG sampler updates simultaneously *block* (or sets) of latent variables then, at each iteration, each block is updated conditionally on the others. Even though the latent variables can be grouped in several ways, we chose to group the latent variables as follows: $\{\mathbf{z}_{\text{pop}}\}$ and $\{\mathbf{z}_i\}_{1\leq i\leq p}$. This grouping being

given by the hierarchical structure of the model. Note that the latent variables also could have been grouped as follows: $\{\mathbf{p}_0, t_0, \mathbf{v}_0\}$, $\{(\beta_{l,k})_{l,k}\}$ and $\{\mathbf{z}_i\}_{1\leq i\leq p}$. In the case of the progression models, the delay variables $(\delta_k)_{1\leq k\leq N-1}$ were grouped with $\mathbf{z}_{\text{pop}}$, although they could also be considered as a block in itself.

For each block, the proposal in the Metropolis-Hastings step is chosen to be a multivariate Gaussian distribution centred at the current state of the block. Each variance-covariance matrix of a proposal distribution is chosen to be diagonal matrix: $\mathbf{D}_{\text{pop}} = \text{Diag}\big(\zeta_{\mathbf{p}_0}^2 \text{I}_N, \zeta_{t_0}^2, \zeta_{\mathbf{v}_0}^2 \text{I}_N, \zeta_{\boldsymbol{\beta}}^2 \text{I}_{(N-1)N_s}\big)$ for the proposal distribution associated to $\mathbf{z}_{\text{pop}}$ and $\mathbf{D}_{\text{indiv}} = \text{Diag}\big(\zeta_\xi^2, \zeta_\tau^2, \zeta_s^2\big)$ for the proposal distribution associated to $\mathbf{z}_i$ $(1 \leq i \leq p)$. The variances parameters $\zeta_{\mathbf{p}_0}^2, \zeta_{t_0}^2, \zeta_{\mathbf{v}_0}^2, \zeta_{\boldsymbol{\beta}}^2$ and $\zeta_\xi^2, \zeta_\tau^2$ are adjusted by hand to ensure an average acceptance rate for each block around 23% (Roberts et al., 1997). The Block MHwG sampler is described in Algorithm 1.

Let $\boldsymbol{\theta}_{\text{hyper}} = (\sigma_{\mathbf{p}_0}^2, \sigma_{t_0}^2, \sigma_{\mathbf{v}_0}^2, \sigma_{\boldsymbol{\beta}}^2)$ denote the fixed hyperparameters which appear in the probability distribution of the latent variables in $\mathbf{z}_{\text{pop}}$. Let $i \in \{1, \dots, p\}$ and $q_{\text{pop}}\big(\cdot \mid \boldsymbol{\theta}\big)$ (respectively $q_i(\cdot \mid \boldsymbol{\theta})$ denote the density function of the joint distribution of the latent variables $\mathbf{z}_{\text{pop}}$ (respectively $\mathbf{z}_i$) as specified in the generative model (equations (3) and (4)):

$$\begin{aligned} q_{\text{pop}}(\mathbf{z}_{\text{pop}} \mid \boldsymbol{\theta}) \propto \exp\Big(-\frac{1}{2\sigma_{\mathbf{p}_0}^2}\|\mathbf{p}_0 - \overline{\mathbf{p}_0}\|^2\Big) \exp\Big(-\frac{1}{2\sigma_{t_0}^2}(t_0 - \overline{t_0})^2\Big) \\ \exp\Big(-\frac{1}{2\sigma_{\mathbf{v}_0}^2}\|\mathbf{v}_0 - \overline{\mathbf{v}_0}\|^2\Big) \exp\Big(-\frac{1}{2\sigma_{\boldsymbol{\beta}}^2}\|\boldsymbol{\beta} - \overline{\boldsymbol{\beta}}\|^2\Big), \end{aligned} \tag{16}$$

and

$$q_i(\mathbf{z}_i \mid \boldsymbol{\theta}) \propto \exp\Big(-\frac{1}{2\sigma_\xi^2}\xi_i^2\Big) \exp\Big(-\frac{1}{2\sigma_\tau^2}\tau_i^2\Big) \exp\Big(-\frac{1}{2}\|\mathbf{s}_i\|^2\Big), \tag{17}$$

with: $\boldsymbol{\beta} = [\beta_{l,k}]_{1\leq l\leq N_s, 1\leq k\leq N-1}$ and for all $i \in \{1, \dots, p\}$, $\mathbf{s}_i = [s_{l,i}]_{1\leq l\leq N_s}$. The probability distributions q_{pop} and q_i $(1 \leq i \leq p)$ are given up to a constant. Indeed, the normalizing constant of q_{pop} or q_i $(1 \leq i \leq p)$ depends only on the parameters $\boldsymbol{\theta}$. Therefore, these constants can be omitted for the computation of the acceptance ratio in Algorithm 1.

4.1.1 Discussion

In order to avoid tuning by hand the parameters $\zeta_{\mathbf{p}_0}^2$, $\zeta_{t_0}^2$, $\zeta_{\mathbf{v}_0}^2$ and $\zeta_{\boldsymbol{\beta}}^2$ of the proposal distribution in the Block MHwG sampler, a possible solution would consist of using an *adaptive* (Atchadé, 2006) version of the Block MHwG sampler, where the algorithm automatically adjusts these variance parameters.

4.2 Stochastic Approximation

The convergence of the MCMC-SAEM was proven in Kuhn and Lavielle (2004) (for bounded latent variables) and in Allassonnière et al. (2010) (for unbounded latent variables) for statistical models which belong to the *curved exponential family*. That is to say, models for which the log complete likelihood $q(\mathbf{y}, \mathbf{z}, \boldsymbol{\theta})$ writes: $\forall \boldsymbol{\theta} \in \Theta$, $\log q(\mathbf{y}, \mathbf{z}, \boldsymbol{\theta}) = -\Phi(\boldsymbol{\theta}) + \langle \mathbf{S}(\mathbf{y}, \mathbf{z}), \Psi(\boldsymbol{\theta}) \rangle$, where Φ, Ψ are smooth functions of the parameters, $\mathbf{S}(\mathbf{y}, \mathbf{z})$ is a measurable function of the observations and latent variables called *sufficient statistic of the model* and

$\langle \cdot, \cdot \rangle$ is an inner product on a product space. The generic spatiotemporal model belongs to the curved exponential family.

Because the generic spatiotemporal model belongs to the curved exponential family, the stochastic approximation can be done on the sufficient statistics of the model. At the kth iteration of the MCMC-SAEM, we have: $\mathbf{S}_k = \mathbf{S}_{k-1} + \varepsilon_k\big(\mathbf{S}(\mathbf{y}, \mathbf{z}^{(k)}) - \mathbf{S}_{k-1}\big)$, where $(\varepsilon_k)_{k\geq 1}$ is a sequence of positive step sizes such that $\sum_k \varepsilon_k = +\infty$ and $\sum_k \varepsilon_k^2 < +\infty$. If $\varepsilon_k = 1$, then $\mathbf{S}_k$ does not depend on $\mathbf{S}_{k-1}$. Intuitively, the sequence $(\mathbf{S}_k)_{k\geq 0}$ has "no memory" as long as $\varepsilon_k = 1$ and the MCMC-SAEM freely explores the parameters space during this period. In practice, we choose $\varepsilon_k = 1$ as long as $k \leq N_b$ and $\varepsilon_k = (k - N_b)^{-0.65}$ if $k > N_b$.

4.2.1 Maximization Step

The maximization step consists of solving the following optimization problem : $\boldsymbol{\theta}^{(k)} = \underset{\boldsymbol{\theta}\in\Theta}{\text{argmax}} \big(-\Phi(\boldsymbol{\theta}) + \langle \mathbf{S}_k, \Psi(\boldsymbol{\theta}) \rangle \big)$, where $\mathbf{S}_k$ denotes the stochastic approximation on the sufficient statistics of the model, obtained in the "stochastic approximation step" of the algorithm. For the generic spatiotemporal model, this optimization problem is solved in closed-form.

4.3 Computational Aspects

For the progression model (9), the MCMC-SAEM would have to estimate the following parameters : $\boldsymbol{\theta} = (\overline{p_0}, \overline{t_0}, \overline{v_0}, \overline{\delta}_1, \ldots, \overline{\delta}_{N-1}, \overline{\beta}_1, \ldots, \overline{\beta}_{(N-1)N_s}, \sigma_\eta, \sigma_\tau, \sigma)$. In this example, we see that the number of parameters to estimate is $6 + (N-1)(N_s+1)$. As the dimension N of the manifold $\mathbb{M}$ increases, the number of parameters increases linearly. Moreover, as N increases, the number N_s of independent sources has a greater impact on the number of parameters to estimate.

The number p of individuals also impacts on the runtime of the MCMC-SAEM. As the number p of individuals increases, the cost of a single computation of the observed likelihood increases. This step is the most expensive step of the MCMC-SAEM algorithm. The overall runtime of the MCMC-SAEM could be improved by sampling the blocks $(\mathbf{z}_i)_{1\leq i\leq p}$ (in the Block MHwG sampler) in parallel.

5. Experiments

The section on numerical experiments begins with a comparison of our implementation of the MCMC-SAEM algorithm with other state-of-the-art algorithms. It is followed by a section entitled "Validation Procedure" which presents an evaluation criteria designed to quantify how well the time reparametrizations of the generic spatiotemporal model allowed to temporally align the progression of individuals. Finally, results obtained on a synthetic dataset of Symmetric Positive Definite (SPD) matrices and on a real dataset of neuropsychological tests scores are given. These datasets are analyzed using the particular cases of the generic spatiotemporal model described in Section 3.

5.1 Convergence of the Algorithm

In this section, we aim to compare our implementation of the MCMC-SAEM with state-of-the-art algorithms. Our algorithm, implemented in MATLAB, is compared with STAN and

Table 1: Relative error on the parameters estimated with the different algorithms. First row: results obtained with our implementation of the MCMC-SAEM. Second row: results obtained with STAN. Third row: results obtained with MONOLIX.

$\lvert\widehat{p_0} - p_0^*\rvert/p_0^*$	$\lvert\widehat{t_0} - t_0^*\rvert/t_0^*$	$\lvert\widehat{v_0} - v_0^*\rvert/v_0^*$	$\lvert\widehat{\sigma_\xi} - \sigma_\xi^*\rvert/\sigma_\xi^*$	$\lvert\widehat{\sigma_\tau} - \sigma_\tau^*\rvert/\sigma_\tau^*$	$\lvert\widehat{\sigma} - \sigma^*\rvert/\sigma^*$
0.0150	0.0050	0.0176	0.0600	0.0545	0.010
0.0917	0.0191	0.1088	0.0600	0.0386	0.010
0.0417	0.0086	0.0412	0.0400	0.0286	0.008

MONOLIX. STAN is a R/C++ library which implements an adaptive Hamiltonian Monte Carlo sampler called "the No U-Turns Sampler" (NUTS, Hoffman and Gelman (2014)). MONOLIX is a software developed by Marc Lavielle and the Lixoft company. It implements the MCMC-SAEM algorithm with some technical improvements (such as a simulated annealing scheme). Note that our implementation of the MCMC-SAEM algorithm differs from MONOLIX in the sense that it can be used with any particular case of the generic spatiotemporal model presented in Section 3. In particular, our implementation can be used to analyze univariate, as well as multivariate (such as covariance matrices), longitudinal observations. In its current version (2016R1), MONOLIX does not allow observations to be passed as matrices or vectors, which is not convenient for the analysis multivariate longitudinal observations.

In order to compare these algorithms, we consider a synthetic longitudinal dataset of observations in $]0, 1[$. The open set $]0, 1[$ is equipped with the Riemannian metric defined in Section 3.2, which generates logistic shaped geodesics. This dataset is generated for $p = 250$ individuals, with an average of 5 time points per individual. Each algorithm is run with the same initialization and the logistic curves model (7) was used to generate the data.

The experimental results given in Table 1 consist in relative errors on the parameters estimated with the different algorithms. The runtime and number of iteration needed for each method to converge are reported in Table 2. The results presented in the first table show that all the different methods succeeded in estimating the parameters which were used to generate the data. The corresponding number of iterations show that STAN is, by far, the most computationally intensive method. Even though our implementation of the MCMC-SAEM requires more iterations to converge than MONOLIX, the overall runtime is similar. The fact that MONOLIX requires fewer iterations to converge can be explained by the fact that the MCMC-SAEM is coupled with a simulated annealing procedure Lavielle and Mentré (2007), which allows for a better and faster exploration of the parameters space.

The results presented in these tables show that the performance of our implementation of the MCMC-SAEM is similar to the one of state-of-the-art methods. Still, the number of iterations needed to converge could be further reduced, for example, by combining several MCMC samplers in the sampling step of the MCMC-SAEM.

Table 2: Number of iterations and runtime corresponding to the experimental results given in Table 1. First row: results obtained with our implementation of the MCMC-SAEM. Second row: results obtained with STAN. Third row: results obtained with MONOLIX.

Iterations	Runtime (for 1000 iterations)	Overall runtime
3500	30 s	90 s
15000	15 min	$\simeq 3,75$ h
400	110 s	45 s

5.2 Validation Procedure

The idea is to assess how well the estimated individual time reparametrizations put in correspondence the progression of the individuals. To this end, we will use additional information which is not used in the model: the time at which a particular event occurs in the life of an individual. For example, the event could be the time at which a disease is diagnosed, or at which a metabolic change occurred for instance. Such an event occurs at a different time point (or age) for each individual.

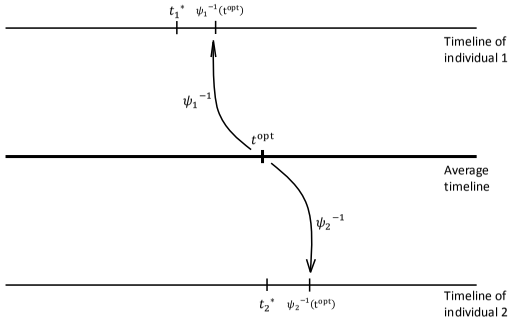


Figure 5: The average time of event t^{opt} is mapped to the individual timelines using ψ_i^{-1}.

The individual time reparametrization ψ_i are supposed to put into correspondence the time at which similar spatiotemporal patterns are found in the individual data. To assess how well this is achieved, we will test whether the time at which a particular event occur in the life of the individuals are mapped to the same time-point in the average trajectory of the model.

For the ith individual, ψ_i maps the timeline of this individual to the "average timeline", namely the one of the average trajectory. Let t_i^* be the time point at which the event occurs in the timeline of the ith individual. We estimate the time-point t^{opt} that corresponds to

the occurrence of the event in the average trajectory $\boldsymbol{\gamma}_0$ by minimizing the sum of errors $E(t) = \sum_i |t_i^* - \psi_i^{-1}(t)|$. Note that t^{opt} can be interpreted as a median of the normalized ages $(\psi_i(t_i^*))_i$, and could therefore not be unique. Then we map t^{opt} back to the individual timelines by using the mappings ψ_i^{-1}, as illustrated in Fig. 5. The value $\psi_i^{-1}(t^{\text{opt}})$ may be thought of as a prediction of the model of the time-point (or age) at which the event occurred for the i*th* individual. Without errors, this time-point would be exactly t_i^*. In practice, the difference $|t_i^* - \psi_i^{-1}(t^{\text{opt}})|$ allows quantifying how well the events the timeline of the ith individual and the average timeline have been put into correspondence.

In the following experiments, the median t^{opt} of $\left(\psi_i(t_i^*)\right)_{1\leq i\leq p}$ is computed unambiguously. To assess how well the individual trajectories and the average trajectory are put into correspondence, we plot an histogram of the errors $\left(|t_i^* - \psi_i^{-1}(t^{\text{opt}})|\right)_{1\leq i\leq p}$.

5.3 Tensors : a Synthetic Dataset

We start by presenting the synthetic dataset of SPD matrices used for this experiment and then present the results obtained with the particular case of the generic spatiotemporal model for SPD matrices (see Section 3.4).

5.3.1 Data

We consider a synthetic dataset, in which we simulate repeated observations of a a symmetric definite positive matrix (also called a diffusion tensor in medical imaging) for one hundred individuals. The observations were not generated from the model. The observations were obtained instead by prescribing an ad hoc hierarchical model on the eigenvalues of the diffusion tensors. At the level of the population, the eigenvalues of the diffusion tensors follow a decreasing piecewise linear evolution with a change point at 50 years old. Observations for a given individual were simulated by randomly shifting the change point (time at which a change occurs in the speed at which eigenvalues decrease) and randomly increasing or decreasing the slopes of each eigenvalue (see Fig.6, left). In this synthetic dataset, the individuals have, on average, five time points.

5.3.2 Results

The results presented below were obtained with $N_s = 1$ source. A greater number of independent sources would have been possible but many more iterations would have been necessary for the MCMC-SAEM to converge. The Bayesian tensor model with the MCMC-SAEM allows estimating an average trajectory of progression in the space SDP(3). This average trajectory is the geodesic which goes through the point $\overline{\mathbf{P}_0}$, at time $\overline{t_0}$, with velocity $\overline{\mathbf{V}_0}$, given by :

$$\overline{\mathbf{P}_0} = \begin{pmatrix} 11.30 & 0.96 & 0.68 \\ 0.96 & 9.53 & 1.21 \\ 0.68 & 1.21 & 10.19 \end{pmatrix}, \quad \overline{t_0} = 53.83 \text{ years},$$

and

$$\overline{\mathbf{V}_0} = \begin{pmatrix} -0.99 & -0.17 & -0.20 \\ -0.17 & -0.75 & -0.27 \\ -0.20 & -0.27 & -0.85 \end{pmatrix} \text{ unit per year.}$$

The evolution of the eigenvalues of the average trajectory, plotted in Fig. 6, is similar to the model used to generate the observations. However, the MCMC-SAEM tends to underestimate the first eigenvalue and overestimate the third eigenvalue. The variability in speed and delay of progression is captured by the estimated parameters $\sigma_\eta = 0.07$ and $\sigma_\tau = 0.5$ year. Fig. 6 (left) shows that the eigenvalues of each individual decrease at a similar pace before and after the change point. This may explain why the model captured small variations in speed of progression. The standard deviation σ_τ on the parameter t_0 is much smaller. The individual acceleration factor, time shift and space shift allow fitting the average trajectory to the observations of an individual. As shown in Fig. 6 (right), the estimated individual trajectory is well-adjusted to the observations of the individual.

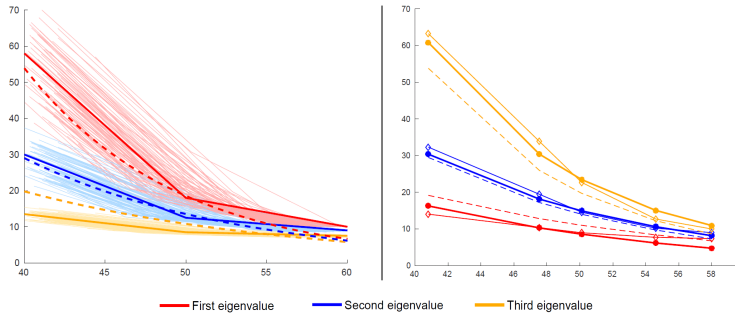


Figure 6: Left : In solid bold line, the average model of eigenvalues evolution for the synthetic dataset of tensors. In solid lines, the evolution of the eigenvalues for all the individuals in the dataset. In dotted line, the evolution of the eigenvalues of the average trajectory, given by the MCMC-SAEM. Right : the evolution of the eigenvalues of an individual. In dotted line, the eigenvalues of the average trajectory estimated by the MCMC-SAEM. With square markers, the eigenvalues of the observations for this individual. With round markers, the eigenvalues of the estimated individual trajectory.

The eigenvalues of the average estimated trajectory are smooth functions of time. Therefore, it would not have been possible to obtain a piecewise-linear progression of the eigenvalues for the average trajectory. Nevertheless, we can still validate the ability of the tensor model to put into correspondence the dynamic of each individual following Sec. 5.2 by using the individual change point t_i^*. For this dataset, the sum of errors $\sum_i |t_i^* - \psi_i^{-1}(t)|$ has a unique minimum at $t^{\text{opt}} = 49.73$ years. This minimum t^{opt} is close to 50 years, the time at which the change point occurs in the average model used to generate the data. Fig. 7 shows that the model made an error of less than 2 years for almost 60% of the population by predicting the individual change point with $\psi_i^{-1}(t^{\text{opt}})$, and less than 4 years for 90% of the population. The change point was generated using a Gaussian distribution centred at

50 years with a standard deviation of 2 years. Therefore, the error is of the same order as the standard deviation of the change point.

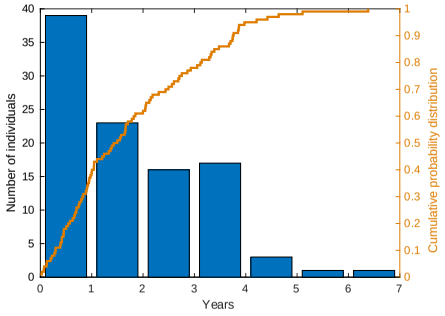


Figure 7: Histogram of $\left(|t_i^* - \psi_i^{-1}(t^{\text{opt}})|\right)_{1 \leq i \leq 100}$ superimposed with the cumulative distribution of this error. Here, t_i^* represents the age of the change point for the ith individual.

5.4 Neuropsychological Tests

This section starts by presenting the real dataset of neuropsychological tests scores from the ADNI database. Then, we present results obtained using the progression model (see Eq. (9)) with logistic curves.

5.4.1 Data

The dataset consists of scores to the modified "ADAS-Cog" test (Mohs et al., 1997) obtained from the ADNI1, ADNIGO and ADNI2 cohorts of the Alzheimer's Disease Neuroimaging Initiative. The 13 items were grouped into 4 categories according to the cognitive function they assess: memory, language, concentration and praxis. For each cognitive function, the scores were added and normalized by the maximum possible value therefore producing measurements in $]0, 1[$. As a consequence, each observation is a point on the manifold $\mathbb{M} =]0, 1[^4$ (note that results without item pooling are presented in Schiratti et al. (2015a)). We use 248 individuals who were included in the study as mild cognitive impaired (MCI) subjects and later converted to Alzheimer's disease (AD). Each individual was observed on average 6 times.

5.4.2 Results

This data set was analyzed using the *progression model* given in Eq. (9) with logistic curves. The number of independent components could be either $1, 2$ or 3, as the manifold is of dimension 4. The model with one independent component estimated a residual noise variance

$\sigma^2 = 0.012$ and explained 79% of the total variance. The model with two (resp. three) independent components estimated a noise variance $\sigma^2 = 0.008$ (resp. $\sigma^2 = 0.0084$) and explained 84% (resp. 85%) of the total variance. Because the results obtained with three independent components are similar to the results obtained with two independent components, we choose, for the sake of clarity, to report the results obtained with two components ($N_s = 2$).

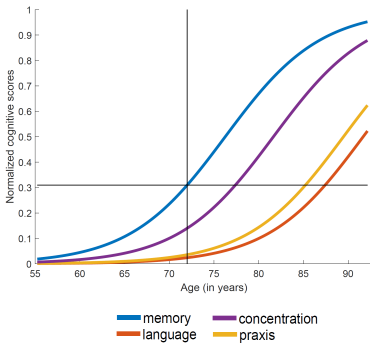


Figure 8: The estimated average trajectory. The estimated parameters p_0 (resp. t_0) are represented by an horizontal (resp. vertical line) at $p_0 = 0.3$ (resp. $t_0 = 72$ years).

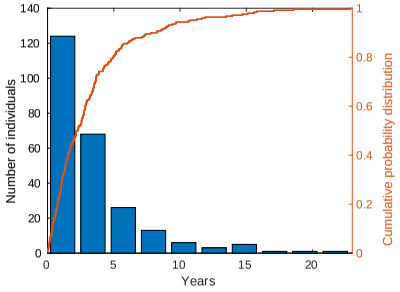


Figure 9: Histogram of $\left(|t_i^{\text{conv}} - \psi_i^{-1}(t^{\text{opt}})|\right)_{1 \leq i \leq 248}$ superimposed with the cumulative distribution of this error.

The average trajectory estimated by the progression model, plotted in Fig. 8, is characterized by the fixed effects $p_0 = 0.3$, $t_0 = 72$ years, $v_0 = 0.04$ unit per year and $\boldsymbol{\delta} = [0; -15; -13; -5]$ years. The first biomarker (memory) reaches the value $p_0 = 0.3$ at 72 years on average, the second one (concentration), at $t_0 + 5 = 77$ years, followed by praxis and language. The fixed effects provide an ordering of the biomarkers and the relative delay between them. The random effects characterize the spatiotemporal variability of the average trajectory among the population. The estimated standard deviation of the time-shift is $\sigma_\tau = 7.5$ years, meaning that age of disease onset ranges between 72 ± 7.5 years for 95% of the individuals. A positive (resp. negative) time-shift means that the individual is evolving behind (resp. ahead) the average trajectory. The estimated standard deviation of the acceleration factors is $\sigma_\eta = 0.9$. As a consequence, most of the individuals are progressing between $e^{\sigma_\eta} \simeq 2.4$ times faster or $e^{-\sigma_\eta} \simeq 0.4$ times slower than the average trajectory (see Fig. 11, first row). Estimates of the individual time-shits and log-acceleration factors are plotted in Fig. 10. This figure shows a clear correspondence between the time shifts and

the estimated age at which individuals were diagnosed with the disease. This fact shows that the normalized age $\psi_i(t)$ is a better temporal marker of disease progression than age. This is confirmed by our validation procedure (Fig. 9), which shows an error in the prediction of age at diagnosis of less than 2.5 years in 50% of the cases.

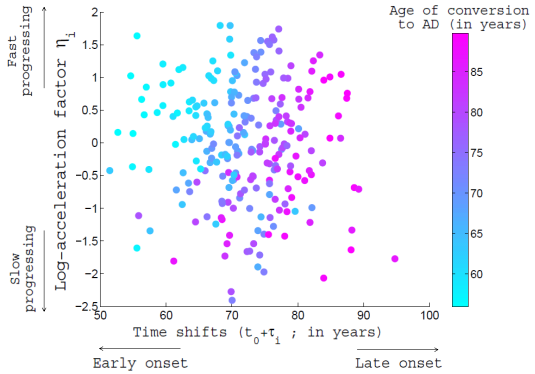


Figure 10: Plot of $t_0 + \tau_i$ with respect to the log-acceleration factor η_i. Each point is colored with respect to the estimated age of conversion to AD.

In this multivariate setting, random effects also include space shifts, which are a combination of two independent components denoted here $\mathbf{c}_1(\mathbf{A})$ and $\mathbf{c}_2(\mathbf{A})$. As shown in lemma 2, these space-shifts perturb the relative delay and the ordering in the progression of biomarkers. Fig. 11 shows that individuals with a space shift of the form $\mathbf{w}_i = \sigma_{s_i}\mathbf{c}_1(\mathbf{A})$ have memory and concentration impaired at nearly the same time, while the language and praxis remains nearly constant. In the opposite direction, impairment in language and praxis nearly coincide for individuals with a space-shift of the form $\mathbf{w}_i = -\sigma_{s_i}\mathbf{c}_1(\mathbf{A})$. The second independent component almost does not change memory and concentration but changes the delay and the ordering between language and praxis. These results show that the biomarkers tend to evolve in pairs : memory & concentration, language & praxis. Space shifts capture here the variability in the profile of cognitive decline at the individual level during the onset of the disease.

5.5 Body Mass Index in Adolescent Girls

We analyzed a longitudinal dataset of body fat percentages from 162 adolescent girls. This dataset is taken from the MIT Growth and Development Study Bandini et al. (2002); Phillips et al. (2003). The data is analyzed using the univariate logistic model (see supplementary material for the details and results). The analysis of this data in
Fitzmaurice et al. (2012) requires the use of the time at menarche to temporally align the data across individuals before the statistical analysis. By contrast, our approach is able to include such an alignment as a random effect of the model. The estimated parameters

produced an error of less than one year for 50% of the individuals in the alignment of the age at which the menarche occurred.

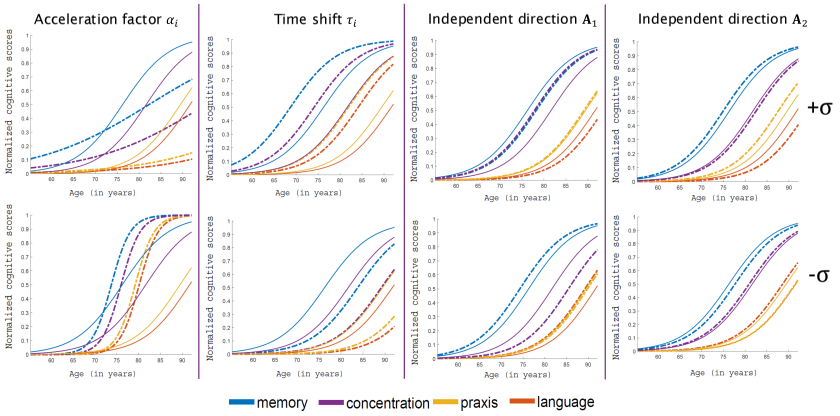


Figure 11: Variability of the average trajectory γ_δ in terms of space shift, time shift and acceleration factor. The solid lines represent the average trajectory, while the dotted lines represent the variability of this average trajectory among the population.

6. Conclusion

We proposed a mixed-effects model for the spatiotemporal analysis of manifold-valued measurements. This generic model can be used to automatically learn the temporal progression of a biological phenomenon from repeated observations of several samples. The model accounts for the fact that each sample has a different appearance, has different trajectory of changes and different pace of changes. This is enabled by the introduction of a hierarchical statistical model whose fixed effects define a group-average trajectory in the space of measurements and random effects account for the spatiotemporal variability of the trajectories of changes at the individual level.

Building the model in the framework of Riemannian geometry allowed us to identify the key orthogonality condition to uniquely decompose temporal and spatial variability across trajectories of changes. Our particular use of parallel transport ensures the invariance of the form of the distribution of spatial random effects in time. Finally, it allows the instantiation of the model for a large variety of data types, as shown by our experiments with univariate and multivariate bounded measurements, as well as symmetric definite positive matrices. This framework is also well suited for analysing data with smooth constraints or highly structured data such as images, networks(Koval et al. (2017)), or shapes for instance.

We rely on the stochastic approximation of the Expectation-Maximization algorithm to estimate model parameters. This algorithm is proven to converge asymptotically, and our experiments confirmed this convergence in several practical situations. This algorithm is very generic and allows a modular implementation of the method which eases its instantiation for any kind of manifold-valued observations. Note, nevertheless, that specific optimizations were needed to match, and even outperform state-of-the-art implementations for real-valued measurements.

The model is particularly useful to analyze biological phenomenon, for which there is no obvious way to temporally align individual data time series. Temporal realignment introduced as random-effect allowed us to automatically predict the age at which patients were diagnosed with Alzheimer's disease or at which menarche occurred in adolescent girls. Although the model assumed a monotonic progression of body fat with age, this hypothesis could be relaxed by replacing the distribution of the acceleration factors and space shifts with a mixture of Gaussian distributions. This could also be achieved by considering non-affine time reparametrizations of the average trajectory. Improvements in prediction and goodness of fit might result from extensions of the model, for instance by introducing mixture models to identify population clusters or by adding a drift in the parallel transport so that individual trajectories do not always remains parallel to the average trajectory.

Acknowledgments

This work was partially supported by the FMJH (Governement Program: ANR-10-CAMP-0151-02). The research leading to these results has received funding from the program Investissements davenir ANR-10-IAIHU-06.

Appendix A. Proof of Lemma 2 and Lemma 3

In this appendix, we will first give a proof of Lemma 2 and we shall then give a proof of Lemma 3.

A.1 Proof of Lemma 2

The definitions and properties given above are useful to prove the result given in 2. Before proving this result, we recall its context. We consider a product manifold $\mathbb{M} = M^N$ where $M \subset \mathbb{R}$ is an open interval. M is equipped with a Riemannian metric g and is geodesically complete. The product manifold $\mathbb{M}$ is equipped with the product metric.

Proof Let $t_0 \in \mathbb{R}$, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N)$ be a geodesic of $\mathbb{M}$ and $\mathbf{w} = (w_1, \dots, w_N) \in \mathrm{T}_{\boldsymbol{\gamma}(t_0)}\mathbb{M}$. Using the previous proposition on product manifolds, the computation of $\eta^{\mathbf{w}}(\boldsymbol{\gamma}, t)$ boils down to the computation of $\mathrm{Exp}_{\gamma_i(t)}\big(\mathrm{P}_{\gamma_i,t_0,t}(w_i)\big)$. This term is computed in three steps.

Let $i \in \{1, \dots, N\}$. The parallel transport $\mathrm{P}_{\gamma_i,t_0,t}(w_i)$ is computed as follows. First, note that M is an open interval of $\mathbb{R}$. Therefore, for all $p \in M$, $\mathrm{T}_pM \simeq \mathbb{R}$. The Riemannian metric g of M is necessarily of the form $p \in M \mapsto g_p$ with : $\forall (u, v) \in \mathrm{T}_pM$, $g_p(u, v) = uvf(p)$ where $f : M \to]0, +\infty[$ is a smooth function.

It follows from the definition of parallel transport along the curve $t \mapsto \gamma_i$ that : $\forall t$, $\mathrm{P}_{\gamma_i,t_0,t}(w_i) \in \mathrm{T}_{\gamma_i(t)}M$. Since, for all t, $\mathrm{T}_{\gamma_i(t)}M$ is a one-dimensional vector space, the tangent vector $\dot\gamma_i(t) \neq 0$ spans this space. As a consequence, there exist a smooth function $\xi_i : \mathbb{R} \to \mathbb{R}$ such that : $\forall t \in \mathbb{R}$, $\mathrm{P}_{\gamma_i,t_0,t}(w_i) = \xi_i(t)\dot\gamma_i(t)$. Because the parallel transport is an isometry and because γ_i is a geodesic, we have :

$$\forall t \in \mathbb{R},\ g_{\gamma_i(t)}\big(\mathrm{P}_{\gamma_i,t_0,t}(w_i), \dot\gamma_i(t)\big) = g_{\gamma_i(t_0)}(w_i, \dot\gamma_i(t_0)). \tag{18}$$

The bilinearity of $g_{\gamma_i(t)}$ gives :

$$\begin{aligned} g_{\gamma_i(t)}\big(\mathrm{P}_{\gamma_i,t_0,t}(w_i), \dot\gamma_i(t)\big) &= g_{\gamma_i(t)}\big(\xi_i(t)\dot\gamma_i(t), \dot\gamma_i(t)\big) \\ &= \xi_i(t) g_{\gamma_i(t)}\big(\dot\gamma_i(t), \dot\gamma_i(t)\big). \end{aligned} \tag{19}$$

Using that $\dot\gamma_i$ is parallel along γ_i, we have :

$$\forall t \in \mathbb{R},\ g_{\gamma_i(t)}\big(\dot\gamma_i(t), \dot\gamma_i(t)\big) = g_{\gamma_i(t_0)}\big(\dot\gamma_i(t_0), \dot\gamma_i(t_0)\big). \tag{20}$$

As a consequence, (18), (19) and (20) give :

$$\forall t \in \mathbb{R},\ \xi_i(t) g_{\gamma_i(t_0)}\big(\dot\gamma_i(t_0), \dot\gamma_i(t_0)\big) = g_{\gamma_i(t_0)}\big(w_i, \dot\gamma_i(t_0)\big). \tag{21}$$

Using the form of the metric on M, (21) writes :

$$\forall t \in \mathbb{R},\ \xi_i(t)\big(\dot\gamma_i(t_0)\big)^2 f\big(\gamma_i(t_0)\big) = w_i \dot\gamma_i(t_0) f\big(\gamma_i(t_0)\big). \tag{22}$$

This last equation gives : $\forall t$, $\xi_i(t) = w_i/\dot\gamma_i(t_0)$. Finally,

$$\forall i \in \{1, \dots, N\},\ \forall t \in \mathbb{R},\ \mathrm{P}_{\gamma_i,t_0,t}(w_i) = \frac{w_i}{\dot\gamma_i(t_0)} \dot\gamma_i(t). \tag{23}$$

The last step consist in computing $\mathrm{Exp}_{\gamma_i(t)}\big(\mathrm{P}_{\gamma_i,t_0,t}(w_i)\big)$ with $t \in \mathbb{R}$ fixed. This is done by introducing the curves

$$c : s \in [0, 1] \mapsto \mathrm{Exp}_{\gamma_i(t)}\big(s\mathrm{P}_{\gamma_i,t_0,t}(w_i)\big),$$

and

$$\tilde c : s \in [0, 1] \mapsto \gamma_i\Big(t + s\frac{w_i}{\dot\gamma_i(t_0)}\Big).$$

Both curves c and $\tilde c$ are geodesics of M which satisfy to : $c(0) = \tilde c(0) = \gamma_i(t)$ and $\dot c(0) = \dot{\tilde c}(0) = \frac{w_i}{\dot\gamma_i(t_0)}\dot\gamma_i(t)$. By unicity, the two curves are equal. As a consequence, for all $i \in \{1, \dots, N\}$ and all $t \in \mathbb{R}$,

$$\mathrm{Exp}_{\gamma_i(t)}\big(\mathrm{P}_{\gamma_i,t_0,t}(w_i)\big) = \gamma_i\Big(t + \frac{w_i}{\dot\gamma_i(t_0)}\Big). \tag{24}$$

This last equation completes the proof of the lemma. ■

A.2 Proof of Lemma 3

First, some notations need to be introduced. Let $m = n(n+1)/2$ denote the dimension of the linear space SYM(n), $(\mathbf{E}_i)_{1\leq i\leq m}$ be the canonical basis of SYM(n) and $(\mathbf{E}_i^*)_{1\leq i\leq m}$ its *dual basis*. The matrices are indexed by a single index, which corresponds to an enumeration of the pairs of integers $\{(k, l),\ 1 \leq k, l \leq n,\ k \leq l\}$. A matrix $\mathbf{V}$ in SYM(n) will be identified to the vector $(v_1, \dots, v_m)$ of its coefficients from the upper triangular part. Using the expression of the Christoffel symbols in terms of the canonical basis of SYM(n) and its dual basis, Lenglet and collaborators prove that if $t \mapsto \boldsymbol{\Sigma}(t) = (\sigma_1(t), \dots, \sigma_m(t))$ is a smooth curve in SPD(n) and $t \mapsto \mathbf{V}(t) = (v_1(t), \dots, v_m(t))$ a vector field along Σ, the *covariant derivative* of $\mathbf{V}$ along $\boldsymbol{\Sigma}$ is given by the expression:

$$D\mathbf{V}/dt = \sum_i^m \frac{dv_i}{dt}(t)\mathbf{E}_i + \sum_{i,j=1}^m v_i(t)\frac{d\sigma_j}{dt}(t)\nabla_{\mathbf{E}_i}\mathbf{E}_j. \tag{25}$$

Taking the Frobenius inner product of the previous equality (on both sides) with $\mathbf{E}_k^*$ ($1 \leq k \leq m$), together with the expression of the Christoffel symbols (see Lenglet et al. (2006), Eq.(3) and Eq.(4)), one gets that the vector field $\mathbf{V}$ is parallel along the curve $\boldsymbol{\Sigma}$ if and only if:

$$\frac{d\mathbf{V}}{dt}(t) - \frac{1}{2}\mathbf{V}(t)\boldsymbol{\Sigma}(t)^{-1}\frac{d\boldsymbol{\Sigma}}{dt}(t) - \frac{1}{2}\frac{d\boldsymbol{\Sigma}(t)}{dt}(t)\boldsymbol{\Sigma}(t)^{-1}\mathbf{V}(t) = 0. \tag{26}$$

We can now give the proof of Lemma 3.

Proof With the expression of the geodesic $\boldsymbol{\gamma}_0$, given in Section 3.4, one can easily see that:

$$\forall t \in \mathbb{R},\ \frac{d\boldsymbol{\gamma}_0(t)}{dt}\boldsymbol{\gamma}_0^{-1}(t) = \mathbf{V}_0\mathbf{P}_0^{-1}. \tag{27}$$

Let $t \mapsto \mathbf{V}(t)$ be a vector field parallel along $\boldsymbol{\gamma}_0$. Eq. (26) is equivalent to:

$$\frac{d\mathbf{V}(t)}{dt} = \frac{1}{2}\mathbf{V}(t)\mathbf{P}_0^{-1}\mathbf{V}_0 - \frac{1}{2}\mathbf{V}_0\mathbf{P}_0^{-1}\mathbf{V}(t). \tag{28}$$

Note that Eq. (28) is a *differential Lyapunov equation*. It can be solved by considering the matrix-valued function $t \mapsto \exp(-t\mathbf{M}^\top)\mathbf{R}(t)\exp(-t\mathbf{M})$, with $\mathbf{M} = -(1/2)\mathbf{P}_0^{-1}\mathbf{V}_0$ and $\mathbf{R}$, any differentiable matrix-valued function. Given that $\mathbf{V}(t_0) = \mathbf{W}$, one has that the parallel transport in $\mathrm{SPD}(n)$ is given by:

$$\mathbf{V}(t) = \exp\Big(\frac{t-t_0}{2}\mathbf{V}_0\mathbf{P}_0^{-1}\Big)\mathbf{W}\exp\Big(\frac{t-t_0}{2}\mathbf{P}_0^{-1}\mathbf{V}_0\Big). \tag{29}$$

■

References

Stéphanie Allassonnière, Estelle Kuhn, and Alain Trouvé. Construction of bayesian deformable models via a stochastic approximation algorithm: a convergence study. *Bernoulli*, 16(3):641–678, 2010.

V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic resonance in medicine*, 56(2):411–421, 2006.

Y. F. Atchadé. An adaptive version for the metropolis adjusted langevin algorithm with a truncated drift. *Methodology and Computing in applied Probability*, 8(2):235–254, 2006.

L.G. Bandini, A. Must, J.L. Spadano, and W.H. Dietz. Relation of body composition, parental overweight, pubertal stage, and race-ethnicity to energy expenditure among premenarcheal girls. *The American Journal Of Clinical Nutrition*, 76(5):1040–1047, 2002.

T. F. Coleman and D. C. Sorensen. A note on the computation of an orthonormal basis for the null space of a matrix. *Mathematical Programming*, 29(2):234–242, 1984.

I. Delor, JE. Charoin, R. Gieschke, S. Retout, and P. Jacqmin. Modeling alzheimer's disease progression using disease onset time and disease trajectory concepts applied to cdr-sob scores from adni. *CPT: pharmacometrics & systems pharmacology*, 2(10):e78, 2013.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

M. P. Do Carmo Valero. *Riemannian geometry*. Birkhäuser, 1992.

S. Durrleman, X. Pennec, A. Trouvé, G. Gerig, and N. Ayache. Spatiotemporal atlas estimation for developmental delay detection in longitudinal datasets. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 297–304. Springer Berlin Heidelberg, 2009.

Stanley Durrleman, Xavier Pennec, Alain Trouvé, José Braga, Guido Gerig, and Nicholas Ayache. Toward a comprehensive framework for the spatiotemporal statistical analysis of longitudinal shape data. *International Journal of Computer Vision*, 103(1):22–59, 2013.

C. Eisenhart. The assumptions underlying the analysis of variance. *Biometrics*, 3(1):1–21, 1947.

G.M. Fitzmaurice, N.M. Laird, and J.H. Ware. *Applied longitudinal analysis*, volume 998. John Wiley & Sons, 2012.

T. Fletcher. Geodesic regression on riemannian manifolds. In *Proceedings of the Third International Workshop on Mathematical Foundations of Computational Anatomy-Geometrical and Statistical Methods for Modelling Biological Shape Variability*, pages 75–86, 2011.

Hubert M Fonteijn, Marc Modat, Matthew J Clarkson, Josephine Barnes, Manja Lehmann, Nicola Z Hobbs, Rachael I Scahill, Sarah J Tabrizi, Sebastien Ourselin, Nick C Fox, et al.

An event-based model for disease progression and its application in familial alzheimer's disease and huntington's disease. *NeuroImage*, 60(3):1880–1889, 2012.

M. D. Hoffman and A. Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Yi Hong, Nikhil Singh, Roland Kwitt, and Marc Niethammer. Time-warped geodesic regression. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 105–112. Springer, 2014.

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*, volume 46. John Wiley & Sons, 2004.

B. M. Jedynak, A. Lang, B. Liu, E. Katz, Y. Zhang, B. T. Wyman, D. Raunig, C. P. Jedynak, B. Caffo, J. L. Prince, et al. A computational neurodegenerative disease progression score: method and results with the alzheimer's disease neuroimaging initiative cohort. *Neuroimage*, 63(3):1478–1486, 2012.

I. Koval, J.-B. Schiratti, A. Routier, M. Bacci, O. Colliot, S. Allassonnière, and S. Durrleman. Statistical learning of spatiotemporal patterns from longitudinal manifold-valued networks. In *Medical Image Computing and Computer-Assisted Intervention*, 2017.

E. Kuhn and M. Lavielle. Coupling a stochastic approximation version of em with an mcmc procedure. *ESAIM: Probability and Statistics*, 8:115–131, 2004.

Nan M Laird and James H Ware. Random-effects models for longitudinal data. *Biometrics*, pages 963–974, 1982.

M. Lavielle and F. Mentré. Estimation of population pharmacokinetic parameters of saquinavir in hiv patients with the monolix software. *Journal of pharmacokinetics and pharmacodynamics*, 34(2):229–249, 2007.

C. Lenglet, M. Rousson, R. Deriche, and O. Faugeras. Statistics on the manifold of multivariate normal distributions: Theory and application to diffusion tensor mri processing. *Journal of Mathematical Imaging and Vision*, 25(3):423–444, 2006.

M. Lorenzi, X. Pennec, G. B Frisoni, and N. Ayache. Disentangling normal aging from alzheimer's disease in structural magnetic resonance images. *Neurobiology of aging*, 36: S42–S52, 2015.

M. Moakher and M. Zéraï. The riemannian geometry of the space of positive-definite matrices and its application to the regularization of positive-definite matrix-valued data. *Journal of Mathematical Imaging and Vision*, 40(2):171–187, 2011.

R. C. Mohs, D. Knopman, R. C Petersen, S. H. Ferris, C. Ernesto, M. Grundman, M. Sano, L. Bieliauskas, D. Geldmacher, C. Clark, et al. Development of cognitive instruments for use in clinical trials of antidementia drugs: additions to the alzheimer's disease assessment scale that broaden its scope. *Alzheimer Disease & Associated Disorders*, 11:13–21, 1997.

X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.

P. Petersen. *Riemannian geometry*, volume 171. Springer, 2006.

S.M. Phillips, L.G. Bandini, D.V. Compton, E.N. Naumova, and A. Must. A longitudinal comparison of body composition by total body water and bioelectrical impedance in adolescent girls. *The Journal of nutrition*, 133(5):1419–1425, 2003.

G. O. Roberts, A. Gelman, W. R. Gilks, et al. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997.

J.-B. Schiratti, S. Allassonnière, O. Colliot, and S. Durrleman. Mixed-effects model for the spatiotemporal analysis of longitudinal manifold-valued data. In *5th MICCAI Workshop on Mathematical Foundations of Computational Anatomy*, 2015a.

J.-B. Schiratti, S. Allassonnière, O. Colliot, and S. Durrleman. Learning spatiotemporal trajectories from manifold-valued longitudinal data. In *Advances in Neural Information Processing Systems*, pages 2404–2412, 2015b.

J.-B. Schiratti, S. Allassonnière, A. Routier, O. Colliot, S. Durrleman, and the ADNI. A mixed-effects model with time reparametrization for longitudinal univariate manifold-valued data. In *International Conference on Information Processing in Medical Imaging*, pages 564–575. Springer, 2015c.

N. Singh, J. Hinkle, S. Joshi, and P. T. Fletcher. An efficient parallel algorithm for hierarchical geodesic models in diffeomorphisms. In *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pages 341–344. IEEE, 2014.

Nikhil Singh, Jacob Hinkle, Sarang Joshi, and P Thomas Fletcher. A hierarchical geodesic model for diffeomorphic longitudinal shape analysis. In *Information Processing in Medical Imaging*, pages 560–571. Springer, 2013.

L. T. Skovgaard. A riemannian geometry of the multivariate normal model. *Scandinavian Journal of Statistics*, pages 211–223, 1984.

J. Su, S. Kurtek, E. Klassen, A. Srivastava, et al. Statistical analysis of trajectories on riemannian manifolds: Bird migration, hurricane tracking and video surveillance. *The Annals of Applied Statistics*, 8(1):530–552, 2014.

G. Verbeke and G. Molenberghs. *Linear mixed models for longitudinal data*. Springer Science & Business Media, 2009.

Eric Yang, Michael Farnum, Victor Lobanov, Tim Schultz, Nandini Raghavan, Mahesh N Samtani, Gerald Novak, Vaibhav Narayan, and Allitia DiBernardo. Quantifying the pathophysiological timeline of alzheimer's disease. *Journal of Alzheimer's Disease*, 26(4):745–753, 2011.

A. L. Young, N. P. Oxtoby, J. Huang, R. V. Marinescu, P. Daga, D. M. Cash, N. C. Fox, S. Ourselin, J. M. Schott, D. C. Alexander, et al. Multiple orderings of events in disease progression. In *International Conference on Information Processing in Medical Imaging*, pages 711–722. Springer, 2015.

Journal of Machine Learning Research 18 (2017) 1-35 Submitted 4/17; Revised 10/17; Published 12/17

Stochastic Gradient Descent as Approximate Bayesian Inference

Stephan Mandt STEPHAN.MANDT@GMAIL.COM
Data Science Institute
Department of Computer Science
Columbia University
New York, NY 10025, USA

Matthew D. Hoffman MATHOFFM@ADOBE.COM
Adobe Research
Adobe Systems Incorporated
601 Townsend Street
San Francisco, CA 94103, USA

David M. Blei DAVID.BLEI@COLUMBIA.EDU
Department of Statistics
Department of Computer Science
Columbia University
New York, NY 10025, USA

Editor: Manfred Opper

Abstract

Stochastic Gradient Descent with a constant learning rate (constant SGD) simulates a Markov chain with a stationary distribution. With this perspective, we derive several new results. (1) We show that constant SGD can be used as an approximate Bayesian posterior inference algorithm. Specifically, we show how to adjust the tuning parameters of constant SGD to best match the stationary distribution to a posterior, minimizing the Kullback-Leibler divergence between these two distributions. (2) We demonstrate that constant SGD gives rise to a new variational EM algorithm that optimizes hyperparameters in complex probabilistic models. (3) We also show how to tune SGD with momentum for approximate sampling. (4) We analyze stochastic-gradient MCMC algorithms. For Stochastic-Gradient Langevin Dynamics and Stochastic-Gradient Fisher Scoring, we quantify the approximation errors due to finite learning rates. Finally (5), we use the stochastic process perspective to give a short proof of why Polyak averaging is optimal. Based on this idea, we propose a scalable approximate MCMC algorithm, the Averaged Stochastic Gradient Sampler.

Keywords: approximate Bayesian inference, variational inference, stochastic optimization, stochastic gradient MCMC, stochastic differential equations

1. Introduction

Stochastic gradient descent (SGD) has become crucial to modern machine learning. SGD optimizes a function by following noisy gradients with a decreasing step size. The classical result of Robbins and Monro (1951) is that this procedure provably reaches the optimum of the function (or local optimum, when it is nonconvex) (Bouleau and Lepingle, 1994). Recent studies investigate the merits of adaptive step sizes (Duchi et al., 2011; Tieleman and Hinton, 2012), gradient or iterate averaging (Toulis et al., 2016; Défossez and Bach, 2015), and constant step-sizes (Bach and Moulines, 2013; Flammarion and Bach, 2015). Stochastic gradient descent has enabled efficient optimization with massive data, since one can often obtain noisy-but-unbiased gradients very cheaply by randomly subsampling a large dataset.

Recently, stochastic gradients (SG) have also been used in the service of scalable Bayesian Markov Chain Monte Carlo (MCMC) methods, where the goal is to generate samples from a conditional distribution of latent variables given a data set. In Bayesian inference, we assume a probabilistic model $p(\theta, \mathbf{x})$ with data $\mathbf{x}$ and hidden variables θ; our goal is to approximate the posterior

$$p(\theta \,|\, \mathbf{x}) = \exp\{\log p(\theta, \mathbf{x}) - \log p(\mathbf{x})\}. \tag{1}$$

These so-called stochastic gradient MCMC algorithms—such as SG Langevin dynamics (Welling and Teh, 2011), SG Hamiltonian Monte Carlo (Chen et al., 2014), SG thermostats (Ding et al., 2014), and SG Fisher scoring (Ahn et al., 2012)—employ stochastic gradients of $\log p(\theta, \mathbf{x})$ to improve convergence and computation of existing sampling algorithms. Also see Ma et al. (2015) for a complete classification of these algorithms.

The similarities between SGD as an optimization algorithm and stochastic gradient MCMC algorithms raise the question of how exactly these two types of algorithms relate to each other. The main questions we try to address in this paper are:

- What is the simplest modification to SGD that yields an efficient approximate Bayesian sampling algorithm?
- How can we construct other sampling algorithms based on variations of SGD such as preconditioning (Duchi et al., 2011; Tieleman and Hinton, 2012), momentum (Polyak, 1964) or Polyak averaging (Polyak and Juditsky, 1992)?

To answer these questions, we draw on the theoretical analysis tools of continuous-time stochastic differential equations (Bachelier, 1900; Gardiner et al., 1985) and variational inference (Jordan et al., 1999).

As a simple example, consider SGD with a constant learning rate (constant SGD). Constant SGD first marches toward an optimum of the objective function and then bounces around its vicinity. (In contrast, traditional SGD converges to the optimum by decreasing the learning rate.) Our analysis below rests on the idea that constant SGD is a stochastic process with a stationary distribution, one that is centered on the optimum and that has a certain covariance structure. The main idea is that we can use this stationary distribution to approximate a posterior. In contrast, stochastic gradient MCMC algorithms take precautions to sample from an asymptotically exact posterior, but at the expense of slower mixing. Our inexact approach enjoys minimal implementation effort and typically faster mixing. It is a hybrid of Monte Carlo and variational algorithms, and complements the toolbox of approximate Bayesian inference.

Here is how it works. We apply constant SGD as though we were trying to minimize the negative log-joint probability $-\log p(\theta, \mathbf{x})$ over the model parameters θ. Constant SGD has several tunable parameters: the constant learning rate, the minibatch size, and the preconditioning matrix (if any) that we apply to the gradient updates. These tuning parameters all affect the stationary distribution of constant SGD; depending on how they are set, this stationary distribution of θ will be closer to or farther from the posterior distribution $p(\theta|\mathbf{x})$. If we set these parameters appropriately, we can perform approximate Bayesian inference by simply running constant SGD.

In more detail, we make the following contributions:

©2017 Stephan Mandt, Matthew D. Hoffman, David M. Blei.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-214.html.

1. First, we develop a variational Bayesian view of stochastic gradient descent. Based on its interpretation as a continuous-time stochastic process—specifically a multivariate Ornstein-Uhlenbeck (OU) process (Uhlenbeck and Ornstein, 1930; Gardiner et al., 1985)—we compute stationary distributions for a large class of SGD algorithms, all of which converge to a Gaussian distribution with a non-trivial covariance matrix. The stationary distribution is parameterized by the learning rate, minibatch size, and preconditioning matrix.

   Results about the multivariate OU process make it easy to minimize the KL divergence between the stationary distribution and the posterior analytically. We can thus relate the optimal step size or preconditioning matrix to the Hessian and noise covariances near the optimum. The optimal preconditioners relate to AdaGrad (Duchi et al., 2011), RMSProp (Tieleman and Hinton, 2012), and classical Fisher scoring (Longford, 1987). We demonstrate how these different optimization methods compare when used for approximate inference.

2. We show that constant SGD gives rise to a new variational EM algorithm (Bishop, 2006) which allows us to use SGD to optimize hyperparameters while performing approximate inference in a Bayesian model. We demonstrate this by fitting a posterior to a Bayesian multinomial regression model.

3. We use our formalism to derive the stationary distribution for SGD with momentum (Polyak, 1964). Our results show that adding momentum only changes the scale of the covariance of the stationary distribution, not its shape. This scaling factor is a simple function of the damping coefficient. Thus we can also use SGD with momentum for approximate Bayesian inference.

4. Then, we analyze scalable MCMC algorithms. Specifically, we use the stochastic-process perspective to compute the stationary distribution of Stochastic-Gradient Langevin Dynamics (SGLD) by Welling and Teh (2011) when using constant learning rates, and analyze stochastic gradient Fisher scoring (SGFS) by Ahn et al. (2012). The view from the multivariate OU process reveals a simple justification for this method: we confirm that the preconditioning matrix suggested in SGFS is indeed optimal. We also derive a criterion for the free noise parameter in SGFS that can enhance numerical stability, and we show how the stationary distribution is modified when the preconditioner is approximated with a diagonal matrix (as is often done in practice for high-dimensional problems).

5. Finally, we analyze iterate averaging (Polyak and Juditsky, 1992), where one successively averages the iterates of SGD to obtain a lower-variance estimator of the optimum. Based on the stochastic-process methodology, we give a shorter derivation of a known result, namely that the convergence speed of iterate averaging cannot be improved by preconditioning the stochastic gradient with any matrix. Furthermore, we show that (under certain assumptions), Polyak iterate averaging can yield an *optimal* stochastic-gradient MCMC algorithm, and that this optimal sampler can generate exactly one effectively independent sample per pass through the dataset. This result is both positive and negative; it suggests that iterate averaging can be used as a powerful Bayesian sampler, but it also argues that no SG-MCMC algorithm can generate more than one useful sample per pass through the data, and so the cost of these algorithms must scale linearly with dataset size.

Our paper is organized as follows. In Section 3 we review the continuous-time limit of SGD, showing that it can be interpreted as an OU process. In Section 4 we present consequences of this perspective: the interpretation of SGD as variational Bayes and results around preconditioning and momentum. Section 5 discusses SG Langevin Dynamics and SG Fisher Scoring. In Section 6 we discuss Polyak averaging for optimization and sampling. In the empirical study (Section 7), we show that our theoretical assumptions are satisfied for different models, that we can use SGD to perform gradient-based hyperparameter optimization, and that iterate averaging gives rise to a Bayesian sampler with fast mixing.

2. Related Work

Our paper relates to Bayesian inference and stochastic optimization.

Scalable MCMC. Recent work in Bayesian statistics focuses on making MCMC sampling algorithms scalable by using stochastic gradients. In particular, Welling and Teh (2011) developed stochastic-gradient Langevin dynamics (SGLD). This algorithm samples from a Bayesian posterior by adding artificial noise to the stochastic gradient which, as the step size decays, comes to dominate the SGD noise. Also see Sato and Nakagawa (2014) for a detailed convergence analysis of the algorithm. Though elegant, one disadvantage of SGLD is that the step size must be decreased to arrive at the correct sampling regime, and as step sizes get small so does mixing speed. Other research suggests improvements to this issue, using Hamiltonian Monte Carlo (Chen et al., 2014) or thermostats (Ding et al., 2014). Shang et al. (2015) build on thermostats and use a similar continuous-time formalism as used in this paper.

Ma et al. (2015) give a complete classification of possible stochastic gradient-based MCMC schemes.

Below, we will analyze properties of stochastic gradient Fisher scoring (SGFS; Ahn et al., 2012), an extention to SGLD. This algorithm speeds up mixing times in SGLD by preconditioning gradients with the inverse gradient noise covariance. Ahn et al. (2012) show that (under some assumptions) SGFS can eliminate the bias associated with running SGLD with non-vanishing learning rates. Our approach to analysis can extend and sharpen the results of Ahn et al. (2012). For example, they propose using a diagonal approximation to the gradient noise covariance as a heuristic; in our framework, we can analyze and rigorously justify this choice as a variational Bayesian approximation.

Maclaurin et al. (2016) also interpret SGD as a non-parametric variational inference scheme, but with different goals and in a different formalism. The paper proposes a way to track entropy changes in the implicit variational objective, based on estimates of the Hessian. As such, the authors mainly consider sampling distributions that are not stationary, whereas we focus on constant learning rates and distributions that have (approximately) converged. Note that their notion of hyperparameters does not refer to model parameters but to parameters of SGD.

Stochastic Optimization. Stochastic gradient descent is an active field (Zhang, 2004; Bottou, 1998). Many papers discuss constant step-size SGD. Bach and Moulines (2013); Flammarion and Bach (2015) discuss convergence rate of averaged gradients with constant step size, while Défossez and Bach (2015) analyze sampling distributions using quasi-martingale techniques. Toulis et al. (2014) calculate the asymptotic variance of SGD for the case of decreasing learning rates, assuming that the data is distributed according to the model. None of these papers consider the Bayesian setting. Dieuleveut et al. (2017) also analyzed SGD with constant step size and its relation to

Markov chains. Their analysis resulted in a novel extrapolation scheme to improve the convergence behavior of iterate averaging.

The fact that optimal preconditioning (using a decreasing Robbins-Monro schedule) is achieved by choosing the inverse noise covariance was first shown in (Sakrison, 1965), but here we derive the same result based on different arguments and suggest a scalar prefactor. Note the optimal scalar learning rate of $2/\mathrm{Tr}(BB^\top)$, where $BB^\top$ is the SGD noise covariance (as discussed in Section 4 or this paper), can also be derived based on stability arguments. This was done in the context of least mean square filters (Widrow and Stearns, 1985).

Finally, Chen et al. (2016) also draw analogies between SGD and scalable MCMC. They suggest annealing the posterior over time to use scalable MCMC as a tool for global optimization. We follow the opposite idea and suggest to use constant SGD as an approximate sampler by choosing appropriate learning rate and preconditioners.

Stochastic differential equations. The idea of analyzing stochastic gradient descent with stochastic differential equations is well established in the stochastic approximation literature (Kushner and Yin, 2003; Ljung et al., 2012). Recent work focuses on dynamical aspects of the algorithm. Li et al. (2015) discuss several one-dimensional cases and momentum. Li et al. (2017) give a mathematically rigorous justification of the continuous-time limit. Chen et al. (2015) analyze stochastic gradient MCMC and study their convergence properties using stochastic differential equations.

Our work makes use of the same formalism but has a different focus. Instead of analyzing dynamical properties, we focus on stationary distributions. Further, our paper introduces the idea of minimizing KL divergence between multivariate sampling distributions and the posterior.

Variational Inference. Variational Inference (VI) denotes a set of methods which aim at approximating a Bayesian posterior by a simpler, typically factorized distribution. This is done by minimizing Kullback-Leibler divergence or related divergences between these distributions (Jordan et al., 1999; Opper and Saad, 2001). For the class of models where the conditional distributions are in the exponential family, the variational objective can be optimized by closed-form updates (Ghahramani and Beal, 2000), but this is a restricted class of models with conjugate priors. A scalable version of VI, termed Stochastic Variational Inference (SVI), relies on stochastic gradient descent for data subsampling (Hoffman et al., 2013). For non-conjugate models, Black-Box variational inference (Ranganath et al., 2014) has enabled SVI for a large class of models, but this approach may suffer from high-variance gradients. A modified form of black-box variational inference relies on re-parameterization gradients (Salimans and Knowles, 2013; Kingma and Welling, 2014; Rezende et al., 2014; Kucukelbir et al., 2015; Ruiz et al., 2016). This version is limited to continuous latent variables but typically has much lower-variance gradients.

In this paper, we compare against the Gaussian reparameterization gradient version of black-box variational inference as used in Kingma and Welling (2014); Rezende et al. (2014); Kucukelbir et al. (2015) which we refer to as BBVI. We find that our approach performs similarly in practice, but it is different in that it does not optimize the parameters of a simple variational distribution. Rather, it controls the shape of the approximate posterior via the parameters of the optimization algorithm, such as the learning rate or preconditioning matrix.

3. Continuous-Time Limit Revisited

We first review the theoretical framework that we use throughout the paper. Our goal is to characterize the behavior of SGD when using a constant step size. To do this, we approximate SGD with a continuous-time stochastic process (Kushner and Yin, 2003; Ljung et al., 2012).

3.1 Problem Setup

Consider loss functions of the following form:

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\ell_n(\theta), \quad g(\theta) \equiv \nabla_\theta \mathcal{L}(\theta). \tag{2}$$

Such loss functions are common in machine learning, where $\mathcal{L}(\theta) \equiv \mathcal{L}(\theta, \mathbf{x})$ is a loss function that depends on data $\mathbf{x}$ and parameters θ. Each $\ell_n(\theta) \equiv \ell(\theta, \mathbf{x}_n)$ is the contribution to the overall loss from a single observation $\mathbf{x}_n$. For example, when finding a maximum-a-posteriori estimate of a model, the contributions to the loss may be

$$\ell_n(\theta) = -\log p(x_n \,|\, \theta) - \frac{1}{N}\log p(\theta), \tag{3}$$

where $p(x_n \,|\, \theta)$ is the likelihood and $p(\theta)$ is the prior. For simpler notation, we will suppress the dependence of the loss on the data.

From this loss we construct stochastic gradients. Let $\mathcal{S}$ be a set of S random indices drawn uniformly at random from the set $\{1, \ldots, N\}$. This set indexes functions $\ell_n(\theta)$, and we call $\mathcal{S}$ a "minibatch" of size S. Based on the minibatch, we used the indexed functions to form a stochastic estimate of the loss and a stochastic gradient,

$$\hat{\mathcal{L}}_S(\theta) = \frac{1}{S}\sum_{n\in\mathcal{S}}\ell_n(\theta), \quad \hat{g}_S(\theta) = \nabla_\theta \hat{\mathcal{L}}_S(\theta). \tag{4}$$

In expectation the stochastic gradient is the full gradient, i.e., $g(\theta) = \mathbb{E}[\hat{g}_S(\theta)]$. We use this stochastic gradient in the SGD update

$$\theta(t+1) = \theta(t) - \epsilon\,\hat{g}_S(\theta(t)). \tag{5}$$

Above and for what follows, we assume a constant (non-decreasing) learning rate ϵ.

Eqs. 4 and 5 define the discrete-time process that SGD simulates from. We will approximate it with a continuous-time process that is easier to analyze.

3.2 SGD as an Ornstein-Uhlenbeck Process

We now show how to approximate the discrete-time Eq. 5 with a continuous-time Ornstein-Uhlenbeck process (Uhlenbeck and Ornstein, 1930). This leads to the stochastic differential equation below in Eq. 11. To justify the approximation, we make four assumptions. We verify its accuracy empirically in Section 7.

Assumption 1 *Observe that the stochastic gradient is a sum of S independent, uniformly sampled contributions. Invoking the central limit theorem, we assume that the gradient noise is Gaussian with covariance $\frac{1}{S}C(\theta)$, hence*

$$\hat{g}_S(\theta) \approx g(\theta) + \frac{1}{\sqrt{S}}\Delta g(\theta), \quad \Delta g(\theta) \sim \mathcal{N}(0, C(\theta)). \tag{6}$$

Assumption 2 *We assume that the covariance matrix $C(\theta)$ is approximately constant with respect to θ. As a symmetric positive-semidefinite matrix, this constant matrix C factorizes as*

$$C(\theta) \approx C = BB^\top. \tag{7}$$

Assumption 2 is justified when the iterates of SGD are confined to a small enough region around a local optimum of the loss (e.g. due to a small ϵ) that the noise covariance does not vary significantly in that region.

We now define $\Delta\theta(t) = \theta(t+1) - \theta(t)$ and combine Eqs. 5, 6, and 7 to rewrite the process as

$$\Delta\theta(t) = -\epsilon\, g(\theta(t)) + \tfrac{\epsilon}{\sqrt{S}} B\, \Delta W, \quad \Delta W \sim \mathcal{N}(0, \mathbf{I})\,. \tag{8}$$

This can be interpreted as a finite-difference equation that approximates the following continuous-time stochastic differential equation:

$$d\theta(t) = -\epsilon g(\theta) dt + \tfrac{\epsilon}{\sqrt{S}} B\, dW(t). \tag{9}$$

Assumption 3 *We assume that we can approximate the finite-difference equation (8) by the stochastic differential equation (9).*

This assumption is justified if either the gradients or the learning rates are small enough that the discretization error becomes negligible.

Assumption 4 *We assume that the stationary distribution of the iterates is constrained to a region where the loss is well approximated by a quadratic function,*

$$\mathcal{L}(\theta) = \tfrac{1}{2}\, \theta^\top A \theta. \tag{10}$$

(Without loss of generality, we assume that a minimum of the loss is at $\theta = 0$.) We also assume that A is positive definite.

The symmetric matrix A is thus the Hessian at the optimum. Assumption 4 makes sense when the loss function is smooth and the stochastic process reaches a low-variance quasi-stationary distribution around a deep local minimum. The exit time of a stochastic process is typically exponential in the height of the barriers between minima, which can make local optima very stable even in the presence of noise (Kramers, 1940).

SGD as an Ornstein-Uhlenbeck process. The four assumptions above result in a specific kind of stochastic process, the multivariate *Ornstein-Uhlenbeck* process (Uhlenbeck and Ornstein, 1930):

$$d\theta(t) = -\epsilon A\, \theta(t) dt + \tfrac{1}{\sqrt{S}} \epsilon B\, dW(t) \tag{11}$$

This connection helps us analyze properties of SGD because the Ornstein-Uhlenbeck process has an analytic stationary distribution $q(\theta)$ that is Gaussian. This distribution will be the core analytic tool of this paper:

$$q(\theta) \propto \exp\left\{-\tfrac{1}{2}\theta^\top \Sigma^{-1} \theta\right\}. \tag{12}$$

The covariance Σ satisfies

$$\Sigma A + A\Sigma = \tfrac{\epsilon}{S} BB^\top. \tag{13}$$

(More details are in Appendix B.) Without explicitly solving this equation, we see that the resulting covariance Σ is proportional to the learning rate ϵ and inversely proportional to the magnitude of A and minibatch size S. This characterizes the stationary distribution of running SGD with a constant step size.

Discussion of Assumptions 1–4. Our analysis suggests that constant SGD and Langevin-type diffusion algorithms (Welling and Teh, 2011) are very similar. Both types of algorithms can be characterized by three regimes. First, there is a search phase where the algorithm approaches the optimum. In this early phase, assumptions 1–4 are often violated and it is hard to say anything general about the behavior of the algorithm. Second, there is a phase where SGD has converged to the vicinity of a local optimum. Here, the objective already looks quadratic, but the gradient noise is small relative to the average gradient $g(\theta)$. Thus SGD takes a relatively directed path towards the optimum. This is the regime where our assumptions should be approximately valid, and where our formalism reveals its use. Finally, in the third phase the iterates are near the local optimum. Here, the average gradient $g(\theta)$ is small and the sampling noise becomes more important. In this final phase, constant SGD begins to sample from its stationary distribution.

Finally, we note that if the gradient noise covariance C does not have full rank, then neither will the stationary covariance Σ. This scenario complicates the analysis in Section 4, so below we will assume that C has full rank. This could be enforced easily by adding very low-magnitude isotropic artificial Gaussian noise to the stochastic gradients.

4. SGD as Approximate Inference

We discussed a continuous-time interpretation of SGD with a constant step size (constant SGD). We now discuss how to use constant SGD as an approximate inference algorithm. To repeat the set-up from the introduction, consider a probabilistic model $p(\theta, \mathbf{x})$ with data $\mathbf{x}$ and hidden variables θ; our goal is to approximate the posterior $p(\theta \mid \mathbf{x})$ in Eq. 1.

We set the loss to be the negative log-joint distribution (Eqs. 2 and 3), which equals the negative log-posterior up to an additive constant. The classical goal of SGD is to minimize this loss, leading us to a maximum-a-posteriori point estimate of the parameters. This is how SGD is used in many statistical models, including logistic regression, linear regression, matrix factorization, and neural networks. In contrast, our goal here is to tune the parameters of SGD so that its stationary distribution approximates the posterior.

Fig. 1 shows an example. Here we illustrate two Bayesian posteriors—from a linear regression problem (top) and a logistic regression problem (bottom)—along with iterates from a constant SGD algorithm. In these figures, we set the parameters of the optimization to values that minimize the Kullback-Leibler (KL) divergence between the stationary distribution of the OU process and the posterior—these results come from our theorems below. The left plots optimize both a preconditioning matrix and the step size; the middle plots optimize only the step size (both are outlined in Section 4.1). We can see that the stationary distribution of constant SGD can be made close to the exact posterior.

Fig. 2 also compares the empirical covariance of the iterates with the predicted covariance in terms of Eq. 13. The close match supports the assumptions of Section 3.

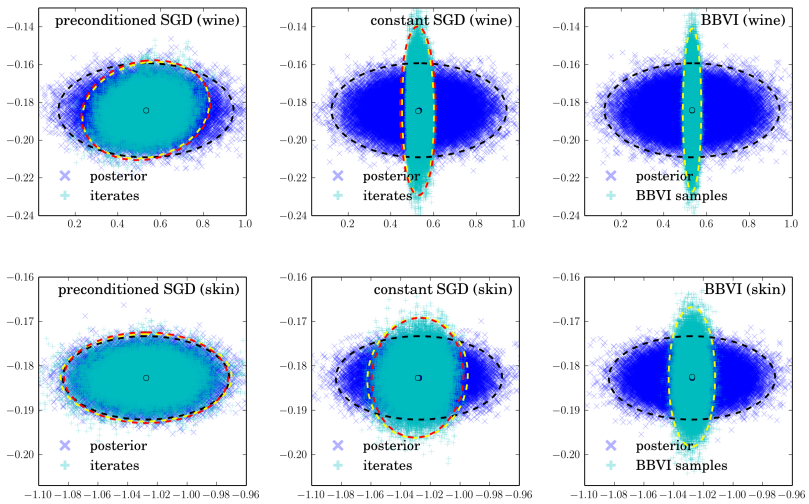


Figure 1: Posterior distribution $f(\theta) \propto \exp\{-N\mathcal{L}(\theta)\}$ (blue) and stationary sampling distributions $q(\theta)$ of the iterates of SGD (cyan) or black box variational inference (BBVI) based on reparameterization gradients. Rows: linear regression (top) and logistic regression (bottom) discussed in Section 7. Columns: full-rank preconditioned constant SGD (left), constant SGD (middle), and BBVI (Kucukelbir et al., 2015) (right). We show projections on the smallest and largest principal component of the posterior. The plot also shows the empirical covariances (3 standard deviations) of the posterior (black), the covariance of the samples (yellow), and their prediction (red) in terms of the Ornstein-Uhlenbeck process, Eq. 13.

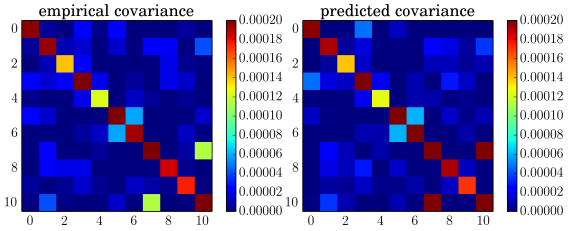


Figure 2: Empirical and predicted covariances of the iterates of stochastic gradient descent, where the prediction is based on Eq. 13. We used linear regression on the wine quality data set as detailed in Section 7.1.

We will use this perspective in three ways. First, we develop optimal conditions for constant SGD to best approximate the posterior, connecting to well-known results around adaptive learning rates and preconditioners (Duchi et al., 2011; Tieleman and Hinton, 2012). Second, we propose an algorithm for hyperparameter optimization based on constant SGD. Third, we use it to analyze the stationary distribution of stochastic gradient descent with momentum (Polyak, 1964).

4.1 Constant Stochastic Gradient Descent

First, we show how to tune constant SGD's parameters to minimize KL divergence to the posterior; this is a type of variational inference (Jordan et al., 1999). The analysis leads to three versions of constant SGD—one with a constant step size, one with a full preconditioning matrix, and one with a diagonal preconditioning matrix. Each yields samples from an approximate posterior, and each reflects a different tradeoff between efficiency and accuracy. Finally, we show how to use these algorithms to learn hyperparameters.

Assumption 4 from Section 3 says that the posterior is approximately Gaussian in the region that the stationary distribution focuses on,

$$f(\theta) \propto \exp\left\{-\tfrac{N}{2}\theta^\top A\theta\right\}. \tag{14}$$

The scalar N corrects the averaging in equation 2. Furthermore, in this section we will consider a more general SGD scheme that may involve a *preconditioning matrix* H instead of a scalar learning rate ϵ:

$$\theta_{t+1} = \theta_t - H\hat{g}_S(\theta(t)).$$

We will set the parameters of SGD to minimize the KL divergence between the stationary distribution $q(\theta)$ (Eqs. 12 and 13) and the posterior $f(\theta)$ (Eq. 14). This involves the the learning rate ϵ or more generally the preconditioning matrix H and the minibatch size S:

$$\{H^*, S^*\} = \arg\min_{H,S} KL(q \mid\mid f).$$

First, consider a scalar learning rate ϵ (or a trivial preconditioner $H = \epsilon\mathbf{I}$). The distributions $f(\theta)$ and $q(\theta)$ are both Gaussians. Their means coincide, at the minimum of the loss, and so their KL divergence is

$$\begin{aligned} KL(q \,||\, f) &= -\mathbb{E}_q[\log f(\theta)] + \mathbb{E}_q[\log q(\theta)] \\ &= \tfrac{1}{2}\left(N\mathbb{E}_q[\theta^\top A\theta] - \log|NA| - \log|\Sigma| - D\right) \\ &= \tfrac{1}{2}\left(N\mathrm{Tr}(A\Sigma) - \log|NA| - \log|\Sigma| - D\right), \end{aligned}$$

where $|\cdot|$ is the determinant and D is the dimension of θ.

We suggest three variants of constant SGD that generate samples from an approximate posterior.

Theorem 1 (Constant SGD) *Under Assumptions A1-A4, the constant learning rate that minimizes KL divergence from the stationary distribution of constant SGD to the posterior is*

$$\epsilon^* = 2\tfrac{S}{N}\tfrac{D}{\mathrm{Tr}(BB^\top)}. \tag{15}$$

Proof To prove this claim, we face the problem that the covariance of the stationary distribution depends indirectly on ϵ through Eq. 13. Inspecting this equation reveals that $\Sigma_0 \equiv \frac{S}{\epsilon}\Sigma$ is independent of S and ϵ. This simplifies the entropy term $\log|\Sigma| = D\log(\epsilon/S) + \log|\Sigma_0|$. Since Σ_0 is constant, we can neglect it when minimizing KL divergence.

We also need to simplify the term $\mathrm{Tr}(A\Sigma)$, which still depends on ϵ and S through Σ. To do this, we again use Eq. 13, from which follows that $\mathrm{Tr}(A\Sigma) = \frac{1}{2}(\mathrm{Tr}(A\Sigma) + \mathrm{Tr}(\Sigma A)) = \frac{\epsilon}{2S}\mathrm{Tr}(BB^\top)$. The KL divergence is therefore, up to constant terms,

$$KL(q||f) \overset{c}{=} \frac{\epsilon N}{2S}\mathrm{Tr}(BB^\top) - D\log(\epsilon/S). \tag{16}$$

Minimizing KL divergence over ϵ/S results in Eq. 15 for the optimal learning rate. ■

Theorem 1 suggests that the learning rate should be chosen inversely proportional to the average of diagonal entries of the noise covariance, and proportional to the ratio between the minibatch size and dataset size.

Since both the posterior f and the variational distribution q are Gaussian, one might wonder if also the reverse KL divergence is viable. While the KL divergence can be computed up to a constant, we cannot remove its dependence on the unknown stationary distribution Σ using Eq. 13, unless A and Σ commute. This setup is discussed in Appendix C.

Instead of using a scalar learning rate, we now consider a positive-definite preconditioning matrix H. This gives us more tuning parameters to better approximate the posterior.

Theorem 2 (Preconditioned constant SGD) *The preconditioner for constant SGD that minimizes KL divergence from the stationary distribution to the posterior is*

$$H^* = \frac{2S}{N}(BB^\top)^{-1}. \tag{17}$$

Proof To prove this, we need the Ornstein-Uhlenbeck process which corresponds to preconditioned SGD. Replacing the constant learning rate in Eq. 11 with a positive-definite preconditioning matrix H results in

$$d\theta(t) = -HA\,\theta(t)dt + \frac{1}{\sqrt{S}}HB\,dW(t).$$

All our results carry over after substituting $\epsilon A \leftarrow HA,\ \epsilon B \leftarrow HB$. Eq. 13, after the transformation and multiplication by H^{-1} from the left, becomes

$$A\Sigma + H^{-1}\Sigma AH = \frac{1}{S}BB^\top H. \tag{18}$$

Using the cyclic property of the trace, this implies that

$$\mathrm{Tr}(A\Sigma) = \frac{1}{2}(\mathrm{Tr}(A\Sigma) + \mathrm{Tr}(H^{-1}A\Sigma H)) = \frac{1}{2S}\mathrm{Tr}(BB^\top H). \tag{19}$$

Consider now the log determinant term, $\log|\Sigma|$, which still has an implicit dependence on H. We first define $Q = \Sigma H^{-1}$, hence $Q^\top = H^{-1}\Sigma$ since Σ, H and H^{-1} are symmetric. Eq. 18 can be written as $A\Sigma H^{-1} + H^{-1}\Sigma A = \frac{1}{S}BB^\top$, which is equivalent to $QA + AQ^\top = \frac{1}{S}BB^\top$.

Thus, we see that Q is independent of H. The log determinant term is up to a constant $\log|\Sigma| = \log|H| + \log|Q|$. Combining Eq. 19 with this term, the KL divergence is up to a constant

$$KL(q||f) \overset{c}{=} \frac{N}{2S}\mathrm{Tr}(BB^\top H) + \log|H| + \log|Q|. \tag{20}$$

Taking derivatives with respect to the entries of H results in Eq. 17. ■

In high-dimensional applications, working with large dense matrices is impractical. In those settings we can constrain the preconditioner to be diagonal. The following corollaries are based on the proof of Theorem 2:

Corollary 3 *The optimal diagonal preconditioner for SGD that minimizes KL divergence to the posterior is* $H^*_{kk} = \frac{2S}{NBB^\top_{kk}}$.

Proof This follows from Eq. 20, where we restrict the preconditioning matrix to be diagonal. ■

Corollary 4 *Under assumptions A1-A4, preconditioning with the full inverse noise covariance as in Theorem 2 results in samples from the exact posterior.*

Proof Consider Eq. 18. Inserting $H = \frac{2S}{N}(BB^\top)^{-1}$ results in $A\Sigma + H^{-1}\Sigma AH = \frac{2}{N}\mathbf{I}$ which is solved by $\Sigma = A^{-1}/N$ which is the posterior covariance. ■

We showed that the optimal diagonal preconditioner is the inverse of the diagonal part of the noise matrix. Similar preconditioning matrices have been suggested earlier in optimal control theory based on very different arguments, see (Widrow and Stearns, 1985). Our result also relates to AdaGrad and its relatives (Duchi et al., 2011; Tieleman and Hinton, 2012), which also adjust the preconditioner based on the square root of the diagonal entries of the noise covariance. In Appendix F we derive an optimal global learning rate for AdaGrad-style diagonal preconditioners. In Section 7, we compare three versions of constant SGD for approximate posterior inference: one with a scalar step size, one with a dense preconditioner, and one with a diagonal preconditioner.

Remark on estimating the noise covariance. In order to use our theoretical insights in practice, we need to estimate the stochastic gradient noise covariance $C \equiv BB^\top$. We do this in an online manner. As before, let g_t be the full gradient, $\hat{g}_{S,t}$ be the stochastic gradient of the full minibatch, and $\hat{g}_{1,t}$ be the stochastic gradient of the first sample in the minibatch at time t (which has a much larger variance if $S \gg 1$). For large S, we can approximate $g_t \approx \hat{g}_{S,t}$, and thus obtain an estimator of the noise covariance by $(\hat{g}_{1,t} - \hat{g}_{S,t})(\hat{g}_{1,t} - \hat{g}_{S,t})^\top$. Following Ahn et al. (2012), we can now build an online estimate C_t that approaches C by the following recursion,

$$C_t = (1 - \kappa_t)C_{t-1} + \kappa_t(\hat{g}_{1,t} - \hat{g}_{S,t})(\hat{g}_{1,t} - \hat{g}_{S,t})^\top. \tag{21}$$

Above, κ_t is a decreasing learning rate. Ahn et al. (2012) have proven that such an online average converges to the noise covariance in the optimum at long times (provided that $\kappa_t \sim 1/t$ and that N is sufficiently large). We found that this online estimator works well in practice, even though our theoretical assumptions would require preconditioning SGD with the true noise covariance in finite

time. Regarding the computational overhead of this procedure, note that similar online estimates of the gradient noise are carried out in adaptive SGD schemes such as AdaGrad (Duchi et al., 2011) or RMSProp (Tieleman and Hinton, 2012). When using a diagonal approximation to the noise covariance, the costs are proportional in the number of dimensions and mini-batch size; this efficiency means that the online estimate does not spoil the efficiency of SGD. Full preconditioning scales quadratically in the dimension and is therefore impractical in many real-word setups.

4.2 Constant SGD as Variational EM

Consider a supervised probabilistic model with joint distribution $p(y, \theta|x, \lambda) = p(y|x, \theta)p(\theta|\lambda)$ that factorizes into a likelihood and prior, respectively. Our goal is to find optimal hyperparameters λ. Jointly point-estimating θ and λ by following gradients of the log joint leads to overfitting or degenerate solutions. This can be prevented in a Bayesian approach, where we treat the parameters θ as latent variables. In Empirical Bayes (or type-II maximum likelihood), we maximize the *marginal* likelihood of the data, integrating out the main model parameters:

$$\lambda^\star = \arg\max_\lambda \log p(y|x, \lambda) = \arg\max_\lambda \log \textstyle\int_\theta p(y, \theta|x, \lambda) d\theta.$$

When this marginal log-likelihood is intractable, a common approach is to use *variational expectation-maximization (VEM)* (Bishop, 2006), which iteratively optimizes a variational lower bound on the marginal log-likelihood over λ. If we approximate the posterior $p(\theta|x, y, \lambda)$ with some distribution $q(\theta)$, then VEM tries to find a value for λ that maximizes the expected log-joint probability $\mathbb{E}_q[\log p(\theta, y|x, \lambda)]$.

Constant SGD gives rise to a simple VEM algorithm that applies to a large class of differentiable models. Define $\mathcal{L}(\theta, \lambda) = -\log p(y, \theta \,|\, x, \lambda)$. If we interpret the stationary distribution of SGD as a variational approximation to a model's posterior, we can justify following a stochastic gradient descent scheme on both θ and λ:

$$\theta_{t+1} = \theta_t - \epsilon^* \nabla_\theta \mathcal{L}(\theta_t, \lambda_t); \quad \lambda_{t+1} = \lambda_t - \rho_t \nabla_\lambda \mathcal{L}(\theta_t, \lambda_t). \tag{22}$$

While the update for θ uses the optimal constant learning rate ϵ^* and therefore samples from an approximate posterior, the λ update uses a decreasing learning rate ρ_t and therefore converges to a local optimum. The result is a type of VEM algorithm.

We stress that the optimal constant learning rate ϵ^* is not unknown, but can be estimated based on an online estimate of the noise covariance $C \equiv BB^\top$, as given in Eq. 21. In Section 7 we show that gradient-based hyperparameter learning is a cheap alternative to cross-validation.

4.3 Stochastic Gradient with Momentum

The continuous-time formalism also allows us to explore extensions of classical SGD. One of the most popular methods is stochastic gradient with momentum (Polyak, 1964; Sutskever et al., 2013). Here, we give a version of this algorithm that allows us to sample from an approximate posterior.

SGD with momentum doubles the dimension of parameter space in introducing an additional *momentum* variable v that has the same dimension as θ. Position and momentum are coupled in such a way that the algorithm keeps memory of some of its past gradients. The updates of SGD with momentum are

$$\begin{aligned} \mathrm{v}(t+1) &= (1-\mu)\mathrm{v}(t) - \epsilon\, \hat{g}_S(\theta(t)) \\ \theta(t+1) &= \theta(t) + \mathrm{v}(t+1). \end{aligned}$$

This involves the damping coefficient $\mu \in [0, 1]$. For $\mu = 1$ (infinite damping or overdamping), the momentum information gets lost and we recover SGD.

As before we assume a quadratic objective $\mathcal{L} = \frac{1}{2}\theta^\top A\theta$. Going through the same steps A1-A4 of Section 3 that allowed us to derive the Ornstein-Uhlenbeck process for SGD, we find

$$\begin{aligned} d\mathrm{v} &= -\mu\mathrm{v}dt - \epsilon A\theta dt + \tfrac{1}{\sqrt{S}}\epsilon B\, dW, \\ d\theta &= \mathrm{v}dt. \end{aligned} \tag{23}$$

We solve this set of stochastic equations asymptotically for the long-time limit. We reformulate the stochastic equation in terms of coupled equations for the moments. (This strategy was also used by Li et al. (2015) in a more restricted setting). The first moments of the set of coupled stochastic differential equations give

$$d\mathbb{E}[\mathrm{v}] = -\mu\mathbb{E}[\mathrm{v}]dt - \epsilon A\mathbb{E}[\theta]dt, \quad d\mathbb{E}[\theta] = \mathbb{E}[\mathrm{v}]dt.$$

Note we used that expectations commute with the differential operators. These deterministic equations have the simple solution $\mathbb{E}[\theta] = 0$ and $\mathbb{E}[\mathrm{v}] = 0$, which means that the momentum has expectation zero and the expectation of the position variable converges to the optimum (at 0).

In order to compute the stationary distribution, we derive and solve similar equations for the second moments. These calculations are carried out in Appendix D, where we derive the following conditions:

$$\begin{aligned} \mathbb{E}[\mathrm{v}\mathrm{v}^\top] &= \tfrac{\epsilon}{2}\mathbb{E}[\theta\theta^\top]A + \tfrac{\epsilon}{2}A\mathbb{E}[\theta\theta^\top], \\ \mu\mathbb{E}[\mathrm{v}\mathrm{v}^\top] &= \tfrac{\epsilon^2}{2S}BB^\top. \end{aligned} \tag{24}$$

Eq. 25 relate to equilibrium thermodynamics, where $\mathbb{E}[\mathrm{v}\mathrm{v}^\top]$ is a matrix of expected kinetic energies, while $\frac{1}{2}(\mathbb{E}[\theta\theta^\top]A + A\mathbb{E}[\theta\theta^\top])$ has the interpretation of a matrix of expected potential energies. The first equation says that energy conservation holds in expectation, which is the case for an equilibrium system which exchanges energy with its environment. The second equation relates the covariance of the velocities with $BB^\top$, which plays the role of a matrix-valued temperature. This is known as the fluctuation-dissipation theorem (Nyquist, 1928). Combining both equations and using $\Sigma = \mathbb{E}[\theta\theta^\top]$ yields

$$\Sigma A + A\Sigma = \tfrac{\epsilon}{\mu S}BB^\top. \tag{25}$$

This is exactly Eq. 13 of SGD without momentum, however, with the difference that the noise covariance is re-scaled by a factor $\frac{\epsilon}{\mu S}$ instead of $\frac{\epsilon}{S}$.

We have shown that ϵ, S, and μ play similar roles: only the combination $\frac{\epsilon}{\mu S}$ affects the KL divergence to the posterior. Thus, no single optimal constant learning rate exists—many combinations of ϵ, μ, and S can yield the same stationary distribution. But different choices of these parameters will affect the dynamics of the Markov chain. For example, Sutskever et al. (2013) observe that, for a given effective learning rate $\frac{\epsilon}{\mu S}$, using a smaller μ sometimes makes the *discretized* dynamics of SGD more stable. Also, using very small values of μ while holding $\frac{\epsilon}{\mu}$ fixed will eventually increase the autocorrelation time of the Markov chain (but this effect is often negligible in practice).

5. Analyzing Stochastic Gradient MCMC Algorithms

We have analyzed well-known stochastic optimization algorithms such as SGD, preconditioned SGD and SGD with momentum. We now investigate Bayesian sampling algorithms. A large class of modern MCMC methods rely on stochastic gradients (Welling and Teh, 2011; Ahn et al., 2012; Chen et al., 2014; Ding et al., 2014; Ma et al., 2015). The central idea is to add artificial noise to the stochastic gradient to asymptotically sample from the true posterior.

In practice, however, the algorithms are used in combination with several heuristic approximations, such as non-vanishing learning rates or diagonal approximations to the Hessian. In this section, we use the variational perspective to quantify these biases and help understanding these algorithms better under more realistic assumptions.

5.1 SGLD with Constant Rates

To begin with, we we analyze the well-known Stochastic Gradient Langevin Dynamics by Welling and Teh (2011). This algorithm has been carefully analyzed in the long time limit where the stochastic gradient noise vanishes as the learning rate goes to zero, and where mixing becomes infinitely slow (Sato and Nakagawa, 2014). Here we analyze an approximate version of the algorithm, SGLD with a constant learning rate. We are interested in the stationary distribution of this approximate inference algorithm.

The discrete-time process that describes Stochastic Gradient Langevin dynamics is

$$\theta_{t+1} \quad = \quad \theta_t - \tfrac{\epsilon}{2} N \hat{\nabla}_\theta \mathcal{L}(\theta_t) + \sqrt{\epsilon}\, V(t),$$

where $V(t) \sim \mathcal{N}(0, \mathbf{I})$ is a vector of independent Gaussian noises. Following assumptions 1–4 of Section 3, V becomes the Wiener noise dV and the corresponding continuous-time process is

$$d\theta \quad = \quad -\tfrac{1}{2}\epsilon N A \theta dt + \sqrt{\epsilon} dV + \epsilon \tfrac{1}{\sqrt{S}} N B\, dW.$$

Above, dV and dW are vectors of independent Wiener noises, i.e. $\mathbb{E}[dW dV^\top] = \mathbb{E}[dV dW^\top] = 0$. The analog of Eq. 13 is then

$$\tfrac{1}{2} N (A\Sigma + \Sigma A) \quad = \quad \mathbf{I} + \tfrac{\epsilon N}{2S} B B^\top.$$

In the limit of $\epsilon \to 0$, we find that $\Sigma^{-1} = NA$, meaning the stationary distribution becomes identical to the posterior. However, for non-zero ϵ, there are discrepancies. These correction-terms are positive. This shows that the posterior covariance is generally overestimated by Langevin dynamics, which can be attributed to non-vanishing learning rates at long times.

5.2 Stochastic Gradient Fisher Scoring

We now investigate Stochastic Gradient Fisher Scoring (Ahn et al., 2012), a scalable Bayesian MCMC algorithm. We use the variational perspective to rederive the Fisher scoring update and identify it as optimal. We also analyze the sampling distribution of the truncated algorithm, one with diagonal preconditioning (as it is used in practice), and quantify the bias that this induces.

The basic idea here is that the stochastic gradient is preconditioned and additional noise is added to the updates such that the algorithm approximately samples from the Bayesian posterior. More precisely, the update can be cast into the following form:

$$\theta(t+1) = \theta(t) - \epsilon H\, \hat{g}(\theta(t)) + \sqrt{\epsilon} H E\, W(t). \tag{26}$$

The matrix H is a preconditioner and $EW(t)$ is Gaussian noise; we control the preconditioner and the covariance $EE^\top$ of the noise. Stochastic gradient Fisher scoring suggests a preconditioning matrix H that leads to samples from the posterior even if the learning rate ϵ is not asymptotically small. We show here that this preconditioner follows from our variational analysis.

Theorem 5 (Stochastic Gradient Fisher Scoring) *Under Assumptions A1-A4, the positive-definite preconditioner H in Eq. 26 that minimizes KL divergence from the stationary distribution of SGFS to the posterior is*

$$H^* = \tfrac{2}{N}(\epsilon B B^\top + E E^\top)^{-1}. \tag{27}$$

Proof To prove the claim, we go through the steps of Section 3 to derive the corresponding Ornstein-Uhlenbeck process, $d\theta(t) = -\epsilon H A \theta(t) dt + H\left[\epsilon B + \sqrt{\epsilon} E\right] dW(t)$. For simplicity, we have set the minibatch size S to 1. In Appendix E, we derive the following KL divergence between the posterior and the sampling distribution:

$$KL(q||p) = -\tfrac{N}{4}\mathrm{Tr}(H(\epsilon B B^\top + E E^\top)) + \tfrac{1}{2}\log|T| + \tfrac{1}{2}\log|H| + \tfrac{1}{2}\log|NA| + \tfrac{D}{2}.$$

(T is constant with respect to H, ϵ, and E.) We can now minimize this KL divergence over the parameters H and E. When E is given, minimizing over H gives Eq. 27. ■

Eq. 27 not only minimizes the KL divergence, but makes it 0, meaning that the stationary sampling distribution *is* the posterior. This solution corresponds to the suggested Fisher Scoring update in the idealized case when the sampling noise distribution is estimated perfectly (Ahn et al., 2012). Through this update, the algorithm thus generates posterior samples without decreasing the learning rate to zero. (This is in contrast to Stochastic Gradient Langevin Dynamics by Welling and Teh (2011).)

In practice, however, SGFS is often used with a diagonal approximation of the preconditioning matrix (Ahn et al., 2012; Ma et al., 2015). However, researchers have not explored how the stationary distribution is affected by this truncation, which makes the algorithm only approximately Bayesian. We can quantify its deviation from the exact posterior and we derive the optimal diagonal preconditioner, which follows from the KL divergence in Theorem 5:

Corollary 6 *When approximating the Fisher scoring preconditioner by a diagonal matrix H^*_{kk} or a scalar H^*_{scalar}, respectively, then*

$$H^*_{kk} = \frac{2}{N}(\epsilon B B^\top_{kk} + E E^\top_{kk})^{-1} \quad \textit{and} \quad H^*_{scalar} = \frac{2D}{N}(\sum_k [\epsilon B B^\top_{kk} + E E^\top_{kk}])^{-1}.$$

Note that we have not made any assumptions about the noise covariance E. We can adjust it in favor of a more stable algorithm. For example, in the interests of stability we might want to set a maximum step size h^{max}, so that $H_{kk} \leq h^{max}$ for all k. We can adjust E such that $H_{kk} \equiv h^{max}$ in Eq. 27 becomes independent of k. Solving for E yields $E E^\top_{kk} = \frac{2}{h^{max} N} - \epsilon B B^\top_{kk}$.

Hence, to keep the learning rates bounded in favor of stability, one can inject noise in dimensions where the variance of the gradient is too small. This guideline is opposite to the advice of Ahn et al. (2012) to choose B proportional to E, but follows naturally from the variational analysis.

An additional benefit of SGFS over simple constant SGD is that the sum of gradient noise and Gaussian noise will always look "more Gaussian" than the gradient noise on its own. An extreme case is when the gradient covariance $BB^\top$ is not full rank; in this situation injecting full-rank Gaussian noise could prevent degenerate behavior.

6. A Bayesian View on Iterate Averaging

We now apply our continuous-time analysis to the technique of iterate averaging (Polyak and Juditsky, 1992). Iterate averaging was derived as an optimization algorithm, and we analyze this algorithm in Section 6.1 from this perspective. In Section 6.2 we show that iterate averaging can also be used as a Bayesian sampling algorithm.

6.1 Iterate Averaging for Optimization

Iterate averaging further refines SGD's estimate of the optimal parameters by averaging the iterates produced by a series of SGD steps. Polyak and Juditsky (1992) proved the remarkable result that averaged SGD achieves the best possible convergence rate among stochastic gradient algorithms[1], including those that use second-order information such as Hessians. This implies that the convergence speed of iterate averaging cannot be improved when premultiplying the stochastic gradient with any preconditioning matrix. We use stochastic calculus to show that the stationary distribution of iterate averaging is the same for any constant preconditioner. Based on slightly stronger-than-usual assumptions, we give a short proof on why this result holds.

To eliminate asymptotic biases, iterate averaging usually requires a slowly decreasing learning rate. We consider a simplified version with a constant rate, also analyzed in (Zhang, 2004; Nemirovski et al., 2009).

Algorithm. Before we begin iterate averaging, we assume that we have run ordinary constant SGD for long enough that it has reached its stationary distribution and forgotten its intialization. In this scenario, we use iterate averaging to "polish" the results obtained by SGD.

Iterate averaging estimates the location of the minimum of $\mathcal{L}$ using a sequence of stochastic gradients $\hat{g}_S$, and then computes an average of the iterates in an online manner,

$$\begin{aligned} \theta_{t+1} &= \theta_t - \epsilon \hat{g}_S(\theta_t), \\ \hat{\mu}_{t+1} &= \tfrac{t}{t+1}\,\hat{\mu}_t + \tfrac{1}{t+1}\,\theta_{t+1}. \end{aligned} \tag{28}$$

After T stochastic gradient steps and going over to continuous times, this average is

$$\hat{\mu} \approx \tfrac{1}{T}\int_0^T \theta(t)\,dt \equiv \hat{\mu}'. \tag{29}$$

The average $\hat{\mu}$ and its approximation $\hat{\mu}'$ are random variables whose expected value is the minimum of the objective. (Again, this assumes $\theta(0)$ is drawn from the SGD process's stationary distribution.)

The accuracy of this estimator $\hat{\mu}'$ after a fixed number of iterations T is characterized by its covariance matrix D. Using stochastic calculus, we can compute D from the autocorrelation matrix of the stationary distribution of the OU process, shown in Appendix G. We prove that

$$D \equiv \mathbb{E}\left[\hat{\mu}'\hat{\mu}'^\top\right] \approx \tfrac{1}{\epsilon T}\left(\Sigma(A^{-1})^\top + A^{-1}\Sigma\right). \tag{30}$$

1. To be more precise, averaged SGD is optimal among methods that only have access to a stochastic gradient oracle—if more is known about the source of the noise then sometimes better rates are possible (e.g., Johnson and Zhang, 2013; Defazio et al., 2014).

This approximation ignores terms of order $1/(T\epsilon)^2$. This term is small since we assume that the number of iterations T is much larger than the inverse learning rate $1/\epsilon$. The covariance shrinks linearly over iterations as we expect from Polyak and Juditsky (1992). We use Eq. 13 to derive

$$D \approx \tfrac{1}{TS} A^{-1} BB^\top (A^{-1})^\top.$$

Note that this covariance depends only on the number of iterations T times the minibatch size S, not on the step size ϵ. Since TS is the total number of examples that are processed, this means that this iterate averaging scheme's efficiency does not depend on either the minibatch size or the step size, proven first in (Polyak and Juditsky, 1992).

We can make a slightly stronger statement. If we precondition the stochastic gradients with a positive-definite matrix H (for example, the inverse of the Hessian evaluated at the minimum), it turns out that the covariance of the estimator remains unchanged. The resulting Ornstein-Uhlenbeck process is $d\theta = -HA\theta(t)dt + \frac{1}{\sqrt{S}} HB\,dW(t)$. The resulting stationary covariance D' of preconditioned iterate averaging is the same as above:

$$\begin{aligned} D' &\approx \tfrac{1}{T}(HA)^{-1}(\tfrac{1}{\sqrt{S}}HB)(\tfrac{1}{\sqrt{S}}HB)^\top((HA)^\top)^{-1} \\ &= \tfrac{1}{TS} A^{-1}H^{-1}HBB^\top HH^{-1}A^{-1} \\ &= \tfrac{1}{TS} A^{-1}BB^\top A^{-1}. \end{aligned} \tag{31}$$

Both the stationary distribution and the autocorrelation matrix change as a result of the preconditioning, and these changes exactly cancel each other out.

This optimality of iterate averaging was first derived by Polyak and Juditsky (1992), using quasi-martingales. Our derivation is based on stronger assumptions, but is shorter.

6.2 Finite-Window Iterate Averaging for Posterior Sampling

Above, we used the continuous-time formalism to quickly rederive known results about the optimality of iterate averaging as an optimization algorithm. Next, as in Section 4, we will analyze iterate averaging as an algorithm for approximate posterior inference.

We will show that (under some optimistic assumptions), iterate averaging requires exactly N gradient calls to generate one sample drawn from the exact posterior distribution, where N is the number of observations. That is, there exist conditions under which iterate averaging generates one true posterior sample per pass over the data. This result is both exciting and discouraging; it implies that, since our assumptions are all optimistic and iterate averaging is known to saturate the Cramér-Rao bound, *no* black-box stochastic-gradient MCMC algorithm can generate samples in time sublinear in the number of data points.

In addition to Assumptions 1–4 of Section 3, we need an additional assumption for our theoretical considerations to hold:

Assumption 5 *Assume that the sample size N is large enough that the Bernstein-von Mises theorem (Le Cam, 1986) applies (hence the posterior is Gaussian). Also assume that the observed dataset was drawn from the model $p(y\,|\theta)$ with parameter $\theta = 0$. Then $A = BB^\top$, that is, the Fisher information matrix equals the Hessian.*

This simplifies equation 13:

$$A\Sigma + \Sigma A = \tfrac{\epsilon}{S} BB^\top \overset{A5}{\Longrightarrow} \Sigma = \frac{\epsilon}{2S}\mathbf{I}. \tag{32}$$

Algorithm 1 The Iterate Averaging Stochastic Gradient sampler (IASG)

input: averaging window $T = N/S$, number of samples M, input for SGD.
for $t = 1$ *to* $M * T$ **do**
  $\theta_t = \theta_{t-1} - \epsilon\, \hat{g}_S(\theta_{t-1})$; // perform an SGD step;
  if $t \bmod T = 0$ **then**
    $\mu_{t/T} = \frac{1}{T} \sum_{t'=0}^{T-1} \theta_{t-t'}$; // average the T most recent iterates
  end
end
output: return samples $\{\mu_1, \ldots, \mu_M\}$.

That is, the sampling distribution of SGD is isotropic.

Stationary distribution. We will consider the sampling distribution of the *average* of T successive samples from the stationary SGD process with step size ϵ. Going to the continuous-time OU formalism, we show in Appendix G that the stationary covariance of the iterate averaging estimator defined in Eq. 30 is

$$D = \frac{1}{ST} A^{-1} + \frac{1}{\epsilon S T^2} U \Lambda^{-2} (e^{-\epsilon T \Lambda} - \mathbf{I}) U^\top, \tag{33}$$

where U is orthonormal, Λ is diagonal, and $U \Lambda U^\top = A$ is the eigendecomposition of the Hessian A. We have previously assumed that the posterior has covariance $\frac{1}{N} A^{-1}$. Thus, to leading order in the ratio $1/(\epsilon T \Lambda)$, the stationary distribution of fixed-window iterate averaging is a scaled version of the posterior.

If we choose $T = N/S$, so that we average the iterates of a single pass through the dataset, then the iterate-averaging sampling distribution will have approximately the same covariance as the posterior,

$$\begin{aligned} D^\star &= \frac{1}{N} A^{-1} + \frac{S}{\epsilon} \frac{1}{N^2} U \Lambda^{-2} (e^{-\frac{\epsilon}{S} N \Lambda} - \mathbf{I}) U^\top \\ &= \frac{1}{N} U \Lambda^{-1} \left(\mathbf{I} + \frac{S}{\epsilon} \frac{1}{N} \Lambda^{-1} (e^{-\frac{\epsilon}{S} N \Lambda} - \mathbf{I}) \right) U^\top. \end{aligned} \tag{34}$$

$D^\star$ and A^{-1} have identical eigenvectors, and their eigenvalues differ by a factor that goes to zero as $\frac{\epsilon}{S}$ becomes large. Conversely, as $\frac{\epsilon}{S}$ approaches zero, all of these eigenvalues approach $\frac{\epsilon}{2S}$ as in Eq. 32. (This can be shown by taking a second-order Maclaurin approximation of $e^{-\frac{\epsilon N}{S} \Lambda}$.)

Our analysis gives rise to the Iterate Averaging Stochastic Gradient sampler (IASG), described in Algorithm 1. We now investigate its approximation error and efficiency.

Approximation error, step size, and minibatch size. We now focus on the correction terms that lead to deviations between the iterate averaging estimator's covariance D^* and the posterior covariance $\frac{1}{N} A^{-1}$.

The analysis above tells us that we can ignore these correction terms if we choose a large enough $\frac{\epsilon}{S}$. But the analysis in previous chapters assumes that $\frac{\epsilon}{S}$ is small enough that assumptions 1–4 hold. These considerations are in tension.

When is $\frac{\epsilon}{S}$ "large enough"? Eq. 34 shows that the relative error is largest in the direction of the smallest eigenvalue $\lambda_{\min} \equiv \min_k \Lambda_{kk}$ of A (corresponding to the least-constrained direction in the posterior). We will focus our analysis on the relative error in this direction, which is given by

$$\mathrm{err}_{\max} \equiv \frac{S}{\epsilon} \frac{1}{N \lambda_{\min}} (e^{-\frac{S}{\epsilon} N \lambda_{\min}} - 1).$$

We are given N and $\lambda_{\min}$, but we can control $\frac{\epsilon}{S}$. To make $\mathrm{err}_{\max}$ small, we must therefore choose $\frac{\epsilon}{S} > \frac{c}{N \lambda_{\min}}$ for some constant c. So larger datasets and lower-variance posteriors let us use smaller stepsizes.

It is hard to say in general how small $\frac{\epsilon}{S}$ needs to be to satisfy assumptions 1–4. But if assumption 4 is satisfied (i.e., the cost is approximately quadratic), then assumption 3 (no discretization error) cannot hold if $\epsilon > \frac{2}{\lambda_{\max}}$. This is the step size at which the discretized noise-free gradient descent process becomes unstable for quadratic costs. (We define $\lambda_{\max} \equiv \max_k \Lambda_{kk}$ analogous to $\lambda_{\min}$.)

Combining this observation with the analysis above, we see that this iterate averaging scheme will not work unless

$$\frac{2}{S \lambda_{\max}} > \frac{\epsilon}{S} > \frac{c}{N \lambda_{\min}} \Rightarrow \frac{2}{c} > \frac{S}{N} \frac{\lambda_{\max}}{\lambda_{\min}}.$$

That is, we need the dataset size N to be large enough relative to the condition number $\frac{\lambda_{\max}}{\lambda_{\min}}$ of the Hessian A if this simple iterate-averaging scheme is to generate good posterior samples. If the condition number is large relative to N, then it may be necessary to replace the scalar step size ϵ with a preconditioning matrix $H \approx A^{-1}$ to reduce the effective condition number of the Hessian A.

Efficiency. Next, we theoreticaly analyze the efficiency with which iterate averaging can draw samples from the posterior, and compare this method to other approaches. We assume that the cost of analyzing a minibatch is proportional to S. We have shown above that we need to average over $T = N/S$ samples of SGD to create a sample of iterate averaged SGD. Since this averaging induces a strong autocorrelation, we can only use every Tth sample of the chain of averaged iterates. Furthermore, every sample of SGD incurs a cost at least proportional to D where D is the dimensionality of θ. This means that we cannot generate an independent sample from the posterior in less than $O(S * T * D) = O(ND)$ time; we must analyze at least N observations per posterior sample.

We compare this result with the more classical strategy of estimating the posterior mode via Newton's method (which has an $O(ND^2 + D^3)$ cost) and then estimating the posterior covariance by computing the inverse-Hessian at the mode, again incurring an $O(ND^2 + D^3)$ cost. By contrast, getting an unbiased full-rank estimate of the covariance using MCMC requires generating at least D samples, which again costs $O(ND^2)$. If $N > D$, then this is within a constant cost of the classical approach.

However, it is conceivable that Polyak averaging (Section 6.2) could be used to estimate the first few principal components of the posterior relatively quickly (i.e., in $O(ND)$ time). This corresponds to finding the smallest principal components of A, which cannot be done efficiently in general. A related question is investigated experimentally in Section 7.2.

The analysis above implies an upper bound on the efficiency of stochastic-gradient MCMC (SGMCMC) methods. The argument is this: given that assumptions 1–5 hold, suppose that there exists an SGMCMC method that, for large N, is able to generate effectively independent posterior samples using $d < O(N)$ operations. Then, if we wanted to estimate the posterior mode, we could simply average some large number M of those samples to obtain an estimator whose covariance would be $\frac{1}{M} A^{-1}$. This approach would require dM operations, whereas iterate averaging would require $O(MN) > dM$ operations to obtain an estimator with the same covariance. But this contradicts the result of Polyak and Juditsky (1992) that *no* stochastic-gradient-oracle algorithm can

Method	Wine	Skin	Protein
constant SGD	18.7	0.471	1000.9
constant SGD-d	14.0	0.921	678.4
constant SGD-f	0.7	0.005	1.8
SGLD (Welling and Teh, 2011)	2.9	0.905	4.5
SGFS-d (Ahn et al., 2012)	12.8	0.864	597.4
SGFS-f (Ahn et al., 2012)	0.8	0.005	1.3
BBVI (Kucukelbir et al., 2015)	44.7	5.74	478.1

Table 1: KL divergences between the posterior and stationary sampling distributions applied to the data sets discussed in Section 7.1. To estimate the KL divergence, we fitted a multivariate Gaussian to the iterates of our sampling algorithms and used a Laplace approximation for the posterior. We compared constant SGD without preconditioning and with diagonal (-d) and full rank (-f) preconditioning against Stochastic Gradient Langevin Dynamics and Stochastic Gradient Fisher Scoring (SGFS) with diagonal (-d) and full rank (-f) preconditioning, and BBVI.

outperform iterate averaging. Thus, the optimality of iterate averaging as an optimization algorithm, taken with assumptions 1–5, implies that *no SGMCMC algorithm can generate posterior samples in sublinear time*[2].

This argument relies on assumptions 1–5 being true, but one can easily construct scenarios in which they are violated. However, these assumptions are all optimistic; there seems (to us) little reason to think that problems that violate assumptions 1–5 will be easier than those that do not.

7. Experiments

We test our theoretical assumptions from Section 3 and find good experimental evidence that they are reasonable in some settings. We also investigate iterate averaging and show that the assumptions outlined in 6.2 result in samples from a close approximation to the posterior. We also compare against other approximate inference algorithms, including SGLD (Welling and Teh, 2011), NUTS (Hoffman and Gelman, 2014), and black-box variational inference (BBVI) using Gaussian reparametrization gradients (Kucukelbir et al., 2015). In Section 7.3 we show that constant SGD lets us optimize hyperparameters in a Bayesian model.

7.1 Confirming the Stationary Distribution's Covariance

In this section, we confirm empirically that the stationary distributions of SGD with KL-optimal constant learning rates are as predicted by the Ornstein-Uhlenbeck process.

Real-world data. We first considered the following data sets.

- The *Wine Quality Data Set*[3], containing $N = 4,898$ instances, 11 features, and one integer output variable (the wine rating).

2. At least, not without exploiting additional knowledge about the source of gradient noise as do methods like SVRG and SAGA (Johnson and Zhang, 2013; Defazio et al., 2014).
3. P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis, 'Wine Quality Data Set', UCI Machine Learning Repository.

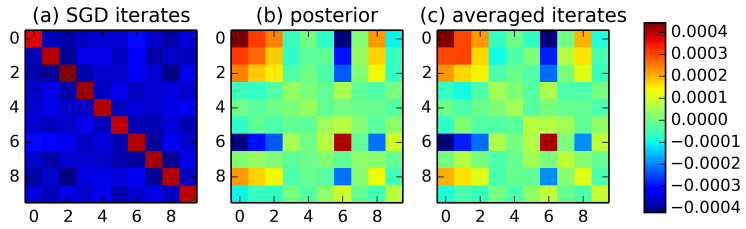


Figure 3: Iterate averaging on linear regression, where we generated artificial data as generated from the model. (a) shows the empirical covariance of the iterates of SGD, whereas (c) shows the averaged iterates with optimally chosen time window. The resulting covariance strongly resembles the true posterior covariance in (b). This shows that iterate averaging may result in posterior sampling.

- A data set of *Protein Tertiary Structure*[4], containing $N = 45,730$ instances, 8 features and one output variable.
- The *Skin Segmentation Data Set*[5], containing $N = 245,057$ instances, 3 features, and one binary output variable.

We applied linear regression on data sets 1 and 2 and applied logistic regression on data set 3. We rescaled the feature to unit length and used a mini-batch of size $S = 100$, $S = 100$ and $S = 10000$ for the three data sets, respectively. The quadratic regularizer was 1. The constant learning rate was adjusted according to Eq. 15.

Fig. 1 shows two-dimensional projections of samples from the posterior (blue) and the stationary distribution (cyan), where the directions were chosen two be the smallest and largest principal component of the posterior. Both distributions are approximately Gaussian and centered around the maximum of the posterior. To check our theoretical assumptions, we compared the covariance of the sampling distribution (yellow) against its predicted value based on the Ornstein-Uhlenbeck process (red), where very good agreement was found. Since the predicted covariance is based on approximating SGD as a multivariate Ornstein-Uhlenbeck process, we conclude that our modeling assumptions are satisfied to a very good extent. Since the wine dataset is smaller than the skin segmentation data set, it has a broader posterior and therefore requires a larger learning rate to match it. For this reason, discretization effects play a bigger role and the stationary distribution of preconditioned SGD on wine does not exactly match the posterior. The unprojected 11-dimensional covariances on wine data are also compared in Fig. 2. The rightmost column of Fig. 1 shows the sampling distributions of black box variational inference (BBVI) using the reparametrization trick (Kucukelbir et al., 2015). Our results show that the approximation to the posterior given by constant SGD is not worse than the approximation given by BBVI.

We also computed KL divergences between the posterior and stationary distributions of various algorithms: constant SGD with KL-optimal learning rates and preconditioners, Stochastic Gradient Langevin Dynamics, Stochastic Gradient Fisher Scoring (with and without diagonal approximation)

4. Prashant Singh Rana, 'Protein Tertiary Structure Data Set', UCI Machine Learning Repository.
5. Rajen Bhatt, Abhinav Dhall, 'Skin Segmentation Dataset', UCI Machine Learning Repository.

and BBVI. For SG Fisher Scoring, we set the learning rate to ϵ^* of Eq. 15, while for Langevin dynamics we chose the largest rate that yielded stable results ($\epsilon = \{10^{-3}, 10^{-6}, 10^{-5}\}$ for data sets 1, 2 and 3, respectively). Table 1 summarizes the results. We found that constant SGD can compete in approximating the posterior with the MCMC algorithms under consideration. This suggests that the most important factor is not the artificial noise involved in scalable MCMC, but rather the approximation of the preconditioning matrix.

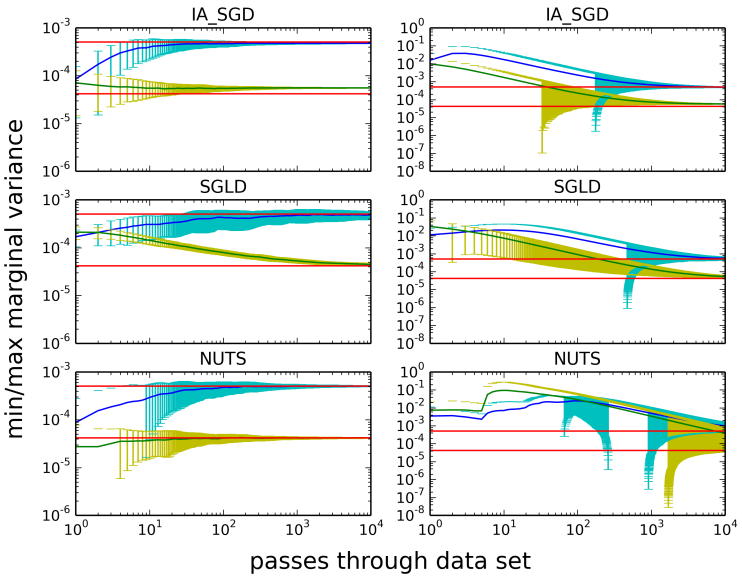


Figure 4: Convergence speed comparison between IASG (top), SGLD (middle), and NUTS (bottom) on linear regression. The plots show minimal (yellow) and maximal (blue) posterior marginal variances, respectively, as a function of iterations, measured in units of passes through the data. Error bars denote one standard deviation. Red solid lines show the ground truth. Left plots were initialized in the posterior maximum, whereas in the right column, we initialized randomly.

7.2 Iterate Averaging as Approximate MCMC

In the following, we show that under the assumptions specified in Section 6.2, iterate averaging with a constant learning rate and fixed averaging window results in samples from the posterior.

Synthetic data. In order to strictly satisfy the assumptions outlined in Section 6.2, we generated artificial data that came from the model. We chose a linear regression model with a Gaussian prior with precision $\lambda = 1$. We first generated $N = 10,000$ covariates by drawing them from a $D = 10$

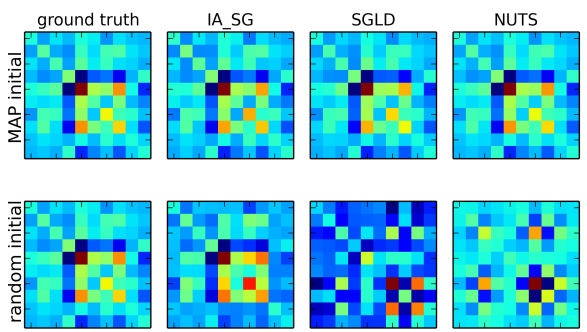


Figure 5: Posterior covariances as estimated by different methods, see also Fig. 4. The top row shows results where we initialized the samplers in the maximum posterior mode. The bottom rows were initialized randomly. For MAP initialization, all samplers find a good estimate of the posterior. When initializing randomly, IASG performs better than NUTS and SGLD.

dimensional Gaussian with unit covariance. We drew the true weight vector from the prior. We then generated the corresponding response variables from the linear regression model.

In a first experiment, we confirmed that iterate averaging may result in samples from the exact posterior, shown in Fig. 3. The left panel shows the empirical covariance matrix of the iterates of SGD. The middle and right panel show the posterior covariance and empirical covariance of the averaged iterates, respectively. As predicted by our theory, there is a very strong resemblance which demonstrates that constant-rate iterate averaging with the right rates and averaging windows may actually result in samples from the posterior. To generate this plot, we then ran constant SGD with a constant learning rate $\epsilon = 0.005$ for $10 * D * N = 10^6$ iterations and a minibatch size $S = 1$ and used an averaging window of $N/S = 10^4$, as predicted by the theory presented earlier in order to achieve samples from the posterior.

Next, we analyzed the convergence speed of Iterate-Averaged Stochastic Gradients (IASG, see Algorithm 1) compared to related methods. Fig. 4 shows these comparisons for three modern scalable MCMC algorithms: SGLD (Welling and Teh, 2011) and NUTS (Hoffman and Gelman, 2014). We investigated how quickly these algorithms could give us estimates of the posterior covariances. To better visualize the convergence behavior, we focussed on diagonal entries of the posterior covariance, the marginal variances. As before, the data were generated from the model such the theoretical assumptions of 6.2 applied. We then ran the samplers for up to $10^3 * D$ effective passes through the data. For IASG and SGLD we used a minibatch size of $S = 10$ and an averaging window of $N/S = 1000$. The constant learning rate of IASG was $\epsilon = 0.003$ and for SGLD we decreased the learning rate according to the Robbins-Monro schedule of $\epsilon_t = \frac{\epsilon_0}{\sqrt{1000+t}}$ where we found $\epsilon_0 = 10^{-3}$ to be optimal. NUTS automatically adjusts its learning rate and uses non-stochastic gradients.

The left column of Fig. 4 shows the results of these experiments. We found the convergence speeds of the samplers to be highly dependent on whether we optimized the samplers in the maximum posterior mode (termed MAP-initialization: this partially eleminates the initial bias and burn-in phase) or whether the samplers were initialized randomly, as in a real-world application. Thus, we show both results: while the the right column shows random initializations, the left one shows MAP-initialization. The rows of Fig. 4 show results of IASG (top), SGLD (middle), and NUTS (bottom). Each entry shows the smallest and largest marginal variance of the posterior over iterations, as estimated from these methods, where we excluded the first 10 iterations to avoid large biases. We also give the standard deviations for these estimates based on 100 independent Markov chains. The solid red lines show the ground truth of these marginal variances. Fig. 5 shows additional results on the same experiments, where we display the posterior estimates of the three different samplers under the two different initializations.

We found that in both initializations, IASG can find a fast approximate solution to the posterior. It uses stochastic gradients which gives it a competitive advantage over NUTS (which uses full gradients) in particular in the early search phase of the sampler. Langevin dynamics behaves similarly at early iterations (it also employs stochastic gradients). However, compared to IASG, we see that the Langevin algorithm has a much larger standard error when estmating the posterior covariances. This is because it uses a decreasing Robbins-Monro learning rate that slows down equilibration at long times. In contrast, IASG uses a constant learning rate and therefore converges fast. Note that in practice a large variance may be as bad as a large bias, especially if posterior estimates are based on a single Markov chain. This is also evident in Fig. 5, which shows that for random initialization, both NUTS and SGLD reveal less of the structure of the true posterior covariance compared to IASG.

When initialized in the posterior maximum, we see that all algorithms perform reasonably well (though SGLD's estimates are still highly variable even after $10,000$ sweeps through the dataset). IASG converges to a stable estimate much faster than SGLD or NUTS, but produces slightly biased estimates of the smallest variance.

7.3 Optimizing Hyperparameters

We test the hypothesis of Section 4.2, namely that constant SGD as a variational algorithm gives rise to a variational EM algorithm where we jointly optimize hyperparameters using gradients while drawing samples form an approximate posterior. To this end, we experimented with a Bayesian multinomial logistic (a.k.a. softmax) regression model with normal priors. The negative log-joint is

$$\begin{aligned}\mathcal{L} \equiv -\log p(y, \theta|x) = \tfrac{\lambda}{2}\textstyle\sum_{d,k}\theta_{dk}^2 - \tfrac{DK}{2}\log(\lambda) + \tfrac{DK}{2}\log 2\pi \\ + \textstyle\sum_n \log \sum_k \exp\{\sum_d x_{nd}\theta_{dk}\} - \sum_d x_{nd}\theta_{dy_n},\end{aligned} \tag{35}$$

where $n \in \{1, \ldots, N\}$ indexes examples, $d \in \{1, \ldots, D\}$ indexes features and $k \in \{1, \ldots, K\}$ indexes classes. $x_n \in \mathbb{R}^D$ is the feature vector for the nth example and $y_n \in \{1, \ldots, K\}$ is the class for that example. Eq. 35 has the degenerate maximizer $\lambda = \infty$, $\theta = 0$, which has infinite posterior density which we hope to avoid in our approach.

Real-world data. In all experiments, we applied this model to the MNIST dataset ($60,000$ training examples, $10,000$ test examples, 784 features) and the cover type dataset ($500,000$ training examples, $81,012$ testing examples, 54 features).

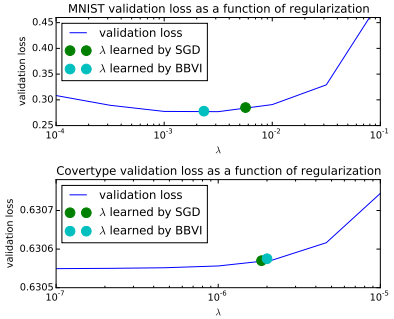


Figure 6: Validation loss as a function of L2 regularization parameter λ. Circles show the values of λ that were automatically selected by SGD and BBVI.

Fig. 6 shows the validation loss achieved by maximizing equation 35 over θ for various values of λ. This gives rise to the continuous blue curve. The value for constant SGD was obtained using Eq. 22, hence using constant learning rates for θ and decreasing learning rates for λ. BBVI was carried out by optimizing a variational lower bound in mean-field variational inference, and optimizing hyperparameters based on this lower bound.

The results suggest that BBVI and constant SGD yield similar results. Thus, constant SGD can be used as an inexpensive alternative to cross-validation or other VEM methods for hyperparameter selection.

8. Conclusions

In this paper, we built on a stochastic process perspective of stochastic gradient descent and various extensions to derive several new results. Under specified assumptions, SGD is approximated by a multivariate Ornstein-Uhlenbeck process, which possesses an analytic solution. We computed the stationary distribution of constant SGD and analyzed its properties.

We analyzed SGD together with several extensions, such as momentum, preconditioning, and iterate averaging. The shape of the stationary distribution is controlled by the parameters of the algorithm such as the constant learning rate, preconditioning matrix, or averaging period. We can thus tune these parameters to minimize the Kullback-Leibler divergence between the stationary distribution and a Bayesian posterior. This view uses these stochastic optimization algorithms as approximate inference. We also analyzed stochastic gradient Langevin dynamics and stochastic gradient Fisher scoring and were able to analyze approximation errors for these algorithms.

The Bayesian view on constant-rate SGD allows us to use this algorithm as a new variational EM algorithm. We suggested and tested a double SGD scheme which uses decreasing learning rates on the hyperparameters and constant learning rates on the main parameters. We showed that this is both easy to implement and prevents us from finding degenerate solutions; it is a cheap alternative to cross-validation for many complex models.

Last, our analysis suggests the many similarities between sampling and optimization algorithms that can be explored using the stochastic process perspective. A future direction might be to explore

similarities in the noise characteristics of black-box variational inference algorithms and Langevin-type MCMC. Further exploring the use of iterate averaging as a Bayesian algorithm is another interesting avenue for further studies.

Acknowledgments

We would like to thank Yingzhen Li and Thomas Hofmann for their valuable feedback on our manuscript.

Appendix A. Examples: Ornstein-Uhlenbeck Formalism

Let us illustrate the Ornstein-Uhlenbeck formalism based on two simple examples. First, consider the following quadratic loss,

$$\mathcal{L}(\theta) = -\tfrac{1}{2N}\sum_{n=1}^{N}||x_n - \theta||^2. \tag{36}$$

Let us define $\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n$ as the empirical mean of the data points. The gradient is $g(\theta) = (\bar{x}-\theta)$, and the stochastic gradient is $\hat{g}(\theta) = \frac{1}{S}\sum_{s=1}^{S}(x_s-\theta)$. Because the gradient is linear in x, the noise covariance is just the covariance of the data: $\Sigma_x \equiv C/S = \frac{1}{S^2}\mathbb{E}\left[\sum_{s,s'}(x_s-\bar{x})(x_{s'}-\bar{x})^\top\right] = \frac{1}{S}\mathbb{E}[(x_n-\bar{x})(x_n-\bar{x})^\top]$. We can shift the parameter $\theta \to \theta + \bar{x}$, resulting in $\theta^* = 0$. Note that the Hessian $A \equiv \mathbf{I}$ is just unity. According to Eq. 38,

$$q(\theta) \;\propto\; \exp\left\{-\tfrac{S}{2\epsilon}(\theta-\bar{x})^\top\Sigma_x^{-1}(\theta-\bar{x})\right\}.$$

as the resulting stationary distribution. Next, consider linear regression, where we minimize

$$\mathcal{L}(\theta) \;=\; -\tfrac{1}{2N}\textstyle\sum_n (y_n - x_n^\top\theta)^2. \tag{37}$$

We can write the stochastic gradient as $\hat{g} = \hat{A}\theta - \hat{\mu}$, where $\hat{\mu} = \frac{1}{S}\sum_s x_s y_s$ and $\hat{A} = \frac{1}{S}\sum_s x_s x_s^\top$ are estimates based on a mini-batch of size S. The sampling noise covariance is $C(\theta) = \mathbb{E}[(\hat{g}-g)(\hat{g}-g)^\top] = \mathbb{E}[\hat{g}\hat{g}^\top] - gg^\top$, where $\mathbb{E}[\hat{g}\hat{g}^\top] = \mathbb{E}[(\hat{A}\theta-\hat{\mu})(\hat{A}\theta-\hat{\mu})^\top]$. We see that the noise covariance is quadratic, but unfortunately it cannot be further simplified.

Fig. 1 shows the objective function of linear regression (blue) and the sampling distribution of stochastic gradient descent (yellow) on simulated data. We see that both distributions do not coincide, because the sampling distribution is also affected by the noise covariance.

Appendix B. Stationary Covariance

The Ornstein-Uhlenbeck process has an analytic solution in terms of the stochastic integral (Gardiner et al., 1985),

$$\theta(t) = \exp(-At)\theta(0) + \sqrt{\tfrac{\epsilon}{S}}\int_0^t \exp[-A(t-t')]B dW(t') \tag{38}$$

Following Gardiner's book and using $A = A^\top$, we derive an algebraic relation for the stationary covariance of the multivariate Ornstein-Uhlenbeck process. Define $\Sigma = \mathbb{E}[\theta(t)\theta(t)^\top]$. Using the

formal solution for $\theta(t)$ given in the main paper, we find

$$\begin{aligned}
A\Sigma + \Sigma A &= \tfrac{\epsilon}{S}\int_{-\infty}^{t} A\exp[-A(t-t')]BB^\top\exp[-A(t-t')]dt' \\
&+ \tfrac{\epsilon}{S}\int_{-\infty}^{t}\exp[-A(t-t')]BB^\top\exp[-A(t-t')]dt'A \\
&= \tfrac{\epsilon}{S}\int_{-\infty}^{t}\frac{d}{dt'}\left(\exp[-A(t-t')]BB^\top\exp[-A(t-t')]\right) \\
&= \tfrac{\epsilon}{S}BB^\top.
\end{aligned}$$

We used that the lower limit of the integral vanishes by the positivity of the eigenvalues of A.

Appendix C. Reverse KL Divergence Setup

It is interesting to also consider the case of trying to minimize the reverse KL divergence, i.e. $KL(f||q)$ instead of $KL(q||f)$. One might assume that this is possible since both the variational distribution and the posterior are assumed to be Gaussian. This turns out to lead only to a feasible algorithm in the special case where the Hessian in the optimum A and the stationary covariance Σ commute. In more detail, the KL-divergence between the posterior and the stationary distribution is (up to constants):

$$\begin{aligned}
KL(f||q) &= \mathbb{E}_f[\log f] - \mathbb{E}_f[\log q] \\
&= \tfrac{1}{2}\mathbb{E}_f\left[\theta^\top\Sigma^{-1}\theta\right] + \tfrac{1}{2}\log|\Sigma| + \text{const.} \\
&= \tfrac{1}{2N}\mathrm{Tr}(A^{-1}\Sigma^{-1}) + \tfrac{1}{2}\log|\Sigma| + \text{const.}
\end{aligned} \tag{39}$$

While we were able to derive this divergence, it turns out that we cannot in general eliminate its dependence in the stationary covariance and re-express it in terms of $BB^\top$, using Eq. 13. However, if A and Σ commute, we can proceed as follows:

$$\begin{aligned}
\mathrm{Tr}(A^{-1}\Sigma^{-1}) &= \mathrm{Tr}((\Sigma A)^{-1}) \\
&\overset{A\Sigma=\Sigma A}{=} 2\mathrm{Tr}((\Sigma A + A\Sigma)^{-1}) \\
&\overset{Eq.\ 13}{=} \tfrac{2S}{\epsilon}\mathrm{Tr}((BB^\top)^{-1})
\end{aligned} \tag{40}$$

Following the logic of Theorems 1 and 2, we find the following result for the optimal learning rate:

$$\epsilon^* = \frac{2S}{ND}\mathrm{Tr}((BB^\top)^{-1}). \tag{41}$$

Interestingly, when comparing Eq. 15 with Eq. 41, we find that the inverse of the trace of the noise covariance gets replaced by the trace of the inverse noise covariance. While $KL(q||f)$ suggests to choose the learning rate inversely proportional to the largest Eigenvalue of the noise, $KL(f||q)$ thus suggests to choose the learning rate proportional to the inverse of the smallest Eigenvalue of the noise. Both approaches have thus a different emphasis on how to fit the posterior mode, in a similar fashion as variational inference and expectation propagation. Note, however, that A and Σ rarely commute in practice, and thus $KL(q||f)$ is the only viable option.

Appendix D. SGD With Momentum

Here, we give more details on the deviations of the results on SGD with momentum. In order to compute the stationary distribution of Eq. 23, we need to solve the equations for the second moments:

$$\begin{aligned}
d\mathbb{E}[\theta\theta^\top] &= \mathbb{E}[d\theta\theta^\top + \theta d\theta^\top] \\
&= (\mathbb{E}[\mathrm{v}\theta^\top] + \mathbb{E}[\theta\mathrm{v}^\top])dt, && (42)\\
d\mathbb{E}[\theta\mathrm{v}^\top] &= \mathbb{E}[d\theta\mathrm{v}^\top + \theta d\mathrm{v}^\top] \\
&= \mathbb{E}[\mathrm{v}\mathrm{v}^\top]dt - \mu\mathbb{E}[\theta\mathrm{v}^\top]dt - \epsilon\mathbb{E}[\theta\theta^\top]Adt, && (43)\\
d\mathbb{E}[\mathrm{v}\theta^\top] &= \mathbb{E}[d\mathrm{v}\theta^\top + \mathrm{v}d\theta^\top] \\
&= \mathbb{E}[\mathrm{v}\mathrm{v}^\top]dt - \mu\mathbb{E}[\mathrm{v}\theta^\top]dt - \epsilon A\mathbb{E}[\theta\theta^\top]dt, && (44)\\
d\mathbb{E}[\mathrm{v}\mathrm{v}^\top] &= \mathbb{E}[d\mathrm{v}\mathrm{v}^\top + \mathrm{v}d\mathrm{v}^\top] + \mathbb{E}[d\mathrm{v}d\mathrm{v}^\top] \\
&= -2\mu\mathbb{E}[\mathrm{v}\mathrm{v}^\top]dt - \epsilon A\mathbb{E}[\theta\mathrm{v}^\top]dt - \epsilon\mathbb{E}[\mathrm{v}\theta^\top]Adt + \tfrac{\epsilon^2}{S}BB^\top dt. && (45)
\end{aligned}$$

In the last equation we used the fact that according to Ito's rule, there is an additional non-vanishing contribution due to the noise, $\mathbb{E}[d\mathrm{v}d\mathrm{v}^\top] = \frac{\epsilon^2}{S}\mathbb{E}[B\,dW dW^\top B^\top] = \frac{\epsilon^2}{S}BB^\top dt$. This contribution does not exist for the other correlators; for more details see e.g. (Gardiner et al., 1985).

Since we are looking for a stationary solution, we set the left hand sides of all equations to zero. Eq. 42 implies that $\mathbb{E}[\mathrm{v}\theta^\top] + \mathbb{E}[\theta\mathrm{v}^\top] = 0$, hence the cross-correlation between momentum and position is anti-symmetric in the stationary state. We can thus add Eqs. 43 and 44 to find $0 = d\mathbb{E}[\mathrm{v}\theta^\top + \theta\mathrm{v}^\top] = 2\mathbb{E}[\mathrm{v}\mathrm{v}^\top]dt - \epsilon A\mathbb{E}[\theta\theta^\top]dt - \epsilon\mathbb{E}[\theta\theta^\top]Adt$. Combining this with Eq. 45 yields

$$\begin{aligned}
\mathbb{E}[\mathrm{v}\mathrm{v}^\top] &= \tfrac{\epsilon}{2}\mathbb{E}[\theta\theta^\top]A + \tfrac{\epsilon}{2}A\mathbb{E}[\theta\theta^\top] \\
\mu\mathbb{E}[\mathrm{v}\mathrm{v}^\top] &= \tfrac{\epsilon^2}{2S}BB^\top - \underbrace{\tfrac{1}{2}\epsilon(A\mathbb{E}[\theta\mathrm{v}^\top] + \mathbb{E}[\mathrm{v}\theta^\top]A)}_{=0}.
\end{aligned}$$

Last, we show that the underbraced term is zero, which gives Eq. 25 in the main paper. Denote $\xi = \mathbb{E}[\theta\mathrm{v}^\top] = -\xi^\top$ which is antisymmetric due to Eq. 42. First of all, $A\xi + \xi^\top A$ is obviously symmetric. It is simultaneously antisymmetric due to the following calculation: $\sum_k A_{ik}\xi_{kj} + \sum_k \xi_{ik}^\top A_{kj}^\top = \sum_k A_{ik}\xi_{kj} - \sum_k \xi_{ki}A_{jk} = \sum_k A_{ik}\xi_{kj} - \sum_k A_{jk}\xi_{ki}$. This term swaps the sign as i and j are interchanged. Being both symmetric and antisymmetric, it is zero.

Appendix E. Stochastic Gradient Fisher Scoring

We start from the Ornstein-Uhlenbeck process with minibatch size $S = 1$,

$$\begin{aligned}
d\theta(t) &= -\epsilon HA\theta(t)dt + H\left[\epsilon B dW(t) + \sqrt{\epsilon}E dV(t)\right] \\
&\Leftrightarrow \\
d\theta(t) &= -A'\theta(t)dt + B'dW(t),
\end{aligned}$$

where we define $A' \equiv \epsilon HA$ and $B' \equiv H\sqrt{\epsilon^2 BB^\top + \epsilon EE^\top}$, and use the fact that the dynamics of $CdW(t) + DdV(t)$ are equivalent to $\sqrt{CC^\top + DD^\top}dW(t)$. Here we are using matrix square roots, so $B'B'^\top = \epsilon H(\epsilon BB^\top + EE^\top)H$.

As derived in the paper, the variational bound is (up to a constant)

$$KL \stackrel{c}{=} \frac{N}{2}\mathrm{Tr}(A\Sigma) - \tfrac{1}{2}\log(|\Sigma|).$$

To evaluate it, the task is to remove the unknown covariance Σ from the bound. To this end, as before, we use the identity for the stationary covariance $A'\Sigma + \Sigma A'^\top = B'B'^\top$. The criterion for the stationary covariance is equivalent to

$$HA\Sigma + \Sigma AH = \epsilon HBB^\top H + HEE^\top H.$$

We can further simplify this expression as follows:

$$\begin{aligned}
A\Sigma + H^{-1}\Sigma AH &= \epsilon BB^\top H + EE^\top H \\
\Rightarrow \mathrm{Tr}(A\Sigma) &= \frac{1}{2}\mathrm{Tr}(H(\epsilon BB^\top + EE^\top)).
\end{aligned}$$

As above, we can also reparameterize the covariance as $\Sigma = TH$, so that T does not depend on H:

$$\begin{aligned}
HA\Sigma + \Sigma AH &= \epsilon HBB^\top H + HEE^\top H \\
A\Sigma H^{-1} + H^{-1}\Sigma A &= \epsilon BB^\top + EE^\top \\
AT + T^\top A &= \epsilon BB^\top + EE^\top.
\end{aligned}$$

The KL divergence is therefore

$$\begin{aligned}
KL &= \frac{N}{2}\mathrm{Tr}(A\Sigma) - \frac{D}{2} - \frac{1}{2}\log(N|A|) - \frac{1}{2}\log|\Sigma| \\
&= \frac{N}{4}\mathrm{Tr}(H(\epsilon BB^\top + EE^\top)) - \frac{D}{2} - \frac{1}{2}\log(N|A|) - \frac{1}{2}\log|T| - \frac{1}{2}\log|H|, \quad (46)
\end{aligned}$$

which is the result we give in the main text.

Appendix F. Square Root Preconditioning

We analyze the case where we precondition with a matrix that is proportional to the square root of the diagonal entries of the noise covariance.

We define

$$G = \sqrt{\mathrm{diag}(BB^\top)}$$

as the diagonal matrix that contains square roots of the diagonal elements of the noise covariance. We use an additional scalar learning rate ϵ .

Theorem 7 (Taking square roots) *Consider SGD preconditioned with G^{-1} as defined above. Under the previous assumptions, the constant learning rate which minimizes KL divergence between the stationary distribution of this process and the posterior is*

$$\epsilon^* = \frac{2DS}{N\mathrm{Tr}(BB^\top G^{-1})}. \quad (47)$$

Proof We read off the appropriate KL divergence from the proof of Theorem 2 with $G^{-1} \equiv H$:

$$KL(q||f) \stackrel{c}{=} \frac{\epsilon N}{2S}\mathrm{Tr}(BB^\top G^{-1}) - \mathrm{Tr}\log(G) + \frac{D}{2}\log\frac{\epsilon}{S} - \frac{1}{2}\log|\Sigma|$$

Minimizing this KL divergence over the learning rate ϵ yields Eq. 47. ■

Appendix G. Iterage Averaging

We now prove our result for the covariance of the averaged iterates. We first need an identity for the non-equal-time covariance in the stationary state:

$$\mathbb{E}[\theta(t)\theta(s)^\top] = \begin{cases} \Sigma e^{-\epsilon A(s-t)} & t < s \\ e^{-\epsilon A(t-s)}\Sigma & t \geq s. \end{cases} \tag{48}$$

To derive it, one uses the formal solution of the Ornstein-Uhlenbeck process for $\theta(t)$ in combination with Eq. 29, see also (Gardiner et al., 1985) for more details. Note that for $t = s$, it simplifies to $\mathbb{E}[\theta(t)\theta(t)^\top] = \Sigma$, as one would expect.

We are averaging over T time steps. Going to the continuous-time OU formalism, we are interested in the following quantity, which is the equal-time covariance of the time-averaged iterates:

$$D \equiv \mathbb{E}\left[\left(\frac{1}{T}\int_0^T \theta(t)dt\right)\left(\frac{1}{T}\int_0^T \theta(s)ds\right)^\top\right].$$

This can be further broken down to two contributions:

$$\begin{aligned} D &= \frac{1}{T^2}\int_0^T\int_0^T \mathbb{E}[\theta(t)\theta(s)^\top]dsdt \\ &= \frac{1}{T^2}\int_0^T\int_0^t \mathbb{E}[\theta(t)\theta(s)^\top]dsdt + \frac{1}{T^2}\int_0^T\int_t^T \mathbb{E}[\theta(t)\theta(s)^\top]dsdt \end{aligned}$$

We now use the eigendecomposition $A = U\Lambda U^\top$ of the Hessian, the autocorrelation Eq. 48, as well as the identity $e^{cA} = Ue^{c\Lambda}U^\top$. The first term becomes

$$\begin{aligned} \frac{1}{T^2}\int_0^T\int_0^t \mathbb{E}[\theta(t)\theta(s)^\top]dsdt &= \frac{1}{T^2}\int_0^T\int_0^t e^{-\epsilon A(t-s)}\Sigma dsdt \\ &= \frac{1}{T^2}\int_0^T\int_0^t Ue^{-\epsilon\Lambda(t-s)}U^\top\Sigma dsdt \\ &= \frac{1}{\epsilon T^2}\int_0^T U\Lambda^{-1}(\mathbf{I} - e^{-\epsilon t\Lambda})U^\top\Sigma dt \\ &= \frac{1}{\epsilon T}A^{-1}\Sigma + \frac{1}{\epsilon^2 T^2}U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^\top\Sigma. \end{aligned} \tag{49}$$

The calculation for the second term goes analogously and yields

$$\frac{1}{T^2}\int_0^T\int_t^T \mathbb{E}[\theta(t)\theta(s)^\top]dsdt = \frac{1}{\epsilon T}\Sigma A^{-1} + \frac{1}{\epsilon^2 T^2}\Sigma U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^\top.$$

Both equations combined give us

$$\begin{aligned} D = &\frac{1}{\epsilon T}(A^{-1}\Sigma + \Sigma A^{-1}) \\ &+ \frac{1}{\epsilon^2 T^2}(U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^\top\Sigma + \Sigma U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^\top). \end{aligned} \tag{50}$$

When $\epsilon T\Lambda \gg 1$ (valid for sufficiently long averaging periods T), we obtain

$$D \approx \frac{1}{\epsilon T}(A^{-1}\Sigma + \Sigma A^{-1}),$$

which is Eq. 30 in the main text. We can also simplify the expression for $\Sigma = \frac{\epsilon}{2S}I$, as motivated in Section 6.2. In this case, Eq. 50 results in

$$D = \frac{1}{ST}A^{-1} + \frac{1}{\epsilon ST^2}(U\Lambda^{-2}(e^{-\epsilon T\Lambda} - \mathbf{I})U^\top).$$

This is exactly Eq. 33 in the main text.

References

Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1591–1598, July 2012.

Francis Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate o (1/n). In *Advances in Neural Information Processing Systems*, pages 773–781, 2013.

Louis Bachelier. *Théorie de la spéculation.* Gauthier-Villars, 1900.

Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer New York, 2006.

Léon Bottou. Online learning and stochastic approximations. *Online Learning in Neural Networks*, 17(9):25, 1998.

Nicolas Bouleau and Dominique Lepingle. *Numerical methods for stochastic processes*, volume 273. John Wiley & Sons, 1994.

Changyou Chen, Nan Ding, and Lawrence Carin. On the convergence of stochastic gradient MCMC algorithms with high-order integrators. In *Advances in Neural Information Processing Systems*, pages 2269–2277, 2015.

Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li, and Lawrence Carin. Bridging the gap between stochastic gradient MCMC and stochastic optimization. In *Artificial Intelligence and Statistics*, 2016.

Tianqi Chen, Emily B Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1683–1691, 2014.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

Alexandre Défossez and Francis Bach. Averaged least-mean-squares: Bias-variance trade-offs and optimal sampling distributions. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 205–213, 2015.

Aymeric Dieuleveut, Alain Durmus, and Francis Bach. Bridging the gap between constant step size stochastic gradient descent and markov chains. *arXiv preprint arXiv:1707.06386*, 2017.

Nan Ding, Youhan Fang, Ryan Babbush, Changyou Chen, Robert D Skeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems*, pages 3203–3211, 2014.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.

Nicolas Flammarion and Francis Bach. From averaging to acceleration, there is only a step-size. In *Proceedings of the International Conference on Learning Theory*, 2015.

Crispin W Gardiner et al. *Handbook of Stochastic Methods*, volume 4. Springer Berlin, 1985.

Zoubin Ghahramani and Matthew J Beal. Propagation algorithms for variational Bayesian learning. In *Advances in Neural Information Processing Systems*, pages 507–513, 2000.

Matthew D Hoffman and Andrew Gelman. The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

Michael Jordan, Zoubin Ghahramani, Tommi Jaakkola, and Lawrence Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *The International Conference on Learning Representations*, 2014.

Hendrik Anthony Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(4):284–304, 1940.

Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David Blei. Automatic variational inference in STAN. In *Advances in Neural Information Processing Systems*, pages 568–576, 2015.

Harold J Kushner and George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

Lucien Le Cam. Asymptotic methods in statistical decision theory. *Springer New York*, 1986.

Qianxiao Li, Cheng Tai, and Weinan E. Dynamics of stochastic gradient algorithms. *arXiv preprint arXiv:1511.06251v2*, 2015.

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2101–2110, 2017.

Lennart Ljung, Georg Ch Pflug, and Harro Walk. *Stochastic approximation and optimization of random systems*, volume 17. Birkhäuser, 2012.

Nicholas T Longford. A fast scoring algorithm for maximum likelihood estimation in unbalanced mixed models with nested random effects. *Biometrika*, 74(4):817–827, 1987.

Yi-An Ma, Tianqi Chen, and Emily B Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, 2015.

Dougal Maclaurin, David Duvenaud, and Ryan P Adams. Early stopping is nonparametric variational inference. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 1070–1077, 2016.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

Harry Nyquist. Thermal agitation of electric charge in conductors. *Physical Review*, 32:110–113, Jul 1928.

Manfred Opper and David Saad. *Advanced mean field methods: Theory and practice*. MIT press, 2001.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Boris Teodorovich Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics*, pages 814–822, 2014.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Francisco Ruiz, Michaelis Titsias, and David Blei. The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*, pages 460–468, 2016.

David J Sakrison. Efficient recursive estimation; application to estimating the parameters of a covariance function. *International Journal of Engineering Science*, 3(4):461–483, 1965.

Tim Salimans and David A Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Issei Sato and Hiroshi Nakagawa. Approximation analysis of stochastic gradient Langevin dynamics by using Fokker-Planck equation and Ito process. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 982–990, 2014.

Xiaocheng Shang, Zhanxing Zhu, Benedict Leimkuhler, and Amos J Storkey. Covariance-controlled adaptive Langevin thermostat for large-scale Bayesian sampling. In *Advances in Neural Information Processing Systems*, pages 37–45, 2015.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1139–1147, 2013.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5—RmsProp: Divide the Gradient by a Running Average of its Recent Magnitude. Coursera: Neural Networks for Machine Learning, 2012.

Panagiotis Toulis, Edoardo Airoldi, and Jason Rennie. Statistical analysis of stochastic gradient methods for generalized linear models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 667–675, 2014.

Panos Toulis, Dustin Tran, and Edoardo M Airoldi. Towards stability and optimality in stochastic gradient descent. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 1290–1298, 2016.

George E Uhlenbeck and Leonard S Ornstein. On the theory of the Brownian motion. *Physical Review*, 36(5):823, 1930.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

Bernard Widrow and Samuel D Stearns. Adaptive signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1985, 491 p.*, 1, 1985.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the 21st International Conference on Machine Learning*, page 116. ACM, 2004.

Journal of Machine Learning Research 18 (2017) 1-37 Submitted 4/17; Revised 9/17; Published 11/17

STORE: Sparse Tensor Response Regression and Neuroimaging Analysis

Will Wei Sun WSUN@BUS.MIAMI.EDU
Department of Management Science
University of Miami School of Business Administration
Miami, FL 33146, USA

Lexin Li LEXINLI@BERKELEY.EDU
Division of Biostatistics
University of California
Berkeley, CA 94720, USA

Editor: Animashree Anandkumar

Abstract

Motivated by applications in neuroimaging analysis, we propose a new regression model, Sparse TensOr REsponse regression (STORE), with a tensor response and a vector predictor. STORE embeds two key sparse structures: element-wise sparsity and low-rankness. It can handle both a non-symmetric and a symmetric tensor response, and thus is applicable to both structural and functional neuroimaging data. We formulate the parameter estimation as a non-convex optimization problem, and develop an efficient alternating updating algorithm. We establish a non-asymptotic estimation error bound for the actual estimator obtained from the proposed algorithm. This error bound reveals an interesting interaction between the computational efficiency and the statistical rate of convergence. When the distribution of the error tensor is Gaussian, we further obtain a fast estimation error rate which allows the tensor dimension to grow exponentially with the sample size. We illustrate the efficacy of our model through intensive simulations and an analysis of the Autism spectrum disorder neuroimaging data.

Keywords: Functional connectivity analysis, High-dimensional statistical learning, Magnetic resonance imaging, Non-asymptotic error bound, Tensor decomposition

1. Introduction

In this article, we study a class of high-dimensional regression models with a tensor response and a vector predictor. Tensor, a.k.a. multidimensional array, is now frequently arising in a wide range of scientific and business applications (Yuan and Zhang, 2016). Our motivation comes from neuroimaging analysis. One example is anatomical magnetic resonance imaging (MRI), where the data takes the form of a three-dimensional array, and image voxels correspond to brain spatial locations. Another example is functional magnetic resonance imaging (fMRI), where the goal is to understand brain functional connectivity that is encoded by a symmetric matrix, with rows and columns corresponding to brain regions, and entries corresponding to interaction between those regions. In both examples, it is of keen scientific interest to compare the scans of brains, or the brain connectivity patterns, between the subjects with neurological disorder and the healthy controls, after adjusting for additional covariates such as age and sex. Both can be more generally formulated as a regression problem, with image tensor or connectivity matrix serving as a response, and the group indicator and other covariates forming a predictor vector.

1.1 Our Proposal

We develop a general class of tensor response regression models, namely STORE, and embed two key sparse structures: element-wise sparsity and low-rankness. Both structures serve to greatly reduce the computational complexity of the estimation procedure. Meanwhile, both are scientifically plausible in plenty of applications, and have been widely employed in high-dimensional multivariate regressions (e.g., Tibshirani, 1996; Zou, 2006; Yuan and Lin, 2006; Peng et al., 2010; Zhou et al., 2013; Raskutti and Yuan, 2016). A unique feature of STORE is that, it can not only handle a non-symmetric tensor response, but also a symmetric tensor response, and thus is applicable to both structural and functional neuroimaging analysis.

We formulate the learning of STORE as a non-convex optimization problem, and accordingly develop an efficient alternating updating algorithm. Our algorithm consists of two major steps, and each step iteratively updates a subset of the unknown parameters while fixing the others. In Step 1, we reformulate the estimation as a sparse tensor decomposition problem and then employ a decomposition algorithm, the truncated tensor power method (Sun et al., 2017), for solution. In Step 2, we utilize the bi-convex structure of the problem to obtain a closed-form solution.

We carry out a non-asymptotic theoretical analysis for the actual estimator obtained from our optimization algorithm. Based upon a set of our newly developed techniques to tackle the non-convexity in estimation, we obtain an explicit error bound of the actual minimizer. Specifically, let $\mathcal{E}_i, i = 1, \ldots, n$, denote the error tensor, and $\mathbf{\Theta}^*$ denote the set of all true parameters. Given an initial parameter with an initialization error ϵ, the finite sample error bound of the t-th step solution $\widehat{\mathbf{\Theta}}^{(t)}$ consists of two parts:

$$D\left(\widehat{\mathbf{\Theta}}^{(t)}, \mathbf{\Theta}^*\right) \leq \underbrace{\kappa^t \epsilon}_{\text{computational error}} + \underbrace{\frac{1}{1-\kappa} \max\left\{C \cdot \eta\left(\frac{1}{n}\sum_{i=1}^{n} \mathcal{E}_i, s\right), \frac{\widetilde{C}}{\sqrt{n}}\right\}}_{\text{statistical error}},$$

with a high probability. Here $\kappa \in (0,1)$ is a contraction coefficient, C and $\widetilde{C}$ are some positive constants, and $\eta\left(n^{-1}\sum_{i=1}^{n} \mathcal{E}_i, s\right)$ represents the s-sparse spectral norm of the averaged error tensor; see (8) for a formal definition of this norm. This error bound portrays the estimation error in each iteration, and reveals an interesting interplay between the computational efficiency and the statistical rate of convergence. Note that the computational error decays geometrically with the iteration number t, whereas the statistical error remains the same when t grows. Therefore, this bound provides a meaningful guidance for the maximal number of iterations T. That is, we stop when the computational error becomes dominated by the statistical error. The resulting estimator falls within the statistical precision of the true parameter $\mathbf{\Theta}^*$. Additionally, this finite sample error bound provides a general theoretical guarantee of our estimator, and the result holds for any distribution of the error tensor $\mathcal{E}_i$, by noting that it relies on $\mathcal{E}_i$ only through its sparse spectral norm $\eta\left(n^{-1}\sum_{i=1}^{n} \mathcal{E}_i, s\right)$. In Sections 4.3 and 4.4, we obtain explicit forms of statistical errors when the distribution of

©2017 Will Wei Sun and Lexin Li.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-203.html.

$\mathcal{E}_i$ is available. In particular, when the third-order error tensor $\mathcal{E}_i \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, $i = 1, \cdots, n$, follows an i.i.d. Gaussian distribution, we have

$$D\left(\widehat{\mathbf{\Theta}}^{(T)}, \mathbf{\Theta}^*\right) = O_p\left(\sqrt{\frac{s^3 \log(d_1 d_2 d_3)}{n}}\right),$$

where s is the cardinality parameter of the decomposed components in the tensor coefficients. This fast estimation error rate allows the tensor dimension to grow exponentially with the sample size. When the order of the tensor is one, STORE reduces to the d-dimensional vector regression and our statistical error reduces to $O_p(\sqrt{s \log d/n})$, which is known to be minimax optimal (Wang et al., 2014).

1.2 Related Works and Our Contributions

Our work is related to but also clearly distinct from a number of recent development in high-dimensional statistical models involving tensor data.

The first is a class of tensor decomposition methods (Chi and Kolda, 2012; Liu et al., 2012; Anandkumar et al., 2014a,b; Yuan and Zhang, 2016; Sun et al., 2017). Tensor decomposition is an *unsupervised* learning method that aims to find the best low-rank approximation of a single tensor. Our proposed tensor response regression, however, is a *supervised* learning method, which aims to estimate the coefficient tensor that characterizes the association between the tensor response and the vector predictor. Although we utilize the sparse tensor decomposition step of Sun et al. (2017) as part of our estimation algorithm, our objective and technical tools involved are completely different. Particularly, because we work with *multiple* tensor samples, the consistency of our estimator is indexed by both the tensor dimension and the sample size. This is different from the sparse tensor decomposition estimator in Sun et al. (2017) that works with a *single* tensor only. Consequently, new large deviation inequalities are needed and derived in our theoretical investigation. Moreover our algorithm is not restricted to any particular decomposition procedure but can be coupled with other decomposition solutions, e.g., Chi and Kolda (2012); Liu et al. (2012).

The second related line of research tackles tensor regression where the response is a scalar and the predictor is a tensor (Zhou et al., 2013; Wang and Zhu, 2017; Yu and Liu, 2016). In those papers, a low-rank structure is often imposed on the coefficient tensor, which is similar to the low-rank principle we also employ. However, they all treated the tensor as a *predictor*, whereas we treat it as a *response*. This difference in modeling approach leads to different focus and interpretation. The tensor predictor regression focuses on understanding the change of a biological outcome as the tensor image varies, while the tensor response regression aims to study the change of the image as the predictors such as the disease status and age vary. When it comes to theoretical analysis, the two models involve utterly different techniques. In a way, their difference is in analogous to that between multi-response regression and multi-predictor regression in the classical vector-valued regression context.

The third and more relevant line of works directly studies regression with a tensor response (Zhu et al., 2009; Li et al., 2011; Rabusseau and Kadri, 2016; Li and Zhang, 2017; Raskutti and Yuan, 2016). Although aiming to address the same type of problem, our proposal is analytically different in several ways. We also numerically compare with some of those alternative solutions in simulations and real data analysis.

In particular, Zhu et al. (2009) considered a 3×3 symmetric positive definite matrix arising from diffusion tensor imaging as a response, and developed an intrinsic regression model by mapping the Euclidean space of covariates to the Riemannian manifold of positive-definite matrices. Unlike their solution, we consider a non-symmetric or symmetric tensor response in the Euclidean space, and allow the dimension of the tensor response to increase with the sample size. Li et al. (2011) estimated regression parameters by building iteratively increasing neighbors around each voxel and smoothing observations within the neighbors with weights. By contrast, we model all the voxels in an image tensor in a joint fashion.

Rabusseau and Kadri (2016) considered a tensor response model with a low-rank structure. However, no sparsity is enforced in their estimator, and thus their method is not directly applicable for selecting brain subregions affected by the disorder. Our approach instead is capable of region selection, and the resulting estimator is much easier to interpret.

Li and Zhang (2017) proposed an envelope-based tensor response model that is notably different from our proposal. First, they utilized a generalized sparsity principle to exploit the redundant information in the tensor response, by seeking linear combinations of the response that are irrelevant to the regression. Our method instead utilizes the element-wise sparsity in terms of the individual entries. As such our method can achieve region selection, whereas their approach cannot. Second, they obtained the $\sqrt{n}$-convergence rate for the *global* minimizer of their objective function when the tensor dimension is fixed. However, their objective function is non-convex and there is no guarantee that the optimization algorithm can find this global minimizer. By contrast, we derive the error rate of the *actual* minimizer of our algorithm at each iteration, and we also permit an increasing tensor dimension. Third, their approach could not directly incorporate the symmetry constraint when the tensor response is symmetric, which is often encountered in functional imaging analysis. To the best of our knowledge, our solution is the first that can simultaneously tackle both a non-symmetric and a symmetric tensor response in a regression setup.

Raskutti and Yuan (2016) developed a class of sparse regression models, under the assumption of Gaussian error, when either or both the response and predictor are tensors. When the error distribution is Gaussian, our error bound matches theirs. However, our bound is obtained for a general error distribution, where the normality is not necessarily required. In addition, they required another crucial condition that the regularizer must be convex and weakly decomposable. We do not impose this assumption, but instead tackle a non-convex optimization problem, and employ a different set of proof techniques. Finally, they achieved the low-rankness of the estimator through a tensor nuclear norm, which is known to be computationally NP-hard (Friedland and Lim, 2014). By contrast, our rate is established for the actual estimator obtained from our optimization algorithm, which we show later is both feasible and scalable.

Our major contribution is two-fold. First, we propose a new class of regression models with tensor as a response. This model is useful for a wide range of scientific applications, but has received only limited attention in the statistics literature, especially for the symmetric tensor response case. Our proposal is shown to exhibit numerous advantages compared to the existing solutions. Second, we develop a set of new tools for the theoretical analysis of non-convex optimization, which is notably different from recent development in this area (Wang et al., 2014; Yi and Caramanis, 2015; Sun et al., 2015; Balakrishnan et al., 2017). A common technique used in the non-convex optimization analysis is to separately

establish the convergence for the population and sample optimizers then combine the two. By contrast, our analysis hinges on exploitation of the bi-convex structure of the objective function, as well as a careful characterization of the impact of the intermediate sparse tensor decomposition step on the estimation error in each iteration step. The bi-convex structure frequently arises in many optimization problems. As such tools we develop are also of independent interest, and enrich the current toolbox of non-convex optimization analysis.

1.3 Notations and Structure

Throughout the article we adopt the following notations. Denote $[d] = \{1, \ldots, d\}$, and $\mathbf{I}_d$ the $d \times d$ identity matrix. Denote $1(\cdot)$ the indicator function, and $\circ$ the outer product between vectors. For a vector $\mathbf{a} = \mathbb{R}^d$, $\|\mathbf{a}\|$ denotes its Euclidean norm, and $\|\mathbf{a}\|_0$ the L_0 norm, i.e., the number of nonzero entries in $\mathbf{a}$. For a matrix $\mathbf{A} \in \mathbb{R}^{d\times d}$, $\|\mathbf{A}\|$ denotes its spectral norm. For a tensor $\mathcal{A} \in \mathbb{R}^{d_1\times\ldots\times d_m}$, and a set of vectors $\mathbf{a}_j \in \mathbb{R}^{d_j}, j = 1, \ldots, m$, the multilinear combination of the tensor entries is defined as $\mathcal{A} \times_1 \mathbf{a}_1 \times_2 \ldots \times_m \mathbf{a}_m := \sum_{i_1\in[d_1]} \cdots \sum_{i_m\in[d_m]} a_{i_1} \ldots a_{i_m} \mathcal{A}_{i_1,\ldots,i_m} \in \mathbb{R}$. The tensor spectral norm is defined as $\|\mathcal{A}\| := \sup_{\|\mathbf{a}_1\|=\ldots=\|\mathbf{a}_m\|=1} |\mathcal{A} \times_1 \mathbf{a}_1 \times_2 \ldots \times_m \mathbf{a}_m|$, and the tensor Frobenius norm as $\|\mathcal{A}\|_F := \sqrt{\sum_{i_1,\ldots,i_m} \mathcal{A}^2_{i_1,\ldots,i_m}}$. For two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{d_1\times\ldots\times d_m}$, their inner product is $\langle \mathcal{A}, \mathcal{B}\rangle = \sum_{i_1,\ldots,i_m} \mathcal{A}_{i_1,\ldots,i_m} \mathcal{B}_{i_1,\ldots,i_m}$. See Kolda and Bader (2009) for more tensor operations. We say $a_n = o(b_n)$ if a_n/b_n converges to 0 as n increases to infinity.

The rest of the article is organized as follows. Section 2 introduces the proposed sparse tensor response regression model, and Section 3 presents the optimization algorithm. Section 4 establishes the estimation error bound. Section 5 presents the simulation results, and Section 6 applies our method to an analysis of the Autism spectrum disorder imaging data. The appendix collects all technical proofs.

2. Model

For an mth-order tensor response $\mathcal{Y}_i \in \mathbb{R}^{d_1\times\cdots\times d_m}$, and a vector of predictors $\mathbf{x}_i \in \mathbb{R}^p$, $i = 1, \ldots, n$, we consider the tensor response regression model of the form,

$$\mathcal{Y}_i = \mathcal{B}^* \times_{m+1} \mathbf{x}_i + \mathcal{E}_i, \tag{1}$$

where $\mathcal{B}^* \in \mathbb{R}^{d_1\times\cdots\times d_m\times p}$ is an $(m+1)$th-order tensor coefficient, and $\mathcal{E}_i \in \mathbb{R}^{d_1\times\cdots\times d_m}$ is an error tensor independent of $\mathbf{x}_i$. Without loss of generality, the intercept is set to zero, by centering the samples $\mathbf{x}_i$ and $\mathcal{Y}_i$. We also use $d_{m+1} := p$ to represent the dimension of $\mathbf{x}_i$. Our goal is to estimate $\mathcal{B}^*$ given n i.i.d. observations $\{(\mathbf{x}_i, \mathcal{Y}_i), i = 1, \ldots, n\}$.

To facilitate estimation of the ultrahigh dimensional unknown parameters under a limited sample size, it is crucial to introduce some sparse structures. Two most commonly used structures are element-wise sparsity and low-rankness (Raskutti and Yuan, 2016). In the context of tensor response regression, we assume that $\mathcal{B}^*$ admits the following rank-K decomposition structure (Kolda and Bader, 2009),

$$\mathcal{B}^* = \sum_{k\in[K]} w^*_k \boldsymbol{\beta}^*_{k,1} \circ \cdots \circ \boldsymbol{\beta}^*_{k,m} \circ \boldsymbol{\beta}^*_{k,m+1}, \quad w^*_k \in \mathbb{R},\ \boldsymbol{\beta}^*_{k,j} \in \mathbb{S}^{d_j}, \tag{2}$$

where $\mathbb{S}^d = \{\mathbf{v} \in \mathbb{R}^d \,|\, \|\mathbf{v}\| = 1\}$, $\|\boldsymbol{\beta}^*_{k,j}\|_0 \le d_{j0} < d_j$, and d_{j0} represents the true sparsity of the individual components $\boldsymbol{\beta}^*_{k,j}$, for $k \in [K]$, $j \in [m+1]$. Consequently, when coupling with the low-rank structure of (2), the element-wise sparsity of $\boldsymbol{\beta}^*_{k,j}$ implies that some individual entries of $\mathcal{B}^*$ are zero. Moreover, we assume $w^*_{\max} = w^*_1 \ge \cdots \ge w^*_K = w^*_{\min} > 0$, and assume each w^*_i to be bounded away from 0 and ∞. Under the structure of (2), estimating $\mathcal{B}^*$ reduces to the estimation of $w^*_k, \beta^*_{k,1}, \ldots, \beta^*_{k,m+1}$ for any $k \in [K]$, and the corresponding parameter space becomes $\mathbb{B} := \{w_k \in \mathbb{R}, \boldsymbol{\beta}_{k,j} \in \mathbb{R}^{d_j}, k \in [K], j \in [m+1] \,|\, c_1 \le |w_k| \le c_2, \|\boldsymbol{\beta}_{k,j}\|_2 = 1, \|\boldsymbol{\beta}_{k,j}\|_0 \le d_{j0}\}$. Accordingly, the number of unknown parameters is reduced from $p\prod_{j=1}^m d_j$ to $K(\sum_{j=1}^m d_j + p)$. To estimate those unknown parameters, we propose to solve the following constrained optimization problem,

$$\min_{\substack{w_k,\boldsymbol{\beta}_{k,1},\ldots,\boldsymbol{\beta}_{k,m+1}\\ k\in[K],j\in[m+1]}} \frac{1}{n}\sum_{i=1}^n \Big\|\mathcal{Y}_i - \sum_{k\in[K]} w_k(\boldsymbol{\beta}^\top_{k,m+1}\mathbf{x}_i)\boldsymbol{\beta}_{k,1} \circ \cdots \circ \boldsymbol{\beta}_{k,m}\Big\|_F^2, \tag{3}$$
$$\text{subject to } \|\boldsymbol{\beta}_{k,j}\|_2 = 1, \|\boldsymbol{\beta}_{k,j}\|_0 \le s_j,$$

where s_j is the cardinality parameter of the j-th component. Here we encourage the sparsity of the decomposed components via a hard-thresholding penalty to control $\|\boldsymbol{\beta}_{k,j}\|_0$. Compared to the lasso type penalized approach in sparse learning, the hard-thresholding method avoids bias and has been shown to be more appealing in numerous high-dimensional learning problems (Shen et al., 2012, 2013; Wang et al., 2014).

3. Estimation

The problem in (3) is a non-convex optimization. The key of our estimation procedure is to explore its *bi-convex* structure. In particular, the objective function in (3) is bi-convex in $(\boldsymbol{\beta}_{k,1}, \ldots, \boldsymbol{\beta}_{k,m+1})$, $k \in [K]$, in that it is convex in $\boldsymbol{\beta}_{k,j}$ when all other parameters are fixed. Utilizing this property, we propose an efficient alternating updating algorithm to solve (3).

3.1 Algorithm

We first summarize our estimation procedure in Algorithm 1, then present the two key alternative steps.

Step 1: The first core step of our algorithm is to update $w_k, \boldsymbol{\beta}_{k,1}, \ldots, \boldsymbol{\beta}_{k,m}$ for each $k = 1, \ldots, K$, given $\boldsymbol{\beta}_{j,m+1}$, $j = 1, \ldots, K$ and $w_{k'}, \beta_{k',1}, \ldots, \beta_{k',m}$, $k' \ne k$. Letting $\alpha_{ik} := \boldsymbol{\beta}^\top_{k,m+1}\mathbf{x}_i$, $i = 1, \ldots, n$, we note that (3) is equivalent to

$$\min_{\substack{w_k,\boldsymbol{\beta}_{k,1},\ldots,\boldsymbol{\beta}_{k,m}\\ \|\boldsymbol{\beta}_{k,j}\|_2=1,\|\boldsymbol{\beta}_{k,j}\|_0\le s_j,j\in[m]}} \frac{1}{n}\sum_{i=1}^n \alpha^2_{ik}\Big\|\mathcal{R}_i - w_k\boldsymbol{\beta}_{k,1} \circ \cdots \circ \boldsymbol{\beta}_{k,m}\Big\|_F^2, \tag{4}$$

where the residual tensor term $\mathcal{R}_i$ is of the form,

$$\mathcal{R}_i := \Big[\mathcal{Y}_i - \sum_{k'\ne k, k'\in[K]} w_{k'}\alpha_{ik'}\boldsymbol{\beta}_{k',1} \circ \cdots \circ \boldsymbol{\beta}_{k',m}\Big]/\alpha_{ik}. \tag{5}$$

The next lemma shows that (4) can be solved via a sparse tensor decomposition procedure.

Algorithm 1 Alternating updating algorithm for STORE.

1: **Input:** data $\{(\mathbf{x}_i, \mathcal{Y}_i), i = 1, \ldots, n\}$, rank K, cardinality vector $(s_1, \ldots, s_m)$.
2: **Initialize** $w_k = 1$ and random unit-norm vectors $\boldsymbol{\beta}_{k,1}, \ldots, \boldsymbol{\beta}_{k,m+1}$ for each $k \in [K]$.
3: **Until** the termination condition is met, **Do**
4: Step 1: For $k = 1$ to K, obtain $\widehat{w}_k, \widehat{\boldsymbol{\beta}}_{k,1}, \ldots, \widehat{\boldsymbol{\beta}}_{k,m}$ by solving (6) via the sparse tensor decomposition procedure of Sun et al. (2017) with parameters K and $(s_1, \ldots, s_m)$.
5: Step 2: For $k = 1$ to K, obtain $\widehat{\boldsymbol{\beta}}_{k,m+1}$ using Lemma 2.
6: **Output:** $\widehat{w}_k, \widehat{\boldsymbol{\beta}}_{k,1}, \ldots, \widehat{\boldsymbol{\beta}}_{k,m+1}$ for each $k \in [K]$.

Lemma 1 *Denote $(\widehat{w}_k, \widehat{\boldsymbol{\beta}}_{k,1}, \ldots, \widehat{\boldsymbol{\beta}}_{k,m})$ as the solution of (4), and denote $(\widetilde{w}_k, \widetilde{\boldsymbol{\beta}}_{k,1}, \ldots, \widetilde{\boldsymbol{\beta}}_{k,m})$ as the solution of*

$$\min_{\substack{w_k, \boldsymbol{\beta}_{k,1}, \ldots, \boldsymbol{\beta}_{k,m} \\ \|\boldsymbol{\beta}_{k,j}\|_2 = 1, \|\boldsymbol{\beta}_{k,j}\|_0 \le s_j, j \in [m]}} \left\| \bar{\mathcal{R}} - w_k \boldsymbol{\beta}_{k,1} \circ \cdots \circ \boldsymbol{\beta}_{k,m} \right\|_F^2, \tag{6}$$

where $\bar{\mathcal{R}} := \frac{1}{n} \sum_{i=1}^n \alpha_{ik}^2 \mathcal{R}_i$. Then it satisfies that $\widehat{w}_k = n \widetilde{w}_k / \sum_{i=1}^n \alpha_{ik}^2$ and $\widehat{\boldsymbol{\beta}}_{k,l} = \widetilde{\boldsymbol{\beta}}_{k,l}$, for $l = 1, \ldots, m$ and $k = 1, \ldots, K$.

Lemma 1 implies that the optimization problem in (4) reduces to a sparse rank-one tensor decomposition on the averaged tensor $\bar{\mathcal{R}}$. To efficiently solve (6), in this paper we employ a truncation-based sparse tensor decomposition procedure (Sun et al., 2017), by first solving the non-sparse tensor decomposition components and then truncating them to achieve the desirable sparsity. It is also noteworthy that our method is flexible in the choice of optimization algorithm for solving (6), and Algorithm 1 can be coupled with many other sparse tensor decomposition algorithms, e.g., Chi and Kolda (2012); Liu et al. (2012).

When the tensor response $\mathcal{Y}_i$ is symmetric, the resulting coefficient $\mathcal{B}$ should be symmetric too. Our algorithm can easily adapt to this scenario, by setting $\boldsymbol{\beta}_{k,1} = \cdots = \boldsymbol{\beta}_{k,m} = \boldsymbol{\beta}_k$ for each $k \in [K]$, and the cardinality parameters $s_1 = \cdots = s_m = s$. That is, we slightly modify Step 1 in Algorithm 1 as: for $k = 1$ to K, compute $\mathcal{R}_i$ in (5) and solve $\widehat{w}_k, \widehat{\boldsymbol{\beta}}_k$ via the symmetric sparse tensor decomposition with parameters K and s.

Step 2: The second core step of our algorithm is to update $\boldsymbol{\beta}_{k,m+1}$ for each $k = 1, \ldots, K$, given $w_j, \beta_{j,1}, \ldots, \boldsymbol{\beta}_{j,m}$, $j = 1, \ldots, K$ and $\boldsymbol{\beta}_{k',m+1}$, $k' \neq k$. Letting $\mathcal{A}_k = w_k \beta_{k,1} \circ \cdots \circ \boldsymbol{\beta}_{k,m}$, then (3) is equivalent to

$$\min_{\alpha} \frac{1}{n} \sum_{i=1}^n \left\| \mathcal{T}_i - \alpha^\top \mathbf{x}_i \mathcal{A}_k \right\|_F^2, \tag{7}$$

where the residual tensor $\mathcal{T}_i = \mathcal{Y}_i - \sum_{k' \neq k, k' \in [K]} w_{k'} (\boldsymbol{\beta}_{k',m+1}^\top \mathbf{x}_i) \boldsymbol{\beta}_{k',1} \circ \cdots \circ \boldsymbol{\beta}_{k',m}$. The next lemma gives a closed-form solution of (7).

Lemma 2 *The solution of (7) is given by*

$$\widehat{\boldsymbol{\beta}}_{k,m+1} = \left(\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \right)^{-1} \frac{n^{-1} \sum_{i=1}^n \langle \mathcal{T}_i, \mathcal{A}_k \rangle \mathbf{x}_i}{\|\mathcal{A}_k\|_F^2}.$$

In our neuroimaging example in Section 6, the dimension of the predictor vector is $p = 3$. For such a small value of p, the sample size n is generally much larger, and we can directly invert the sample covariance matrix $n^{-1} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$. If $n < p$, this matrix is not invertible. Then one may employ sparse graphical model estimation (Yuan and Lin, 2007; Friedman et al., 2008; Zhang and Zou, 2014) and replace this inverse with a sparse precision matrix estimator. Alternatively, one may also introduce additional sparsity constraint on the parameter $\boldsymbol{\beta}_{k,m+1}$ and resort to a regularized estimation approach to update $\boldsymbol{\beta}_{k,m+1}$.

Finally, we terminate the alternating update of Steps 1 and 2 when the new estimates are close to the ones from the previous iteration. The termination condition is set as

$$\max_{j \in [m+1], k \in [K]} \min \left\{ \|\widehat{\boldsymbol{\beta}}_{k,j}^{(t)} - \widehat{\boldsymbol{\beta}}_{k,j}^{(t-1)}\|, \|\widehat{\boldsymbol{\beta}}_{k,j}^{(t)} + \widehat{\boldsymbol{\beta}}_{k,j}^{(t-1)}\| \right\} \le 10^{-4}.$$

3.2 Tuning Parameter Selection

In Algorithm 1, the rank K and the cardinality $s_1, \ldots, s_m$ are tuning parameters. We propose to select those parameters via a BIC-type criterion. Specifically, given a pre-specified set of rank values $\mathcal{K}$ and a pre-specified set of cardinality values $\mathcal{S}_1, \ldots, \mathcal{S}_m$, we choose the combination of parameters $(\widehat{K}, \widehat{s}_1, \ldots, \widehat{s}_m)$ that minimizes

$$\mathrm{BIC} = \log \left(\sum_{i=1}^n \|\mathcal{Y}_i - \widehat{\mathcal{B}} \times_{m+1} \mathbf{x}_i\|_F^2 \right) + \frac{\log(n \prod_{j=1}^m d_m)}{n \prod_{j=1}^m d_m} \sum_{k=1}^K \sum_{j=1}^m \|\widehat{\boldsymbol{\beta}}_{k,j}\|_0.$$

This criterion balances between model fitting and model sparsity, and a similar version has been commonly employed in rank estimation (Zhou et al., 2013).

4. Theory

Next we establish the error bound of the actual STORE estimator obtained from our Algorithm 1. The resulting error bound consists of two quantities: a computational error and a statistical error. The computational error captures the error caused by the non-convexity of the optimization, whereas the statistical error measures the error due to finite samples.

In order to compute the distance between the estimator and the truth, we define the distance measure between two unit vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$ as $D(\mathbf{u}, \mathbf{v}) := \sqrt{1 - (\mathbf{u}^\top \mathbf{v})^2}$. We then have $D(\mathbf{u}, \mathbf{v}) \le \min\{\|\mathbf{u} - \mathbf{v}\|, \|\mathbf{u} + \mathbf{v}\|\} \le \sqrt{2} D(\mathbf{u}, \mathbf{v})$. The distance function $D(\mathbf{u}, \mathbf{v})$ resolves the sign issue in the decomposition components since changing the signs of any two component vectors does not affect the generated tensor.

4.1 Assumptions

We first introduce the technical assumptions required to guarantee the desirable error bound. The first assumption is about the structure of the true model.

Assumption 3 (Model Assumption) *In model (1), we assume the true coefficient $\mathcal{B}^*$ is sparse and low-rank satisfying (2) and such decomposition is unique up to a permutation. Assume $\|\mathcal{B}^*\| \le C_1 w_{\max}^*$, and for each i, assume $\|\mathbf{x}_i\| \le C_2$ and $|\boldsymbol{\beta}_{k,m+1}^{*\top} \mathbf{x}_i| \ge C_3$ almost surely, for some positive constants C_1, C_2, C_3. Furthermore, we require the eigenvalues of*

the sample covariance $\Sigma := n^{-1}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top$ *to satisfy* $c_0 < \lambda_{\min}(\Sigma) \le \lambda_{\max}(\Sigma) < \widetilde{c}_0$ *for some constants* $c_0, \widetilde{c}_0$.

The above unique decomposition condition is to ensure the identifiability of tensor decomposition. Kruskal (1976, 1977) provided the classical conditions for such identifiability if the sum of the Kruskal ranks of the m decomposed component matrices is larger than $2K+2$. In our model, the rank K is fixed, and hence the identifiability of our tensor decomposition (2) is guaranteed when the decomposed components are not highly dependent. Moreover, the conditions on the predictor vector $\mathbf{x}_i$ are mild and trivially hold when the dimension of $\mathbf{x}_i$ is fixed. In Section 6, the dimension of $\mathbf{x}_i$ is 3 in our neuroimaging data example.

The second assumption is about the initialization in Algorithm 1.

Assumption 4 (Initialization Assumption) *Define the initialization error*

$$\epsilon := \max\left\{\max_k \|\widehat{w}_k^{(0)} - w_k^*\|_2, \max_{k,j} D\left(\widehat{\boldsymbol{\beta}}_{k,j}^{(0)}, \boldsymbol{\beta}_{k,j}^*\right)\right\}.$$

We assume that

$$\epsilon < \min\left\{\sqrt{\frac{w_{\min}^*}{2(w_{\min}^* + w_{\max}^* C_1)}}, \frac{w_{\min}^{*2}}{8\sqrt{5}w_{\max}^* C_1(6\sqrt{2}w_{\min}^* + 2)}, \frac{C_3}{2C_2}, \frac{w_{\min}^*}{2}\right\},$$

where the constants C_1, C_2, C_3 *are as defined in Assumption 3.*

Note that our assumption on the initialization parameters only requires the error to be bounded by some constant; i.e., the initial estimators are not too far away from the true parameters. This assumption is necessary to handle the non-convexity of the optimization, and has been commonly imposed in non-convex optimization (Wang et al., 2014; Balakrishnan et al., 2017). It is also noteworthy that this assumption is satisfied by the estimators from sparse singular value decomposition of the unfolding matrix from the original tensor in Step 1 of Algorithm 1; see Sun et al. (2017) for more discussion on the theoretical guarantee of such an initialization procedure.

The third assumption requires that the error tensor $\mathcal{E}_i$ is controlled. Before stating the assumption, we introduce a critical concept to measure the noise level in the tensor response regression. In particular, we define the sparse spectral norm of $\mathcal{E}$ as

$$\eta(\mathcal{E}, d_{01}, \cdots, d_{0m}) := \sup_{\substack{\|\mathbf{u}_1\|=\cdots=\|\mathbf{u}_m\|=1 \\ \|\mathbf{u}_1\|_0\le d_{01},\ldots,\|\mathbf{u}_m\|_0\le d_{0m}}} \left|\mathcal{E}\times_1\mathbf{u}_1\times_2\cdots\times_m\mathbf{u}_m\right|. \tag{8}$$

This quantifies the perturbation error in a sparse scenario, and in the sparse case with $d_{0j} \ll d_j$ $(j = 1, \ldots, m)$, it is much smaller than the spectral norm $\|\mathcal{E}\|$, i.e., $\eta(\mathcal{E}, d_1, \cdots, d_m)$. Denoting $d_0 = \max_j d_{0j}$, we have $\eta(\mathcal{E}, d_{01}, \cdots, d_{0m}) \le \eta(\mathcal{E}, d_0, \cdots, d_0)$. We write $\eta(\mathcal{E}, d_0) := \eta(\mathcal{E}, d_0, \cdots, d_0)$.

Assumption 5 (Error Assumption) *Let* $s := \max\{s_1, \ldots, s_m\}$, *with* s_j *being the cardinality parameter in Algorithm 1. We assume the error tensor* $\mathcal{E}_i$, $i = 1, \ldots, n$, *satisfies*

$$\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right) \le \frac{C_3 w_{\min}^*}{8},$$

where C_3 *is as defined in Assumption 3.*

This assumption requires that the observed tensor responses, $\mathcal{Y}_i$, $i = 1, \ldots, n$, are not too noisy. It is a minor condition, since we only require the spectral norm of the error tensor to be bounded. As we later show in Section 4.3, when $\mathcal{E}_i$ is a standard Gaussian tensor, its spectral norm converges to zero when the sample size n increases.

4.2 Main Result

Next we present our main theorem that describes the estimation error of the actual estimator at each iteration from Algorithm 1.

Theorem 6 *Assuming Assumptions 3, 4, and 5, the estimator in the t-th iteration of Algorithm 1, with* $s \ge d_0 := \max_j d_{j0}$, *satisfies that*

$$\begin{aligned}&\max\left\{\max_k \|\widehat{w}_k^{(t)} - w_k^*\|_2, \max_{k,j} D\left(\widehat{\boldsymbol{\beta}}_{k,j}^{(t)}, \boldsymbol{\beta}_{k,j}^*\right)\right\} \\ \le\ & \underbrace{\kappa^t\epsilon}_{\text{computational error}} + \underbrace{\frac{1}{1-\kappa}\max\left\{\frac{4\sqrt{5}}{C_3 w_{\min}^*}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right), \frac{\widetilde{C}}{\sqrt{n}}\right\}}_{\text{statistical error}},\end{aligned} \tag{9}$$

where the contraction coefficient $\kappa = (\kappa_1 + \kappa_2)\kappa_3 \in (0, 1)$, *with* $\kappa_1 := 4\sqrt{5}w_{\max}^* C_3\epsilon/w_{\min}^*$, $\kappa_2 := 2C_2\eta\left(\bar{\mathcal{E}}, s\right)/C_3^2$, *and* $\kappa_3 := 2/w_{\min}^* + 6\sqrt{2}$. *Here* C_1, C_2, C_3 *are constants as defined in Assumption 3, and* $\widetilde{C}$ *is some positive constant.*

Theorem 6 reveals the interplay between the computational error and the statistical error for our estimator, and has some interesting implications. First, it provides a theoretical guidance to terminate the iterations of the alternating updating algorithm. That is, when the computational error is dominated by the statistical error, in that

$$t \ge T := \log_{1/\kappa}\left(\frac{(1-\kappa)\epsilon}{\max\left\{\frac{4\sqrt{5}}{C_3 w_{\min}^*}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right), \frac{\widetilde{C}}{\sqrt{n}}\right\}}\right), \tag{10}$$

the estimator from our algorithm achieves an error at the rate

$$O_p\left(\max\left\{\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right), \frac{1}{\sqrt{n}}\right\}\right). \tag{11}$$

The constants C_3 and $\widetilde{C}$ in (10) are unknown in practice. However, due to the fact both constants are independent of the sample size, the sparsity parameters, and the dimensionality, the expression (10) gives an order of the optimal termination T in terms of the parameters that go to infinity. Second, Theorem 6 also provides a general theoretical guarantee for our estimator for any distribution of the error tensor $\mathcal{E}_i$, as the error rate depends on the distribution of the error tensor only through $\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right)$. Later we obtain the explicit forms of the error rates for some special error distributions.

We also remark that, Theorem 6 is obtained assuming the true rank K is known. This is a common assumption in theoretical analysis in the tensor literature (Zhou et al., 2013; Li and Zhang, 2017). To our knowledge, none of the exiting works has provided a provable estimation of the rank, due to that the exact tensor rank calculation is NP hard (Kolda and Bader, 2009). The theory of rank estimation will be our future research.

4.3 Gaussian Tensor Error

We derive the explicit form of the statistical error in Theorem 6 when $\mathcal{E}_i, i = 1, \ldots, n$, are i.i.d. Gaussian tensors; i.e., the entries of $\mathcal{E}_i$ are i.i.d. standard Gaussian random variables.

Corollary 7 *Under the assumptions of Theorem 6, and assuming $\mathcal{E}_i \in \mathbb{R}^{d_1 \times d_2 \times d_3}, i = 1, \ldots, n$, are i.i.d. Gaussian tensors, we have*

$$\eta\left(\frac{1}{n}\sum_{i=1}^{n}\mathcal{E}_i, s\right) = O_p\left(\sqrt{\frac{s^3\log(d_1d_2d_3)}{n}}\right).$$

Consequently the estimator at the T-th iteration of Algorithm 1 satisfies

$$\max\left\{\max_k \|\widehat{w}_k^{(T)} - w_k^*\|_2, \max_{k,j} D\left(\widehat{\boldsymbol{\beta}}_{k,j}^{(T)}, \boldsymbol{\beta}_{k,j}^*\right)\right\} = O_p\left(\sqrt{\frac{s^3\log(d_1d_2d_3)}{n}}\right).$$

We provide some insight about the statistical error $O_p(\sqrt{s^3\log(d_1d_2d_3)/n})$. Note that, s is the maximal cardinality of vectors $\widehat{\boldsymbol{\beta}}_{k,j}$ for $k \in [K]$ and $j \in [m]$, and hence there are at most $O(s^3)$ non-zero elements and $O(d_1d_2d_3)$ free parameters in the true tensor coefficient $\mathcal{B}^*$. This error rate allows the tensor dimension to grow exponentially with the sample size and matches the one established by Raskutti and Yuan (2016) for sparse tensor regression. When the order of tensor is one, i.e., $m = 1$, it reduces to the statistical error $O_p(\sqrt{s\log d/n})$ for the high-dimensional vector regression $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$ with $\boldsymbol{\beta} \in \mathbb{R}^d$ and $\|\boldsymbol{\beta}\|_0 \le s$, and is known to be minimax optimal (Wang et al., 2014).

4.4 Symmetric Matrix Error

Next we consider the case when the response $\mathcal{Y}_i$ is a symmetric matrix ($m = 2$). Such a scenario is often encountered in functional neuroimaging analysis, where the target of interest is the brain connectivity pattern encoded in the form of a symmetric correlation matrix. The symmetry of the coefficient $\mathcal{B}^*$ requires that $\boldsymbol{\beta}_{k,1}^* = \boldsymbol{\beta}_{k,2}^* = \boldsymbol{\beta}_k^*$. Henceforth, $\mathcal{B}^* = \sum_{k\in[K]} w_k^* \cdot \boldsymbol{\beta}_k^* \circ \boldsymbol{\beta}_k^* \circ \boldsymbol{\beta}_{k,3}^*$, where $w_k^* \in \mathbb{R}$, $\|\boldsymbol{\beta}_k^*\|_2 = 1, \|\boldsymbol{\beta}_k^*\|_0 \le d_0$, and $\boldsymbol{\beta}_{k,3}^* \in \mathbb{S}^p$. To facilitate the derivation of the explicit form of $\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right)$, we assume that the error matrix $\mathcal{E}_i$ satisfies

$$\mathcal{E}_i = (\widetilde{\mathcal{E}}_i + \widetilde{\mathcal{E}}_i^\top)/2, \tag{12}$$

where $\widetilde{\mathcal{E}}_i \in \mathbb{R}^{d\times d}$ is a matrix whose entries are i.i.d. standard Gaussian. This assumption is mainly for technical reasons, and the theoretical analysis of a more general symmetric tensor response is left as future work.

Corollary 8 *Under the assumptions of Theorem 6, and assuming $\mathcal{E}_i, i = 1, \ldots, n$ are i.i.d. and of the form* (12)*, we have*

$$\eta\left(\frac{1}{n}\sum_{i=1}^{n}\mathcal{E}_i, s\right) = O_p\left(\sqrt{\frac{s^2\log(d^2)}{n}}\right).$$

5. Simulations

In this section, we investigate the numerical performance of our method, and demonstrate its superior performance when compared to some alternative solutions.

To evaluate the estimation accuracy, we report the estimation error $\|\widehat{\mathcal{B}} - \mathcal{B}\|_F$. To evaluate the variable selection accuracy, we compute the true positive rate, $\text{TPR} := m^{-1}\sum_{j=1}^m \text{TPR}_j$, and the false positive rate, $\text{FPR} := m^{-1}\sum_{j=1}^m \text{FPR}_j$, where $\text{TPR}_j := K^{-1}\sum_{k=1}^K\sum_i 1([\beta_{k,j}]_i \neq 0, [\widehat{\beta}_{k,j}]_i \neq 0)/\sum_i 1([\beta_{k,j}]_i \neq 0)$, $\text{FPR}_j := K^{-1}\sum_{k=1}^K\sum_i 1([\beta_{k,j}]_i = 0, [\widehat{\beta}_{k,j}]_i \neq 0)/\sum_i 1([\beta_{k,j}]_i = 0)$. We also report the F1 score, $F_1 := 2(\text{recall}^{-1} + \text{precision}^{-1})^{-1}$, where $\text{precision} = m^{-1}K^{-1}\sum_{j=1}^m\sum_{k=1}^K\sum_i 1([\beta_{k,j}]_i \neq 0, [\widehat{\beta}_{k,j}]_i \neq 0)/\sum_i 1([\widehat{\beta}_{k,j}]_i = 0)$ and recall = TPR. The TPR, FPR and F1 score are all between 0 and 1, with a larger value of TPR and F1 and a smaller value of FPR indicating a better selection performance.

We compare our STORE method with four alternative solutions. The first is ordinary least squares (OLS) that fits one response variable at a time. Although a naive solution, OLS is widely used in neuroimaging analysis due to its simplicity. The second is the sparse ordinary least squares method (Sparse OLS) of Peng et al. (2010) that first vectorizes the tensor response then fits a regularized multivariate regression with the lasso penalty. The third is the envelope-based tensor response regression method (ENV) of Li and Zhang (2017) that utilizes a generalized sparsity principle. It is noteworthy that, in Li and Zhang (2017), ENV has been shown to clearly outperform the tensor predictor regression method of Zhou et al. (2013). Hence, we directly compare with ENV but no longer with Zhou et al. (2013). The fourth is the higher-order low-rank regression method (HOLRR) of Rabusseau and Kadri (2016) that imposes a low-rank tensor structure. We also comment that, OLS and sparse OLS both ignore the tensor structure of the data, whereas neither of OLS, ENV, and HOLRR performs region selection.

5.1 3D Tensor Response Example

We first consider an example of a third-order tensor response ($m = 3$). The data was generated from model (1) with x_i a scalar taking values 0 or 1 with an equal probability, and the coefficient tensor $\mathcal{B} \in \mathbb{R}^{d_1\times d_2\times d_3}$ of the form, $\mathcal{B} = \sum_{k\in[K]} w_k\beta_{k,1}\circ\beta_{k,2}\circ\beta_{k,3}\circ\beta_{k,4}$, $w_k \in \mathbb{R}$, $\beta_{k,j} \in \mathbb{S}^{d_j}$. For each $k \in [K]$, we first generated i.i.d. standard Gaussian entries of the three vectors $\beta_{k,1}, \beta_{k,2}, \beta_{k,3}$, then truncated each vector with the cardinality parameters s_{01}, s_{02}, s_{03} accordingly. Next we normalized each vector and aggregated the coefficients as w_k. We set $\beta_{k,4}$ as one. In all simulations, we fixed the tensor dimensions $d_1 = 100, d_2 = 50, d_3 = 20$. We set the true cardinality $s_{0j} = s * d_j$ ($j = 1, 2, 3$), and varied the sparsity level $s \in \{0.3, 0.5\}$ and rank $K \in \{2, 5\}$. We considered the sample size $n \in \{20, 100\}$. Based on our theoretical finding, a larger sample size leads to a better estimation performance. We have chosen a relatively small sample size to mimic the situation where the number of subjects is limited. Such a situation is commonly encountered in neuroimaging studies. In total, there are 8 combinations of different scenarios. For STORE, the tuning parameters were chosen according to the BIC criterion in Section 3.2 by setting $\mathcal{K} = \{1, 2, \ldots, 10\}$ and $\mathcal{S} = \{0.1, 0.2, \ldots, 0.9\}$. The tunings for Sparse OLS and HOLRR were conducted based on their recommended approaches.

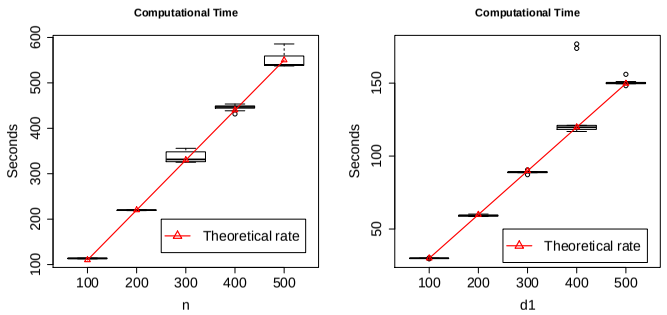


Figure 1: The computational time of our algorithm as the sample size and the tensor dimension varies. The theoretical linear rate is shown in red triangle.

Table 1 reports simulation results based on 20 replications. It is seen that our STORE method clearly achieves a superior performance than all the competing methods in terms of both estimation accuracy and variable selection accuracy. The same pattern holds for different sample size n, rank K, and sparsity level s. It is also noted that, since OLS, ENV, and HOLRR do not incorporate entry-wise sparsity, their corresponding TPR and FPR values are always one, and their F1 scores are undefined due to zero in the denominator of the precision.

We also studied the computational time of our algorithm when the sample size and the tensor dimension increase. The code is written in R and is implemented on a laptop with 2.5 GHz Intel Core i7 processor. Figure 5.1 reports the results based on 20 replications. Specifically, we fixed the rank $K = 2$ and the sparsity $s = 0.3$. In the left panel, we fixed the tensor dimensions $d_1 = 100, d_2 = 50, d_3 = 20$, but varied the sample size $n \in \{100, 200, 300, 400, 500\}$. In the right panel, we fixed $n = 20$, $d_2 = 50$, $d_3 = 20$, but varied the dimension $d_1 \in \{100, 200, 300, 400, 500\}$. This scale of sample size and tensor dimension is typical in neuroimaging applications. From the plot we see that the computational time of our Algorithm 1 is roughly linear in terms of the sample size and the tensor dimension when other parameters are fixed. This shows that our algorithm is scalable and feasible with moderate to large sample size and dimensionality. Meanwhile, it is possible to further improve our current algorithm. For instance, in each iteration, one could employ the recently developed parallel tensor decomposition (Papalexakis et al., 2012), or the sketching tensor decomposition (Wang et al., 2015), to accelerate the computation.

5.2 2D Symmetric Matrix Response Example

Next we considered an example of a second-order symmetric tensor (matrix, $m = 2$). This type of problem is commonly encountered in functional neuroimaging and brain connectivity analysis. The data was generated from the model, $\mathcal{Y}_i = \mathcal{B} \times_{m+1} x_i + \mathcal{E}_i$, where $\mathcal{Y}_i \in \mathbb{R}^{100 \times 100}$

n	K	s	Method	Error	TPR	FPR	F1
20	2	0.3	OLS	103.923 (2.330)	1 (0)	1 (0)	NA
			Sparse OLS	57.208 (1.971)	0.995 (0.001)	0.085 (0.007)	0.638 (0.019)
			ENV	8.603 (0.142)	1 (0)	1 (0)	NA
			HOLRR	7.311 (0.169)	1 (0)	1 (0)	NA
			STORE	**4.438** (0.098)	1 (0)	0 (0)	1 (0)
		0.5	OLS	103.923 (2.330)	1 (0)	1 (0)	NA
			Sparse OLS	76.709 (1.730)	0.960 (0.004)	0.372 (0.011)	0.604 (0.005)
			ENV	8.406 (0.109)	1 (0)	1 (0)	NA
			HOLRR	7.278 (0.204)	1 (0)	1 (0)	NA
			STORE	**5.827** (0.101)	1 (0)	0 (0)	1 (0)
	5	0.3	OLS	103.923 (2.330)	1 (0)	1 (0)	NA
			Sparse OLS	75.516 (1.319)	0.996 (0)	0.224 (0.041)	0.657 (0.044)
			ENV	13.593 (0.113)	1 (0)	1 (0)	NA
			HOLRR	10.696 (0.307)	1 (0)	1 (0)	NA
			STORE	**7.086** (0.151)	1 (0)	0 (0)	1 (0)
		0.5	OLS	103.923 (2.330)	1 (0)	1 (0)	NA
			Sparse OLS	93.409 (2.038)	0.988 (0.001)	0.647 (0.008)	0.738 (0.002)
			ENV	13.625 (0.145)	1 (0)	1 (0)	NA
			HOLRR	10.972 (0.294)	1 (0)	1 (0)	NA
			STORE	**9.140** (0.103)	1 (0)	0 (0)	1 (0)
100	2	0.3	OLS	44.250 (0.333)	1 (0)	1 (0)	NA
			Sparse OLS	30.851 (1.626)	1 (0)	0.045 (0.020)	0.831 (0.068)
			ENV	3.721 (0.090)	1 (0)	1 (0)	NA
			HOLRR	3.066 (0.068)	1 (0)	1 (0)	NA
			STORE	**1.979** (0.047)	1 (0)	0 (0)	1 (0)
		0.5	OLS	44.250 (0.333)	1 (0)	1 (0)	NA
			Sparse OLS	38.504 (0.443)	0.993 (0)	0.149 (0.003)	0.801 (0.003)
			ENV	3.803 (0.078)	1 (0)	1 (0)	NA
			HOLRR	2.999 (0.063)	1 (0)	1 (0)	NA
			STORE	**2.609** (0.029)	1 (0)	0 (0)	1 (0)
	5	0.3	OLS	44.250 (0.333)	1 (0)	1 (0)	NA
			Sparse OLS	32.584 (0.351)	0.998 (0)	0.149 (0.003)	0.721 (0.004)
			ENV	5.954 (0.049)	1 (0)	1 (0)	NA
			HOLRR	4.335 (0.051)	1 (0)	1 (0)	NA
			STORE	**3.131** (0.033)	1 (0)	0 (0)	1 (0)
		0.5	OLS	44.250 (0.333)	1 (0)	1 (0)	NA
			Sparse OLS	41.487 (0.349)	0.994 (0)	0.459 (0.003)	0.801 (0.001)
			ENV	6.037 (0.027)	1 (0)	1 (0)	NA
			HOLRR	4.328 (0.039)	1 (0)	1 (0)	NA
			STORE	**4.077** (0.037)	1 (0)	0 (0)	1 (0)

Table 1: Third-order tensor response example with different sample size n, rank K, and sparsity s. Reported are the average estimation error, TPR, FPR, and F1 score, with the standard error shown in the parenthesis. The minimal error in each case is shown in boldface.

is the symmetric matrix response, x_i is a scalar taking values 0 or 1, and the error term $\mathcal{E}_i$ is a symmetric matrix whose upper triangle entries are generated randomly from a standard normal distribution. The sample size is $n = 20$. To mimic some common connectivity patterns, we have simulated three symmetric coefficient matrices $\mathcal{B}$ that lead to three graph patterns: a randomly generated sparse and low-rank graph (random), a hub graph (hub),

Graph Pattern	Method	Estimation Error	TPR	FPR	F1
random	OLS	22.168 (0.560)	1 (0)	1 (0)	NA
	Sparse OLS	15.855 (0.543)	1 (0)	0.232 (0.012)	0.646 (0.012)
	ENV	10.751 (0.352)	1 (0)	1 (0)	NA
	HOLRR	8.318 (0.199)	1 (0)	1 (0)	NA
	STORE	**2.497** (0.075)	1 (0)	0 (0)	1 (0)
hub	OLS	22.168 (0.560)	1 (0)	1 (0)	NA
	Sparse OLS	12.924 (0.197)	0.338 (0.060)	0 (0)	0.443 (0.073)
	ENV	17.811 (0.222)	1 (0)	1 (0)	NA
	HOLRR	11.685 (0.110)	1 (0)	1 (0)	NA
	STORE	**7.448** (0.453)	0.824 (0.043)	0.079 (0.004)	0.272 (0.010)
small-world	OLS	23.481 (0.635)	1 (0)	1 (0)	NA
	Sparse OLS	22.024 (0.819)	0.821 (0.063)	0.010 (0.001)	0.816 (0.062)
	ENV	28.173 (0.427)	1 (0)	1 (0)	NA
	HOLRR	**18.794** (0.319)	1 (0)	1 (0)	NA
	STORE	21.309 (0.154)	0.901 (0.014)	0.118 (0.004)	0.610 (0.002)

Table 2: Symmetric matrix response example with different graph structures. Reported are the average estimation error, TPR, FPR, and F1 score, with the standard error shown in the parenthesis. The minimal error in each example is shown in boldface.

and a small-world graph (small-world). Note that the coefficient tensor $\mathcal{B}$ is symmetric, though the corresponding graph pattern is not necessarily so. The random and hub graphs are both of a low-rank structure, with the true rank $K = 2$ and $K = 10$, respectively. The small-world graph is not of an exact low-rank structure, and our method essentially provides a low-rank *approximation*. This scenario also allows us to investigate the performance of our method under potential model mis-specification. Besides, as none of the competitive solution enforces the symmetry property of the estimator $\widehat{\mathcal{B}}$, for comparison, we symmetrized their final estimators by $(\widehat{\mathcal{B}} + \widehat{\mathcal{B}}^\top)/2$.

Table 2 reports the results over 20 data replications. Figure 2 further illustrates the estimated graph patterns of all methods in one replicate. The first column shows the true graphs. The second column shows the estimates from OLS, ENV, and HOLRR. Note that, although those three methods yield different numerical estimates, none of them impose any sparsity. As such, their estimated graph patterns look the same. The third column shows the estimate from Sparse OLS, and the last column from our method STORE. In the first scenario where the true graph is sparse and low-rank, our method substantially outperforms the competing solutions, with a much smaller estimation error and perfect selection accuracy. By contrast, OLS, ENV, and HOLRR yield much larger estimation error and could only generate dense graph estimates. Sparse OLS, on the other hand, produce many false positive links and a large error too. In the second scenario where the truth is a hub graph, our method again outperforms the alternative solutions. This time Sparse OLS misses many true positive links, which again results in a much larger estimation error. In the third scenario of the small-world graph, our model assumption is not satisfied. However our method still manages to achieve a reasonable estimation error and graph recovery. Although HOLRR obtains the smallest estimation error for this

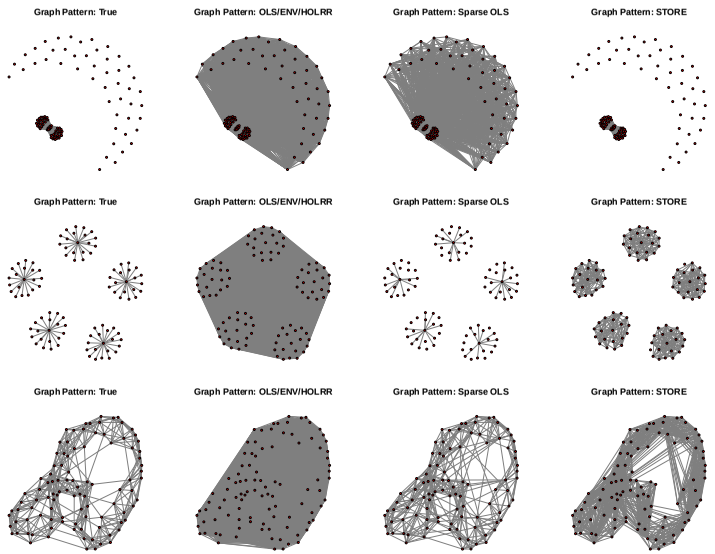


Figure 2: Symmetric matrix response example. Top row: the random graph, middle row: the hub graph, bottom row: the small-world graph. The first column: the true graph pattern; The second column: the estimated graph from OLS, ENV, and HOLRR; The third column: the estimated graph from Sparse OLS; The forth column: the estimated graph from our method STORE.

example, its graph estimate is completely dense and the corresponding FPR is one. Sparse OLS recoveres the graph pattern reasonably well, but still misses many true positive links, and yields a smaller TPR than our method.

6. Real Data Analysis

In this section, we analyze a real neuroimaging data to illustrate our proposed method. The data is from the Autism Brain Imaging Data Exchange (ABIDE), a study for autism spectrum disorder (ASD) (Di Martino et al., 2014). ASD is an increasingly prevalent neurodevelopmental disorder. It is characterized by symptoms such as social difficulties, communication deficits, stereotyped behaviors and cognitive delays (Rudie et al., 2013). It is of scientific importance to understand how brain functional architecture and brain connectivity pattern differ between subjects with ASD and the normal controls. After removing those with missing values and poor image quality, the data set consists of the resting-state

functional magnetic resonance imaging (fMRI) of 795 subjects, of which 362 have ASD, and 433 are the normal controls. We took the fMRI image as the response, and the ASD status indicator (1 = ASD and 0 = normal control), age and sex as covariates. For each fMRI image, there are two forms of data: one is a 3D tensor and another is a 2D symmetric matrix. Correspondingly, we have fitted two separate tensor response regression models.

6.1 3D FALFF Tensor Response

The first form of the fMRI data is a third-order tensor of fractional amplitude of low-frequency fluctuations (fALFF). fALFF is a metric reflecting the percentage of power spectrum within low-frequency domain ($0.01 - 0.1$ Hz). It characterizes the intensity of spontaneous brain activities, and provides a measure of functional architecture of the brain (Shi and Kang, 2015). It is calculated at each individual image voxel, and forms a 3D tensor with dimension $91 \times 109 \times 91$. We applied our STORE method as well as four competing methods (OLS, Sparse OLS, ENV, and HOLRR) to this general 3D tensor response.

Figure 3 shows the estimated coefficient tensors, overlaid on a brain image of a randomly selected subject, by all five methods. Among those, OLS, ENV, and HOLRR perform no region selection, whereas Sparse OLS yields a sparse estimate that is difficult to interpret. By contrast, the STORE estimate reveals a small number of brain regions that exhibit a clear difference between the ASD and normal control groups. Furthermore, we mapped those nonzero entries of our estimate to the AAL atlas. Table 3 reports the names of those identified regions and the corresponding coefficient entries in each region. Our results in general agree with the autism literature. For instance, we have found that multiple cerebellum (`Cerebellum`) regions show distinctive patterns between the ASD and control groups. The cerebellum has long been known for its importance in motor learning, coordination, and more recently, cognitive functions and affective regulation. It has emerged as one of the key brain regions affected in autism (Becker and Stoodley, 2013). Moreover, we identified the superior parietal lobule (`Parietal_Sup`) and precuneus (`Precuneus`), which agrees with Travers et al. (2015), in that they found individuals with ASD showed decreased activation in the superior parietal lobule and precuneus relative to individuals with typical development, suggesting that the superior parietal lobule may play an important role in motor learning and repetitive behavior in individuals with ASD.

6.2 2D Symmetric Partial Correlation Matrix Response

The second form of the fMRI data is a second-order symmetric partial correlation matrix. The partial correlation between brain regions reveals the synchronization of brain systems, and offers an alternative measure of the intrinsic brain functional architecture. It is evaluated, however, at a different scale from fALFF. Specifically, it is computed between pairs of pre-specified brain regions, each of which is a collection of brain voxels. It forms a 2D symmetric matrix with dimension 116×116, which corresponds to 116 regions-of-interest under the commonly used Anatomical Automatic Labeling (AAL) atlas (Tzourio-Mazoyer et al., 2002). We applied our STORE method to this 2D symmetric matrix response. As none of those competing methods incorporates the symmetry, we chose not to apply them in this analysis.

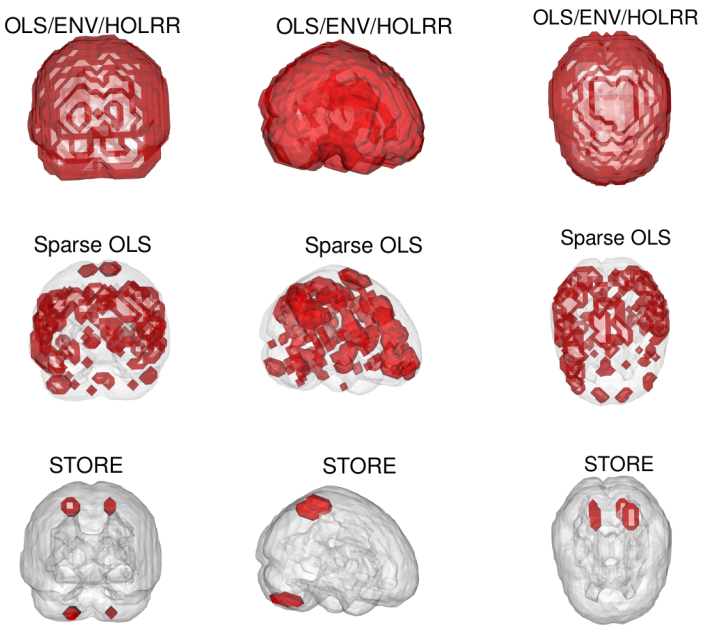


Figure 3: Analysis of the ABIDE data. The response is a 3D brain image tensor. Shown are the estimated coefficient tensor overlaid on a randomly selected brain image. Top row: OLS/ENV/HOLRR, middle row: Sparse OLS, bottom row: STORE. Left column: front view, middle column: side view, right column: top view.

Figure 4 reports top 20 identified links from our method. The red links correspond to those that are more likely to be absent in the ASD group, whereas the black ones are those more likely to be present in ASD. Table 4 reports the names of those links and their relative connection strengthens in the network, as reflected by the corresponding entries in the matrix coefficient $\widehat{\mathcal{B}}$. Many different connectivity patterns concentrate on the left middle frontal gyrus (`Frontal_Mid_L`) and the temporal lobe (`Temporal`), and such findings again agrees with the literature (Kana et al., 2006; Di Martino et al., 2014; Ha et al., 2015).

Acknowledgments

We would like to thank the Action Editor Animashree Anandkumar and the anonymous referees for helpful comments on an earlier manuscript. Li's research was partially supported by NSF grant DMS-1613137 and NIH grant AG034570.

Important Regions	$\widehat{\mathcal{B}}_{i,j,k}$
Postcentral_L	0.0033
Cerebelum_9_L	0.0028
Precuneus_L	0.0025
Occipital_Sup_R	0.0025
Parietal_Inf_L	0.0025
Cerebelum_8_L	0.0020
Cerebelum_8_R	0.0020
Precuneus_R	0.0017
Parietal_Sup_R	0.0017
Cerebelum_9_R	0.0016
Parietal_Sup_L	0.0016

Table 3: Analysis of the ABIDE data. The response is the third-order fALFF tensor. Reported are the identified brain regions by our method that exhibit clear difference in fALFF between the ASD and normal control.

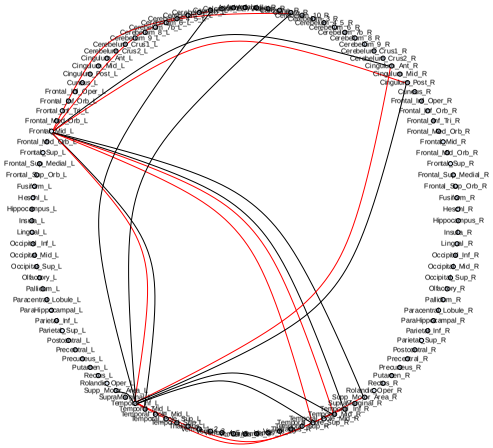


Figure 4: Analysis of the ABIDE data. The response is a symmetric partial correlation matrix. Shown are the top 20 links found by our method. Red links are more likely to be absent in the ASD group and black link are more likely to be present in the ASD group.

Connected Regions	$\widehat{\mathcal{B}}_{i,j}$
Frontal_Mid_L ⟷ Temporal_Inf_L	-0.0034
Cingulum_Ant_R ⟷ Temporal_Inf_L	-0.0027
Temporal_Inf_L ⟷ Temporal_Inf_R	-0.0024
Supp_Motor_Area_R ⟷ Temporal_Inf_L	-0.0023
Temporal_Mid_L ⟷ Temporal_Inf_L	-0.0023
Frontal_Mid_L ⟷ Temporal_Mid_R	-0.0020
Frontal_Mid_L ⟷ Caudate_R	-0.0019
Frontal_Mid_L ⟷ Temporal_Pole_Sup_R	-0.0018
Frontal_Mid_L ⟷ Cingulum_Post_R	-0.0017
Frontal_Mid_L ⟷ Caudate_L	-0.0016
Frontal_Mid_L ⟷ Supp_Motor_Area_R	0.0017
Frontal_Mid_L ⟷ Temporal_Mid_L	0.0017
Frontal_Mid_L ⟷ Temporal_Inf_R	0.0019
Frontal_Mid_L ⟷ Cingulum_Ant_R	0.0021
Supp_Motor_Area_L ⟷ Temporal_Inf_L	0.0021
Caudate_L ⟷ Temporal_Inf_L	0.0021
Cingulum_Post_R ⟷ Temporal_Inf_L	0.0023
Temporal_Pole_Sup_R ⟷ Temporal_Inf_L	0.0024
Caudate_R ⟷ Temporal_Inf_L	0.0026
Temporal_Mid_R ⟷ Temporal_Inf_L	0.0026

Table 4: Analysis of the ABIDE data. The response is a symmetric partial correlation matrix. Reported are the identified top 20 links by our method.

Appendix A.

We divide the appendix into three parts. First we begin with a number of auxiliary lemmas. Next we provide detailed proofs of the main theorem. Then we present the technical proofs of all lemmas and corollaries.

A1 Auxiliary Lemmas

Lemma 9 *For any matrix $\mathbf{Y}_i \in \mathbb{R}^{d_1 \times d_2}$, with $i = 1, \ldots, n$, and vectors $\gamma = (\gamma_1, \ldots, \gamma_n)^\top \in \mathbb{R}^n$, $\alpha \in \mathbb{R}^{d_1}$ and $\beta \in \mathbb{R}^{d_2}$, we have*

$$\arg \min_{\beta, \|\beta\|_2=1} \frac{1}{n} \sum_{i=1}^n \gamma_i^2 \left\| \mathbf{Y}_i - \alpha\beta^\top \right\|_F^2 = \arg \min_{\beta, \|\beta\|_2=1} \left\| \frac{1}{n} \sum_{i=1}^n \gamma_i^2 \mathbf{Y}_i - \alpha\beta^\top \right\|_F^2.$$

The proof of Lemma 9 is provided in Section A3.3 in the online supplement.

We next introduce the Slepian's lemma (Slepian, 1962), which provides a comparison between the supremums of two Gaussian processes.

Lemma 10 *(Slepian, 1962, Slepian's Lemma) Denote two centered Gaussian processes $\{G_s, s \in \mathcal{S}\}$ and $\{H_s, s \in \mathcal{S}\}$. Assume that both processes are almost surely bounded and for each $s, t \in \mathcal{S}$, $\mathbb{E}(G_s - G_t)^2 \leq E(H_s - H_t)^2$, then we have*

$$\mathbb{E}\left[\sup_{s \in \mathcal{S}} G_s\right] \leq \mathbb{E}\left[\sup_{s \in \mathcal{S}} H_s\right].$$

Moreover, if $\mathbb{E}(G_s^2) = E(H_s^2)$ for all $s \in \mathcal{S}$, then we have, for each $x > 0$,

$$\mathbb{P}\left[\sup_{s \in \mathcal{S}} G_s > x\right] \leq \mathbb{P}\left[\sup_{s \in \mathcal{S}} H_s > x\right].$$

The next result provides a concentration of Lipschitz functions of Gaussian random variables (Massart, 2003) .

Lemma 11 *(Massart, 2003, Theorem 3.4) Let $\mathbf{v} \in \mathbb{R}^d$ be a Gaussian random variable such that $\mathbf{v} \sim N(0, \mathbf{I}_d)$. Assuming $g(\mathbf{v}) \in \mathbb{R}$ to be a Lipschitz function such that $|g(\mathbf{v}_1) - g(\mathbf{v}_2)| \leq L\|\mathbf{v}_1 - \mathbf{v}_2\|_2$ for any $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^d$, then we have, for each $t > 0$,*

$$\mathbb{P}\left[|g(\mathbf{v}) - \mathbb{E}[g(\mathbf{v})]| \geq t\right] \leq 2 \exp\left(-\frac{t^2}{2L^2}\right).$$

The next lemma provides an upper bound of the Gaussian width of the unit ball for the sparsity regularizer (Raskutti and Yuan, 2016).

Lemma 12 *(Raskutti and Yuan, 2016, Lemma 2) For a tensor $\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, denote its regularizer $R(\mathcal{T}) = \sum_{j_1} \sum_{j_2} \sum_{j_3} |\mathcal{T}_{j_1, j_2, j_3}|$. Define the unit ball of this regularizer as $B_R(1) := \{\mathcal{T} \in \mathbb{R}^{d_1 \times d_2 \times d_3} | R(\mathcal{T}) \leq 1\}$. For a Gaussian tensor $\mathcal{G} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ whose entries are independent standard normal random variables, we have*

$$\mathbb{E}\left[\sup_{\mathcal{T} \in B_R(1)} \langle \mathcal{T}, \mathcal{G} \rangle\right] \leq c\sqrt{\log(d_1 d_2 d_3)},$$

for some bounded constant $c > 0$.

The next lemma links the hard thresholding sparsity and the L_1-penalized sparsity.

Lemma 13 *For any vectors $\mathbf{u} \in \mathbb{R}^{d_1}, \mathbf{v} \in \mathbb{R}^{d_2}, \mathbf{w} \in \mathbb{R}^{d_3}$ satisfying $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = \|\mathbf{w}\|_2 = 1$, $\|\mathbf{u}\|_0 \leq s, \|\mathbf{v}\|_0 \leq s$, and $\|\mathbf{w}\|_0 \leq s$, denoting $\mathcal{A} := \mathbf{u} \circ \mathbf{v} \circ \mathbf{w}$, we have the bound of the L_1-norm regularizer*

$$\|\mathcal{A}\|_1 := \sum_{j_1} \sum_{j_2} \sum_{j_3} |\mathcal{A}_{j_1 j_2 j_3}| \leq s^{3/2}.$$

A2 Proof of Theorem 6

We divide the proof of Theorem 6 into two major steps: characterization of the estimation error in Step 1 in Algorithm 1, then the estimation error in Step 2. Each leads to a new theorem. Then we complete the proof of Theorem 6 by iteratively applying those results.

A2.1 Estimation Error in Step 1 of Algorithm 1

We first derive the estimation error in Step 1 of our Algorithm 1. The key idea is to transform the problem into a standard sparse tensor decomposition problem, then incorporate the existing contracting results obtained in Sun et al. (2017) to derive the final error bound of the estimator in Step 1. In the following derivation, for simplicity, we assume $K = 1$. This does not lose generality, since K is assumed to be a constant, and it does not affect the final error rate. In this case, the true model reduces to $\mathcal{Y}_i = w^*(\boldsymbol{\beta}_{k,m+1}^{*\top} \mathbf{x}_i) \boldsymbol{\beta}_{k,1}^* \circ \cdots \circ \boldsymbol{\beta}_{k,m}^* + \mathcal{E}_i$, for each $i = 1, \ldots, n$. Based on the Step 1 of our algorithm, if the true parameter $\boldsymbol{\beta}_{k,m+1}^*$ is available, we are solving the following sparse tensor decomposition problem,

$$\bar{\mathcal{R}}_k = \mathcal{T} + \bar{\mathcal{E}}, \tag{A1}$$

where the true tensor $\mathcal{T} = w_k^* \boldsymbol{\beta}_{k,1}^* \circ \cdots \circ \boldsymbol{\beta}_{k,m}^*$, the oracle response and the oracle error are, respectively,

$$\bar{\mathcal{R}}_k := \frac{1}{n} \sum_{i=1}^n \mathcal{R}_{ik} = \frac{1}{n} \sum_{i=1}^n \frac{\mathcal{Y}_i}{\boldsymbol{\beta}_{k,m+1}^{*\top} \mathbf{x}_i} \text{ and } \bar{\mathcal{E}} = \frac{1}{n} \sum_{i=1}^n \frac{\mathcal{E}_i}{\boldsymbol{\beta}_{k,m+1}^{*\top} \mathbf{x}_i}.$$

In practice, however, we only have an estimator $\widehat{\boldsymbol{\beta}}_{k,m+1}$. Hence the sparse tensor decomposition method in Step 1 of our algorithm is actually applied to

$$\widehat{\mathcal{R}}_k = \mathcal{T} + \widehat{\mathcal{E}} \tag{A2}$$

with the response tensor

$$\widehat{\mathcal{R}}_k := \frac{1}{n} \sum_{i=1}^n \frac{\mathcal{Y}_i}{\widehat{\boldsymbol{\beta}}_{k,m+1}^\top \mathbf{x}_i}$$

Therefore, according to $(A1)$ and $(A2)$, we have the explicit form of $\widehat{\mathcal{E}}$,

$$\begin{aligned} \widehat{\mathcal{E}} &= \widehat{\mathcal{R}}_k - \bar{\mathcal{R}}_k + \bar{\mathcal{E}} \\ &= \underbrace{\frac{1}{n} \sum_{i=1}^n \frac{(\boldsymbol{\beta}_{k,m+1}^* - \widehat{\boldsymbol{\beta}}_{k,m+1})^\top \mathbf{x}_i}{\widehat{\boldsymbol{\beta}}_{k,m+1}^\top \mathbf{x}_i \boldsymbol{\beta}_{k,m+1}^{*\top} \mathbf{x}_i} \mathcal{Y}_i}_{I_1} + \underbrace{\frac{1}{n} \sum_{i=1}^n \frac{\mathcal{E}_i}{\boldsymbol{\beta}_{k,m+1}^{*\top} \mathbf{x}_i}}_{I_2} \end{aligned} \tag{A3}$$

Before we derive the estimation error of the estimator based on $(A2)$, we introduce a lemma for deriving the error bound of a general sparse tensor decomposition.

Assumption 14 *The decomposition components are incoherent such that*

$$\zeta := \max_{i\neq j}\{|\langle \boldsymbol{\beta}^*_{i,1}, \boldsymbol{\beta}^*_{j,1}\rangle|, |\langle \boldsymbol{\beta}^*_{i,2}, \boldsymbol{\beta}^*_{j,2}\rangle|, |\langle \boldsymbol{\beta}^*_{i,3}, \boldsymbol{\beta}^*_{j,3}\rangle|\} \leq \frac{C_0}{\sqrt{d_0}},$$

with $d_0 = \max\{d_{01}, d_{02}, d_{03}\}$*, and for any* j*,* $\|\sum_{i\neq j} w_i \langle \boldsymbol{\beta}^*_{i,1}, \boldsymbol{\beta}^*_{j,1}\rangle \langle \boldsymbol{\beta}^*_{i,2}, \boldsymbol{\beta}^*_{j,2}\rangle \boldsymbol{\beta}^*_{i,3}\| \leq C_1 w^*_{\max}\sqrt{K}\zeta$*. Moreover, the matrices* $\mathbf{A} := [\boldsymbol{\beta}^*_{i,1}, \cdots, \boldsymbol{\beta}^*_{K,1}]$*,* $\mathbf{B} := [\boldsymbol{\beta}^*_{i,2}, \cdots, \boldsymbol{\beta}^*_{K,2}]$*, and* $\mathbf{C} := [\boldsymbol{\beta}^*_{i,3}, \cdots, \boldsymbol{\beta}^*_{K,3}]$ *satisfy that* $\max\{\|\mathbf{A}\|, \|\mathbf{B}\|, \|\mathbf{C}\|\} \leq 1 + C_2\sqrt{K/d_0}$ *for some positive constants* C_0, C_1, C_2*.*

Define a function $f(\epsilon; K, d_0)$ as

$$f(\epsilon; K, d_0) := \frac{2C_0}{\sqrt{d_0}}\Big(1 + C_2\sqrt{\frac{K}{d_0}}\Big)^2\epsilon + C_1\frac{\sqrt{K}}{d_0} + C_3\epsilon^2,$$

for some constants $C_0, C_1, C_2, C_3 > 0$. When $K = o(d_0^{3/2})$, the first two terms of $f(\epsilon; K, d_0)$ converge to 0 and the last term is the contracting term.

Lemma 15 *(Sun et al., 2017, Lemma S.4.1) Consider the model* $\widehat{\mathcal{T}} = \mathcal{T} + \mathcal{E}$ *where the low-rank and sparse components of* $\mathcal{T}$ *satisfy Assumption 14, and assume* $\|\mathcal{T}\| \leq C_3 w^*_{\max}$ *and* $K = o(d_0^{3/2})$*. In addition, assume the estimators* $\widehat{\boldsymbol{\beta}}_{j,1}$ *and* $\widehat{\boldsymbol{\beta}}_{j,2}$ *satisfy* $D(\widehat{\boldsymbol{\beta}}_{j,1}, \boldsymbol{\beta}^*_{j,1}) \leq \epsilon$ *and* $D(\widehat{\boldsymbol{\beta}}_{j,2}, \boldsymbol{\beta}^*_{j,2}) \leq \epsilon$ *for some* $j \in [K]$*. If the perturbation error* $\eta(\mathcal{E}, d_0 + s)$*, with* $s \geq d_0$*, is small enough such that* $\eta(\mathcal{E}, d_0 + s) < w_j(1-\epsilon^2) - w^*_{\max} f(\epsilon; K, d_0)$*, then the update* $\widehat{\boldsymbol{\beta}}_{j,3}$ *satisfies, with high probability,*

$$D(\widehat{\boldsymbol{\beta}}_{j,3}, \boldsymbol{\beta}^*_{j,3}) \leq \frac{\sqrt{5}w^*_{\max} f(\epsilon; K, d_0) + \sqrt{5}\eta(\mathcal{E}, d_0 + s)}{w_j(1-\epsilon^2) - w^*_{\max} f(\epsilon; K, d_0) - \eta(\mathcal{E}, d_0 + s)}.$$

If we further assume $D(\widehat{\boldsymbol{\beta}}_{j,3}, \boldsymbol{\beta}^*_{j,3}) \leq \epsilon$*, then the update* $\widehat{w} = \widehat{\mathcal{T}} \times_1 \widehat{\boldsymbol{\beta}}_{j,1} \times_2 \widehat{\boldsymbol{\beta}}_{j,2} \times_3 \widehat{\boldsymbol{\beta}}_{j,3}$ *satisfies, with high probability,* $|\widehat{w} - w_j| \leq 2w_j\epsilon^2 + w^*_{\max} f(\epsilon; K, d_0) + \eta(\mathcal{E}, d_0 + s)$*.*

By verifying the conditions in Lemma 15, we are able to compute the estimation error in Step 1 of our algorithm. For simplicity, we consider the case when $m = 3$, while the extension to a more general m is straightforward.

Lemma 16 *Assume* $\|\mathcal{T}\| \leq C_1 w^*_{\max}$*,* $D(\widehat{\boldsymbol{\beta}}_{k,1}, \boldsymbol{\beta}^*_{k,1}) \leq \epsilon$*,* $D(\widehat{\boldsymbol{\beta}}_{k,2}, \boldsymbol{\beta}^*_{k,2}) \leq \epsilon$*, and*

$$\epsilon < \min\left\{\sqrt{\frac{w^*_{\min}}{2(w^*_{\min} + w^*_{\max}C_1)}}, \frac{w^*_{\min}}{4\sqrt{5}w^*_{\max}C_1}\right\}.$$

Assume the error tensor satisfies $\eta(\widehat{\mathcal{E}}, d_0 + s) \leq w^*_{\min}/4$ *with* $s \geq d_0$*, where* $\widehat{\mathcal{E}}$ *is as defined in* $(A3)$*. Then we have*

$$D(\widehat{\boldsymbol{\beta}}_{k,3}, \boldsymbol{\beta}^*_{k,3}) \leq \kappa_1\epsilon + \frac{4\sqrt{5}}{w^*_{\min}}\eta(\widehat{\mathcal{E}}, d_0 + s),$$

where the contraction coefficient

$$\kappa_1 := \frac{4\sqrt{5}w^*_{\max}C_1}{w^*_{\min}}\epsilon < \frac{4\sqrt{5}w^*_{\max}C_1}{w^*_{\min}}\min\left\{\sqrt{\frac{w^*_{\min}}{2(w^*_{\min} + w^*_{\max}C_1)}}, \frac{w^*_{\min}}{4\sqrt{5}w^*_{\max}C_1}\right\} \in (0,1).$$

The proof of Lemma 16 is provided in Section A3.5. Based on Lemma 16, to compute the closed-form error rate in Step 1 of our Algorithm, the remaining step is to compute $\eta(\widehat{\mathcal{E}}, s)$, since $\eta(\widehat{\mathcal{E}}, d_0 + s) \leq 2\eta(\widehat{\mathcal{E}}, s)$ by noting that $s \geq d_0$. Here the explicit form of $\widehat{\mathcal{E}}$ is defined in $(A3)$. Again we only consider $m = 3$ for simplicity, and the proof for a general m follows straightforwardly.

Lemma 17 *Assume the conditions in Lemma 16 hold. Assume that* $\|w^*_k \boldsymbol{\beta}^*_{k,1} \circ \cdots \circ \boldsymbol{\beta}^*_{k,m}\| \leq C_1$*,* $\|\mathbf{x}_i\| \leq C_2$*, and* $|\boldsymbol{\beta}^{*\top}_{k,m+1}\mathbf{x}_i| \geq C_3$ *for each* $i = 1, \ldots, n$*, for some positive constants* C_1, C_2, C_3*. If* $\epsilon \leq C_3/(2C_2)$*, then we have*

$$\eta(\widehat{\mathcal{E}}, s) \leq \underbrace{\frac{2C_2\eta\left(\bar{\mathcal{E}}, s\right)}{C_3^2}}_{\kappa_2}\epsilon + \frac{1}{C_3}\eta\left(\bar{\mathcal{E}}, s\right).$$

where $\bar{\mathcal{E}} := \frac{1}{n}\sum_{i=1}^n \mathcal{E}_i$*.*

The proof of Lemma 17 is provided in Section A3.6. Combining Lemma 16 and Lemma 17, we obtain the final contraction result of Step 1 in Algorithm 1.

Theorem 18 *(Contraction result in Step 1 in Algorithm 1) Assume* $D(\widehat{\boldsymbol{\beta}}_{k,1}, \boldsymbol{\beta}^*_{k,1}) \leq \epsilon$*,* $D(\widehat{\boldsymbol{\beta}}_{k,2}, \boldsymbol{\beta}^*_{k,2}) \leq \epsilon$*, and* $D(\widehat{\boldsymbol{\beta}}_{k,4}, \boldsymbol{\beta}^*_{k,4}) \leq \epsilon$*, with*

$$\epsilon < \min\left\{\sqrt{\frac{w^*_{\min}}{2(w^*_{\min} + w^*_{\max}C_1)}}, \frac{w^*_{\min}}{4\sqrt{5}w^*_{\max}C_1}, \frac{C_3}{2C_2}\right\}.$$

Assume the error tensor satisfies $\eta(n^{-1}\sum_{i=1}^n \mathcal{E}_i, d_0 + s) \leq w^*_{\min}/4$ *with* $s \geq d_0$*. Then we have*

$$D(\widehat{\boldsymbol{\beta}}_{k,3}, \boldsymbol{\beta}^*_{k,3}) \leq (\kappa_1 + \kappa_2)\epsilon + \frac{4\sqrt{5}}{C_3 w^*_{\min}}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right).$$

A2.2 Estimation Error in Step 2 of Algorithm 1

Next we derive the estimation error in Step 2 of our algorithm. That is, we aim to bound $D(\widehat{\boldsymbol{\beta}}_{k,m+1}, \boldsymbol{\beta}^*_{k,m+1})$ given the estimators $\widehat{w}_k, \widehat{\boldsymbol{\beta}}_{k,1}, \ldots, \widehat{\boldsymbol{\beta}}_{k,m}$.

Denote $\widehat{\mathcal{A}}_k = \widehat{w}_k\widehat{\boldsymbol{\beta}}_{k,1} \circ \cdots \circ \widehat{\boldsymbol{\beta}}_{k,m}$, and $\widehat{\mathcal{T}}_i = \mathcal{Y}_i - \sum_{k'\neq k, k'\in[K]} \widehat{w}_{k'}(\widehat{\boldsymbol{\beta}}^\top_{k',m+1}\mathbf{x}_i)\widehat{\boldsymbol{\beta}}_{k',1} \circ \cdots \circ \widehat{\boldsymbol{\beta}}_{k',m}$, and the closed-form estimator in Step 2 of our algorithm is

$$\widehat{\boldsymbol{\beta}}_{k,m+1} = \left(\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top\right)^{-1}\frac{n^{-1}\sum_{i=1}^n \langle \widehat{\mathcal{T}}_i, \widehat{\mathcal{A}}_k\rangle \mathbf{x}_i}{\|\widehat{\mathcal{A}}_k\|_F^2}.$$

Theorem 19 *(Contraction result in Step 2 in Algorithm 1) Under Assumption 3, and the assumption that the initialization error satisfies* $\epsilon \le w^*_{\min}/2$, *if* $|\widehat{w}_k - w^*_k| \le \epsilon$, $D(\widehat{\boldsymbol{\beta}}_{k,1}, \boldsymbol{\beta}^*_{k,1}) \le \epsilon, \ldots, D(\widehat{\boldsymbol{\beta}}_{k,m}, \boldsymbol{\beta}^*_{k,m}) \le \epsilon$, *then we have*

$$D(\widehat{\boldsymbol{\beta}}_{k,m+1}, \boldsymbol{\beta}^*_{k,m+1}) \le \kappa_3 \epsilon + \frac{\widetilde{C}}{\sqrt{n}},$$

where $\kappa_3 := 2/w^*_{\min} + 6\sqrt{2}$ *and* $\widetilde{C}$ *is a positive constant as defined in* (A7).

Proof: For simplicity, we only prove for $K = 1$ and $m = 3$. The derivation for a general K and m follows similarly.

Denote $\mathcal{A}^*_k := w^*_k \boldsymbol{\beta}^*_{k,1} \circ \boldsymbol{\beta}^*_{k,2} \circ \boldsymbol{\beta}^*_{k,3}$. The true model reduces to $\mathcal{Y}_i = (\boldsymbol{\beta}^{*\top}_{k,4}\mathbf{x}_i)\mathcal{A}^*_k + \mathcal{E}_i$, for each $i = 1, \ldots, n$. Denote $\Omega := (n^{-1}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i^\top)^{-1}$. Given $\widehat{\mathcal{A}}_k = \widehat{w}_k \widehat{\boldsymbol{\beta}}_{k,1} \circ \cdots \circ \widehat{\boldsymbol{\beta}}_{k,3}$, and $\widehat{\mathcal{T}}_i = \mathcal{Y}_i$ when $K = 1$, we have the following simplification of $\widehat{\boldsymbol{\beta}}_{k,4}$,

$$\begin{aligned}
\widehat{\boldsymbol{\beta}}_{k,4} &= \Omega \frac{\sum_{i=1}^n \langle \widehat{\mathcal{T}}_i, \widehat{\mathcal{A}}_k\rangle \mathbf{x}_i}{n\|\widehat{\mathcal{A}}_k\|_F^2} = \Omega \frac{\sum_{i=1}^n \langle (\boldsymbol{\beta}^{*\top}_{k,4}\mathbf{x}_i)\mathcal{A}^*_k + \mathcal{E}_i, \widehat{\mathcal{A}}_k\rangle \mathbf{x}_i}{n\|\widehat{\mathcal{A}}_k\|_F^2} \\
&= \frac{\langle \mathcal{A}^*_k, \widehat{\mathcal{A}}_k\rangle}{\|\widehat{\mathcal{A}}_k\|_F^2}\boldsymbol{\beta}^*_{k,4} + \frac{\Omega\sum_{i=1}^n \langle \mathcal{E}_i, \widehat{\mathcal{A}}_k\rangle \mathbf{x}_i}{n\|\widehat{\mathcal{A}}_k\|_F^2}, \qquad \text{(A4)}
\end{aligned}$$

where the first part in (A4) is due to the fact that $n^{-1}\sum_{i=1}^n (\boldsymbol{\beta}^{*\top}_{k,4}\mathbf{x}_i)\Omega\mathbf{x}_i = n^{-1}\sum_{i=1}^n \Omega\mathbf{x}_i\mathbf{x}_i^\top\boldsymbol{\beta}^*_{k,4} = \boldsymbol{\beta}^*_{k,4}$. Therefore, the error bound of $\widehat{\boldsymbol{\beta}}_{k,4}$ can be simplified as

$$\begin{aligned}
\left\|\widehat{\boldsymbol{\beta}}_{k,4} - \boldsymbol{\beta}^*_{k,4}\right\|_2 &= \left\| \frac{\langle \mathcal{A}^*_k, \widehat{\mathcal{A}}_k\rangle}{\|\widehat{\mathcal{A}}_k\|_F^2}\boldsymbol{\beta}^*_{k,4} - \boldsymbol{\beta}^*_{k,4} + \frac{\Omega\sum_{i=1}^n \langle \mathcal{E}_i, \widehat{\mathcal{A}}_k\rangle \mathbf{x}_i}{n\|\widehat{\mathcal{A}}_k\|_F^2} \right\|_2, \\
&\le \underbrace{\left| \frac{\langle \mathcal{A}^*_k, \widehat{\mathcal{A}}_k\rangle - \|\widehat{\mathcal{A}}_k\|_F^2}{\|\widehat{\mathcal{A}}_k\|_F^2} \right|}_{(I)} + \underbrace{\frac{\|\Omega\sum_{i=1}^n \langle \mathcal{E}_i, \widehat{\mathcal{A}}_k\rangle \mathbf{x}_i\|_2}{n\|\widehat{\mathcal{A}}_k\|_F^2}}_{(II)}. \qquad \text{(A5)}
\end{aligned}$$

In the following lemma, we bound the two terms in (A5) to obtain the final error bound.

Lemma 20 *Under the Conditions in Theorem 19, we have*

$$\begin{aligned}
(I) &\le \left[\frac{2}{w^*_{\min}} + 6\sqrt{2}\right]\epsilon, \qquad \text{(A6)} \\
(II) &\le \underbrace{\frac{\max_i \|\Omega\mathbf{x}_i\|_2 \cdot \mathbb{E}\left[\|\mathcal{G}\|_F\right]}{\|\widehat{\mathcal{A}}_k\|_F}}_{\widetilde{C}} \cdot \frac{1}{\sqrt{n}}. \qquad \text{(A7)}
\end{aligned}$$

Finally, combining the results in (A6) and (A7) leads to the final bound of $\|\widehat{\boldsymbol{\beta}}_{k,4} - \boldsymbol{\beta}^*_{k,4}\|_2$, and hence the bound of $D(\widehat{\boldsymbol{\beta}}_{k,4}, \boldsymbol{\beta}^*_{k,4})$. □

A2.3 Proof of Theorem 6

Now we complete the proof of Theorem 6, by iteratively applying the contraction results in Theorems 18 and 19 for the two steps of Algorithm 1. In iteration $t = 1$, given the initializations $\widehat{\boldsymbol{\beta}}^{(0)}_{k,j}$ and $\widehat{w}^{(0)}$ with initialization error ϵ, Theorem 18 implies that

$$D(\widehat{\boldsymbol{\beta}}^{(1)}_{k,3}, \boldsymbol{\beta}^*_{k,3}) \le (\kappa_1 + \kappa_2)\epsilon + \frac{4\sqrt{5}}{C_3 w^*_{\min}}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right),$$

where $\kappa_1 + \kappa_2 < 1$ according to Assumptions 4 and 5. The second term converges to zero as sample size increases. Therefore, for a sufficiently large sample size, the above error bound is smaller than ϵ. By a similar derivation, the same error bound holds for $D(\widehat{\boldsymbol{\beta}}^{(1)}_{k,1}, \boldsymbol{\beta}^*_{k,1})$, $D(\widehat{\boldsymbol{\beta}}^{(1)}_{k,2}, \boldsymbol{\beta}^*_{k,2})$, and $|\widehat{w}^{(1)}_k - w^*_k|$. In Step 2 of the algorithm, applying Theorem 19 based on the above estimators in iteration $t = 1$, we obtain that

$$D(\widehat{\boldsymbol{\beta}}^{(1)}_{k,4}, \boldsymbol{\beta}^*_{k,4}) \le \kappa_3(\kappa_1 + \kappa_2)\epsilon + \kappa_3\frac{4\sqrt{5}}{C_3 w^*_{\min}}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right) + \frac{\widetilde{C}}{\sqrt{n}}.$$

Again, the contraction coefficient $\kappa_3(\kappa_1 + \kappa_2) < 1$ according to Assumptions 4 and 5, and the remaining term converges to zero as sample size increases. For $\kappa = (\kappa_1 + \kappa_2)\kappa_3 \in (0, 1)$, by repeatedly applying these derivations, in the t iteration, we obtain that

$$\begin{aligned}
&\max\left\{\max_k \|\widehat{w}^{(t)}_k - w^*_k\|_2, \max_{k,j}\left\{\|\widehat{\boldsymbol{\beta}}^{(t)}_{k,j} - \boldsymbol{\beta}^*_{k,j}\|_2\right\}\right\} \\
\le\ & \kappa^t\epsilon + \frac{1-\kappa^t}{1-\kappa}\frac{4\sqrt{5}}{C_3 w^*_{\min}}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right) + \frac{1-\kappa^{t-1}}{1-\kappa}\frac{\widetilde{C}}{\sqrt{n}} \\
\le\ & \kappa^t\epsilon + \frac{1}{1-\kappa}\max\left\{\frac{4\sqrt{5}}{C_3 w^*_{\min}}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right), \frac{\widetilde{C}}{\sqrt{n}}\right\}.
\end{aligned}$$

This completes the proof of Theorem 6. □

A3 Proofs of Lemmas and Corollaries

This subsection provides all the supporting lemmas and corollaries as well as their proofs.

A3.1 Proof of Lemma 1

To solve (4), we can use the alternating updating method to update one parameter at a time. In particular, for each $j = 1, \ldots, m$, given $\beta^{(j')}_k$ with $j' \ne j$, we solve

$$\widehat{\beta}_{k,j} := \arg \min_{\substack{\beta_{k,j} \\ \|\beta_{k,j}\|_2 = 1, \|\beta_{k,j}\|_0 \le s_j}} \frac{1}{n}\sum_{i=1}^n \alpha_{ik}^2 \left\|\mathcal{R}_i - w_k \beta_{k,1} \circ \cdots \circ \beta_{k,m}\right\|_F^2.$$

According to the matrix representation of tensor operations (Kolda and Bader, 2009; Kim et al., 2014), this optimization problem is equivalent to solve

$$\min_{\substack{\beta_{k,j} \\ \|\beta_{k,j}\|_2 = 1, \|\beta_{k,j}\|_0 \le s_j}} \frac{1}{n}\sum_{i=1}^n \alpha_{ik}^2 \left\|[\mathcal{R}_i]^\top_{(j)} - \mathbf{h}^{(j)}_k \beta^{(j)\top}_k\right\|_F^2, \qquad \text{(A8)}$$

where $[\mathcal{R}_i]_{(j)}$ is the mode-j matricization of the tensor $\mathcal{R}_i$, and

$$\mathbf{h}_k^{(j)} := \beta_{k,m} \odot \cdots \odot \beta_k^{(j+1)} \odot \beta_k^{(j-1)} \odot \cdots \odot \beta_{k,1} \in \mathbb{R}^{\prod_{j' \neq j} d_{j'}}$$

with $\odot$ the Khatri-Rao product. According to Lemma 9, we can conclude that the minimizer of $(A8)$ is also the solution of

$$\min_{\substack{\beta_{k,j} \\ \|\beta_{k,j}\|_2=1, \|\beta_{k,j}\|_0 \leq s_j}} \Big\| [\frac{1}{n} \sum_{i=1}^n \alpha_{ik}^2 \mathcal{R}_i]_{(j)}^\top - \mathbf{h}_k^{(j)} \beta_k^{(j)\top} \Big\|_F^2.$$

This is equivalent to solve

$$\min_{\substack{\beta_{k,j} \\ \|\beta_{k,j}\|_2=1, \|\beta_{k,j}\|_0 \leq s_j}} \Big\| \frac{1}{n} \sum_{i=1}^n \alpha_{ik}^2 \mathcal{R}_i - w_k \beta_{k,1} \circ \cdots \circ \beta_{k,m} \Big\|_F^2.$$

given other parameters $\beta_k^{(j')}$ with $j' \neq j$. Finally, simple algebra leads to the estimator $\widehat{w}_k$. This proves the desirable result in Lemma 1. □

A3.2 Proof of Lemma 2

Denote $[t_i]_{i_1,\ldots,i_m}$ as the $(i_1,\ldots,i_m)$-th entry of the tensor $\mathcal{T}_i$, and denote $a_{i_1,\ldots,i_m}$ as the $(i_1,\ldots,i_m)$-th entry of the tensor $\mathcal{A}_k$. By the definition of the tensor Frobenius norm, we have

$$\frac{\partial \frac{1}{n} \sum_{i=1}^n \Big\| \mathcal{T}_i - \alpha^\top \mathbf{x}_i \mathcal{A}_k \Big\|_F^2}{\partial \alpha} = \frac{1}{n} \sum_{i=1}^n \sum_{i_1,\ldots,i_m} 2([t_i]_{i_1,\ldots,i_m} - \alpha^\top \mathbf{x}_i a_{i_1,\ldots,i_m})(-a_{i_1,\ldots,i_m} \mathbf{x}_i).$$

Setting the above gradient to zero implies that

$$\frac{1}{n} \sum_{i=1}^n \{ \sum_{i_1,\ldots,i_m} [t_i]_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m} \} \mathbf{x}_i = \frac{1}{n} \sum_{i=1}^n \sum_{i_1,\ldots,i_m} \alpha^\top \mathbf{x}_i a_{i_1,\ldots,i_m}^2 \mathbf{x}_i,$$

where the left-hand-side equals $n^{-1} \sum_{i=1}^n \|\mathcal{T}_i * \mathcal{A}_k\|_+ \mathbf{x}_i$, and the right-hand-side equals

$$\{ \sum_{i_1,\ldots,i_m} a_{i_1,\ldots,i_m}^2 \} \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \alpha = \|\mathcal{A}_k\|_F^2 \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \alpha.$$

This proves the desirable conclusion in Lemma 2. □

A3.3 Proof of Lemma 9

We first derive the solution of

$$\arg\min_\beta \frac{1}{n} \sum_{i=1}^n \gamma_i^2 \Big\| \mathbf{Y}_i - \alpha \beta^\top \Big\|_F^2,$$

and then connect it with the optimization problem on the right-hand-side. Denote $[\mathbf{Y}_i]_{sj} \in \mathbb{R}$ as the (s,j)-th entry of the matrix $\mathbf{Y}_i$, and denote $[\mathbf{Y}_i]_j \in \mathbb{R}^{d_1}$ as the j-th column of the matrix $\mathbf{Y}_i$. Note that, for each $j = 1,\ldots,d_2$, solving

$$\frac{\partial \frac{1}{n} \sum_{i=1}^n \gamma_i^2 \Big\| \mathbf{Y}_i - \alpha \beta^\top \Big\|_F^2}{\partial \beta_j} = \frac{1}{n} \sum_{i=1}^n \sum_{s=1}^{d_1} 2\gamma_i^2 ([\mathbf{Y}_i]_{sj} - \alpha_s \beta_j)(-\alpha_s) = 0$$

leads to the solution

$$\widehat{\beta}_j = \frac{\frac{1}{n} \sum_{i=1}^n \sum_{s=1}^{d_1} \gamma_i^2 [\mathbf{Y}_i]_{sj} \alpha_s}{\frac{1}{n} \sum_{i=1}^n \gamma_i^2 \sum_{s=1}^{d_1} \alpha_s^2} = \frac{\frac{1}{n} \sum_{i=1}^n \gamma_i^2 [\mathbf{Y}_i]_j^\top \alpha}{(\frac{1}{n} \sum_{i=1}^n \gamma_i^2) \alpha^\top \alpha}.$$

The solution $\widehat{\beta} = (\widehat{\beta}_1,\ldots,\widehat{\beta}_{d_2})^\top$ is indeed a minimizer to the original optimization problem by noting that the second order derivative of β_j is positive. By solving a similar equation, we get the minimizer of the optimization problem $\min_{\beta_j} \big\| \frac{1}{n} \sum_{i=1}^n \mathbf{Y}_i - \alpha \beta^\top \big\|_F^2$ as

$$\widetilde{\beta}_j = \frac{\frac{1}{n} \sum_{i=1}^n \gamma_i^2 [\mathbf{Y}_i]_j^\top \alpha}{\alpha^\top \alpha}.$$

Note that $\widehat{\beta}_j$ equals to $\widetilde{\beta}_j$ up to a constant $n^{-1} \sum_{i=1}^n \gamma_i^2$. Therefore, normalizing $\widehat{\beta}_j$ and $\widetilde{\beta}_j$ to unit norm leads to the same solution. This completes the proof of Lemma 9. □

A3.4 Proof of Lemma 13

According to the Cauchy-Schwarz inequality, we have $\|\mathbf{u}\|_1 \leq \sqrt{s}\|\mathbf{u}\|_2 = \sqrt{s}$, and $\|\mathbf{v}\|_1 \leq \sqrt{s}$, $\|\mathbf{w}\|_1 \leq \sqrt{s}$. Therefore, $\|\mathcal{A}\|_1 = \|\mathbf{u} \circ \mathbf{v} \circ \mathbf{w}\|_1 \leq \|\mathbf{u}\|_1 \cdot \|\mathbf{v}\|_1 \cdot \|\mathbf{w}\|_1 \leq s^{3/2}$. □

A3.5 Proof of Lemma 16

The key step is to verify the conditions in Lemma 15. In our example, the incoherence condition, Assumption 14, trivially holds, since the incoherence parameter $\zeta = 0$. Therefore, the function $f(\epsilon; K, d_0) = C_3 \epsilon^2$ in our example. According to Lemma 15, we have

$$D(\widehat{\boldsymbol{\beta}}_{k,3}, \boldsymbol{\beta}_{k,3}^*) \leq \frac{\sqrt{5} w_{\max}^* C_3 \epsilon^2 + \sqrt{5} \eta(\widehat{\mathcal{E}}, d_0 + s)}{w_j(1-\epsilon^2) - w_{\max}^* C_3 \epsilon^2 - \eta(\widehat{\mathcal{E}}, d_0 + s)}.$$

We next simplify the denominator $w_j(1-\epsilon^2) - w_{\max}^* C_3 \epsilon^2 - \eta(\widehat{\mathcal{E}}, d_0+s)$. By the assumption on ϵ, we have

$$w_j(1-\epsilon^2) - w_{\max}^* C_3 \epsilon^2 \geq w_{\min}^*/2,$$

which, together with the condition on the error tensor $\eta(\widehat{\mathcal{E}}, d_0 + s)$, implies that

$$w_j(1-\epsilon^2) - w_{\max}^* C_3 \epsilon^2 - \eta(\widehat{\mathcal{E}}, d_0 + s) \geq w_{\min}^*/4.$$

Therefore, we have

$$D(\widehat{\boldsymbol{\beta}}_{k,3}, \boldsymbol{\beta}_{k,3}^*) \leq \frac{4\sqrt{5} w_{\max}^* C_3}{w_{\min}^*} \epsilon^2 + \frac{4\sqrt{5}}{w_{\min}^*} \eta(\widehat{\mathcal{E}}, d_0 + s),$$

which completes the proof of Lemma 16. □

A3.6 Proof of Lemma 17

According to the explicit form in $(A3)$, we have

$$\eta(\widehat{\mathcal{E}}, s) \leq \eta(I_1, s) + \eta(I_2, s).$$

Next we bound the two terms $\eta(I_1, s)$ and $\eta(I_2, s)$ separately.

Bound $\eta(I_1, s)$: Denote $\mathbb{S}_q^d = \{\mathbf{v} \in \mathbb{R}^d \,|\, \|\mathbf{v}\|_2 = 1, \|\mathbf{v}\|_0 \leq q\}$. According to the definiteness of $\eta(\cdot)$, we have

$$\begin{aligned}\eta(I_1, s) &= \sup_{\mathbf{u}\in\mathbb{S}_s^{d_1},\mathbf{v}\in\mathbb{S}_s^{d_2},\mathbf{w}\in\mathbb{S}_s^{d_3}} |I_1 \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w}| \\ &\leq \max_i \left|\frac{(\boldsymbol{\beta}_{k,4}^* - \widehat{\boldsymbol{\beta}}_{k,4})^\top \mathbf{x}_i}{\widehat{\boldsymbol{\beta}}_{k,4}^\top \mathbf{x}_i \boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i}\right| \sup_{\mathbf{u}\in\mathbb{S}_s^{d_1},\mathbf{v}\in\mathbb{S}_s^{d_2},\mathbf{w}\in\mathbb{S}_s^{d_3}} \left|\left(\frac{1}{n}\sum_{i=1}^n \mathcal{Y}_i\right) \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w}\right|.\end{aligned}$$

By the Cauchy Schwarz inequality, we have $\|(\boldsymbol{\beta}_{k,4}^* - \widehat{\boldsymbol{\beta}}_{k,4})^\top \mathbf{x}_i\| \leq \|\boldsymbol{\beta}_{k,4}^* - \widehat{\boldsymbol{\beta}}_{k,4}\|\|\mathbf{x}_i\| \leq C_2\|\boldsymbol{\beta}_{k,4}^* - \widehat{\boldsymbol{\beta}}_{k,4}\| \leq C_2\epsilon$. Moreover, note that $|\widehat{\boldsymbol{\beta}}_{k,4}^\top \mathbf{x}_i \boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i| = |\boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i + (\widehat{\boldsymbol{\beta}}_{k,4} - \boldsymbol{\beta}_{k,4}^*)^\top \mathbf{x}_i||\boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i|$. When the element of $\mathbf{x}_i$ is bounded for each $i = 1, \ldots, n$ and the dimension d_{m+1} of $\mathbf{x}_i$ is fixed, we have $\|\mathbf{x}_i\| \leq C_2$. Then by the assumption that $|\boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i| \geq C_3$ and $\|\mathbf{x}_i\| \leq C_2$, we have

$$|\widehat{\boldsymbol{\beta}}_{k,4}^\top \mathbf{x}_i \boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i| \geq (C_3 - C_2\|\widehat{\boldsymbol{\beta}}_{k,4} - \boldsymbol{\beta}_{k,4}^*\|)C_3 \geq (C_3 - C_2\epsilon)C_3 \geq C_3^2/2,$$

where the last inequality is due to the assumption that $\epsilon \leq C_3/(2C_2)$. Then,

$$\max_i \left|\frac{(\boldsymbol{\beta}_{k,4}^* - \widehat{\boldsymbol{\beta}}_{k,4})^\top \mathbf{x}_i}{\widehat{\boldsymbol{\beta}}_{k,4}^\top \mathbf{x}_i \boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i}\right| \leq \frac{2C_2\epsilon}{C_3^2}.$$

Moreover, according to the true model $\mathcal{Y}_i = w^*(\boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i)\boldsymbol{\beta}_{k,1}^* \circ \boldsymbol{\beta}_{k,2}^* \circ \boldsymbol{\beta}_{k,3}^* + \mathcal{E}_i$,

$$\begin{aligned}&\sup_{\mathbf{u}\in\mathbb{S}_s^{d_1},\mathbf{v}\in\mathbb{S}_s^{d_2},\mathbf{w}\in\mathbb{S}_s^{d_3}} \left|\left(\frac{1}{n}\sum_{i=1}^n \mathcal{Y}_i\right) \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w}\right| \\ \leq &\sup_{\mathbf{u}\in\mathbb{S}_s^{d_1},\mathbf{v}\in\mathbb{S}_s^{d_2},\mathbf{w}\in\mathbb{S}_s^{d_3}} \left|\left(w^*\left(\boldsymbol{\beta}_{k,4}^{*\top}\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\right)\boldsymbol{\beta}_{k,1}^* \circ \boldsymbol{\beta}_{k,2}^* \circ \boldsymbol{\beta}_{k,3}^*\right) \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w}\right| + \eta\left(\bar{\mathcal{E}}, s\right) \\ = &\eta\left(\bar{\mathcal{E}}, s\right),\end{aligned}$$

where the last equality is due to the fact that the covariate vector $\mathbf{x}_i$ is centered such that $\frac{1}{n}\sum_{i=1}^n \mathbf{x}_i = \mathbf{0}$. Therefore, we have,

$$\eta(I_1, s) \leq \underbrace{\frac{2C_2\eta\left(\bar{\mathcal{E}}, s\right)}{C_3^2}}_{\kappa_2}\epsilon.$$

Bound $\eta(I_2, s)$: By the assumption that, for each i, $|\boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i| \geq C_3$, we have that $\max_i(|\boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i|)^{-1} \leq 1/C_3$, and hence

$$\begin{aligned}\eta(I_2, s) &= \sup_{\mathbf{u}\in\mathbb{S}_s^{d_1},\mathbf{v}\in\mathbb{S}_s^{d_2},\mathbf{w}\in\mathbb{S}_s^{d_3}} \left|\left(\frac{1}{n}\sum_{i=1}^n \frac{\mathcal{E}_i}{\boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i}\right) \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w}\right| \\ &\leq \max_i(|\boldsymbol{\beta}_{k,4}^{*\top} \mathbf{x}_i|)^{-1}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right) \\ &\leq \frac{1}{C_3}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right).\end{aligned}$$

Combining the above two results, we obtain the desirable upper bound of $\eta(\widehat{\mathcal{E}}, s)$. □

A3.7 Proof of Lemma 20

Bound (I)**:** For any two tensors $\mathcal{A}, \mathcal{B}$ of the same dimension, the Cauchy-Schwarz inequality implies that $\langle \mathcal{A}, \mathcal{B} \rangle \leq \|\mathcal{A}\|_F\|\mathcal{B}\|_F$. Therefore, we have

$$(I) \leq \frac{\|\widehat{\mathcal{A}}_k - \mathcal{A}_k^*\|_F}{\|\widehat{\mathcal{A}}_k\|_F}.$$

We next simplify the above numerator $\|\widehat{\mathcal{A}}_k - \mathcal{A}_k^*\|_F$. By definition,

$$\begin{aligned}&\|\widehat{\mathcal{A}}_k - \mathcal{A}_k^*\|_F = \left\|\widehat{w}_k\widehat{\boldsymbol{\beta}}_{k,1} \circ \widehat{\boldsymbol{\beta}}_{k,2} \circ \widehat{\boldsymbol{\beta}}_{k,3} - w_k^*\boldsymbol{\beta}_{k,1}^* \circ \boldsymbol{\beta}_{k,2}^* \circ \boldsymbol{\beta}_{k,3}^*\right\|_F \\ \leq &\underbrace{\left\|\widehat{w}_k\widehat{\boldsymbol{\beta}}_{k,1} \circ \widehat{\boldsymbol{\beta}}_{k,2} \circ \widehat{\boldsymbol{\beta}}_{k,3} - w_k^*\widehat{\boldsymbol{\beta}}_{k,1} \circ \widehat{\boldsymbol{\beta}}_{k,2} \circ \widehat{\boldsymbol{\beta}}_{k,3}\right\|_F}_{I_1} \\ + &\underbrace{\left\|w_k^*\widehat{\boldsymbol{\beta}}_{k,1} \circ \widehat{\boldsymbol{\beta}}_{k,2} \circ \widehat{\boldsymbol{\beta}}_{k,3} - w_k^*\boldsymbol{\beta}_{k,1}^* \circ \widehat{\boldsymbol{\beta}}_{k,2} \circ \widehat{\boldsymbol{\beta}}_{k,3}\right\|_F}_{I_2} \\ + &\underbrace{\left\|w_k^*\boldsymbol{\beta}_{k,1}^* \circ \widehat{\boldsymbol{\beta}}_{k,2} \circ \widehat{\boldsymbol{\beta}}_{k,3} - w_k^*\boldsymbol{\beta}_{k,1}^* \circ \boldsymbol{\beta}_{k,2}^* \circ \widehat{\boldsymbol{\beta}}_{k,3}\right\|_F}_{I_3} \\ + &\underbrace{\left\|w_k^*\boldsymbol{\beta}_{k,1}^* \circ \boldsymbol{\beta}_{k,2}^* \circ \widehat{\boldsymbol{\beta}}_{k,3} - w_k^*\boldsymbol{\beta}_{k,1}^* \circ \boldsymbol{\beta}_{k,2}^* \circ \boldsymbol{\beta}_{k,3}^*\right\|_F}_{I_4}.\end{aligned}$$

We next bound each term $I_j/\|\widehat{\mathcal{A}}_k\|_F$ for $j = 1, 2, 3, 4$. First, according to the assumptions $|\widehat{w}_k - w_k^*| \leq \epsilon$ and $\epsilon \leq w_{\min}^*/2$, we have $|\widehat{w}_k| \geq w_k^* - \epsilon \geq w_{\min}^*/2$. Therefore, we have

$$\frac{I_1}{\|\widehat{\mathcal{A}}_k\|_F} = \left|\frac{\widehat{w}_k - w_k^*}{\widehat{w}_k}\right| \leq \frac{\epsilon}{|\widehat{w}_k|} \leq \frac{2}{w_{\min}^*}\epsilon.$$

In addition, note that for any vectors $\mathbf{a} \in \mathbb{R}^m, \mathbf{b} \in \mathbb{R}^n$, since the rank of the matrix $\mathbf{a} \circ \mathbf{b}$ is 1, we have $\|\mathbf{a} \circ \mathbf{b}\|_F = \|\mathbf{a} \circ \mathbf{b}\|_2$. This equality and the fact that $\|\boldsymbol{\beta}_{k,j}^*\|_2 = \|\widehat{\boldsymbol{\beta}}_{k,j}\|_2 = 1$ for

any $j = 1, 2, 3$ imply that

$$\frac{I_2}{\|\widehat{\mathcal{A}}_k\|_F} = \frac{\left\|w_k^*(\widehat{\boldsymbol{\beta}}_{k,1} - \boldsymbol{\beta}_{k,1}^*) \circ \widehat{\boldsymbol{\beta}}_{k,2} \circ \widehat{\boldsymbol{\beta}}_{k,3}\right\|_F}{\left\|\widehat{w}_k \widehat{\boldsymbol{\beta}}_{k,1} \circ \widehat{\boldsymbol{\beta}}_{k,2} \circ \widehat{\boldsymbol{\beta}}_{k,3}\right\|_F} \leq \frac{|w_k^*| \|\widehat{\boldsymbol{\beta}}_{k,1} - \boldsymbol{\beta}_{k,1}^*\|_2}{|\widehat{w}_k|} \leq 2\sqrt{2}\epsilon,$$

where the last inequality is due to the inequality $|\widehat{w}_k| \geq w_{\min}^*/2$ as we show above, as well as the fact that $D(\mathbf{u}, \mathbf{v}) \leq \min\{\|\mathbf{u} - \mathbf{v}\|_2, \|\mathbf{u} + \mathbf{v}\|_2\} \leq \sqrt{2} D(\mathbf{u}, \mathbf{v})$ for unit vectors $\mathbf{u}, \mathbf{v}$. By applying similar proof techniques, we also have

$$\frac{I_3}{\|\widehat{\mathcal{A}}_k\|_F} \leq 2\sqrt{2}\epsilon; \quad \frac{I_4}{\|\widehat{\mathcal{A}}_k\|_F} \leq 2\sqrt{2}\epsilon.$$

Combing the above four inequalities, we obtain the desirable bound for (I) in $(A6)$.

Bound (II)**:** It is easy to see that

$$(II) \leq \frac{\frac{1}{n}\sum_{i=1}^n \langle \mathcal{E}_i, \widehat{\mathcal{A}}_k \rangle \|\Omega \mathbf{x}_i\|_2}{\|\widehat{\mathcal{A}}_k\|_F^2}.$$

According to the assumptions that Ω is positive definite with bounded eigenvalues and $\|\mathbf{x}_i\| \leq C_2$ for each i, we have $\max_i \|\Omega \mathbf{x}_i\|_2$ is bounded by some constant $C_1 > 0$. In addition, according to the Gaussian comparison inequality Lemma 10, as well as the large deviation bound, we have that, with high probability

$$\frac{1}{n}\sum_{i=1}^n \langle \mathcal{E}_i, \widehat{\mathcal{A}}_k \rangle \leq \mathbb{E}\left[\frac{1}{\sqrt{n}} \langle \mathcal{G}, \widehat{\mathcal{A}}_k \rangle\right] \leq \frac{1}{\sqrt{n}} \mathbb{E}\left[\|\mathcal{G}\|_F \|\widehat{\mathcal{A}}_k\|_F\right],$$

for some Gaussian tensor $\mathcal{G} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ whose entries are i.i.d. standard normal random variables. Therefore, we have, with high probability, the desirable inequality in $(A7)$ holds, where the whole term $\widetilde{C}$ is bounded by noting that $\max_i \|\Omega \mathbf{x}_i\|_2$ is bounded and all entries of tensors $\mathcal{G}$, and $\widehat{\mathcal{A}}_k$ of the same dimension are bounded. □

A3.8 Proof of Corollary 7

The derivation of $\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right)$ in the Gaussian error tensor scenario consists of three parts. In Stage 1, we apply Lemma 10 to show that

$$\mathbb{P}\left[\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right) > x\right] \leq \mathbb{P}\left[n^{-1/2}\eta(\mathcal{G}, s) > x\right],$$

for some Gaussian tensor $\mathcal{G}$. In Stage 2, we show via the large deviation bound inequality that

$$\mathbb{P}\left(|\eta(\mathcal{G}, s) - \mathbb{E}[\eta(\mathcal{G}, s)]| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2L^2}\right)$$

with some Lipschitz constant L. In Stage 3, by incorporating Lemma 12, and exploring the sparsity constraint, we can compute

$$\mathbb{E}[\eta(\mathcal{G}, s)]| \leq C\sqrt{s^3 \log(d_1 d_2 d_3)},$$

for some constant $C > 0$.

Stage 1: According to the definition of $\eta(\cdot)$ in (8), we have

$$\begin{aligned}\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right) &= \sup_{\substack{\|\mathbf{u}\|=\|\mathbf{v}\|=\|\mathbf{w}\|=1 \\ \|\mathbf{u}\|_0 \leq s, \|\mathbf{v}\|_0 \leq s, \|\mathbf{w}\|_0 \leq s}} \left|\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w}\right| \\ &= \sup_{\substack{\|\mathbf{u}\|=\|\mathbf{v}\|=\|\mathbf{w}\|=1 \\ \|\mathbf{u}\|_0 \leq s, \|\mathbf{v}\|_0 \leq s, \|\mathbf{w}\|_0 \leq s}} \left\langle \frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} \right\rangle.\end{aligned}$$

Clearly, for any unit-norm vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$, when $\mathcal{E}_i, i = 1, \ldots, n$ are i.i.d. Gaussian tensors, we have

$$\mathbb{E}\left[\left\langle \frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} \right\rangle\right] = 0.$$

Since the variance of each entry of $\mathcal{E}_i$ is 1, we also have

$$\begin{aligned}&\mathrm{var}\left[\left\langle \frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} \right\rangle\right] = \mathrm{var}\left[\frac{1}{n}\sum_{i=1}^n \langle \mathcal{E}_i, \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} \rangle\right] \\ = \;& \frac{1}{n^2}\sum_{i=1}^n \sum_{r,s,t} [\mathbf{u} \circ \mathbf{v} \circ \mathbf{w}]_{r,s,t}^2 \mathrm{var}([\mathcal{E}_i]_{r,s,t}) = \frac{\|\mathbf{u} \circ \mathbf{v} \circ \mathbf{w}\|_F^2}{n}.\end{aligned}$$

For some Gaussian tensor $\mathcal{G}$ of the same dimension as $\mathcal{E}_i$, it is easy to show that

$$\begin{aligned}\mathbb{E}\left[\frac{1}{\sqrt{n}} \langle \mathcal{G}, \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} \rangle\right] &= 0 \\ \mathrm{var}\left[\frac{1}{\sqrt{n}} \langle \mathcal{G}, \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} \rangle\right] &= \frac{\|\mathbf{u} \circ \mathbf{v} \circ \mathbf{w}\|_F^2}{n}.\end{aligned}$$

Moreover, for any $\mathcal{A}_1 := \mathbf{u}_1 \circ \mathbf{v}_1 \circ \mathbf{w}_1$ and $\mathcal{A}_2 := \mathbf{u}_2 \circ \mathbf{v}_2 \circ \mathbf{w}_2$, we have

$$\mathrm{var}\left[\left\langle \frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, \mathcal{A}_1 - \mathcal{A}_2 \right\rangle\right] = \frac{\|\mathcal{A}_1 - \mathcal{A}_2\|_F}{n} = \mathrm{var}\left[\frac{1}{\sqrt{n}} \langle \mathcal{G}, \mathcal{A}_1 - \mathcal{A}_2 \rangle\right].$$

Therefore, according to Lemma 10, we have, for each $x > 0$,

$$\mathbb{P}\left[\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right) > x\right] \leq \mathbb{P}\left[\frac{1}{\sqrt{n}}\eta(\mathcal{G}, s) > x\right]. \tag{A9}$$

Stage 2: We show that the function $\eta(\cdot, s)$ is a Lipschitz function in its first argument. For any two tensors $\mathcal{G}_1, \mathcal{G}_2 \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, denote $\mathcal{A}^* = \sup_{\mathcal{A}} \langle \mathcal{G}_1, \mathcal{A} \rangle$. We have

$$\sup_{\mathcal{A}} \langle \mathcal{G}_1, \mathcal{A} \rangle - \sup_{\mathcal{A}} \langle \mathcal{G}_2, \mathcal{A} \rangle \leq \langle \mathcal{G}_1, \mathcal{A}^* \rangle - \sup_{\mathcal{A}} \langle \mathcal{G}_2, \mathcal{A} \rangle \leq \langle \mathcal{G}_1, \mathcal{A}^* \rangle - \langle \mathcal{G}_2, \mathcal{A}^* \rangle \leq \langle \mathcal{G}_1 - \mathcal{G}_2, \mathcal{A}^* \rangle.$$

Therefore, we have

$$\begin{aligned}
& |\eta(\mathcal{G}_1, s) - \eta(\mathcal{G}_2, s)| \\
= & \sup_{\substack{\|\mathbf{u}\|=\|\mathbf{v}\|=\|\mathbf{w}\|=1 \\ \|\mathbf{u}\|_0\le s, \|\mathbf{v}\|_0\le s, \|\mathbf{w}\|_0\le s}} \left|\mathcal{G}_1 \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w}\right| - \sup_{\substack{\|\mathbf{u}\|=\|\mathbf{v}\|=\|\mathbf{w}\|=1 \\ \|\mathbf{u}\|_0\le s, \|\mathbf{v}\|_0\le s, \|\mathbf{w}\|_0\le s}} \left|\mathcal{G}_2 \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w}\right| \\
\le & \sup_{\substack{\|\mathbf{u}\|=\|\mathbf{v}\|=\|\mathbf{w}\|=1 \\ \|\mathbf{u}\|_0\le s, \|\mathbf{v}\|_0\le s, \|\mathbf{w}\|_0\le s}} \left\langle \mathcal{G}_1 - \mathcal{G}_2, \mathbf{u}\circ\mathbf{v}\circ\mathbf{w} \right\rangle \\
\le & \sup_{\|\mathbf{u}\|=\|\mathbf{v}\|=\|\mathbf{w}\|=1} \|\mathbf{u}\circ\mathbf{v}\circ\mathbf{w}\|_F \cdot \|\mathcal{G}_1 - \mathcal{G}_2\|_F \\
\le & \|\mathcal{G}_1 - \mathcal{G}_2\|_F,
\end{aligned}$$

where the second inequality is due to the fact that $\langle \mathcal{A}, \mathcal{B}\rangle \le \|\mathcal{A}\|_F \|\mathcal{B}\|_F$, and the third inequality is because $\|\mathbf{u}\circ\mathbf{v}\circ\mathbf{w}\|_F = \|\mathbf{u}\circ\mathbf{v}\circ\mathbf{w}\|_2 \le \|\mathbf{u}\|_2\|\mathbf{v}\|_2\|\mathbf{w}\|_2 = 1$ for any unit-norm vectors $\mathbf{u}, \mathbf{v}, \mathbf{w}$.

Applying the concentration result of Lipschitz functions of Gaussian random variables in Lemma 11 with $L = 1$, for the Gaussian tensor $\mathcal{G}$, we have

$$\mathbb{P}\left(|\eta(\mathcal{G}, s) - \mathbb{E}[\eta(\mathcal{G}, s)]| \ge t\right) \le 2\exp\left(-\frac{t^2}{2}\right). \tag{A10}$$

Stage 3: We aim to bound $\mathbb{E}[\eta(\mathcal{G}, s)]|$. For a tensor $\mathcal{T} \in \mathbb{R}^{d_1\times d_2\times d_3}$, denote its L_1-norm regularizer $R(\mathcal{T}) = \sum_{j_1}\sum_{j_2}\sum_{j_3}|\mathcal{T}_{j_1,j_2,j_3}|$. Define the ball of this regularizer as $B_R(\delta) := \{\mathcal{T} \in \mathbb{R}^{d_1\times d_2\times d_3} | R(\mathcal{T}) \le \delta\}$.

For any vectors $\mathbf{u} \in \mathbb{R}^{d_1}, \mathbf{v} \in \mathbb{R}^{d_2}, \mathbf{w} \in \mathbb{R}^{d_3}$ satisfying $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = \|\mathbf{w}\|_2 = 1$, $\|\mathbf{u}\|_0 \le s, \|\mathbf{v}\|_0 \le s$, and $\|\mathbf{w}\|_0 \le s$, denote $\mathcal{A} := \mathbf{u}\circ\mathbf{v}\circ\mathbf{w}$. Lemma 13 implies that $R(\mathcal{A}) \le s^{3/2}$.

Therefore, we have

$$\begin{aligned}
\mathbb{E}[\eta(\mathcal{G}, s)] &= \mathbb{E}\left[\sup_{\substack{\|\mathbf{u}\|=\|\mathbf{v}\|=\|\mathbf{w}\|=1 \\ \|\mathbf{u}\|_0\le s, \|\mathbf{v}\|_0\le s, \|\mathbf{w}\|_0\le s}} \langle \mathcal{G}, \mathbf{u}\circ\mathbf{v}\circ\mathbf{w}\rangle\right] \\
&\le \mathbb{E}\left[\sup_{\mathcal{A}\in B_R(s^{3/2})} \langle \mathcal{G}, \mathcal{A}\rangle\right] = s^{3/2}\mathbb{E}\left[\sup_{\mathcal{A}\in B_R(1)} \langle \mathcal{G}, \mathcal{A}\rangle\right].
\end{aligned}$$

This result, together with Lemma 12, implies that

$$\mathbb{E}[\eta(\mathcal{G}, s)]| \le C\sqrt{s^3\log(d_1d_2d_3)}. \tag{A11}$$

Finally, combing (A10) and (A11), and setting $t = \sqrt{s^3\log(d_1d_2d_3)}$, we have, with probability $1 - 2\exp(-s^3\log(d_1d_2d_3)/2)$,

$$|\eta(\mathcal{G}, s) - \mathbb{E}[\eta(\mathcal{G}, s)]| \le \sqrt{s^3\log(d_1d_2d_3)}.$$

Henceforth

$$\eta(\mathcal{G}, s) \le (C+1)\sqrt{s^3\log(d_1d_2d_3)},$$

which together with (A9), implies that

$$\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right) = O_p\left(\sqrt{\frac{s^3\log(d_1d_2d_3)}{n}}\right).$$

This completes the proof of Corollary 7. □

A3.9 Proof of Corollary 8

The proof follows by incorporating the result in Lemma 7 and the structure assumption in (12). In particular, Lemma 7 implies that, when $\widetilde{\mathcal{E}}_i \in \mathbb{R}^{d\times d}$ is a matrix whose entries are i.i.d. standard Gaussian,

$$\eta\left(\frac{1}{n}\sum_{i=1}^n \widetilde{\mathcal{E}}_i, s\right) = O_p\left(\sqrt{\frac{s^2\log(d^2)}{n}}\right).$$

Moreover, under the structure assumption in (12), we have

$$\eta\left(\frac{1}{n}\sum_{i=1}^n \mathcal{E}_i, s\right) = \eta\left(\frac{1}{n}\sum_{i=1}^n \widetilde{\mathcal{E}}_i, s\right),$$

which completes the proof of Corollary 8. □

References

A. Anandkumar, R. Ge, D. Hsu, S.M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832, 2014a.

A. Anandkumar, R. Ge, and M. Janzamin. Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates. *arXiv preprint arXiv:1402.5180*, 2014b.

S. Balakrishnan, M.J. Wainwright, and B. Yu. Statistical guarantees for the em algorithm: From population to sample-based analysis. *Annals of Statistics*, 45:77–120, 2017.

E.B.E. Becker and C.J. Stoodley. Chapter one - autism spectrum disorder and the cerebellum. In Genevieve Konopka, editor, *Neurobiology of Autism*, volume 113 of *International Review of Neurobiology*, pages 1–34. Academic Press, 2013.

E. Chi and T. Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, 2012.

A. Di Martino, C.G. Yan, Q. Li, E. Denio, F.X. Castellanos, K. Alaerts, J.S. Anderson, M. Assaf, S.Y. Bookheimer, M. Dapretto, B. Deen, S. Delmonte, I. Dinstein, B. Ertl-Wagner, D.A. Fair, L. Gallagher, D.P. Kennedy, C.L. Keown, C. Keysers, J.E. Lainhart, C. Lord, B. Luna, V. Menon, N.J. Minshew, C.S. Monk, S. Mueller, R.A. Muller, M.B. Nebel, J.T. Nigg, K. O'Hearn, K.A. Pelphrey, S.J. Peltier, J.D. Rudie, S. Sunaert, M. Thioux, J.M. Tyszka, L.Q. Uddin, J.S. Verhoeven, N. Wenderoth, J.L. Wiggins, S.H.

Mostofsky, and M.P. Milham. The autism brain imaging data exchange: Towards a large-scale evaluation of the intrinsic brain architecture in autism. *Molecular Psychiatry*, 19 (6):659–667, 2014. ISSN 1359-4184.

S. Friedland and L.H. Lim. Computational complexity of tensor nuclear norm. *arXiv*, 2014.

J. Friedman, H. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical Lasso. *Biostatistics*, 9:432–441, 2008.

S. Ha, I. Sohn, N. Kim, H. Sim, and K. Cheon. Characteristics of brains in autism spectrum disorder: Structure, function and connectivity across the lifespan. *Experimental Neurobiology*, 24(4):273–284, 12 2015.

R. Kana, T. Keller, V.L. Cherkassky, N.J. Minshew, and M.A. Just. Sentence comprehension in autism: thinking in pictures with decreased functional connectivity. *Brain: a journal of neurology*, 129(9):2484–2493, 9 2006.

J. Kim, Y. He, and H. Park. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58:285–319, 2014.

T. Kolda and B. Bader. Tensor decompositions and applications. *SIAM Review*, 51:455–500, 2009.

J.B. Kruskal. More factors than subjects, tests and treatments: an indeterminacy theorem for canonical decomposition and individual differences scaling. *Psychometrika*, 41(3): 281–293, 1976.

J.B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.

L. Li and X. Zhang. Parsimonious tensor response regression. *Journal of the American Statistical Association*, 112:1131–1146, 2017.

Y. Li, H. Zhu, D. Shen, W. Lin, J. H. Gilmore, and J. G. Ibrahim. Multiscale adaptive regression models for neuroimaging data. *Journal of the Royal Statistical Society: Series B*, 73:559–578, 2011.

J. Liu, J. Liu, P. Wonka, and J. Ye. Sparse non-negative tensor factorization using columnwise coordinate descent. *Pattern Recognition*, 45:649–656, 2012.

P. Massart. Concentration inequalties and model selection. *Ecole dEte de Probabilites de Saint-Flour 23*, 2003.

E. Papalexakis, C. Faloutsos, and N.D. Sidiropoulos. Parcube: Sparse parallelizable tensor decompositions. In *ECML PKDD 2012: Machine Learning and Knowledge Discovery in Databases*, pages 521–536, 2012.

J. Peng, J. Zhu, A. Bergamaschi, W. Han, D.Y. Noh, J.R. Pollack, and P. Wang. Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer. *Annals of Applied Statistics*, 4:53–77, 2010.

G. Rabusseau and H. Kadri. Low-rank regression with tensor responses. In *Advances in Neural Information Processing Systems*, 2016.

G. Raskutti and M. Yuan. Convex regularization for high-dimensional tensor regression. *arXiv*, 2016.

J.D. Rudie, J.A. Brown, D. Beck-Pancer, L.M. Hernandez, E.L. Dennis, P.M. Thompson, S.Y. Bookheimer, and M. Dapretto. Altered functional and structural brain network organization in autism. *NeuroImage: Clinical*, 2:79 – 94, 2013. ISSN 2213-1582.

X. Shen, W. Pan, and Y. Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of American Statistical Association*, 107:223–232, 2012.

X. Shen, W. Pan, Y. Zhu, and H. Zhou. On constrained and regularized high-dimensional regression. *The Annals of the Institute of Statistical Mathematics*, 1:1–26, 2013.

R. Shi and J. Kang. Thresholded multiscale gaussian processes with application to bayesian feature selection for massive neuroimaging data. *arXiv preprint arXiv:1504.06074v2*, 2015.

D. Slepian. The one-sided barrier problem for gaussian noise. *The Bell System Technical Journal*, 41:463–501, 1962.

W. Sun, Z. Wang, H. Liu, and G. Cheng. Non-convex statistical optimization for sparse tensor graphical model. *Advances in Neural Information Processing Systems*, 2015.

W. Sun, J. Lu, H. Liu, and G. Cheng. Provable sparse tensor decomposition. *Journal of the Royal Statistical Society, Series B*, 79:899–916, 2017.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B.*, 58:267–288, 1996.

B.G. Travers, R.K. Kana, L.G. Klinger, C.L. Klein, and M.R. Klinger. Motor learning in individuals with autism spectrum disorder: Activation in superior parietal lobule related to learning and repetitive behaviors. *Autism Research*, 8(1):38–51, 2015. ISSN 1939-3806.

N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *NeuroImage*, 15(1):273–289, 2002. ISSN 1053-8119.

X. Wang and H. Zhu. Generalized scalar-on-image regression models via total variation. *Journal of the American Statistical Association*, 112:1156–1168, 2017.

Y. Wang, H. Tung, A. Smola, and A. Anandkumar. Fast and guaranteed tensor decomposition via sketching. *Advances in Neural Information Processing Systems*, 2015.

Z. Wang, Q. Gu, Y. Ning, and H. Liu. High dimensional expectation-maximization algorithm: Statistical optimization and asymptotic normality. *arXiv preprint arXiv:1412.8729*, 2014.

X. Yi and C. Caramanis. Regularized em algorithms: A unified framework and statistical guarantees. *arXiv preprint*, 2015.

R. Yu and Y. Liu. Learning from multiway data: Simple and efficient tensor regression. In *International Conference on Machine Learning*, 2016.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B.*, 68(1):49–67, 2006. ISSN 1369-7412.

M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

M. Yuan and C. Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, 16:1031–1068, 2016.

T. Zhang and H. Zou. Sparse precision matrix estimation via lasso penalized d-trace loss. *Biometrika*, 101:103–120, 2014.

H. Zhou, L. Li, and H. Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108:540–552, 2013.

H. Zhu, Y. Chen, J. G. Ibrahim, Y. Li, C. Hall, and W. Lin. Intrinsic regression models for positive-definite matrices with applications to diffusion tensor imaging. *Journal of the American Statistical Association*, 104(487):1203–1212, 2009.

H. Zou. The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association*, 101(476):1418–1429, December 2006.

Journal of Machine Learning Research 18 (2017) 1-46 Submitted 12/16; Revised 11/17; Published 12/17

A Survey of Preference-Based Reinforcement Learning Methods

Christian Wirth cwirth@ke.tu-darmstadt.de
Knowledge Engineering Group, Technische Universität Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany

Riad Akrour riad@robot-learning.de
Computational Learning for Autonomous Systems, Technische Universität Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany

Gerhard Neumann gneumann@lincoln.ac.uk
Computational Learning, School of Computer Science, University of Lincoln
Brayford Pool, Lincoln, LN6 7TS, Great Britain

Johannes Fürnkranz juffi@ke.tu-darmstadt.de
Knowledge Engineering Group, Technische Universität Darmstadt
Hochschulstraße 10, 64289 Darmstadt, Germany

Editor: George Konidaris

Abstract

Reinforcement learning (RL) techniques optimize the accumulated long-term reward of a suitably chosen reward function. However, designing such a reward function often requires a lot of task-specific prior knowledge. The designer needs to consider different objectives that do not only influence the learned behavior but also the learning progress. To alleviate these issues, preference-based reinforcement learning algorithms (PbRL) have been proposed that can directly learn from an expert's preferences instead of a hand-designed numeric reward. PbRL has gained traction in recent years due to its ability to resolve the reward shaping problem, its ability to learn from non numeric rewards and the possibility to reduce the dependence on expert knowledge. We provide a unified framework for PbRL that describes the task formally and points out the different design principles that affect the evaluation task for the human as well as the computational complexity. The design principles include the type of feedback that is assumed, the representation that is learned to capture the preferences, the optimization problem that has to be solved as well as how the exploration/exploitation problem is tackled. Furthermore, we point out shortcomings of current algorithms, propose open research questions and briefly survey practical tasks that have been solved using PbRL.

Keywords: Reinforcement Learning, Preference Learning, Qualitative Feedback, Markov Decision Process, Policy Search, Temporal Difference Learning, Preference-based Reinforcement Learning

1. Introduction

Due to its solid theoretical foundations, *reinforcement learning* (RL) has become a very attractive tool that has seen many successful applications. It achieved a world first success of machine learning, when TD-Gammon learned to play backgammon (Tesauro, 1995), and its applications which used to focus on games and robotics (Mnih et al., 2015; Kober et al., 2013) now extend to diverse problems in the industrial and medical domain (Wang and Usher, 2005; Zhao et al., 2011). However, the success of reinforcement learning applications often crucially depends on the prior knowledge that is put into the definition of the reward function. Although a reward function is a transferable and compact definition of the learning task, the learned policy may be sensitive to small changes of the reward, possibly yielding very different behaviors, depending on the relative values of the rewards. Thus, the choice of the reward function may have a crucial impact on the success of the learner and many tasks, in particular in robotics, require a high amount of reward-engineering. Four problems are mainly responsible for the difficulties in reward engineering.

(1) Reward Hacking: The agent may maximize the given reward, without performing the intended task (Amodei et al., 2016). For example, consider a robotic vacuum cleaner obtaining a positive reward in case all dirt has been removed. As the robot only observes dirt via camera, it may try to cover up the dirt, effectively removing it from its sensor space

(2) Reward Shaping: In many applications, the reward does not only define the goal but also guides the agent to the correct solution, which is also known as reward shaping (Ng et al., 1999). Striking a balance between the goal definition and the guidance task is also often very difficult for the experimenter.

(3) Infinite rewards: Some applications require infinite rewards, for instance, if certain events should be avoided at any cost. For example, in the cancer treatment problem (Zhao et al., 2009), the death of a patient should be avoided at any cost. However, an infinitely negative reward breaks classic reinforcement learning algorithms and arbitrary, finite values have to be selected.

(4) Multi-objective trade-offs: If it is required to aggregate multiple numeric reward signals, a trade-off is required but the trade-off may not be explicitly known. Hence, different parameterizations have to be evaluated for obtaining a policy that correlates with the expected outcome.

The *reward shaping* problem has especially gained attention in recent years, as it is closely related to intrinsic vs. extrinsic motivation problem (Barto, 2013). An optimal reward would define an extrinsic motivation, e.g., for achieving a defined goal, while also motivating the agent to perform useful actions in states where extrinsic motivation is absent. Hence, the intrinsic motivation can be seen as a guidance method. Several methods have been proposed to learn such a reward automatically, but usually only using an extrinsic goal description (Singh et al., 2009, 2004). However, expert's may have an intuition about useful actions or intrinsic motivation that should be used to guide the agent. Defining such a motivation can be challenging, especially in terms of scalar, numeric rewards and several publications show that an agent's performance can be very sensitive to the used values (Singh et al., 2009; Lu et al., 2016).

Preference-based reinforcement learning (PbRL) is a paradigm for learning from non-numerical feedback in sequential domains. Its key idea is that the requirement for a numerical feedback signal is replaced with the assumption of a preference-based feedback signal

©2017 Christian Wirth, Riad Akrour, Gerhard Neumann and Johannes Fürnkranz.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-634.html.

that indicates relative instead of absolute utility values. Preferences enable a definition of feedback that is not subject to arbitrary reward choices, reward shaping, reward engineering or predefined objective trade-offs. PbRL aims at rendering reinforcement learning applicable to a wider spectrum of tasks and non-expert users. Several algorithms have been developed for solving this class of problems, however, the field still lacks a coherent framework. Furthermore, most publications do not explicitly state the assumptions that are relevant for the application of the algorithms. In this paper, we will survey and categorize preference-based formulations of reinforcement learning and make these assumptions explicit. We also show the relation to other RL-based settings that deviate from the basic setting, such as inverse reinforcement learning or learning with advice.

This paper is based on a preliminary survey (Wirth and Fürnkranz, 2013c), but has been extended in almost every aspect. In particular, the design principles have been reworked to better capture all relevant aspects, we provide substantially more depth about the differences, advantages and disadvantages of the surveyed methods, and we considerably broadened the scope of the survey. The paper starts with the problem setting and the task description in Sec. 2, including a short discussion of related approaches. We then explain the design principles of the algorithms that fit in this framework in Sec. 3. Experimental domains, including some real-world problems that have been solved with PbRL, are described in Sec. 4. Open research questions are pointed out in Sec. 5 before the paper concludes.

2. Preliminaries

Preference-based reinforcement learning algorithms try to solve the *reinforcement learning* problem (Sutton and Barto, 1998) using preferences between states, actions or trajectories.[1] The goal is to learn a policy that is most consistent with the preferences of the expert. Hence, a PbRL algorithm must determine the intention of the expert and find a policy that is maximally consistent with it. For improving its estimate, the algorithms create trajectories which are evaluated by the expert, thereby, exploring the state-action space. Therefore, many algorithms use the *policy evaluation/policy improvement* cycle (Sutton and Barto, 1998) encountered in classical reinforcement learning. Yet, this cycle differs in the evaluation method from classical RL as we can't obtain and propagate numeric feedback, but can only use pairwise preferences.

2.1 Preference Learning

Preference learning is a recent addition to the suite of learning tasks in machine learning (Fürnkranz and Hüllermeier, 2010). Roughly speaking, preference learning is about inducing predictive preference models from empirical data, thereby establishing a link between machine learning and research fields related to preference modeling and decision making. The key difference to conventional supervised machine learning settings is that the training information is typically not given in the form of scalar target values (like in classification and regression), but instead in the form of pairwise comparisons expressing *preferences* between different objects or labels. Thus, in a way, preference learning is comparable to reinforcement learning in that in both settings, the learner may not receive information about the correct target choice, but instead feedback about a single option (in reinforcement learning) or a comparison between two options (in preference learning).

A preference $\boldsymbol{z}_i \succ \boldsymbol{z}_j$ denotes that $\boldsymbol{z}_i$ is preferred over $\boldsymbol{z}_j$ for two choices $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ in a set of possible choices $\mathcal{Z}$. Several short-hand notations can be used (Kreps, 1988):

- $\boldsymbol{z}_i \succ \boldsymbol{z}_j$: The first choice is *strictly preferred*.
- $\boldsymbol{z}_i \prec \boldsymbol{z}_j$: The second choice is *strictly preferred*, i.e., $\boldsymbol{z}_j \succ \boldsymbol{z}_i$.
- $\boldsymbol{z}_i \sim \boldsymbol{z}_j$: The choices are *indifferent*, meaning neither $\boldsymbol{z}_i \succ \boldsymbol{z}_j$ nor $\boldsymbol{z}_j \succ \boldsymbol{z}_i$ holds.
- $\boldsymbol{z}_i \succeq \boldsymbol{z}_j$: The first choice is *weakly preferred*, i.e., $\boldsymbol{z}_j \succ \boldsymbol{z}_i$ does not hold.
- $\boldsymbol{z}_i \preceq \boldsymbol{z}_j$: The second choice is *weakly preferred*, i.e., $\boldsymbol{z}_i \succ \boldsymbol{z}_j$ does not hold.

Depending on what constitutes the set of choices $\mathcal{Z}$, Fürnkranz and Hüllermeier (2010) distinguish between *learning from object preferences*, where the training data is given in the form of pairwise comparisons between data objects, and *learning from label preferences*, where the training data is given in the form of pairwise comparisons between labels that are attached to the objects. In the former case, a common performance task is to rank a new set of objects (*object ranking*; Kamishima et al. 2010), whereas in the latter case, the performance task is to rank the set of labels for a new object (*label ranking*; Vembu and Gärtner 2010). For example, in a reinforcement learning context, states may be considered as objects, and actions may be considered as labels, so that the decision problem of ranking the states according to their desirability is an object ranking problem, whereas ranking the available actions in a given state may be considered as a label ranking problem (Fürnkranz et al., 2012).

There are two main approaches to learning representations of preferences, namely *utility functions*, which evaluate individual alternatives, and *preference relations*, which compare pairs of competing alternatives. From a machine learning point of view, the two approaches give rise to two kinds of learning problems, to which we will return in Sec. 3.2.

2.2 Markov Decision Processes with Preferences

In this section, we will introduce *Markov decision processes with preferences (MDPP)*, a formal, mathematical framework for preference-based reinforcement learning. It is based upon Markov decision processes without reward (MDP\R), which have been proposed by Abbeel and Ng (2004), but extends the setting to preference based feedback. An MDPP is defined by a sextuple $(S, A, \mu, \delta, \gamma, \rho)$ with a *state space* S and *action space* A. The states and actions can be discrete or continuous. $A(s)$ defines the set of actions available in state s and $\mu(s)$ is the distribution of possible initial states. The states can also be represented by feature vectors $\boldsymbol{\phi}(s)$ and state action pairs by the feature vector $\boldsymbol{\varphi}(s, a)$. We will denote a trajectory $\boldsymbol{\tau}$ as a sequence of states and actions, i.e.,

$$\boldsymbol{\tau} = \{s_0, a_0, s_1, a_1, \ldots, s_{n-1}, a_{n-1}, s_n\} .$$

Some algorithms (Wirth et al., 2016; Wilson et al., 2012; Akrour et al., 2011, 2012; Jain et al., 2013, 2015; Gritsenko and Berenson, 2014; Zucker et al., 2010; Jain et al., 2013, 2015;

1. State and action preferences can be mapped to trajectory preferences, as we discuss in Sec. 3.1

Gritsenko and Berenson, 2014) also require the definition of a feature-based description of a trajectory, which we denote as $\boldsymbol{\psi}(\boldsymbol{\tau})$.

As in the standard MDP definition, the *state transition model* $\delta(s' \mid s, a)$ is assumed to be stochastic and the parameter $\gamma \in [0, 1)$ is the discount factor. A *policy* $\pi(a \mid s)$ is a conditional distribution that assigns probabilities to action choices based on the current state. The policy is typically given by a parametric distribution $\pi(a \mid s, \boldsymbol{\theta})$, however, non-parametric approaches are also considered (Akrour et al., 2012; Wirth et al., 2016; Busa-Fekete et al., 2013, 2014).

In contrast to conventional MDPs, we do not assume a numeric reward signal $r(s, a)$. Instead, the agent can observe a preference relation over trajectories $\boldsymbol{\tau}_i \succ \boldsymbol{\tau}_j$. Later, in Sec. 3.1, we will also consider preferences between states and actions, and show that they can be reduced to preferences between trajectories. We further assume that a preference for a given pair of trajectories is generated stochastically with a probability distribution ρ because the expert can err and therefore introduce noise. We use $\rho(\boldsymbol{\tau}_i \succ \boldsymbol{\tau}_j)$ for denoting the probability with which $\boldsymbol{\tau}_i \succ \boldsymbol{\tau}_j$ holds for a given pair of trajectories $(\boldsymbol{\tau}_i, \boldsymbol{\tau}_j)$. The distribution ρ is typically not known, yet the agent can observe a set of preferences

$$\zeta = \{\zeta_i\} = \{\boldsymbol{\tau}_{i1} \succ \boldsymbol{\tau}_{i2}\}_{i=1\dots N}\,, \tag{1}$$

which has been sampled using ρ. $\boldsymbol{\Upsilon}$ denotes the set of all trajectories that are part of ζ. A key problem in PbRL is to obtain a representative set of preferences ζ. Two common simplifications are to disregard the stochasticity of the expert and assume strict preferences. The strict preference assumption implies a total order, i.e., for each pair $\boldsymbol{\tau}_i$ and $\boldsymbol{\tau}_j$, exactly one of the two strict preference relations holds. Phrased otherwise, $\rho(\boldsymbol{\tau}_i \succ \boldsymbol{\tau}_j) = 1 - \rho(\boldsymbol{\tau}_j \succ \boldsymbol{\tau}_i)$. As a consequence, there are no incomparable pairs of trajectories. Incomparability can occur when evaluating trajectories based on multiple criteria, where it is impossible to improve one criteria while decreasing another one. If there are incomparable pairs where no preference relation can be defined, the preferences form a partial order.

2.3 Objective

The general objective of the agent is to find a policy π^* that maximally complies with the given set of preferences ζ. A preference $\boldsymbol{\tau}_1 \succ \boldsymbol{\tau}_2 \in \zeta$ is satisfied if

$$\boldsymbol{\tau}_1 \succ \boldsymbol{\tau}_2 \Leftrightarrow \Pr_\pi(\boldsymbol{\tau}_1) > \Pr_\pi(\boldsymbol{\tau}_2),$$

with

$$\Pr_\pi(\boldsymbol{\tau}) = \mu(s_0) \prod_{t=0}^{|\boldsymbol{\tau}|} \pi(a_t \mid s_t) \delta(s_{t+1} \mid s_t, a_t),$$

as the probability of realizing a trajectory $\boldsymbol{\tau}$ with a policy π. However, satisfying this condition is typically not sufficient as the difference can be infinitesimal. Moreover, due to the stochasticity of the expert, the observed preferences can also contradict each other.

We can now reformulate the objective for a single preference as a maximization problem concerning the difference of the probabilities of trajectories $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$, e.g.,

$$\boldsymbol{\tau}_1 \succ \boldsymbol{\tau}_2 \Leftrightarrow \pi^* = \arg\max_\pi \left(\Pr_\pi(\boldsymbol{\tau}_1) - \Pr_\pi(\boldsymbol{\tau}_2)\right), \tag{2}$$

which is equivalent to creating, if feasible, only dominating trajectories. However, this definition disregards the question of how to deal with multiple preferences.

In general, the goal of all algorithms is to minimize a single preference loss $L(\pi, \zeta_i)$, e.g., $L(\pi, \boldsymbol{\tau}_1 \succ \boldsymbol{\tau}_2) = -\left(\Pr_\pi(\boldsymbol{\tau}_1) - \Pr_\pi(\boldsymbol{\tau}_2)\right)$. The domain of L differs, depending on the approach. Yet, in the multi-preference case, minimizing the loss for one preference might interfere with minimizing the loss for another preference. Hence, without further information, we are not able to define a single loss function but can only specify a multi-objective criterion for the optimal policy where each preference defines a loss function

$$\boldsymbol{L}(\pi, \zeta) = (L(\pi, \zeta_0), L(\pi, \zeta_1), \dots, L(\pi, \zeta_n))$$

Optimizing this multi-objective loss results in several Pareto-optimal solutions. However, we typically want to obtain a single optimal solution. To do so, we can define a scalar objective as the *weighted pairwise disagreement loss* $\mathcal{L}$, as introduced by Duchi et al. (2010),

$$\mathcal{L}(\pi, \zeta) = \sum_{i=1}^{N} \alpha_i L(\pi, \zeta_i), \tag{3}$$

where α_i is the weight or importance assigned to ζ_i. Using such a weighting is a common strategy in preference learning where different evaluation criteria, like the Kendall or the Spearmen distance (Kamishima et al., 2010), are available for different domains. We introduce the weighting α_i to illustrate all preference evaluation criteria in a unified framework.

In PbRL, we want to find a policy that maximizes the realization probability of undominated trajectories while not generating dominated trajectories, if possible. Therefore, preferences concerning the least dominated trajectories should have the highest weight, because the according preferences are most important. Although in other settings, it might be more relevant to realize preferences involving high risk trajectories for preventing major accidents. Hence, different weights α_i can be required. However, all available approaches disregard this problem and use a uniform weight distribution, but focus on requesting preferences over (nearly) undominated trajectories, which has a similar effect as a weighting. In Sec. 3, we explain the different definitions of the single preference loss $L(\pi, \zeta_i)$.

2.4 Preference-Based Reinforcement Learning Algorithms

Learning from preferences is a process that involves two actors: an *agent*, that acts according to a given policy and an *expert* evaluating its behavior. The process is typically composed of several components, as illustrated in Fig. 1. The learning usually starts with a set of trajectories, either predefined or sampled from a given policy. An expert evaluates one or more trajectory pairs $(\boldsymbol{\tau}_{i1}, \boldsymbol{\tau}_{i2})$ and indicates her preference. The method for indicating a preference can differ (cf. Sec. 3.1). For computing a preference-based policy π, three different learning approaches can be found in the literature:

- *learning a policy* computes a policy that tries to maximally comply with the preferences;
- *learning a preference model* learns a model for approximating the expert's preference relation, and

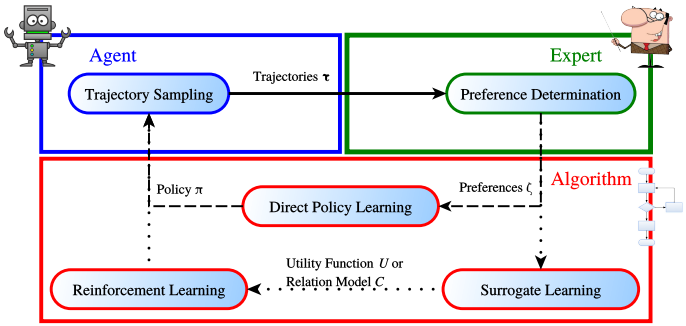


Figure 1: PbRL: Learning policies from preferences via direct (dashed path) and surrogate-based (dotted path) approaches.

- *learning a utility function* estimates a numeric function for the expert's evaluation criterion.

The dashed path in Fig. 1 relates to policy learning. Both other approaches, learning either a preference model or a utility function, are *surrogate-based* in that they learn a surrogate from the obtained preferences, which is in turn used for deriving a maximizing policy (cf. the dotted path in Fig. 1). For this optimization, often conventional reward-based *policy improvement* (PI) algorithms can be used. All three cases are explained in more detail in Sec. 3.2, but the result is always a new policy π_{new}, that can be used to sample new trajectories.

Typically, new trajectories are evaluated repeatedly, and hence, learning from preferences becomes an interactive process between the expert and the algorithm, where the algorithm presents new trajectory pairs to the expert and the expert evaluates them. However, in the interactive setting, a criterion has to be defined for the selection of the preference queries that should be presented to the expert, as described in Sec. 3.4. This criterion must also resolve the exploration/exploitation trade-off as we can only obtain feedback for explicitly requested preferences.

A single pass through the PbRL cycle is often performed in a batch setting, where available trajectories and preference data are used to compute a policy offline.

2.5 Related Problem Settings

Preference-based reinforcement learning is closely related to several other learning settings, which we will briefly discuss in this section.

2.5.1 Learning with Advice

Advice-based algorithms (Torrey and Taylor, 2013; Knox and Stone, 2010, 2012; Griffith et al., 2013) differ from PbRL in that the (possibly preference-based) advice is given in addition to a numeric reward signal. Hence, the main source of feedback is still numeric, not preference-based, and preferences are only used as additional constraints for speeding up the learning process. In addition, not all advice-taking algorithms use preferences. The advice can also be defined via other means, such as rules (Maclin et al., 2005).

Faust et al. (2016) assume that advice is only given in terms of attracting and repulsing states, without an additional reward signal. Such advice is similar to state preferences, but the states are demonstrated by the expert and not evaluated in a pairwise manner.

2.5.2 Ordinal Feedback

PbRL methods can also be used to solve related problem settings. For example, domains with *ordinal feedback signals* can be cast as preference problems by deriving pairwise preferences from the ranking of the ordinal symbols. However, conversely, trying to determine a grading for the ordinal symbols (Weng, 2011) is not feasible if only pairwise preferences are available. Daniel et al. (2014) assume a numeric rating of trajectories. A numeric rating also implicitly defines a ranking which can be again reduced to preferences. While a ranking implicitly contains more information than a binary preference it is not clear whether this additional information can also be transmitted reliably by the expert. Kupcsik et al. (2015) introduced an approach that can use preference as well as numeric feedback. The presented experiments indicate that mixing preference feedback and numeric feedback queries lead to a better performance. However, more experiments with human subjects are needed to shed light on the question of how to combine these feedback types.

2.5.3 Inverse Reinforcement learning

In *inverse reinforcement learning* (IRL; Ng and Russell 2000; Zhifei and Meng Joo 2012), the goal is not to find a policy, but the reward signal that explains the expert's behavior. Stated differently, IRL assumes that the expert maximizes an internal reward function and the goal is to identify this function. However, the expert may use an unknown, true reward which cannot be reconstructed as it may not be subject to the Markov property. Although, it is usually sufficient to find a reward function that produces the same output in each state.

In this setting, the trajectories are supplied by the expert, not by the algorithm. There are also no explicit pairwise preferences, yet they can be derived based on implicit assumptions. In the IRL setting, the demonstrated sequences can be assumed to be preferred over all other trajectories. Hence, the reward-based utility function learning approaches may also be used to solve IRL problems. Many IRL approaches are based on preferences that are defined by implicit assumptions and can be applied to reward-based utility PbRL by adapting them to explicit preferences (see Sec. 3.2.3). Such implicit preferences are also used by Knox and Stone (2009), however, the approach is based on positive (preferred) and negative (dominated) feedback. All positively evaluated sequences are implicitly preferred over all negatively evaluated ones. While most IRL settings assume optimal demonstrations, a few methods (Rothkopf and Dimitrakakis, 2011; Ziebart et al., 2008) relax this assumption and allow sub-optimal demonstrations, based on a parametric optimality prior like a softmax

function. Furthermore, as all trajectories are given by an expert, IRL approaches are not able to obtain new feedback for trajectories and, hence, can not improve upon the expert demonstrations.

2.5.4 Unsupervised Learning

A different line of work (Meunier et al., 2012; Mayeur et al., 2014) uses preferences for learning without a supervised feedback signal. It is assumed that the performance of the policy degrades over time, which is reasonable in some domains. They apply their approach to a robot following task, where a robot needs to follow a second robot based on visual information. Without an optimal policy, the path of the follower will deviate from the leader over time. Therefore, states encountered early in the trajectory are assumed to have higher quality than later states, which again defines an implicit preference. These preferences are used to learn the value function, which induces the policy.

2.5.5 Dueling Bandits

Another related setting is the *dueling bandit* (Yue et al., 2012), which uses preferences in an interactive, but non sequential problem setting. Feedback concerns comparisons of arms of a k-armed bandit. However, PbRL can be phrased as a dueling bandit problem, when using policies as arms. Busa-Fekete et al. (2013, 2014) realized that idea, which we discuss in Sec. 3.2.1. Approaches that learn a preference model from action preferences also share similarities (cf. Sec. 3.2.2). Each state defines a bandit with the actions as arms, but it is required to optimize over sequences of bandits. Furthermore, the mentioned approaches extend the dueling bandit setting by either generalizing over multiple states (bandits) or by deriving (multiple) action preferences from trajectory preferences. Related to dueling bandits is the use of two-point feedback for online convex optimization (Agarwal et al., 2010; Shamir, 2017), which has recently been extended to structured prediction problems (Sokolov et al., 2016).

3. Design Principles for PbRL

Preference-based reinforcement learning algorithms are subject to different design choices, which we will review in this section. First, we consider the different types of preferences, i.e., *trajectory*, *action* or *state* preferences. These types introduce different levels of complexity for the expert and algorithm (Sec. 3.1). The second design choice is how the learning problem is phrased, i.e, *directly learning a policy*, *learning a preference relation model* or a *utility function* as surrogate (Sec. 3.2). The learned representation is connected to the problem of assigning preferences to states and actions, given a delayed feedback signal. We explain this problem and possible approaches in Sec. 3.3. The next design principle is how to collect new feedback, i.e., how new trajectories are generated and selected to obtain new preferences, which we discuss in Sec. 3.4. Having obtained a representation, we need to derive an optimized policy. In Sec. 3.5, we discuss different optimization strategies that can be used to this end. Furthermore, PbRL algorithms may employ different techniques for capturing the transition dynamics, which come with different assumptions, as explained in Sec. 3.6. Sec. 3.7 shows a tabular overview of all algorithms.

3.1 Types of Feedback

There are three different types of preference feedback that can be found in the literature, action, state and trajectory preferences. The most important difference of the preference types is that they impose different challenges for the expert and the algorithm.

3.1.1 Action Preferences

An *action preference* compares two actions for the same state. The action preference $a_{i1} \succ_s a_{i2}$ expresses that in state s, action a_{i1} should be preferred to a_{i2}.

We need to distinguish between short-term optimality (e.g. optimal, immediate reward in terms of reinforcement learning) and long-term optimality (optimal, expected return). Knox and Stone (2012) and Thomaz and Breazeal (2006) analyzed this problem in an advice setting and observed that feedback relating to immediate rewards are difficult to use and feedback concerning the expected, long-term return should be preferred. This observation can be explained by the argument that it is difficult to aggregate short-term action preferences, as they are only valid for a given state. It is unclear how the short-term preferences relate to long-term optimality as we are not able to trade-off preferences for different states. Action preferences are demanding for the expert, she needs to be familiar with the expected long-term outcome. Computationally, this problem is fairly simple as it is sufficient to select the most preferred action in every state, which already implies the best long-term outcome. Hence, only generalization issues remain. The action-preference-based approach of Fürnkranz et al. (2012) also uses long-term action preferences, but not concerning an unknown, optimal policy. They supply a policy for computing the long term expectation. Hence, the expert does not need to be familiar with the expected, optimal policy but can consider samples from the provided policy for its comparison.

3.1.2 State Preferences

A *state preference* $s_{i1} \succ s_{i2}$ determines that state s_{i1} is preferred to state s_{i2}. A state preference is equivalent to saying that there is an action in state s_{i1} that is preferred to all actions available in state s_{i2}.

State preferences are more informative than action preferences as they define relations between parts of the global state space. Yet, they also suffer from the long-term/short-term optimality problem. Long-term state preferences define a clearly defined setting as it is sufficient to discover the most preferred successor state for each state and maximize its selection probability, as analyzed by Wirth and Fürnkranz (2012, 2015). Short-term state preferences do not define a trade-off, because it is unclear whether visiting an undominated state once should be preferred over visiting a rarely dominated state multiple times. Short-term state preferences are used by Zucker et al. (2010) where they try to resolve the trade-off problem using an approximated reward.

State preferences are slightly less demanding for the expert as it is not required to compare actions for a given state. However, the expert still needs to estimate the future outcome of the policy for a given state. State preferences do not directly imply a policy, but it can be easily derived with knowledge of the transition model, as explained in Sec. 3.2.3.

3.1.3 Trajectory Preferences

A *trajectory preference* $\boldsymbol{\tau}_{i1} \succ \boldsymbol{\tau}_{i2}$ specifies that the trajectory $\boldsymbol{\tau}_{i1}$ should be preferred over the dominated trajectory $\boldsymbol{\tau}_{i2}$. Trajectory preferences are the most general form of feedback and the most widely used.

Trajectory preferences are arguably the least demanding preferences type for the expert as she can directly evaluate the outcomes of full trajectories. Trajectories can be evaluated in terms of the resulting behavior (e.g., a jerky vs. a straight movement towards a goal state), or by considering the occurrences of states that are known to be good. Usually, the goal is to keep complexity away from the expert, rendering trajectory preferences the most common approach. Yet, a difficulty with trajectory preferences is that the algorithm needs to determine which states or actions are responsible for the encountered preferences, which is also known as the *temporal credit assignment problem*. This assignment problem is particularly difficult if not all trajectories are starting in the same state. Trajectories with different starting states are solutions to different initial situations. Hence, the preference could be attributed to the initial situation or the applied policy. In practice, no algorithm known to the authors is capable of dealing with preferences between trajectories with different initial states.

Trajectory preferences are the most general form of preference-based feedback because, formally, all preferences can be mapped to trajectory preferences when we admit trajectories with only a single element. An action preference $a_{i1} \succ_s a_{i2}$ is equivalent to the trajectory preference $\boldsymbol{\tau}_{i1} \succ \boldsymbol{\tau}_{i2}, \boldsymbol{\tau}_{i1} = \{s, a_{i1}\}, \boldsymbol{\tau}_{i2} = \{s, a_{i2}\}$. The case of a state preference $s_{i1} \succ s_{i2}$ can be reformulated as indicated above, yielding the set of trajectory preferences $\{\forall a_{i2} \in A(s_{i2}) : (s_{i1}, \pi^*(s_{i1})) \succ (s_{i2}, a_{i2})\}$.

3.2 Defining the Learning Problem

As discussed in Sec. 2.4, preferences can be used to directly learn a policy, or to learn a qualitative preference model or a quantitative utility function, both of which can then be used for deriving a policy. In the following, we will discuss different options for representing a policy, a preference model or a utility function, which have been used in various algorithms in the literature. Unless required otherwise, we only consider the loss for the preferred relation $\succ$, as explained in Sec. 2.2. We use the symbol $L^{\succ}$ for a loss that is only valid for the $\succ$ relation to differentiate from the general, single preference loss L, introduced in Sec. 2.3

3.2.1 Learning a Policy

Direct policy learning assumes a parametric policy space. The learning task is to find a parametrization that maximizes the correspondence of the policies with the observed preferences. Two different approaches have been tried, namely to induce a distribution over a parametric policy space (Wilson et al., 2012), or to compare and rank policies (Busa-Fekete et al., 2013, 2014).

Approximating the Policy Distribution. Wilson et al. (2012) approximate the policy distribution via a *Bayesian likelihood function* subject to the preference-based data probability. A parameterized policy space $\pi(a \mid s, \boldsymbol{\theta})$ is used for defining the policy distribution $\Pr(\pi \mid \zeta)$. Algorithm 1 shows how to collect the preferences. The basic strategy is to sample two policies, described by a parameter vector, from the posterior distribution $\Pr(\pi \mid \zeta)$ induced by the preferences. Each policy pair is used to create one or more trajectories, defining a potential preference query by pairing the resulting trajectories. One of multiple selection criteria, that we discuss in Sec. 3.4.3, is used to select the query to pose to the expert. Variants of the criteria also depend on the policy that created the trajectory or allow to stop the sampling process before m policy pairs are created.

Algorithm 1 Policy Likelihood

Require: prior $\Pr(\pi)$, step limit k, sample limit m, iteration limit n

$\zeta = \emptyset$ ▷ Start with empty preference set
for $i = 0$ **to** n **do**
  $\tilde{\zeta} = \emptyset$ ▷ Clear potential preference queries
  for $j = 0$ **to** m **do**
    $s \sim \mu(s)$ ▷ Draw initial state s
    $\pi_1, \pi_2 \sim \Pr(\pi \mid \zeta)$ ▷ Draw two policies from posterior
    for $j = 0$ **to** n **do**
      $\boldsymbol{\tau}^{\pi_1} = \text{ROLLOUT}(s, \pi_1, k)$ ▷ Create k-step trajectory using policy π_1
      $\boldsymbol{\tau}^{\pi_2} = \text{ROLLOUT}(s, \pi_2, k)$ ▷ Create k-step trajectory using policy π_2
      $\tilde{\zeta} \leftarrow \tilde{\zeta} \cup (\boldsymbol{\tau}^{\pi_1}, \boldsymbol{\tau}^{\pi_2})$ ▷ Add trajectory pair to potential preference queries
    end for
  end for
  $(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2) = \text{SELECTQUERY}(\tilde{\zeta})$ ▷ Select a preference query
  $\zeta \leftarrow \zeta \cup \text{OBTAINTRAJECTORYPREFERENCES}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$ ▷ Query expert
end for
return $\arg\max_\pi \Pr(\pi \mid \zeta)$ ▷ Return maximum-a-posterior

The likelihood $\Pr(\pi \mid \zeta)$ is modeled by comparing trajectories $\boldsymbol{\tau}^\pi$ that are realized by a policy π with the preferred and dominated trajectories in ζ. The likelihood is high if the realized trajectories $\boldsymbol{\tau}^\pi$ of the policy π are closer to preferred trajectory, i.e.,

$$\Pr(\boldsymbol{\tau}_1 \succ \boldsymbol{\tau}_2 \mid \pi) = \Phi\left(\frac{\mathbb{E}[d(\boldsymbol{\tau}_1, \boldsymbol{\tau}^\pi)] - \mathbb{E}[d(\boldsymbol{\tau}_2, \boldsymbol{\tau}^\pi)]}{\sqrt{2}\sigma_p}\right), \tag{4}$$

where the function Φ is the c.d.f of $\mathcal{N}(0, 1)$, which resembles a sigmoidal squashing function. The parameter σ_p accounts for feedback noise to allow the expert to err. The function d is a distance function comparing two trajectories. The policy distribution is then given by applying Bayes theorem, i.e,

$$\Pr(\pi \mid \zeta) \propto \Pr(\pi) \prod_i \Pr(\boldsymbol{\tau}_1 \succ \boldsymbol{\tau}_2 \mid \pi), \tag{5}$$

and the posterior is approximated using *Markov Chain Monte Carlo* simulation (Andrieu et al., 2003). This approach requires the specification of a meaningful distance function, which is hard to define in many domains and requires a large amount of domain knowledge. Wilson et al. (2012) use an Euclidean distance function. Yet, Euclidean distances are hard to use in many domains such as those with high-dimensional continuous state spaces. Furthermore, if the state space is not continuous it may be hard to define a suitable distance

Algorithm 2 Policy Ranking

Require: candidate policies Π_0, step limit k, sample limit m, iteration limit n

for $i = 0$ **to** n **do**
  $\zeta = \emptyset$
  for 0 **to** m **do**
    $\pi_1, \pi_2 = \text{SELECTPOLICIES}(\Pi_i, \zeta)$ ▷ Select policies to compare
    $s \sim \mu(s)$ ▷ Draw initial state s
    $\boldsymbol{\tau}^{\pi_1} = \text{ROLLOUT}(s, \pi_1, k)$ ▷ Create k-step trajectory using policy π_a
    $\boldsymbol{\tau}^{\pi_2} = \text{ROLLOUT}(s, \pi_2, k)$ ▷ Create k-step trajectory using policy π_2
    $\zeta \leftarrow \zeta \cup \text{OBTAINTRAJECTORYPREFERENCES}(\boldsymbol{\tau}^{\pi_1}, \boldsymbol{\tau}^{\pi_2})$ ▷ Query expert
  end for
  $\Pi_{i+1} = \text{EDPS}(\Pi_i, \zeta)$ ▷ Compute candidate policies, based on preferences
end for
return $\arg\max_\pi \sum_{\pi,\pi' \in \Pi_n} \Pr(\pi \succ \pi')$ ▷ Return highest ranked policy

measure d. The definition of a parametric policy also requires domain knowledge, although good parametric policy representations are known for many domains, such as robotics (Kober et al., 2013). Often, such policy representations reduce the dimensionality of the policy and therefore the learning complexity.

Comparing and Ranking Policies. Alternatively, we can directly compare two policies π_1 and π_2 by querying the expert for different trajectory pairs $\boldsymbol{\tau}^{\pi_1}$ and $\boldsymbol{\tau}^{\pi_2}$ that have been realized by the two policies (Busa-Fekete et al., 2013, 2014). The method maintains a set of policies Π_i, described by their parameter vectors. The preference set ζ is used to compute a ranking over policies, as shown in Algorithm 2. A confidence bound method that we discuss in Sec. 3.4.1 determines which policies to select for generating the next trajectory pair for which a preference query is posed.

The outcome of this comparison is then used to compute new policies that are more likely to realize preferred trajectories. A policy π_1 is preferred over a policy π_2 if the generated trajectories $\boldsymbol{\tau}^{\pi_1}$ are on expectation preferred over the trajectories $\boldsymbol{\tau}^{\pi_2}$ realized by the second policy, i.e.,

$$\Pr(\pi_1 \succ \pi_2) \;\Leftrightarrow\; \mathbb{E}_{\boldsymbol{\tau}^{\pi_1}, \boldsymbol{\tau}^{\pi_2}}[\rho(\boldsymbol{\tau}^{\pi_1} \succ \boldsymbol{\tau}^{\pi_2})]. \tag{6}$$

For an observed set of preferences ζ, the expectation in (6) can be approximated as

$$\Pr(\pi_1 \succ \pi_2) \approx \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\boldsymbol{\tau}_i^{\pi_1} \succ \boldsymbol{\tau}_i^{\pi_2}). \tag{7}$$

The objective is to find the set of optimal polices $\pi^* = \arg\max_\pi \sum_{\pi,\pi' \in \Pi} \Pr(\pi \succ \pi')$. In contrast to Wilson et al. (2012), no distance function is required. However, preferences can not be reused as we need to obtain new preferences for each policy pair and, hence, a high amount of preferences is required. Busa-Fekete et al. (2013, 2014) deal with incomparabilities and indifference by assuming

$$\boldsymbol{\tau}_{i1} \sim \boldsymbol{\tau}_{i2} \;\Leftrightarrow\; \rho(\boldsymbol{\tau}_{i1} \prec \boldsymbol{\tau}_{i2}) = \rho(\boldsymbol{\tau}_{i1} \succ \boldsymbol{\tau}_{i2}) = 0.5.$$

Algorithm 3 Preference-based Approximate Policy Iteration

Require: initial policy π_0, iteration limit m, state sample limit k, rollout limit n

for $i = 0$ **to** m **do**
  for 0 **to** k **do**
    $s \sim \mu(s)$ ▷ Sample k states per iteration
    $\boldsymbol{\Upsilon} = \emptyset, \zeta = \emptyset$
    for $\forall a \in A(s)$ **do**
      for 0 **to** n **do**
        $\boldsymbol{\Upsilon} \leftarrow \boldsymbol{\Upsilon} \cup \text{ROLLOUT}(s, a, \pi_i)$ ▷ Create trajectory, starting with (s, a)
      end for
    end for
    $\zeta \leftarrow \text{OBTAINACTIONPREFERENCES}(\boldsymbol{\Upsilon})$ ▷ Query expert
  end for
  $C^{\pi_i} = \text{COMPUTEPREFERNCEMODEL}(\zeta)$ ▷ Compute the preference function C
  $\pi_{i+1} = \text{COMPUTEPOLICY}(C^{\pi_i})$ ▷ Compute greedy policy
end for
return improved policy π_m

The optimization itself can be performed with algorithms similar to *evolutionary direct policy search* (EDPS; Heidrich-Meisner and Igel 2009). Policy comparison approaches following the ideas of Busa-Fekete et al. (2013, 2014) can be seen as cases of preference-based multi-armed bandit optimization, where each arm represents one policy. For a survey of such preference-based bandit methods, we refer to Busa-Fekete and Hüllermeier (2014).

3.2.2 Learning a Preference Model

Instead of directly learning a policy, one can also try to learn a model $C(a \succ a' \mid s)$ that predicts the expected preference relation between a and a' for a given state s. The preference relation between actions can be used to obtain a ranking for actions given a state, from which a policy can be derived in turn. The problem of learning $C(a \succ a' \mid s)$ can be phrased as a classification problem trying to correctly predict all observed preferences. As shown in Algorithm 3, *preference-based approximate policy iteration* obtains multiple trajectories for each action in k sampled states (Fürnkranz et al., 2012). This allows to derive multiple action preferences by comparing trajectories, based on the initial actions. Action preferences are directly used as training data $\{s_i, a_{i1}\} \succ \{s_i, a_{i2}\} \Leftrightarrow (a_{i1} \succ a_{i2} \mid s_i)$ for C. From these, a separate classifier $C_{ij}(s)$ is trained, which predicts whether $a_i \succ a_j$ will hold in a state s.

For deriving a ranking over all actions in a given state, the pairwise preference predictions of the trained classifiers C_{ij} may be combined via voting or weighted voting (Hüllermeier et al., 2008), where each prediction issues a vote for its preferred action. The resulting count $k(s, a)$ for each action a in state s

$$k(s, a) = \sum_{\forall a_i \in A(s), a_j \neq a} C(a_i \succ a_j \mid s) = \sum_{\forall a_i \in A(s), a_j \neq a} C_{ij}(s) \tag{8}$$

correlates with $\rho(\,(s,a_i) \succ (s,a_j)\,)$ if the classifiers C approximate ρ well. Hence, we can derive a policy

$$\pi^*(a \mid s) = \begin{cases} 1 & \text{if } a = \arg\max_{a'} k(s,a') \\ 0 & \text{else} \end{cases},$$

that maximizes the realization probability for the most preferred action, satisfying (2).

Wirth and Fürnkranz (2013a,b) also use action preferences in a similar way. However, they treat C as a continuous function, returning a weighted vote for each action instead of a binary vote. Hence, $k(s,a)$ now also contains the uncertainty of the estimated preference, which allows for more efficient exploration. In this case, the function C is defined by a tabular representation with a single value for each state/action/action triplet, and updated using a gradient-based method. Fürnkranz et al. (2012) use a neural network (Bishop, 1995) that is retrained completely in every iteration, allowing for generalization at the cost of potential approximation errors.

3.2.3 Learning a Utility Function

Utility-based approaches estimate a surrogate utility $U(\boldsymbol{\tau})$ for a given trajectory. In many cases, this trajectory utility can be decomposed into state-action utilities $U(s,a)$, i.e, $U(\boldsymbol{\tau}) = \sum_t U(s_t,a_t)$. Note that this surrogate function is not directly comparable to an approximated reward or return function because it may be subject to concept drift if the estimate of the expert's optimality criterion can change over time. As in the IRL case, the expert may derive the preferences from an unknown, true reward which cannot be reconstructed as it may not be subject to the Markov property. However, in the PbRL case it is sufficient to find a reward function that induces the same, optimal policy as the true reward function. Consequently, we call this function a *utility function* to distinguish it from a fixed reward function[2].

Given the surrogate utility, the policy should then maximize the expected trajectory utility

$$\pi^* = \max_{\pi} \mathbb{E}_{\Pr_\pi(\boldsymbol{\tau})}[U(\boldsymbol{\tau})], \tag{9}$$

which is comparable to the expected return in classic *reinforcement learning*. Besides exploring the policy space, as in classic reinforcement learning, we also need to explore the utility space in preference-based RL algorithms. Estimating a utility function has the advantage that we can evaluate new trajectories without asking the expert for explicit feedback. Yet, utility function approaches may suffer from the approximation errors induced by the estimated utility function. A common utility-based approach is to assume a scalar utility for the trajectories, i.e.,

$$\boldsymbol{\tau}_{i1} \succ \boldsymbol{\tau}_{i2} \Leftrightarrow U(\boldsymbol{\tau}_{i1}) > U(\boldsymbol{\tau}_{i2}).$$

A scalar utility function always exists provided that there are no incomparable trajectory pairs (von Neumann and Morgenstern, 1944).

Algorithm 4 summarizes a general process of utility-based PbRL, which forms the basis of a variety of different realizations. Preference queries are generated by sampling one or more trajectories from a policy and derive queries by exhaustive or selective comparison (cf. Sec. 3.4.1 & 3.4.2). The preferences are then used to compute a utility function (Friedman and Savage, 1952) which, in turn, can be used to optimize policies according to the expected utility. Different learning algorithms can be used for modeling the utility function. In the following, we primarily discriminate between linear and non-linear utility models.

2. The definition of a utility function may differ from the functions encountered in utility theory, but is inline with existing PbRL approaches and the preference learning literature.

Algorithm 4 Utility-based PbRL

Require: initial policy π_0, iteration limit m, state sample limit k, rollout limit n

$\zeta = \emptyset$
for $i = 0$ **to** m **do**
  $\boldsymbol{\Upsilon} = \emptyset$
  for 0 **to** k **do**
    $s \sim \mu(s)$ ▷ Sample k states per iteration
    for 0 **to** n **do**
      $\boldsymbol{\Upsilon} \leftarrow \boldsymbol{\Upsilon} \cup \text{ROLLOUT}(s, \pi_i)$ ▷ Create trajectory, starting with state s
    end for
  end for
  $(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2) = \text{createQuery}(\boldsymbol{\Upsilon})$ ▷ Create a preference query
  $\zeta \leftarrow \zeta \cup \text{obtainTrajectoryPreferences}(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$ ▷ Query expert
  $U^{\pi_i} = \text{computeUtilityFunction}(\zeta)$ ▷ Compute the utility function
  $\pi_{i+1} = \text{computePolicy}(U^{\pi_i})$ ▷ Compute a policy
end for
return improved policy π_m

Linear utility functions. The most common approach is to use utility functions that are linear in a feature vector. We may use state action features $\boldsymbol{\varphi}(s,a)$ resulting in a utility $U(s,a) = \boldsymbol{\theta}^T \varphi(s,a)$, or trajectory features $\boldsymbol{\psi}(\boldsymbol{\tau})$ yielding $U(\boldsymbol{\tau}) = \boldsymbol{\theta}^T \boldsymbol{\psi}(\boldsymbol{\tau})$. In order to find such a linear utility function, we can define a loss function $\mathcal{L}$ which is given by the (weighted) sum of the pairwise disagreement loss L, i.e.,

$$\mathcal{L}(\boldsymbol{\theta}, \zeta) = \sum_{i=1}^{|\zeta|} \alpha_i L(\boldsymbol{\theta}, \zeta_i), \tag{10}$$

as described in Sec. 2.3. Using the linearity of the utility function, we can define the utility difference

$$d(\boldsymbol{\theta}, \zeta_i) = \boldsymbol{\theta}^T (\boldsymbol{\psi}(\boldsymbol{\tau}_{i1}) - \boldsymbol{\psi}(\boldsymbol{\tau}_{i2})) \tag{11}$$

for two trajectories. Different definitions of the pairwise disagreement loss $L(\boldsymbol{\theta}, \zeta_i)$ have been used in the literature and most of them use the utility difference. An intuitive loss directly correlating with the obtained binary feedback is the indicator loss

$$L^{\succ}(\boldsymbol{\theta}, \zeta_i) = \mathbb{I}\left(d(\boldsymbol{\theta}, \zeta_i) \leq \epsilon\right), \tag{12}$$

which states that the utility difference has to be always larger then ϵ, as used by Sugiyama et al. (2012). However, such a binary loss function is typically difficult to optimize and there is no notion of preference violation (or fulfillment) for trading off multiple preferences.

Name	Loss Equation	Reference
ranking SVM	$L^{\succ}(\boldsymbol{\theta}, \zeta_i) = \max(0, 1 - d(\boldsymbol{\theta}, \zeta_i))$	Akrour et al. (2011, 2012) Zucker et al. (2010) Wirth and Fürnkranz (2012, 2015) Runarsson and Lucas (2012, 2014)
IRL	$L^{\succ}(\boldsymbol{\theta}, \zeta_i) = c(d(\boldsymbol{\theta}, \zeta_i))$ $c(x) = \begin{cases} -2x & \textbf{if } x \leq 0 \\ -x & \textbf{otherwise} \end{cases}$	Wirth et al. (2016)
cumulative	$p_{\boldsymbol{\theta}}(\zeta_i) = \Phi\left(\frac{d(\boldsymbol{\theta}, \zeta_i)}{\sqrt{(2)}\sigma_p}\right)$	Kupcsik et al. (2015)
sigmoid	$p_{\text{sig }\boldsymbol{\theta}}(\zeta_i) = \frac{1}{1+\exp(-m \cdot d(\boldsymbol{\theta}, \zeta_i))}$	Christiano et al. (2017)
0/1	$p_{01\ \boldsymbol{\theta}}(\zeta_i) = \mathbb{I}\big(d(\boldsymbol{\theta}, \zeta_i) < 0\big)$	Sugiyama et al. (2012)
combined	$p_{\boldsymbol{\theta}}(\zeta_i) = \frac{\lvert\zeta\rvert - 1}{\lvert\zeta\rvert} p_{01\ \boldsymbol{\theta}}(\zeta_i) + \frac{1}{\lvert\zeta\rvert} p_{\text{sig }\boldsymbol{\theta}}(\zeta_i)$	Wirth et al. (2016)
integrated piecewise	$p_{\boldsymbol{\theta}}(\zeta_i) = \int_0^{\epsilon_{\max}} c(d(\boldsymbol{\theta}, \zeta_i), \epsilon)$ $c(x, \epsilon) = \begin{cases} 0 & \textbf{if } x < -\epsilon \\ 1 & \textbf{if } x > \epsilon \\ \frac{\epsilon + x}{2\epsilon} & \textbf{else}, \end{cases}$	Akrour et al. (2013, 2014)

Table 1: Loss functions for linear utility functions

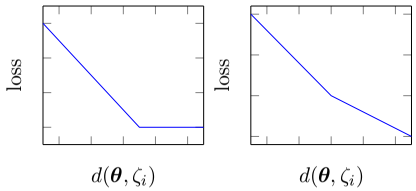
Therefore, continuous loss functions are more often used. A number of continuous loss functions have been proposed in the literature.

Tbl. 1 shows a summary of loss functions and likelihood functions that can be found in the literature. All these approaches try to find a good trade-off between potentially violating preferences and optimizing the utility difference of preferred trajectories. Some algorithms (Akrour et al., 2011, 2012; Zucker et al., 2010; Wirth and Fürnkranz, 2012, 2015; Runarsson and Lucas, 2012, 2014) define the loss function $L^{\succ}(\boldsymbol{\theta}, \zeta_i)$ as hinge loss (Fig. 2a) that can be optimized using SVM ranking algorithms (Joachims, 2002; Herbrich et al., 1999), others (Wirth et al., 2016) as a piecewise linear loss function (Fig. 2b) inspired by inverse reinforcement learning algorithms (Ng and Russell, 2000).

Other algorithms use sigmoidal loss functions to saturate the effect of a high utility differences, in order to overcome the limitations of a linear loss. Such sigmoidal loss functions are often modeled as likelihood functions $p_{\boldsymbol{\theta}}(\zeta_i)$ for the preferences. In this case, we have to optimize the log likelihood, i.e.,

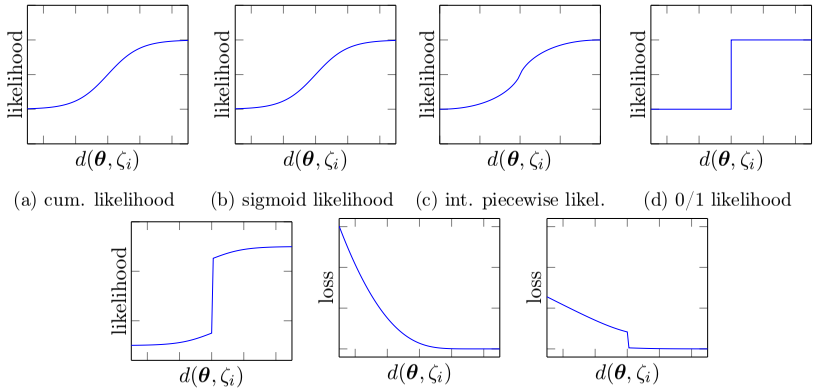
$$\mathcal{L}(\boldsymbol{\theta}, \zeta) = -\log \prod_{i=1}^{|\zeta|} p_{\boldsymbol{\theta}}(\zeta_i) = -\sum_{i=1}^{|\zeta|} \log\left(p_{\boldsymbol{\theta}}(\zeta_i)\right) = \sum_{i=1}^{|\zeta|} \mathcal{L}(\boldsymbol{\theta}, \zeta_i),$$

with $\mathcal{L}(\boldsymbol{\theta}, \zeta) = -\log\left(p_{\boldsymbol{\theta}}(\zeta_i)\right)$. Most algorithms (Akrour et al., 2014; Wirth et al., 2016; Kupcsik et al., 2015) do not directly maximize the likelihood, but obtain the posterior



(a) hinge (b) IRL

Figure 2: Shape of loss functions



(a) cum. likelihood (b) sigmoid likelihood (c) int. piecewise likel. (d) 0/1 likelihood

(e) comb. likelihood (f) cum. log loss (g) combined log loss

Figure 3: Shape of likelihood functions and the according negative log likelihood

distribution

$$\Pr(\boldsymbol{\theta} \mid \zeta) \propto \Pr(\boldsymbol{\theta}) \prod_i p_{\boldsymbol{\theta}}(\zeta_i). \tag{13}$$

The expected utility of a new trajectory is then obtained by computing the expectation over the posterior. The posterior is computed using MCMC (Andrieu et al., 2003) by Akrour et al. (2014) and Wirth et al. (2016), whereas the latter uses *elliptic slice sampling* (ELS; Murray et al. 2010). These procedures are sampling based and can only obtain a costly approximation of the posterior. The utility function's posterior is also computed by Kupcsik et al. (2015), but using more efficient *convex optimization*. Wirth et al. (2016) also compare with a direct maximization of the likelihood. The direct maximum likelihood approach is more optimistic in the sense that it disregards the uncertainty.

The only approach that does not use an explicit cost function for obtaining the utility is by Jain et al. (2013, 2015). It uses the preferences to compute a gradient for the parameter vector of the utility function, i.e., $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha\big(\boldsymbol{\psi}(\boldsymbol{\tau}_{k1}) - \boldsymbol{\psi}(\boldsymbol{\tau}_{k2})\big)$, where α is a step-

size. In this case, the parameter vector is updated such that it is correlated with the feature vector of the preferred trajectory. Such a gradient can be mapped to the simple loss function $L(\boldsymbol{\theta}, \zeta_i) = -d(\boldsymbol{\theta}, \zeta_i)$. In general, it is unclear how the aggregated utility loss $\mathcal{L}(\boldsymbol{\theta}, \zeta)$ is related to the policy loss $\mathcal{L}(\pi, \zeta)$ (see Sec. 2.3), as the policy is subject to the system dynamics whereas the utility is not. Nevertheless, it is assumed that maximizing the expected utility (9) yields more preferred trajectories.

Non-Linear Utility Functions. Few approaches allow the use of non-linear utility functions. The utility feature space is usually assumed to be defined in advance, shifting the problem to the user. However, defining such a feature space usually requires expert knowledge. Methods that can use non-linear utilities are easier to apply but may require a higher amount of preferences or trajectory samples, because the learning problem is more complex.

Gritsenko and Berenson (2014) convert the preferences into a ranking and do not operate directly on the preference level. Instead, their algorithm tries to learn a utility distance $d(U, \zeta_i)$, which models the rank difference k_i of a trajectory pair $\boldsymbol{\tau}_{i1}$ and $\boldsymbol{\tau}_{i1}$, i.e.,

$$L^{\succ}(U, \zeta_i) = \|d(U, \zeta_i) - k_i\|_2, \tag{14}$$

This loss function can be minimized by any regression algorithm. For example, Gritsenko and Berenson (2014) use M5' (Wang and Witten, 1997).

The rank values are subject to a somewhat arbitrary scale (e.g., the difference between mispredictions of neighboring ranks is always the same). As an alternative, the rank values can be encoded as classes instead of numeric values (Gritsenko and Berenson, 2014). Such a multi-class problem can be learned by any conventional classification algorithm, Gritsenko and Berenson (2014) use C4.5 (Quinlan, 1993). The classification model is again used as the utility function where the class is used as utility value. The multi-class loss can not be directly mapped to a preference loss $L^{\succ}(\boldsymbol{\theta}, \zeta_i)$ as the classification problem is discrete and disregards the rank or utility difference.

Christiano et al. (2017) use deep neural networks for approximating a non-linear utility function. They directly minimize the negative log likelihood $-\sum_{i=1}^{N} \log\left(p_{\boldsymbol{\theta}}(\zeta_i)\right)$ with a sigmoid likelihood function. The utility function is not linear in the features, but convex. Hence, it is possible to compute a gradient for the parameters $\boldsymbol{\theta}$ and use common backpropagation techniques for minimizing the loss.

Wirth et al. (2016) and Kupcsik et al. (2015) also allow for non-linear utility functions by using kernel-based feature spaces. While the likelihood function is linear in the resulting, infinite feature space, the effective feature space depends non-linearly on the used samples. Wirth et al. (2016) use Gaussian kernel functions whereas Kupcsik et al. (2015) employ Gaussian process preference learning (Chu and Ghahramani, 2005).

In case the utility function should be applicable to states or state/action pairs, the linearity of the MDPs reward has to be considered (Wirth et al., 2016). Hence, the utility must be linear over the state/action pairs within a trajectory and non-linearity has to be achieved by mapping the state/action features itself, as we will discuss in Sec. 3.3.3.

3.3 The Temporal Credit Assignment Problem

As in all sequence learning problems, a key problem is that it is usually not known which states or actions are responsible for the obtained preference. This *temporal credit assignment* problem is comparable to the delayed reward problem in classic reinforcement learning. It is possible to circumvent it by directly estimating a policy's return in order to approximate the policy value. Methods that provide explicit solutions to the credit assignment typically come with the advantage that standard, reward-based RL methods can be employed. Yet, if we try to solve the credit assignment problem explicitly, we also require the expert to comply with the Markov property. This assumption can easily be violated if we do not use a full state representation, i.e., if the expert has more knowledge about the state than the policy. Depending on how the considered algorithm solves the credit assignment problem, different types of utility functions or preferences can be inferred, which we will discuss in the next subsections.

3.3.1 Value-based Utility

Value-based utility functions estimate the expected long-term utility for a given state, similar to a value function in reinforcement learning. Wirth and Fürnkranz (2012, 2015); Runarsson and Lucas (2012, 2014) use state-preferences $s_{i1} \succ s_{i2}$, which are assumed to originate from the expected utility of the long-term behavior, if we are in the current state and follow an optimal policy. The temporal credit assignment problem is in this case left to the human expert and also not solved explicitly by the algorithm, as we can simply select actions, leading to states with maximal (long-term) utility. To compute the utility $U(s) = \boldsymbol{\theta}^T \boldsymbol{\phi}(s)$ from a state preference, we can again make use of the utility difference $d(\boldsymbol{\theta}, \zeta_i) = \boldsymbol{\theta}^T(\boldsymbol{\phi}(s_{i1}) - \boldsymbol{\phi}(s_{i2}))$ using one of the specified loss functions from Tbl. 1. Using the utility function, we can define the policy

$$\pi^*(a \mid s) = \mathbb{I}(a = \arg\max_{a'} \mathbb{E}_{\delta}[U(s') \mid s, a']), \tag{15}$$

as a greedy policy maximizing the expected value of the next state s'. However, this expectation can only be computed exactly if the transition model δ is known and the state space is discrete. Moreover, value-based utilities always depend on a specific, optimal policy. Hence, it is difficult to transfer the learned value-utility function to other domains.

3.3.2 Return-based Utility

Many approaches that use trajectory preferences circumvent the temporal credit assignment problem by disregarding the temporal structure of the solution. They directly optimize the policy return (Akrour et al., 2011, 2012; Jain et al., 2013, 2015; Gritsenko and Berenson, 2014; Kupcsik et al., 2015) as defined by a return-utility function $U(\boldsymbol{\tau})$. The return utility function $U(\boldsymbol{\tau}) = \boldsymbol{\theta}^T \boldsymbol{\psi}(\boldsymbol{\tau})$ is typically defined by a low dimensional trajectory feature space $\boldsymbol{\psi}(\boldsymbol{\tau})$.

This feature space is usually assumed to be domain-specific (Zucker et al., 2010; Jain et al., 2013, 2015; Gritsenko and Berenson, 2014), however, Akrour et al. (2011, 2012) proposed to cluster the state space using ϵ-means clustering (Duda and Hart, 1973) for defining trajectory features. The trajectory features are given by the relative amount of states in a trajectory belonging to a cluster. Such features can not be directly described as state features as they also depend on the length of the trajectory.

Kupcsik et al. (2015) also learn a return-based utility $U(\boldsymbol{\tau})$ but represent a trajectory $\boldsymbol{\tau} = [\boldsymbol{s}, \boldsymbol{\omega}]$ with a context vector $\boldsymbol{s}$ that describes the current task and a parameter vector $\boldsymbol{\omega}$

that specifies the policy. Instead of using a linear feature representation, a Gaussian process is used to model the return utility $U(\boldsymbol{\tau})$, see Sec. 3.2.3.

3.3.3 Reward-based Utility

Many approaches obtain a state-action utility function that resembles a reward function in classical RL methods, i.e., it evaluates the immediate quality of executing action a in state s without considering the long-term behavior. Using reward-based utilities allows us to apply standard RL methods to improve the policy.

Zucker et al. (2010) propose a simple solution by assuming state preferences $s_1 \succ s_2$ that can be directly used for learning a state utility $U(s)$. For learning a reward-based utility from trajectory preferences, Akrour et al. (2013, 2011, 2014) and Wirth et al. (2016) use the *feature average* (Ng and Russell, 2000) along a trajectory

$$\boldsymbol{\psi}(\boldsymbol{\tau}) = \sum_{t=0}^{|\boldsymbol{\tau}|} \gamma^t \boldsymbol{\varphi}(s_t, a_t), \tag{16}$$

to define trajectory features. Hence, the utility return is defined by

$$U(\boldsymbol{\tau}) = \boldsymbol{\theta}^T \boldsymbol{\psi}(\boldsymbol{\tau}) = \sum_{t=0}^{|\boldsymbol{\tau}|} \gamma^t U(s_t, a_t),$$

with $U(s_t, a_t) = \boldsymbol{\theta}^T \boldsymbol{\varphi}(s_t, a_t)$, which coincides with the definition of the return in reinforcement learning. The trajectory utility $U(\boldsymbol{\tau})$ is a linear sum over the state/action utilities but the state/action utilities themselves can be non-linear. As an example, Wirth et al. (2016) use a kernel function to obtain such a non-linear utility by applying the kernel on a state/action level. Sugiyama et al. (2012) proposed an alternative learning method, based on the sum of future utilities

$$\begin{aligned} U(s,a) &= \sum_{s' \in S} \left(\delta(s' \mid s, a) \left(U(s,s') + \max_{a' \in A} \sum_{s'' \in S} \delta(s'', s', a') U(s' \mid s'') \right) \right), \\ U(s,s') &= \boldsymbol{\theta}^T \boldsymbol{\varphi}(s,s'), \end{aligned} \tag{17}$$

with $U(s, s')$ as the utility for the state transition s to s'. However, this idea is only applicable with a known transition function and a discrete action space.

More recently, a method was proposed that does not require the utility to be linear in the state/action features. Christiano et al. (2017) define a learning method for a trajectory utility $U(\boldsymbol{\tau}) = \sum_{t=0}^{|\boldsymbol{\tau}|} U(s_t, a_t)$, that uses a deep neural network for $U(s_t, a_t)$. This method greatly improves on the scalability of the learning process and allows more expressive utility functions, but requires a high amount of preferences.

In contrast to conventional reinforcement learning, a discount factor of $\gamma = 1$ is used for $U(\boldsymbol{\tau})$ in all approaches because the expert should not need to consider the effects of decay. However, using an undiscounted return is not a major problem as all considered trajectories have a finite length. The parameters $\boldsymbol{\theta}$ can now be estimated by using one of the loss functions presented in Sec. 3.2.3, where trajectory preferences are used as constraints for

the utility return $U(\boldsymbol{\tau})$. The underlying assumption of using immediate utility functions is that the expert's evaluation function complies with the Markov property. If this is not the case, the decomposition of the utility return in immediate utilities is error prone. Instead of using state-action utilities $U(s, a)$, some algorithms (Wirth et al., 2016) also disregard action costs and just use a state utility function, i.e., $U(s, a) = U(s)$.

In contrast to value-based utility functions, reward-based utility functions can be easily transferred across domains. Moreover, a reward-based utility is also independent of the system dynamics and, therefore, often has a simpler structure than value based utilities rendering them simpler to learn and generalize.

3.3.4 Intermediate Action Preferences

Wirth and Fürnkranz (2013a) extend the action preference algorithm of Fürnkranz et al. (2012) by deriving additional action preferences for intermediate states in trajectory preferences, defining an approximate solution to the temporal credit assignment problem. As in the original algorithm, preferences are assumed to relate to the long-term expectation. The assignment problem is solved by attributing preferences to an (unknown) subset of action choices made in states occurring in both trajectories, when following a fixed policy in all other states. This enables the computation of the probability that a single state/action pair is responsible for the encountered preference.

This approach may converge faster than the one of Fürnkranz et al. (2012) using a lower amount of preferences. However, the speedup comes at the cost of possible approximation errors due to possibly incorrect, heuristic estimates of the preferences. Wirth and Fürnkranz (2013b) extend this idea further by obtaining more reliable, intermediate preferences based on the assumption that each time step contributed equally to the return of the trajectory. As a result, an even lower amount of preferences is sufficient. However, many domains do not satisfy this uniform reward assumption.

3.4 Trajectory Preference Elicitation

PbRL may be viewed as an interactive process that iteratively queries the expert's preference function to obtain more knowledge about the objectives of the expert while also exploring the system dynamics. In decision making, this process is also known as *preference elicitation*.

Several approaches disregard this problem and only work with an initial set of preferences (Wirth and Fürnkranz, 2012, 2015; Runarsson and Lucas, 2012, 2014; Gritsenko and Berenson, 2014; Zucker et al., 2010; Sugiyama et al., 2012). Fürnkranz et al. (2012) collect new preference feedback, but assume exhaustive samples of the system dynamics and, hence, no additional exploration is needed.

Interactive PbRL algorithms need to decide how to generate new trajectories and how to use these trajectories to query the expert's preference function. Typically, an exploration method is used that reduces the uncertainty of the transition function as well as the expert's preference function. Yet, some approaches apply two distinct methods for exploring the system dynamics and the preference space.

Usually, we want to minimize the amount of expert queries as they require costly, human interaction while generating new trajectories can be a cheaper, mostly autonomous process. Moreover, a trajectory can give us useful information about the system dynamics even if

it is not used for preference generation. Hence, the number of trajectories can be considerably higher than the number of generated preferences. Yet, many algorithms ignore the potentially higher costs of preference generation in comparison to trajectory generation, as we will elaborate in the following discussion.

3.4.1 Trajectory Generation

Interactive PbRL algorithms need to generate diverse trajectories. In order to be informative, the obtained preferences should be different from existing trajectories. Yet, the trajectories should also be close to optimal in order to obtain useful information. Furthermore, the trajectories need to contain sufficient information about the transition function to compute an optimal policy.

Undirected Homogeneous Exploration. *Homogeneous exploration* employs a single exploration to generate diverse solutions, either directed or undirected. Such methods are undirected if they only compute stochastic policies that allow deviations from the optimal strategy while directed methods implement a separate criterion for guiding the exploration in areas of the state space where uncertainty can be reduced. Directed methods usually work better in low-dimensional policy spaces but do not scale well.

Most approaches (Busa-Fekete et al., 2013, 2014; Jain et al., 2013, 2015; Wirth and Fürnkranz, 2013a,b; Kupcsik et al., 2015; Wirth et al., 2016; Christiano et al., 2017) only use undirected exploration for the system dynamics and make the assumption that this will also explore the expert's preference function sufficiently well. Busa-Fekete et al. (2013, 2014) use a *covariance matrix adaptation evolution strategy* (CMA-ES; Hansen and Kern 2004) for optimization and exploration. They implement a criterion for limiting the amount of samples that have to be evaluated to obtain a ranking. The algorithm maintains a collection of candidate policies and discards candidates once they can be considered suboptimal, according to the Hoeffding bound (Hoeffding, 1963). Jain et al. (2013, 2015) use *rapidly-exploring random trees* (LaValle and Kuffner, 1999) whereas Wirth and Fürnkranz (2013a,b) employ approximate policy iteration (Dimitrakakis and Lagoudakis, 2008) with an EXP3-like exploration method (Auer et al., 1995).

Wirth et al. (2016) use *relative entropy policy search* (REPS; Peters et al. 2010) for computing a stochastic policy based on the policy's return, estimated with *least squares temporal difference (LSTD)* learning (Boyan, 1999). Kupcsik et al. (2015) employ *contextual REPS* (Kupcsik et al., 2014), but directly in parameter space instead of state action space. This variant is able to evaluate parametric policies without requiring transition samples due to the black-box character of the algorithm. The result of the policy improvement step is a distribution over parametric policies that is used to sample new trajectories for querying new preferences. In turn, the distribution over possible utility functions is updated for computing an improved estimate of the expected utilities for the next policy update step.

Christiano et al. (2017) use Trust Region Policy Optimization (TRPO; Schulman et al. 2015) and (synchronous) A3C (Mnih et al., 2016), which are policy optimization algorithms for RL with deep neural networks. Both algorithms also ensure exploration by defining stochastic policies, hence they are undirected.

Directed Homogeneous Exploration. In contrast to undirected approaches, Wilson et al. (2012) propose two directed exploration criteria that are derived from the preference learning method explained in Sec. 3.4.3. However, they also only use a single, homogeneous exploration method. The policy improvement is computed by updating the posterior distribution of a parametric policy space, given a preference-based likelihood function. A new preference query is then created by applying two policies to the MDP, selected by the directed exploration criterion. The computation of these criteria requires multiple trajectories, but they are not used for the queries themselves or for the policy improvement step.

Heterogeneous Exploration. *Heterogeneous exploration* methods use two distinct strategies for exploring the system dynamics and the preference function. For example, Akrour et al. (2011, 2012, 2014) employ different criteria that are well suited for the two tasks. A policy search strategy with an intrinsic exploration method is used to optimize a given utility function. In addition, the utility function implements a criterion for directed exploration of the preference function. Therefore, the resulting policy can be used to sample trajectories for preference queries that reduce the uncertainty over the expert's feedback. Such a strategy comes at the cost of sampling new trajectories for multiple policy update steps before each new preference query. Akrour et al. (2014) propose an alternative version using *least squares policy iteration* (Lagoudakis and Parr, 2003), that does not require additional samples for updating the policy in an off-policy manner. Yet, this approach requires a sufficient amount of samples collected in advance which is unrealistic in many scenarios.

User-Guided Exploration. *User-guided exploration* allows the expert to provide additional trajectories to guide the exploration process. The algorithm by Zucker et al. (2010) assumes a set of initial state preferences and optimize the resulting utility function using a randomized planner (*anytime A**; Likhachev et al. 2003). The planning algorithm samples new trajectories and the expert can select any encountered state to provide additional preferences, i.e., the expert is guiding the exploration of its preference space. In case the expert does not provide additional preferences, no further exploration of the preference function is used. The approach of Jain et al. (2013, 2015) extends the used homogeneous strategy with user-guided exploration. Given a trajectory that has been autonomously generated, the expert may correct the trajectory which simultaneously also provides a preference.

3.4.2 Preference Query Generation

Having generated the trajectories, we need to decide which trajectories to use to query the expert's evaluation function. Typically, the newly generated trajectories are compared to so far undominated trajectories (Akrour et al., 2012, 2011; Wirth et al., 2016; Akrour et al., 2014). Yet, this solution is only feasible if all trajectories are directly comparable, e.g., they start in the same initial state, which is a standard assumption for most PbRL algorithms. Alternatively, both trajectories are explicitly generated by the algorithm (Busa-Fekete et al., 2013, 2014; Jain et al., 2013, 2015; Wirth and Fürnkranz, 2013a,b; Kupcsik et al., 2015; Wilson et al., 2012) or the user selects the samples for a preference query (Zucker et al., 2010).

Exhaustive Preference Query Generation. Some approaches require that each trajectory is evaluated by the expert by defining preferences regarding every possible trajectory pair, either for comparing and selecting policies (Busa-Fekete et al., 2013, 2014) or for evaluating and updating a single policy (Jain et al., 2013, 2015; Wirth and Fürnkranz, 2013a,b;

Kupcsik et al., 2015). These approaches ignore that generating preferences can be more costly than generating trajectories and, hence, typically require a large amount of expert queries.

Greedy Preference Query Generation. In greedy approaches (Akrour et al., 2014), we fully optimize a given objective, described in Sec. 3.4.3, which can be either an undirected or directed exploration objective. After the optimization is finished, we use a trajectory generated from the optimized policy to query the expert's evaluation function. While this approach is typically much more efficient in the number of expert queries, it may need many trajectories to be generated in order to optimize the posed objective. Furthermore, many of these trajectories might have been unnecessarily generated as the posed objective is typically quite error-prone in the beginning of the learning process. Hence, the optimization might explore poor trajectories which could have been avoided if new preferences would have been requested earlier before reaching the maximum of the given objective.

Interleaved Preference Query Generation. The above-mentioned problem was recognized by several authors (Akrour et al., 2012, 2011; Wirth et al., 2016; Wilson et al., 2012; Christiano et al., 2017). A simple solution to alleviate it is to prematurely stop the optimization of the given objective and request new preferences. The preferences are subsequently used to update the given optimization objective (see Sec. 3.4.3). This approach is followed by Akrour et al. (2012, 2011) where the search in parameter space is stopped early, after a given number of iterations. Wirth et al. (2016); Christiano et al. (2017) request new preferences after a single iteration of the used policy optimization algorithm. In each iteration, several trajectories are generated by the current policy but only a subset (typically one) is used for an expert query. Wirth et al. (2016) propose to use the trajectory with the highest expected accumulated utility, whereas Christiano et al. (2017) use a variance estimate, based on an ensemble technique. Wilson et al. (2012) apply the same idea to a parametric policy space, using a selection criterion based on expected belief change. An alternative method selects two policies that are used for creating trajectories for a new preference query. In contrast to Wirth et al. (2016), this algorithm completely discards trajectories that are not used for preference queries, as it is only able to use explicitly evaluated trajectories. Hence, not all trajectories are evaluated, but only evaluated trajectories are used to update the policy.

The resulting algorithms allow to choose the trade-off between the cost of trajectory generation and preference generation by controlling the ratio of generated trajectories to generated preferences. Akrour et al. (2012, 2011); Wirth et al. (2016) can also efficiently explore the trajectory space as well as the preference space as they can use trajectories not evaluated by the expert for policy optimization.

3.4.3 Objectives for Exploring the Preference Function Space

A few algorithms formulate an explicit objective in order to efficiently reduce the uncertainty over the expert's preference function. Different versions have been proposed, but no exhaustive comparison of all exploration methods can be found in the literature. A limited, comparative evaluation was conducted by Wilson et al. (2012) who compared two criteria proposed in the paper.

They suggest to sample multiple policies from the posterior distribution of the policy space and use them to compute trajectories for preference queries. The first method only

considers the expected difference of the trajectories generated by two policies

$$\int_{(\boldsymbol{\tau}_i,\boldsymbol{\tau}_j)} \Pr(\boldsymbol{\tau}_i \mid \pi_i) \Pr(\boldsymbol{\tau}_j \mid \pi_j) d(\boldsymbol{\tau}_i, \boldsymbol{\tau}_j), \tag{18}$$

where $d(\boldsymbol{\tau}_i, \boldsymbol{\tau}_j)$ is a trajectory difference function (see (4)). If two policies disagree greatly on what trajectories to create, it is assumed to be beneficial to rule out one. Therefore, samples from π_i and π_j are evaluated and a query is posed in case the value of the criterion exceeds a given threshold.

The second criterion computes the expected belief change over the policy space that results from adding a preference

$$V(\Pr(\pi \mid \zeta) || \Pr(\pi \mid \zeta \cup \{\boldsymbol{\tau}_{i1}, p_i, \boldsymbol{\tau}_{i2}\}), \tag{19}$$

using potential queries by sampling policies from the current distribution. Different measures could be used as V, e.g. the Kullback-Leibler divergence. Wilson et al. (2012) use an approximation of the variational distance. The query maximizing the criterion, based on a finite trajectory sample set, is then posed to the expert.

Akrour et al. (2011, 2012, 2014) define a utility function that includes an exploration term. The simplest form

$$\begin{aligned} \tilde{U}(\pi) &= U(\pi) + \alpha E(\pi), \\ U(\pi) &= \mathbb{E}_{\Pr_\pi(\boldsymbol{\tau})}[U(\boldsymbol{\tau})], \\ E(\pi) &= \min \Delta(\pi, \pi'), \pi' \in \Pi_t, \\ \Delta(\pi, \pi') &= \frac{|S_1| + |S_2| - |S_3|}{\sqrt{|S_1| + |S_3|}\sqrt{|S_2| + |S_3|}}, \end{aligned} \tag{20}$$

used by Akrour et al. (2011), adds a diversity term to the expected utility of a policy, comparing the states in S_1, only visited by π, the states in S_2, only visited by π' and the states in S_3 visited by both policies. Π_t are all policies obtained up to the current iteration. Candidate policies are generated by an evolutionary strategy and $\mathbb{E}_{\Pr_\pi(\boldsymbol{\tau})}[U(\boldsymbol{\tau})]$ is approximated via samples. The diversity function $\Delta(\pi, \pi')$ rewards differences in the states visited and is only applicable to discrete state spaces, however, a discrete description can be obtained using a clustering algorithm (see Sec. 3.3.2). Akrour et al. (2012, 2014) propose a different selection criterion that maximizes the expected utility of selection

$$\tilde{U}(\boldsymbol{\tau}) = \Pr(\boldsymbol{\tau} \succ \boldsymbol{\tau}^* \mid U) \mathbb{E}_{\zeta \cup \{\boldsymbol{\tau} \succ \boldsymbol{\tau}^*\}}[U(\boldsymbol{\tau})] + \Pr(\boldsymbol{\tau} \prec \boldsymbol{\tau}^* \mid U) \mathbb{E}_{\zeta \cup \{\boldsymbol{\tau} \prec \boldsymbol{\tau}^*\}}[U(\boldsymbol{\tau}^*)], \tag{21}$$

with $\boldsymbol{\tau}^*$ as the currently undominated trajectory. It computes the expected utility for the undominated trajectory, differentiating the cases where the new trajectory $\boldsymbol{\tau}$ is preferred over the current best and where it does not improve. $\Pr(\boldsymbol{\tau} \succ \boldsymbol{\tau}^* \mid U)$ is the probability of improvement, given the current distribution over utility functions. The expected utility $\mathbb{E}_{\zeta \cup \{\boldsymbol{\tau} \succ \boldsymbol{\tau}^*\}}[U(\boldsymbol{\tau})]$ is then computed with the assumption that the new preference is added to the preference set ζ. The two publications (Akrour et al., 2012, 2014) differ in the method used to approximate the criterion.

Wirth et al. (2016) also maintains the current, best trajectory for preference queries. The trajectory maximizing the expected utility

$$\boldsymbol{\tau} = \arg\max_{\boldsymbol{\tau} \in \boldsymbol{\tau}^{[i]}} \mathbb{E}\left[U_\zeta(\boldsymbol{\tau})\right], \tag{22}$$

is selected to obtain the second trajectory for the preference query. $\boldsymbol{\tau}^{[i]}$ is a set of trajectories obtained by sampling from the current policy π_i. Trajectories obtained from sampling former policies are not considered.

Christiano et al. (2017) also use trajectories obtained from the current policy, but evaluate pairs based on the variance of the utility. They approximate the variance by obtaining utility samples from an ensemble of utility functions. Each utility function uses a randomly sampled subset of the available preferences, hence, the learned functions may differ.

3.5 Policy Optimization

A major consideration is how the policy is optimized and how the policy optimization method affects the optimality of the learned policy and the sample requirements of the algorithm.

3.5.1 Direct Policy Search

Direct policy search approaches typically directly search in the parameter space of the policy and do not assume the Markov property or use dynamic programming such as standard reinforcement learning approaches. They maximize the objective defined by (2) and (3), however, this is difficult to compute directly due to the dependence on the transition dynamics. Hence, methods like policy likelihood or ranking (cf. Sec. 3.2.1) are used. Alternatively, common return-based optimization can be employed by defining an utility-based return, as introduced in Sec. 3.3.2.

Direct policy search methods are typically not very sample-efficient, as the temporal structure of the problem is ignored, but they can obtain good results because of their simplicity. These methods can be typically used for policies with a moderate number of parameters, typically less than one hundred.

Wilson et al. (2012) employ *Markov Chain Monte Carlo* (MCMC; Andrieu et al. 2003) sampling to obtain the posterior distribution of a parametric policy space. The sampling is computational costly, but does not require new trajectory samples. A sampling approach is also used by Busa-Fekete et al. (2013, 2014), but within an *evolutionary algorithm* where policies are described with a set of parameters. Akrour et al. (2011, 2012, 2014) also use an evolutionary strategy ($1 + \lambda$ ES; Auger 2005) for optimization in a parametric policy space, which requires a high amount of trajectory samples.

The contextual REPS (Kupcsik et al., 2014) algorithm, as employed by Kupcsik et al. (2015), is also a strategy for optimizing parameters of the policy. In the contextual settings, we can learn to adapt the policy parameters to the context of the task. For example, it can learn to adapt the robot's trajectory if we need to throw a ball for a larger distance (Kupcsik et al., 2014).

3.5.2 Value-based Reinforcement Learning

Value-based approaches exploit the temporal structure of the problem by using the Markov property, which, in principle allows for more sample efficient policy optimization. However, these methods might suffer from approximation errors which can lead to instabilities and divergence. Typically, value-based methods use some variant of policy iteration that is again composed of a policy evaluation step and a policy improvement step. In the policy evaluation step, the estimated reward-based utility (Akrour et al. 2014; Wirth et al. 2016; Christiano et al. 2017; cf. Sec. 3.2.3) is used to compute a value function for the policy. This value function can, for example, be estimated by temporal difference methods such as *least squares temporal difference learning* (LSTD; Boyan 1999). Given the value function, a new policy can be obtained in the policy improvement step. For example, Wirth et al. (2016) uses a variant of the *relative entropy policy search* (REPS) algorithm (Peters et al., 2010) to update the policy. The REPS algorithm performs a soft-greedy policy update that stabilizes the policy iteration process and is also applicable in continuous action spaces. The TRPO (Schulman et al., 2015) algorithm employed by Christiano et al. (2017) works comparably, but uses deep neural networks for approximating the value function and the policy. A separate algorithm for computing the value function is not required.

In discrete action spaces, greedy updates using the max-operator have been used (Akrour et al., 2014). However, this approach requires a sufficient amount of transition samples to be obtained beforehand which again stabilizes the policy iteration process.

Other authors (Fürnkranz et al., 2012; Wirth and Fürnkranz, 2012, 2013a,b, 2015; Runarsson and Lucas, 2012, 2014; Sugiyama et al., 2012) assume a utility function directly defining the expected outcome of a state or action. Hence, an optimal policy is derived by directly selecting the action maximizing the utility. In case of state values (Wirth and Fürnkranz, 2012, 2015; Runarsson and Lucas, 2012, 2014), this step requires the transition model to be known.

3.5.3 Planning

Many other approaches (Zucker et al., 2010; Jain et al., 2013, 2015; Gritsenko and Berenson, 2014) use planning algorithms for optimizing policies. Planning-based algorithms use utility-based approaches to derive an evaluation measure for trajectories (or state/actions). Furthermore, a model of the system dynamics is required to determine the effects of actions. It is then possible to plan a sequence of actions that maximizes the given utility. However, a model of the system dynamics is in many cases not available with a sufficient accuracy.

The use of sampling-based *rapidly-exploring random trees* (RRT; LaValle and Kuffner 1999) and their *constrained bi-directional* variant (CBiRRT; Berenson et al. 2009) was explored in (Jain et al., 2013, 2015; Gritsenko and Berenson, 2014). RRTs are guaranteed to find a solution to a planning problem, but provide no guarantees for the optimality of the found solution. Other planning algorithms come with such guarantees, such as *anytime A** (ARA*; Likhachev et al. 2003), as used by Zucker et al. (2010). Planning methods require an accurate model of the system dynamics and inherently suffer from model errors. Small inaccuracies in the model might be exploited by the planning method which may result in solutions that are not feasible on the real system.

Learning a Policy	Preferences	Traj. Gen.	Pref. Gen.	Surrogate Opt.	Policy Opt.	Require.
Busa-Fekete et al. (2013, 2014)	Trajectory	Homogen	Exhaustive	-	CMA-ES	PP
Wilson et al. (2012)	Trajectory	Homogen	Interleaved	-	MCMC	PP,M/S_0
Learning a Preference Model	**Preferences**	**Traj. Gen.**	**Pref. Gen.**	**Surrogate Opt.**	**Policy Opt.**	**Require.**
Fürnkranz et al. (2012)	Action	-	Exhaustive	MLP	Direct Max.	M/S_0, DA
Wirth and Fürnkranz (2013a,b)	Trajectory	Homogen	Exhaustive	Gradient	Direct Max.	DS, DA
Learning a Utility Function	**Preferences**	**Traj. Gen.**	**Pref. Gen.**	**Surrogate Opt.**	**Policy Opt.**	**Require.**
Akrour et al. (2011, 2012)	Trajectory	Heterogen	Interleaved	SVM	$1+\lambda$-ES	PP
Akrour et al. (2014)	Trajectory	Heterogen	Greedy	MCMC	LSTD, CMA-ES	MA or PP
Christiano et al. (2017)	Trajectory	Homogen	Interleaved	Gradient	TRPO, A3C	
Gritsenko and Berenson (2014)	Trajectory	-	-	C4.5, M5P	CBiRRT	M
Jain et al. (2013, 2015)	Trajectory	Homogen+Guided	Exhaustive	Gradient	RRT	M
Kupcsik et al. (2015)	Trajectory	Homogen	Exhaustive	convex solver	C-REPS	PP
Runarsson and Lucas (2012, 2014)	State	-	-	SVM	Direct Max.	M
Sugiyama et al. (2012)	Trajectory	-	-	convex solver	Direct Max.	M, DS, DA
Wirth and Fürnkranz (2012, 2015)	State	-	-	SVM, LP	Direct Max.	M
Wirth et al. (2016)	Trajectory	Homogen	Interleaved	LP, ELS	LSTD, REPS	
Zucker et al. (2010)	State	Homogen	Guided	SVM	ARA*	M

Table 2: Overview of PbRL approaches by design principle

Algorithms: $1+\lambda$-*ES*: $1+\lambda$ Evolution Strategy (Auger, 2005), *A3C*: Asynchronous Advantage Actor-Critic (Mnih et al., 2016), *ARA**: Anytime A* (Likhachev et al., 2003), *C-REPS*: Contextual Relative Entropy Policy Search (Kupcsik et al., 2014), *C4.5*: C4.5 Classifier (Quinlan, 1993), *CBiRRT*: Constrained Bi-directional Rapidly-Exploring Random Tree (Berenson et al., 2009), *CMA-ES*: Covariance Matrix Adaptation Evolution Strategy (Hansen and Kern, 2004), *ELS*: Elliptic Slice Sampling (Murray et al., 2010), *LP*: Linear Programming, *LSTD*: Least Squares Temporal Difference Learning (Boyan, 1999), *M5P*: M5P Regression Model Trees (Wang and Witten, 1997), *MCMC*:Markov Chain Monte Carlo (Andrieu et al., 2003), *MLP*: Multi-layer Perceptron (Bishop, 1995), *REPS*: Relative Entropy Policy Search (Peters et al., 2010), *RRT*: Rapidly-exploring Random Tree (LaValle and Kuffner, 1999), *SVM*: Ranking Support Vector Machine (Joachims, 2002; Herbrich et al., 1999), *TRPO*: Trust Region Policy Optimization (Schulman et al., 2015)

Requirements: *DA*: discrete action space, *DS*: discrete state space, *M*: transition model is known, *MA*: pre-generated samples required, *PP*: parametric policy, S_0: all states can be used as initial state

3.6 Modelling the Transition Dynamics

The reviewed algorithms also differ with respect to the amount of available model knowledge, which has a big influence on which tasks the approaches are applicable. *Model-based approaches* assume that the system dynamics are known in advance, which allows the use of planning algorithms (Jain et al., 2013, 2015; Gritsenko and Berenson, 2014; Zucker et al., 2010) or to directly derive an optimal policy (Wirth and Fürnkranz, 2012, 2015; Runarsson and Lucas, 2012, 2014; Sugiyama et al., 2012). In addition, some approaches require a model to be able to simulate the system initialized at arbitrary states (Fürnkranz et al., 2012; Wilson et al., 2012).

The transition dynamics of real world problems are usually not known, as they are subject to a wide variate of (unknown) factors. Hence, many model-based approaches try to learn the dynamics based on observed transitions (Grünewälder et al., 2012). However, model learning can be a complicated task in particular for continuous systems and is a research field on its own (Nguyen-Tuong and Peters, 2011). It might require a high amount of samples, we need to choose a suitable model class and the resulting model errors often decrease the quality of the resulting policy.

Alternatively, *model-free approaches* neither assume a model, nor approximate it (Fürnkranz et al., 2012; Wirth and Fürnkranz, 2013a,b; Busa-Fekete et al., 2013, 2014; Wirth et al., 2016; Akrour et al., 2014). For example, Akrour et al. (2014) assume a large database of pre-generated samples which can be subsequently used for model-free, off-policy reinforcement learning algorithms. Most model-free PbRL approaches, with the exception of Wirth et al. (2016), are not able to reuse samples, and therefore require a large amount of samples for comparing or evaluating policies. All algorithms that employ direct policy search strategies are in principle model-free, as the only requirement of such algorithms is that a policy can be executed on the real system. However, as these algorithms often require a lot of policy executions in order to find an optimal policy, it is often assumed that a simulator is available to avoid the costly optimization on the real system. A notable exception is the algorithm by Kupcsik et al. (2015) where the policy optimization is performed without requiring the transition dynamics. Trajectories from the real system are only used to compute an evaluation function for parametric policies.

3.7 Summary of PbRL Algorithms

Tbl. 2 shows a tabular summary of the discussed algorithms, grouped by the different learning problems, introduced in Sec. 3.2. We list the introduced key concepts and the different *requirements*, stated throughout this paper. The *preferences* column is related to the different types of preferences mentioned in Sec. 3.1. The *trajectory generation* and *preference (query) generation* columns are based on the subsections of Sec. 3.4, where we mention how trajectories and preference queries are generated to resolve the exploration problem. The various approaches use different optimization strategies for performing policy optimization and surrogate loss minimization, if applicable. The columns *surrogate optimization* and *policy optimization* show the strategies mentioned in Sec. 3.2 and Sec. 3.5 explicitly.

4. Experiment Domains

Most practical applications for PbRL focus on robot teaching tasks and on board game domains. These domains can be subject to complex reward shaping problems (Ng et al., 1999) as it is required to not only solve the task itself, but also consider constraints like motor limitations or obstacle avoidance. Furthermore, a robot acting in uncontrolled environments needs to cope with changing environments, interact with non expert users and adapt to new tasks. For such tasks, PbRL is a suitable tool as it allows robots to learn without expert knowledge or predefined rewards. In the following, we highlight a few domains where preference-based approaches have been successfully employed in the past.

4.1 Object Handover

Kupcsik et al. (2015) use a preference-based approach to learn a robot handover task where a robot has to pass a bottle to a person sitting, walking or running, as shown in Fig. 4. The handover should be performed as comfortably as possible for the human which approaches at different speed. Hence, the stiffness as well as the trajectory of the robot's end-effector need to be adapted by the learning procedure.

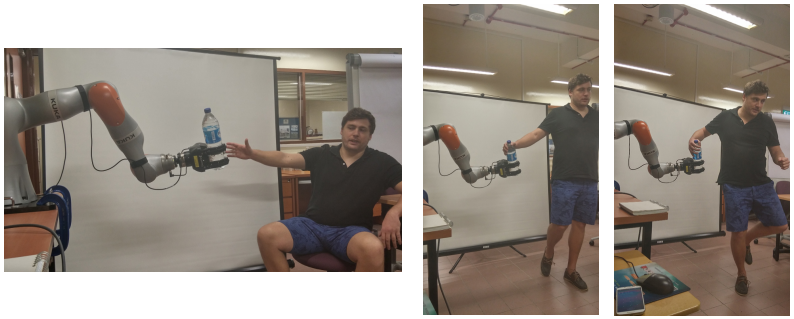


Figure 4: The robot handover task with a sitting, standing and running person

As described in Strabala et al. (2013), this task consists of three stages: approach, signal and transfer. First, the robot needs to approach the human and get ready for the actual handover. In the signaling phase, the robot and the human need to exchange information for determining the intended joint position for the handover. This information exchange is often non-verbal. Lastly, the joint contact needs to be established. This physical transfer should be dynamic and should not require a stationary contact phase. The focus of Kupcsik et al. (2015) is on this last phase. Experiments are performed with a real *7-DoF KUKA LBR arm* and a *Robotiq 3-finger hand* in a laboratory setup, as shown in Fig. 5. The robot performs several handover trials with a human receiving the object. Feedback is obtained in terms of trajectory preferences, but can be enhanced with an ordinal evaluation on a 1–10 scale. The trajectories are considered to be samples of a parametric policy and the related feedback is used to learn an evaluation function for policies, as described in Sec. 3.2.3 and 3.4.1. The evaluation function uses the policy parameters as input, allowing to define a

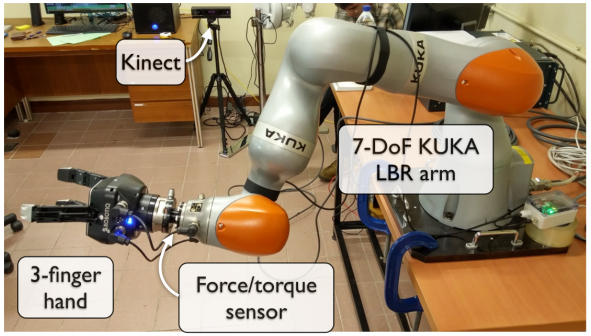


Figure 5: The robot KUKA robot with a 3-finger hand

likelihood function for the feedback, given a parametric policy. Using Bayesian inference and a Gaussian process prior, it is possible to infer the policy evaluation function explicitly. Policies can be evaluated without obtaining trajectory samples by directly evaluating the policy parameters. Hence, no costly environment calls are needed for optimizing the policy.

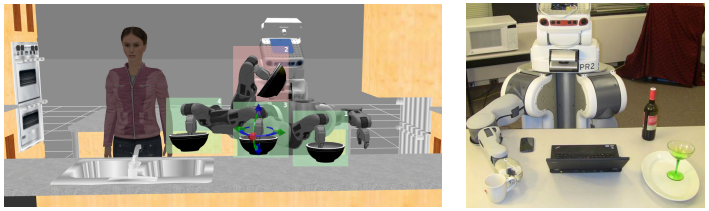
The performed evaluation shows a steady improvement, averaged over 5 trials. Furthermore, the return of the policy approaches a plateau after 80 policy updates. However, the exact amount of used preferences as well as the performance difference to an optimal policy remain unclear.

In general, the approach is very interesting for domains where (low-dimensional) parametric policies are available and trajectory generation is expensive. Only learning the utility function requires trajectory sampling. Furthermore, the process can probably be improved further by considering specific preference function exploration criteria, as introduced in Sec. 3.4.3. However, the approach is most likely not suited for high-dimensional policy spaces because of the Gaussian process. Gaussian process-based methods are computational expensive and require a high number of samples to sufficiently capture complex policies. Hence, a high number of trajectory samples and preferences would be required.

4.2 Path Planning

Jain et al. (2013, 2015) also tries to solve robotic tasks, but use a simulated setup, shown in Fig. 6a. Due to the simulated setup, the authors were able to evaluate their system on a wide range of path planning tasks, namely:

- pick and place, e.g., moving a glass of water without spilling water;
- inserting, e.g., placing or knife inside a holder;
- pouring, e.g., pouring water into a cup;
- heavy-fragile movement, e.g., moving a heavy object with a fragile object close by;
- liquid-electronic movement, e.g., moving a glass of water with a laptop in vicinity;



(a) The simulation environment

(b) The real robot

Figure 6: The experiment setup of Jain et al. (2013, 2015)

- sharp-object movement, e.g., moving a knife with a human in range;
- hot-object movement, e.g., serving a hot drink.

Overall, for evaluating their approach, they defined 35 tasks by changing the objects and/or environment. The robot demonstrates a set of trajectories, which are ranked by the algorithm. The user can then define preferences by re-ranking them or by supplying new trajectories, which are assumed to be an improvement over all trajectories demonstrated by the robot. Hence, they are added to the ranking as the top-ranked element. Preferences are then obtained by computing all pairwise preferences contained in the ranking.

For evaluating the approach, an expert scored 2100 trajectories in total, based on a 1–5 scale. From these scores, the authors derive a ranking and compare it with a ranking based on the learned utility. They achieve an nDCG@1(nDCG@3) ranking error (Manning et al., 2008) of 0.78(0.66) to 0.93(0.92) after obtaining feedback 20 times. The ranking error is averaged over multiple tasks, grouped into three categories. The used feedback is only a re-ranking of 5 obtained trajectories by picking the most preferred one. The exact amount of derived preferences is unclear.

The work of Jain et al. (2013, 2015) is especially interesting because it was demonstrated to be applicable in a wide range of domains. In contrast to Kupcsik et al. (2015), a low-dimensional policy space is not required, but a predefined trajectory feature space. Arguably, this requirement is easier to satisfy and apply. However, the utility function has to be linear in the given feature space but the method could be combined with non-linear utility functions, as introduced by Wirth et al. (2016).

4.3 Chess Agent

An application of PbRL to board games was demonstrated by Wirth and Fürnkranz (2012, 2015). They use already available data from the chess domain to learn an artificial chess agent. Professional chess games are commonly stored in large databases, where many games annotate move and position with a unified symbol set, as shown in Fig. 7. The annotation symbols define the quality of a position or move, e.g., the white/black player has a decisive/moderate/slight advantage over her opponent[3]. In the given example, the annotator

3. http://portablegamenotation.com/Symbols.html

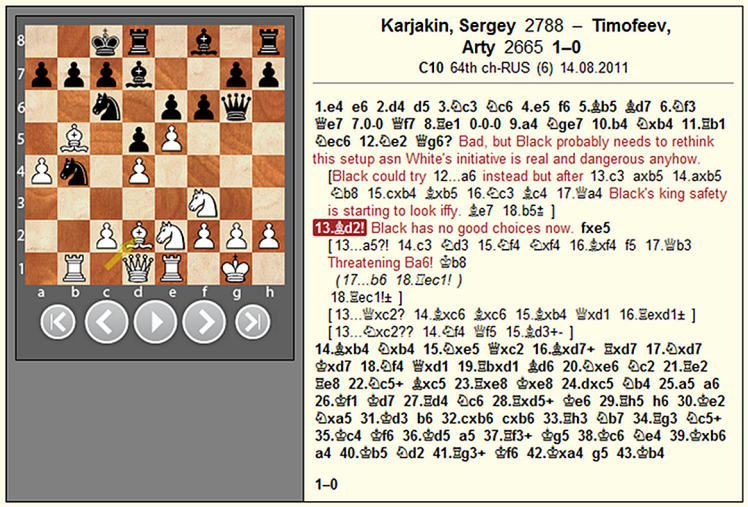


Figure 7: A chess position with move history and annotations.

remarked that White's move 13.♗d2! was a very good move (indicated by the exclamation mark at the end of the move), and provides three alternative replies for the move 13...fxe5 that was played in the game: 13...a5?! is dubious, 13...♕×c2? is a mistake, and 13...♘×c2?? is a losing move. The positions at the end of these variations are also annotated with symbols like ±, indicating a small advantage for White, or +− indicating a decisive advantage for White. It is straight-forward to derive preferences over moves and positions from such annotations. The preferences can then be used to learn a state utility function which estimates the expected outcome of a position. A policy is then derived by maximizing the utility, using the already known chess transition function.

The results in this domain show that it is possible to approach the performance of a near-optimal handcrafted utility function. However, the best results require 2.12M preferences and use a pre-defined utility feature space with only 15 dimensions.

It can be concluded that this approach is interesting for domains where it is easily possible to derive high number of (noisy) preferences. However, if optimal policies are of importance, it should probably only be used for computing an initial solution that can be further refined by applying one of the other PbRL approaches.

4.4 Other Domains

Nearly all other approaches are evaluated on classical reinforcement learning toy domains, where qualitative preferences are derived from the quantitative returns of the trajectories. Commonly used domains are the *mountain car* (Sutton and Barto, 1998), *pendulum swing-up* (Åström and Furuta, 2000), *cart-pole balancing* (Wang et al., 1996), *acrobot* (Spong, 1994),

and the *bicycle balance* (Randløv and Alstrøm, 1998) domains. Two approaches also use a simple grid world defined by Akrour et al. (2014). This *grid world* is easy to solve from a reinforcement learning point of view, but difficult for preference learning as the ground-truth reward values are very similar for some states. Typical state-of-the-art approaches require $5 - 40$ preferences to compute an optimal policy in these domains. Several algorithms are also evaluated on a *cancer treatment* task, derived from a reward based setting by Zhao et al. (2009). More complex toy domains have been used by Christiano et al. (2017), as they employ the ATARI learning environment (Bellemare et al., 2013) and the MuJoCo framework (Todorov et al., 2012) for learning movement and other robotic tasks. Interestingly, they have been able to outperform state-of-the-art RL approaches with numeric rewards in several domains, however they required several hundred preferences.

Runarsson and Lucas (2012, 2014) use the board game Othello, in a setting that is similar to the experiments by Wirth and Fürnkranz (2012, 2015). However, preferences are not derived from human annotations, but by assuming that observed state/action choices are preferable to unseen alternatives. This relates the approach closely to inverse reinforcement learning, although, the system directly learns a value function instead of a reward function. Sugiyama et al. (2012) derive preferences from numeric ratings and also learn a value function, however, in a human dialog setting. They show improvements over classic RL and IRL approaches. A detailed overview of experiment domains and evaluated approaches is given in Tbl. 3. Domain setups are not always identical and may differ in details such as the noise level, used feature space or action limits. Hence, even in the few cases where two papers do publish results of the same evaluation domain, the results are usually not directly comparable.

5. Discussion

While there has been much progress in the recent years in PbRL, many open questions and shortcomings of the approaches remain. For example, none of the utility-based approaches are able to produce a set of Pareto-optimal policies in case of incomparabilities. Only the *direct policy learning* method of Busa-Fekete et al. (2014) can handle incomparabilities by assuming that the incomparable trajectories should be evaluated equally, i.e., that both trajectories receive the same probability. A potential solution to cope with incomparabilities would be to learn a non-scalar utility function, allowing the utility functions to differ in multiple dimensions. A set of Pareto-optimal policies could then be learned using *multi-objective reinforcement learning* (Liu et al., 2015).

The question of preference-space exploration has also not yet been addressed satisfactorily. While a few approaches for preference-space exploration have been proposed (Sec. 3.4.3), a principled integration into the policy-space exploration strategy is still missing. Both tasks should be combined efficiently to reduce the amount of trajectory samples and preferences that are required. In addition, very few utility-based approaches are able to learn non-linear utility functions. Gritsenko and Berenson (2014) learn non-linear utilities but require that the PbRL problem is turned into an ordinal ranking problem first (Sec. 3.2.3). Ordinal rankings are problematic in that arbitrary distances between ranks are introduced when they are treated as regression problems, whereas the classification case suffers from a severe loss of information. Wirth et al. (2016); Kupcsik et al. (2015) employ

	classic discrete			classic continuous					robotics						games				
	End of Maze	**Grid world**	**River Swim**	**Acrobot**	**Bicycle Balance**	**Inverted Pendulum**	**Mountain Car**	**Swing Up**	**MuJoCo**	**NAO Robot**	**path planning**	**unhang wheel**	**terrain locomotion**	**object handover**	**ATARI**	**Chess**	**Othello**	**Cancer**	**Dialog Data**
Akrour et al. (2011)	✗		✗																
Akrour et al. (2012)							✗											✗	
Akrour et al. (2014)		✗			✗	✗				✗									
Busa-Fekete et al. (2013, 2014)																		✗	
Christiano et al. (2017)									✗						✗				
Fürnkranz et al. (2012)						✗	✗											✗	
Gritsenko and Berenson (2014)												✗							
Jain et al. (2013, 2015)											✗								
Kupcsik et al. (2015)														✗					
Runarsson and Lucas (2012, 2014)																	✗		
Sugiyama et al. (2012)																			✗
Wilson et al. (2012)				✗	✗	✗	✗												
Wirth and Fürnkranz (2013a,b)				✗	✗	✗													
Wirth and Fürnkranz (2012, 2015)																✗			
Wirth et al. (2016)		✗		✗	✗			✗											
Zucker et al. (2010)													✗						

Table 3: Overview of evaluation domains

kernels for overcoming this problem, an idea applicable to all linear utility function learning problems. However, this results in a high-dimensional utility space and is computational costly. Hence, the approaches do not scale well to high dimensional spaces. The deep neural network based approach by Christiano et al. (2017) overcomes the scalability issues of kernel-based methods but requires a high amount of preferences. Further research is required to either scale preference-based learning with kernels to larger problems spaces or to improve the results from DNN-based approaches with low amounts of training data.

Another important research question is to find principled ways for combining various types of feedback from the human expert, such as preferences, ordinal or numeric feedback, and demonstrations. In particular, inverse reinforcement learning from demonstrated trajectories could be used to provide good initializations for learning a utility function. This utility could then be refined by preferences of the experts.

Furthermore, PbRL can currently not be used to perform risk-averse optimization (Coraluppi and Marcus, 1999; Geibel, 2001). Risk-adverse optimization guarantees that

the worst case performance of the algorithm is bounded. In many cases, it is required that certain trajectories are not realized, for example, trajectories that would damage the robot or its environment. Such risk-adverse behavior could be realized with weighted preference loss functions such as (3), i.e., by substantially increasing the weights of preferences that concern risky trajectories.

Another important open issue, which would substantially extend the range of possible applications of PbRL, is to develop techniques that are able to use preferences between trajectories starting from different initial states. A potential solution could be to take the expected value of initial states into account and compute the preference based on the advantage function.

As mentioned in Secs. 3.4 and 3.5, most approaches are lacking in terms of sample efficiency. While sample efficiency could already be improved by using reward-based utilities (see Sec. 3.2.3) which allow sample reuse for policy optimization (Wirth et al., 2016), more efficient model-based policy search strategies such as the PILCO algorithm (Deisenroth and Rasmussen, 2011) or guided policy search (Levine and Koltun, 2013) could be employed that learn a system dynamics model from the data and employ it for policy optimization. Moreover, we have seen that the complexity of the tasks that have been used for learning is limited. More complex policies, which, e.g., can directly use sensory feedback as inputs, could be acquired by incorporating recent successes from deep reinforcement learning (Mnih et al., 2015, 2016).

Furthermore, PbRL is still lacking a unified evaluation framework. The different publications use a wide variety of testing domains, usually derived from classic RL problems. However, the specific setups depend on the publication and usually modified the basic configuration by adding noise or removing the terminal conditions. Moreover, many algorithms disregard specific subproblems like policy generalization or utility exploration. Hence, a satisfactory comparison of algorithms is currently not possible.

6. Conclusion

Preference-based reinforcement learning (PbRL) is a suitable tool for learning from qualitative, non-numeric rewards. On the one hand, it can provide solutions in domains where numeric feedback is not readily available, and, on the other hand, it may reduce the demands and prior knowledge that is needed for applying RL to real-world tasks where a numeric feedback signal has to be shaped. This survey provided a first overview over the design choices that have to be considered for PbRL and presented current work in a coherent framework, shedding light on the advantages and limitations of various approaches.

The most substantial problem seems to be the complexity of the learning problem. The presented approaches require a high amount of preferences or well defined feature spaces, either on a policy or trajectory level. The definition of suitable, low-dimensional feature spaces requires expert knowledge, if possible at all. In many domains, high-dimensional feature spaces are required to approximate optimal policies to an acceptable degree. Furthermore, preferences are costly to obtain in an incremental setting with human evaluations. Even in cases where it is possible to obtain preferences from non-expert users, human labor is usually quite expensive. Hence, it is very important to use the available information as efficiently as possible. Utility-based approaches allow to generalize the obtained preference feedback and are an important step towards practical algorithms. Especially reward-based utilities seem to be appropriate for domains without well defined policy spaces, because they allow the application of efficient value-based reinforcement learning methods. Hence, it is possible to use the substantial progress that has already been achieved in this field. However, it is also important to consider the novel problems introduced by using preference-based feedback. Foremost, efficient methods for exploring the preference function space are required.

Recently, interesting real-world applications could be solved by preference-based methods. Yet, several open questions remain. Foremost, sample efficiency and scalibility issues still prevent applications to more complex tasks. DNN-based approaches can tackle the scalibility issues to some degree, but improved solutions to the preference-space exploration problem are required to limited the amount of required preferences. Furthermore, PbRL is still limited by several assumptions like the total order assumption, the uniform preference weighting and the single starting state. Without these assumptions, it would be possible to apply PbRL in multi-objective domains, for risk-adverse problems and in settings that can not be simulated. Methods that can use multiple feedback forms, like preferences, demonstrations or rewards, are also of interested as they are possible able to reduce the cognitive demand even further.

We believe that this line of research could provide the foundation to make reinforcement learning applicable to the non-scientific community because despite many recent successes in reinforcement learning, we believe that the required expertise in machine learning is still a key limitation when it comes to applying RL to real-world tasks.

Acknowledgments

This work was supported by the German Research Foundation (DFG) as part of the Priority Programme 1527 "Autonomous Learning", the research project FU 580/10, and under the EU H2020 project 'RoMaNS', Ref. 645582. We would like to thank Eyke Hüllermeier, Robert Busa-Fekete and Stefan Riezler for interesting and stimulating discussions that greatly influenced this paper.

References

P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the 21st International Conference on Machine learning (ICML-04)*, volume 69. 2004.

A. Agarwal, O. Dekel, and L. Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Proceedings of the 23rd Conference on Learning Theory (COLT-10)*, pages 28–40, 2010.

R. Akrour, M. Schoenauer, and M. Sebag. Preference-based policy learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD-11)*, volume 6911 of *LNCS*, pages 12–27. 2011.

R. Akrour, M. Schoenauer, and M. Sebag. APRIL: Active preference learning-based reinforcement learning. In *Proceedings of the European Conference on Machine Learning*

and Knowledge Discovery in Databases (ECML-PKDD-12), volume 7524 of *LNCS*, pages 116–131. 2012.

R. Akrour, M. Schoenauer, and M. Sebag. Interactive Robot Education. In *Proceedings of the ECML/PKDD Workshop on Reinforcement Learning with Generalized Feedback: Beyond Numeric Rewards*, 2013.

R. Akrour, M. Schoenauer, M. Sebag, and J.-C. Souplet. Programming by Feedback. In *Proceedings of the 31nd International Conference on Machine Learning (ICML-14)*, volume 32, pages 1503–1511. 2014.

D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

K. J. Åström and K. Furuta. Swinging up a pendulum by energy control. *Automatica*, 36 (2):287–295, 2000.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-arm bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS-95)*, pages 322–331. 1995.

A. Auger. Convergence Results for the (1,λ)-SA-ES using the Theory of φ-irreducible Markov Chains. *Theoretical Computer Science*, 334(1-3):35–69, 2005.

A. G. Barto. Intrinsic motivation and reinforcement learning. In G. Baldassarre and M. Mirolli, editors, *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 17–47. Springer Berlin Heidelberg, 2013.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The Arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47 (1):253–279, 2013.

D. Berenson, S. Srinivasa, D. Ferguson, and J. Kuffner, Jr. Manipulation planning on constraint manifolds. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-09)*, pages 625–632, 2009.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

J. Boyan. Least-Squares Temporal Difference Learning. In *Proceedings of the 16th International Conference on Machine Learning (ICML-99)*, pages 49–56. 1999.

R. Busa-Fekete and E. Hüllermeier. A survey of preference-based online learning with bandit algorithms. In *Proceedings of the 25th Algorithmic Learning Theory International Conference (ALT-14)*, volume 8776 of *LNCS*, pages 18–39. 2014.

R. Busa-Fekete, B. Szörényi, P. Weng, W. Cheng, and E. Hüllermeier. Preference-based evolutionary direct policy search. In *Proceedings of the ICRA Workshop on Autonomous Learning*, 2013.

R. Busa-Fekete, B. Szörényi, P. Weng, W. Cheng, and E. Hüllermeier. Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm. *Machine Learning*, 97(3):327–351, 2014.

P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. *CoRR*, abs/1706.03741, 2017.

W. Chu and Z. Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*, pages 137–144. ACM, 2005.

S. P. Coraluppi and S. I. Marcus. Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes. *Automatica*, 35(2):301 – 309, 1999.

C. Daniel, M. Viering, J. Metz, O. Kroemer, and J. Peters. Active reward learning. In *Proceedings of Robotics: Science & Systems X (R:SS-14)*, 2014.

M. P. Deisenroth and C. E. Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 465–472. 2011.

C. Dimitrakakis and M. G. Lagoudakis. Rollout sampling approximate policy iteration. *Machine Learning*, 72(3):157–171, 2008.

J. Duchi, L. W. Mackey, and M. I. Jordan. On the consistency of ranking algorithms. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 327–334, 2010.

R. O. Duda and P. E. Hart. *Pattern Recognition and Scene Analysis*. John Wiley and Sons, 1973.

A. Faust, H.-T. Chiang, N. Rackley, and L. Tapia. Avoiding moving obstacles with stochastic hybrid dynamics using PEARL: Preference appraisal reinforcement learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-16)*, pages 484–490. 2016.

M. Friedman and L. J. Savage. The expected-utility hypothesis and the measurability of utility. *Journal of Political Economy*, 60, 1952.

J. Fürnkranz and E. Hüllermeier, editors. *Preference Learning*. Springer-Verlag, 2010.

J. Fürnkranz, E. Hüllermeier, W. Cheng, and S.-H. Park. Preference-based reinforcement learning: A formal framework and a policy iteration algorithm. *Machine Learning*, 89 (1-2):123–156, 2012. Special Issue of Selected Papers from ECML/PKDD-11.

P. Geibel. Reinforcement learning with bounded risk. In *In Proceedings of the 18th International Conference on Machine Learning (ICML-01)*, pages 162–169. 2001.

S. Griffith, K. Subramanian, J. Scholz, C. Isbell, and A. L. Thomaz. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems 26 (NIPS-13)*, pages 2625–2633, 2013.

A. Gritsenko and D. Berenson. Learning cost functions for motion planning from human preferences. In *Proceedings of the IROS 2014 Workshop on Machine Learning in Planning and Control of Robot Motion*, volume 1, pages 48–6, 2014.

S. Grünewälder, G. Lever, L. Baldassarre, M. Pontil, and A. Gretton. Modelling transition dynamics in MDPs with RKHS embeddings. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 535–542, 2012.

N. Hansen and S. Kern. Evaluating the CMA evolution strategy on multimodal test functions. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN-04)*, volume 3242 of *LNCS*, pages 282–291. 2004.

V. Heidrich-Meisner and C. Igel. Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In *Proceedings of the 26th International Conference on Machine Learning (ICML-09)*, pages 401–408. 2009.

R. Herbrich, T. Graepel, and K. Obermayer. Support vector learning for ordinal regression. In *Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN-99)*, volume 1, pages 97–102, 1999.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16–17):1897–1916, 2008.

A. Jain, B. Wojcik, T. Joachims, and A. Saxena. Learning trajectory preferences for manipulators via iterative improvement. In *Advances in Neural Information Processing Systems 26 (NIPS-13)*, pages 575–583, 2013.

A. Jain, S. Sharma, T. Joachims, and A. Saxena. Learning preferences for manipulation tasks from online coactive feedback. *International Journal of Robotics Research*, 34(10): 1296–1313, 2015.

T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-02)*, pages 133–142. 2002.

T. Kamishima, H. Kazawa, and S. Akaho. A survey and empirical comparison of object ranking methods. In Fürnkranz and Hüllermeier (2010), pages 181–201.

W. B. Knox and P. Stone. Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the 5th International Conference on Knowledge Capture (K-CAP-09)*, pages 9–16, 2009.

W. B. Knox and P. Stone. Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-10)*, pages 5–12. 2010.

W. B. Knox and P. Stone. Reinforcement learning from simultaneous human and MDP reward. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-12)*, pages 475–482. 2012.

J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

D. Kreps. *Notes on the theory of choice*. Underground classics in economics. Westview Press, 1988.

A. Kupcsik, M. P. Deisenroth, J. Peters, A. P. Loh, P. Vadakkepat, and G. Neumann. Model-based contextual policy search for data-efficient generalization of robot skills. *Artificial Intelligence*, 247:415–439, 2014.

A. Kupcsik, D. Hsu, and W. S. Lee. Learning dynamic robot-to-human object handover from human feedback. In *Proceedings of the 17th International Symposium on Robotics Research (ISRR-15)*, 2015.

M. G. Lagoudakis and R. Parr. Least-Squares Policy Iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.

S. M. LaValle and J. Kuffner, Jr. Randomized kinodynamic planning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-99)*, pages 473–479, 1999.

S. Levine and V. Koltun. Guided policy search. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1–9, 2013.

M. Likhachev, G. Gordon, and S. Thrun. ARA*: Anytime A* search with provable bounds on sub-optimality. In *Advances in Neural Information Processing Systems 16 (NIPS-03)*, pages 767–774. 2003.

C. Liu, X. Xu, and D. Hu. Multiobjective reinforcement learning: A comprehensive overview. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3):385–398, 2015.

C. X. Lu, Z. Y. Sun, Z. Z. Shi, and B. X. Cao. Using emotions as intrinsic motivation to accelerate classic reinforcement learning. In *Proceedings of the International Conference on Information System and Artificial Intelligence (ISAI-16)*, pages 332–337, 2016.

R. Maclin, J. W. Shavlik, L. Torrey, T. Walker, and E. W. Wild. Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *Proceedings of the 20th National Conference on Artificial Intelligence (AAAI-05)*, pages 819–824, 2005.

C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

B. Mayeur, R. Akrour, and M. Sebag. Direct value learning: a rank-invariant approach to reinforcement learning. In *Proceedings of the NIPS Workshop on Autonomously Learning Robots*, 2014.

D. Meunier, Y. Deguchi, R. Akrour, E. Suzuki, M. Schoenauer, and M. Sebag. Direct value learning: A preference-based approach to reinforcement learning. In *Proceedings of the ECAI Workshop on Preference Learning: Problems and Applications in AI*, pages 42–47, 2012.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning (ICML-16)*, pages 1928–1937, 2016.

I. Murray, R. P. Adams, and D. J. C. MacKay. Elliptical slice sampling. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS-10)*, volume 9, pages 541–548. 2010.

A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning (ICML-00)*, pages 663–670, 2000.

A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning (ICML-99)*, pages 278–287, 1999.

D. Nguyen-Tuong and J. Peters. Model learning for robot control: a survey. *Cognitive Processing*, 12(4):319–340, 2011.

J. Peters, K. Muelling, and Y. Altun. Relative entropy policy search. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI-10)*, pages 1607–1612. 2010.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

J. Randløv and P. Alstrøm. Learning to drive a bicycle using reinforcement learning and shaping. In *Proceedings of the 15th International Conference on Machine Learning (ICML-98)*, pages 463–471, 1998.

C. A. Rothkopf and C. Dimitrakakis. Preference elicitation and inverse reinforcement learning. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD-11), Part III*, volume 6913 of *Lecture Notes in Computer Science*, pages 34–48, 2011.

T. P. Runarsson and S. M. Lucas. Preference learning for move prediction and evaluation function approximation in othello. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(3):300–313, 2014.

T. P. Runarsson and S. M. Lucas. Imitating play from game trajectories: Temporal difference learning versus preference learning. In *IEEE Conference on Computational Intelligence and Games (CIG-12)*, pages 79–82. IEEE, 2012.

J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1889–1897, 2015.

O. Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *Journal of Machine Learning Research*, 18:1–11, 2017.

S. P. Singh, A. G. Barto, and N. Chentanez. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems 17 (NIPS-04)*, volume 17, pages 1281–1288, 2004.

S. P. Singh, R. L. Lewis, and A. G. Barto. Where do rewards come from? In *Proceedings of the 31th Conference of the Cognitive Science Society (CogSci-09)*, pages 2601–2606, 2009.

A. Sokolov, J. Kreutzer, C. Lo, and S. Riezler. Learning structured predictors from bandit feedback for interactive NLP. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-16)*, 2016.

M. W. Spong. Swing up control of the acrobot. In *IEEE International Conference on Robotics and Automation (ICRA-94)*, pages 2356–2361. IEEE, 1994.

K. Strabala, M. K. Lee, A. Dragan, J. Forlizzi, S. Srinivasa, M. Cakmak, and V. Micelli. Towards seamless human-robot handovers. *Journal of Human-Robot Interaction*, 2013.

H. Sugiyama, T. Meguro, and Y. Minami. Preference-learning based inverse reinforcement learning for dialog control. In *Proceedings of the 13th Annual Conference of the International Speech Communication Association (INTERSPEECH-12)*, pages 222–225, 2012.

R. S. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

G. Tesauro. Temporal Difference Learning and TD-gammon. *Communications ACM*, 38(3): 58–68, 1995.

A. L. Thomaz and C. Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the 21th National Conference on Artificial Intelligence (AAAI-06)*, pages 1000–1006. 2006.

E. Todorov, T. Erez, and Y. Tassa. MuJoCo: A physics engine for model-based control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-12)*, pages 5026–5033, 2012.

L. Torrey and M. E. Taylor. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-13)*, pages 1053–1060. 2013.

S. Vembu and T. Gärtner. Label ranking algorithms: A survey. In Fürnkranz and Hüllermeier (2010), pages 45–64.

J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

H. O. Wang, K. Tanaka, and M. F. Griffin. An approach to fuzzy control of nonlinear systems: stability and design issues. *IEEE Transactions on Fuzzy Systems*, 4(1):14–23, 1996.

Y. Wang and I. H. Witten. Induction of model trees for predicting continuous classes. In *Poster papers of the 9th European Conference on Machine Learning (ECML-97)*. 1997.

Y.-C. Wang and J. M. Usher. Application of reinforcement learning for agent-based production scheduling. *Engineering Applications of Artificial Intelligence*, 18(1):73–82, 2005.

P. Weng. Markov decision processes with ordinal rewards: Reference point-based preferences. In *Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS-11)*, 2011.

A. Wilson, A. Fern, and P. Tadepalli. A bayesian approach for policy learning from trajectory preference queries. In *Advances in Neural Information Processing Systems 25 (NIPS-12)*, pages 1142–1150. 2012.

C. Wirth and J. Fürnkranz. First steps towards learning from game annotations. In *Proceedings of the ECAI Workshop on Preference Learning: Problems and Applications in AI*, pages 53–58, 2012.

C. Wirth and J. Fürnkranz. A policy iteration algorithm for learning from preference-based feedback. In *Advances in Intelligent Data Analysis XII: 12th International Symposium (IDA-13)*, volume 8207 of *LNCS*, pages 427–437. 2013a.

C. Wirth and J. Fürnkranz. EPMC: Every visit preference Monte Carlo for reinforcement learning. In *Proceedings of the 5th Asian Conference on Machine Learning, (ACML-13)*, volume 29 of *JMLR Proceedings*, pages 483–497. 2013b.

C. Wirth and J. Fürnkranz. Preference-based reinforcement learning: A preliminary survey. In *Proceedings of the ECML/PKDD-13 Workshop on Reinforcement Learning from Generalized Feedback: Beyond Numeric Rewards*, 2013c.

C. Wirth and J. Fürnkranz. On learning from game annotations. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(3):304–316, 2015.

C. Wirth, J. Fürnkranz, and G. Neumann. Model-free preference-based reinforcement learning. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 2222–2228, 2016.

Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The k-armed dueling bandits problem. *Journal of Computer System Sciences*, 78(5):1538–1556, 2012.

Y. Zhao, M. Kosorok, and D. Zeng. Reinforcement learning design for cancer clinical trials. *Statistics in Medicine*, 28(26):3294–3315, 2009.

Y. Zhao, D. Zeng, M. A. Socinski, and M. Kosorok. Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer. *Biometrics*, 67(4):1422–1433, 2011.

S. Zhifei and E. Meng Joo. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics*, 5(3):293–311, 2012.

B. D. Ziebart, A. Maas, J. A. Bagnell, and A. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 1433–1438. 2008.

M. Zucker, J. A. Bagnell, C. Atkeson, and J. Kuffner, Jr. An optimization approach to rough terrain locomotion. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA-10)*, pages 3589–3595. 2010.

Journal of Machine Learning Research 18 (2017) 1-50 Submitted 5/16; Revised 10/17; Published 11/17

Generalized SURE for optimal shrinkage of singular values in low-rank matrix denoising

Jérémie Bigot JEREMIE.BIGOT@U-BORDEAUX.FR
Institut de Mathématiques de Bordeaux
Université de Bordeaux
351, cours de la Libération, 33405 Talence, France

Charles Deledalle CHARLES-ALBAN.DELEDALLE@MATH.U-BORDEAUX.FR
Institut de Mathématiques de Bordeaux
Université de Bordeaux
351, cours de la Libération, 33405 Talence, France

Delphine Féral DELPHINE.FERAL@U-BORDEAUX.FR
Institut de Mathématiques de Bordeaux
Université de Bordeaux
351, cours de la Libération, 33405 Talence, France

Editor: Sara van de Geer

Abstract

We consider the problem of estimating a low-rank signal matrix from noisy measurements under the assumption that the distribution of the data matrix belongs to an exponential family. In this setting, we derive generalized Stein's unbiased risk estimation (SURE) formulas that hold for any spectral estimators which shrink or threshold the singular values of the data matrix. This leads to new data-driven spectral estimators, whose optimality is discussed using tools from random matrix theory and through numerical experiments. Under the spiked population model and in the asymptotic setting where the dimensions of the data matrix are let going to infinity, some theoretical properties of our approach are compared to recent results on asymptotically optimal shrinking rules for Gaussian noise. It also leads to new procedures for singular values shrinkage in finite-dimensional matrix denoising for Gamma-distributed and Poisson-distributed measurements.

Keywords: Matrix denoising, singular value decomposition, low-rank model, Gaussian spiked population model, spectral estimator, Stein's unbiased risk estimate, random matrix theory, exponential family, optimal shrinkage rule, degrees of freedom.

1. Introduction

1.1 Low rank matrix denoising in an exponential family

In various applications, it is of interest to estimate a signal matrix from noisy data. Typical examples include the case of data that are produced in a matrix form, while others are concerned with observations from multiple samples that can be organized in a matrix form. In such setting, a typical inference problem involves the estimation of an unknown (non-random) signal matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ from a noisy data matrix $\boldsymbol{Y}$ satisfying the model:

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}, \tag{1}$$

©2017 Jérémie Bigot, Charles Deledalle and Delphine Féral.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-266.html.

where $\boldsymbol{W}$ is an $n \times m$ noise matrix with real entries $\boldsymbol{W}_{ij}$ assumed to be independent random variables with $\mathbb{E}[\boldsymbol{W}_{ij}] = 0$ and $\mathrm{Var}(\boldsymbol{W}_{ij}) = \tau_{ij}^2$ for $1 \le i \le n$ and $1 \le j \le m$. In this paper, we focus on the situation where the signal matrix $\boldsymbol{X}$ is assumed to have a low rank structure, and we consider the general setting where the distribution of $\boldsymbol{Y}$ belongs to a continuous exponential family parametrized by the entries of the matrix $\boldsymbol{X} = \mathbb{E}[\boldsymbol{Y}]$. For discrete observations (count data), we also consider the specific case of Poisson noise.

The low rank assumption on $\boldsymbol{X}$ is often met in practice when there exists a significant correlation between the columns of $\boldsymbol{X}$. This can be the case when the columns of $\boldsymbol{X}$ represent 2D images at different wavelength of hyperspectral data, since images at nearby wavelengths are strongly correlated (Candès et al., 2013). Further applications, where low-rank modeling of $\boldsymbol{X}$ is relevant, can be found in genomics (Wall et al., 2001; Alter et al., 2000), NMR spectroscopy (Nguyen et al., 2011), collaborative filtering (Candès and Recht, 2009) or medical imaging (Bydder and Du, 2006; Lam et al., 2012), among many others.

Low-rank matrix estimation is classically done in the setting where the additive noise is Gaussian with homoscedastic variance. The more general case of observations sampled from an exponential family is less developed, but there exists an increasing research interest in the study of low rank matrix recovery beyond the Gaussian case. Examples of low-rank matrix recovering from Poisson distributed observations can be found in applications with count data such as network traffic analysis (Bazerque et al., 2013) or call center data (Haipeng and Jianhua, 2005). A theory for low-rank matrix recovery and completion in the case of Poisson observations has also been recently proposed in Cao and Xie (2016). Matrix completion under a low rank assumption with additive errors having a sub-exponential distribution and belonging to an exponential family has also been considered in Lafond (2015). The recent work in Udell et al. (2016) proposes a novel framework to approximate, by a low rank matrix, a tabular data set made of numerical, Boolean, categorical or ordinal observations.

1.2 The class of spectral estimators

A standard approach to estimate a low rank matrix relies on the singular value decomposition (SVD) of the data matrix

$$\boldsymbol{Y} = \sum_{k=1}^{\min(n,m)} \tilde{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t, \tag{2}$$

where $\tilde{\sigma}_1 \ge \tilde{\sigma}_2 \ge \ldots \ge \tilde{\sigma}_{\min(n,m)} \ge 0$ denote its singular values, and $\tilde{\boldsymbol{u}}_k, \tilde{\boldsymbol{v}}_k$ denote the associated singular vectors. In this paper, we propose to consider the class of spectral estimators $\hat{\boldsymbol{X}}^f = f(\boldsymbol{Y})$, where $f : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ is a (possibly data-dependent) mapping that acts on the singular values of the data matrix $\boldsymbol{Y}$ while leaving its singular vectors unchanged. More precisely, these estimators take the form

$$\hat{\boldsymbol{X}}^f = f(\boldsymbol{Y}) = \sum_{k=1}^{\min(n,m)} f_k(\boldsymbol{Y}) \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t, \tag{3}$$

where, for each $1 \le k \le \min(n,m)$, $f_k(\boldsymbol{Y})$ are real positive values that may depend only on $\tilde{\sigma}_k$ (hence we write $f_k(\tilde{\sigma}_k)$) or on the whole matrix $\boldsymbol{Y}$.

1.3 Investigated spectral estimators

Typical examples of spectral estimators include the classical principal component analysis (PCA) applied to matrix denoising defined, for some $1 \leq r \leq \min(n,m)$, as

$$\hat{\boldsymbol{X}}^r = \sum_{k=1}^{r} \hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \quad \text{with} \quad \hat{\sigma}_k = f_k(\tilde{\sigma}_k) = \tilde{\sigma}_k \tag{4}$$

for all $1 \leq k \leq r$ and where it is implicitely understood that $f_k(\tilde{\sigma}_k) = 0$ for $k \geq r+1$. Another typical spectral estimator in matrix denoising with Gaussian measurements is the soft-thresholding (Candès et al., 2013) which corresponds to the choice

$$\hat{\boldsymbol{X}}_{\text{soft}} = \sum_{k=1}^{\min(m,n)} \hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \quad \text{with} \quad \hat{\sigma}_k = f_k(\boldsymbol{Y}) = \left(1 - \frac{\lambda(\boldsymbol{Y})}{\tilde{\sigma}_k}\right)_+ \tilde{\sigma}_k, \tag{5}$$

for all $1 \leq k \leq \min(n,m)$ and where $\lambda(\boldsymbol{Y}) > 0$ is a possibly data-dependent threshold parameter, and $(x)_+ = \max(x,0)$ for any $x \in \mathbb{R}$. Finaly, we will consider a more general class of shrinkage estimators, encompassing the PCA and the soft-thresholding, that perform

$$\hat{\boldsymbol{X}}_w = \sum_{k=1}^{\min(m,n)} \hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \quad \text{with} \quad \hat{\sigma}_k = f_k(\boldsymbol{Y}) = w_k(\boldsymbol{Y}) \tilde{\sigma}_k, \tag{6}$$

where $w_k(\boldsymbol{Y}) \in [0,1]$ is a possibly data-dependent shrinking weight.

1.4 Main contributions

Under the assumption that the distribution of $\boldsymbol{Y}$ belongs to an exponential family, the goal of this paper is to derive data-driven choices for the weights $w_k(\boldsymbol{Y})$ in (3). We construct estimators via a two-step procedure. First, an active set of non-zero singular values is defined. Then, in a second step, weights $w_k(\boldsymbol{Y})$ associated with non-zero singular values are optimized, and shown to reach desired asymptotical properties in the Gaussian spiked population model. The main contributions of the paper are then the following ones.

1.4.1 An AIC inspired criterion for rank and singular values locations estimation

When no *a priori* is available on the rank of the signal matrix $\boldsymbol{X}$, optimizing for the weights w_k, for all $1 \leq k \leq \min(m,n)$, can lead to estimators with large variance (*i.e.*, overfitting the noise). We propose an automatic rule to prelocalize the subset of non-zero singular values. An active set $s^\star \subseteq \mathcal{I} = \{1,2,\ldots,\min(n,m)\}$ of singular values is defined as the minimizer of a penalized log-likelihood criterion that is inspired by the Akaike information criterion (AIC)

$$s^* \in \arg\min_{s \subseteq \mathcal{I}} \; -2 \log q(\boldsymbol{Y}; \tilde{\boldsymbol{X}}^s) + 2|s| p_{n,m} \quad \text{with} \quad p_{n,m} = \frac{1}{2}\left(\sqrt{m} + \sqrt{n}\right)^2, \tag{7}$$

where $\tilde{\boldsymbol{X}}^s = \sum_{k \in s} \tilde{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t$, $|s|$ is the cardinal of s, and $q(\boldsymbol{Y}; \tilde{\boldsymbol{X}}^s)$ is the likelihood of the data in a given exponential family with estimated parameter $\tilde{\boldsymbol{X}}^s$. For the case of Gaussian measurements with homoscedastic variance τ^2, one has that $q(\boldsymbol{Y}; \tilde{\boldsymbol{X}}^s) = \|\boldsymbol{Y} - \tilde{\boldsymbol{X}}^s\|_F^2 / 2\tau^2$, where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, and we show that the active set of singular values boils down to

$$s^\star = \{k \; ; \; \tilde{\sigma}_k > c_+^{n,m}\}, \tag{8}$$

where $c_+^{n,m} = \tau(\sqrt{m} + \sqrt{n})$. For Gamma and Poisson measurements, we resort to a greedy optimization procedure described in Section 4.

Once the active set has been determined, the subsequent shrinkage estimator is obtained by optimizing only for the weights within this subset while setting the other ones to zero.

1.4.2 Novel data-driven shrinkage rules minimizing SURE-like formulas

We use the principle of Stein's unbiased risk estimation (SURE) (Stein, 1981) to derive unbiased estimation formulas for the mean squared error (MSE) risk and mean Kullback-Leibler (MKL) risks of spectral estimators. Minimizing such SURE-like formulas over an appropriate class of spectral estimators is shown to lead to novel data-driven shrinkage rules of the singular values of the matrix $\boldsymbol{Y}$. In particular, our approach leads to novel spectral estimators in situations where the variances τ_{ij}^2 of the entries $\boldsymbol{W}_{ij}$ of the noise matrix are not necessarily equal, and may depend on the signal matrix $\boldsymbol{X}$.

As an illustrative example, let us consider spectral estimators of the form

$$\hat{\boldsymbol{X}}_w^1 = f(\boldsymbol{Y}) = w_1(\boldsymbol{Y}) \tilde{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t, \tag{9}$$

which only act on the first singular value $\tilde{\sigma}_1$ of the data while setting all the other ones to zero. In this paper, examples of data-driven choices for the weight $w_1(\boldsymbol{Y})$ are the following ones:

- for Gaussian measurements with $n \leq m$ and known homoscedastic variance τ^2

$$w_1(\boldsymbol{Y}) = \left(1 - \frac{\tau^2}{\tilde{\sigma}_1^2}\left(1 + |m-n| + 2\sum_{\ell=2}^{n} \frac{\tilde{\sigma}_1^2}{\tilde{\sigma}_1^2 - \tilde{\sigma}_\ell^2}\right)\right)_+ \mathbb{1}_{\{\tilde{\sigma}_1 > c_+^{n,m}\}}, \tag{10}$$

- for Gamma measurements with $\tau_{ij}^2 = \boldsymbol{X}_{ij}^2 / L$ and $L > 2$ (see Section 2.1 for a precise definition),

$$w_1(\boldsymbol{Y}) = \min\left[1, \left(\frac{L-1}{Lmn}\sum_{i=1}^{n}\sum_{j=1}^{m} \frac{\hat{\boldsymbol{X}}_{ij}^1}{\boldsymbol{Y}_{ij}} + \frac{1}{Lmn}\left(1 + |m-n| + 2\sum_{\ell=2}^{\min(n,m)} \frac{\tilde{\sigma}_1^2}{\tilde{\sigma}_1^2 - \tilde{\sigma}_\ell^2}\right)\right)^{-1}\right] \mathbb{1}_{\{1 \in s^*\}}, \tag{11}$$

- for Poisson measurements with $\tau_{ij}^2 = \boldsymbol{X}_{ij}$ (see Section 2.1 for a precise definition)

$$w_1(\boldsymbol{Y}) = \min\left[1, \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} \boldsymbol{Y}_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} \hat{\boldsymbol{X}}_{ij}^1}\right] \mathbb{1}_{\{1 \in s^*\}}. \tag{12}$$

Beyond the case of rank one, closed-form solutions for the weights cannot be obtained, except for the case of Gaussian measurements with homoscedastic variance τ^2. In this latter case, the rule for $w_1(\boldsymbol{Y})$ in (10) generalizes to other eigenvalues $w_k(\boldsymbol{Y})$ as

$$w_k(\boldsymbol{Y}) = \left(1 - \frac{\tau^2}{\tilde{\sigma}_k^2}\left(1 + |m-n| + 2\sum_{\ell=1;\ell\neq k}^{\min(n,m)} \frac{\tilde{\sigma}_k^2}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2}\right)\right)_+ \mathbb{1}_{\{\tilde{\sigma}_k > c_+^{n,m}\}}. \tag{13}$$

For Gamma or Poisson distributed measurements, we propose fast algorithms to get numerical approximations of the weights $w_k(\boldsymbol{Y})$ (see Section 5.2 for more details).

1.4.3 Asymptotic properties in the Gaussian spiked population model

Another contribution of the paper is to discuss the optimality of the shrinking weights (13) for Gaussian noise in the asymptotic setting where the dimensions of the matrix $\boldsymbol{Y}$ are let going to infinity. These theoretical results are obtained for the so-called *spiked population model* that has been introduced in the literature on random matrix theory and high-dimensional covariance matrix estimation (see e.g. Baik and Silverstein (2006); Benaych-Georges and Nadakuditi (2012); Dozier and Silverstein (2007); Shabalin and Nobel (2013)). All the theoretical and asymptotic results of the paper (other than derivation of proposed estimators) assume this model.

Definition 1 *The Gaussian spiked population model corresponds to the following setting:*

- *the $\boldsymbol{W}_{ij}$ in (1) are iid Gaussian random variables with zero mean and variance $\tau^2 = 1/m$,*
- *the $\boldsymbol{X}_{ij}$'s in (1) are the entries of an unknown $n \times m$ matrix $\boldsymbol{X}$ that has a low rank structure, meaning that it admits the SVD $\boldsymbol{X} = \sum_{k=1}^{r^*} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^t$, where $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ are the left and right singular vectors associated to the singular value $\sigma_k > 0$, for each $1 \leq k \leq r^*$, with $\sigma_1 > \sigma_2 > \ldots > \sigma_{r^*}$,*
- *the rank r^* of the matrix $\boldsymbol{X}$ is assumed to be fixed,*
- *the dimensions of the data matrix $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ are let going to infinity in the asymptotic framework where the sequence $m = m_n \geq n$ is such that $\lim_{n\to+\infty} \frac{n}{m} = c$ with $0 < c \leq 1$.*

In the Gaussian spiked population model, the asymptotic locations of the empirical singular values $\tilde{\sigma}_1 \geq \ldots \geq \tilde{\sigma}_{\min(n,m)}$ are well understood in the random matrix theory (further details are given in Section 3.1). Note that the setting where the rank r^* is not held fixed but allowed to grow with $\min(n,m)$ is very different (see e.g. Ledoit and Wolf (2012) and references therein).

Under the Gaussian spiked population model, our contributions are then as follows:

- we prove the convergence of the SURE formula when the dimensions of $\boldsymbol{Y}$ tend to infinity,
- it is shown that minimizing the asymptotic value of SURE leads to the same estimator as the limiting value of the estimator obtained by minimizing the SURE,

- this model allows to show that the novel data-driven spectral estimators derived in this paper are asymptotically connected to existing optimal shrinkage rules (Shabalin and Nobel, 2013; Gavish and Donoho, 2017; Nadakuditi, 2014) for low-rank matrix denoising,
- in this setting, we are also able to connect the choice of the penalty function $2|s|p_{n,m}$ in (7) with Stein's notion of degrees of freedom (see e.g. Bradley (2004)) for spectral estimators.

1.4.4 Numerical experiments and publicly available source code

As the theoretical properties of our estimators are studied in an asymptotic setting, we report the results of various numerical experiments to analyze the performances of the proposed estimators for finite-dimensional matrices. These experiments allow the comparison with existing shrinkage rules for Gaussian-distributed measurements and they are also used to shed some lights on the finite sample properties of the method for Gamma-distributed or Poisson-distributed measurements. We also exhibit the settings where the signal matrix $\boldsymbol{X}$ is either easy or more difficult to recover. From these experiments, the main findings are the following ones:

- the use of an appropriate active set s of singular values is an essential step for the quality of shrinkage estimators whose weights are data-driven by SURE-like estimators; taking $s = \{1, \ldots, \min(n,m)\}$ leads to poor results while the choice of $s = s^*$ minimizing the AIC criterion (7) appears to yield the best performances,
- for Gaussian noise, the performances of our approach are similar to those obtained by the asymptotically optimal spectral estimator proposed in Gavish and Donoho (2017) when the true rank r^* of the signal matrix $\boldsymbol{X}$ is sufficiently small, but for large to moderate values of the signal-to-noise ratio our approach may perform better than existing methods in the literature,
- for Gamma or Poisson distributed measurements, the spectral estimators proposed in this paper give better results than estimators based on PCA (restricted to the active set s^*) or soft-thresholding of singular values.

Beyond the case of Gaussian noise, the implementation of the estimators is not straightforward, and we thus provide publicly available source code at

`https://www.math.u-bordeaux.fr/~cdeledal/gsure_low_rank`

to reproduce the figures and the numerical experiments of this paper.

1.5 Related results in the literature

Early work on singular value thresholding began with the work in Eckart and Young (1936) on the best approximation of fixed rank to the data matrix $\boldsymbol{Y}$. Spectral estimators with different amounts of shrinkage for each singular value of the data matrix have then been proposed in Efron and Morris (1972, 1976). In the case of Gaussian measurements with

homoscedastic variance, the problem of estimating $\boldsymbol{X}$ under a low-rank assumption has recently received a lot of attention in the literature on high-dimensional statistics (see e.g. Candès et al. (2013); Donoho and Gavish (2014); Josse and Sardy (2015); Shabalin and Nobel (2013)). Recent works (Gavish and Donoho, 2017; Nadakuditi, 2014) also consider the more general setting where the distribution of the additive noise matrix $\boldsymbol{W}$ is orthogonally invariant, and such that its entries are iid random variables with zero mean and finite fourth moment. In all these papers, the authors have focused on spectral estimators which shrink or threshold the singular values of $\boldsymbol{Y}$, while its singular vectors are left unchanged. In this setting, the main issue is to derive optimal shrinkage rules that depends on the class of spectral estimators that is considered, on the loss function used to measure the risk of an estimator of $\boldsymbol{X}$, and on appropriate assumptions for the distribution of the additive noise matrix $\boldsymbol{W}$.

1.6 Organization of the paper

Section 2 is devoted to the analysis of a data matrix whose entries are distributed according to a continuous exponential family. SURE-like formula are first given for the mean squared error risk, and then for the Kullback-Leibler risk. As an example of discrete exponential family, we also derive such risk estimators for Poisson distributed measurements. The computation of data-driven shrinkage rules is then discussed for Gaussian, Gamma and Poisson noises. In Section 3, we restrict our attention to the Gaussian spiked population model in order to derive asymptotic properties of our approach. We study the asymptotic behavior of the SURE formula proposed in Candès et al. (2013); Donoho and Gavish (2014) for spectral estimators using tools from RMT. This result allows to make a connection between data-driven spectral estimators minimizing the SURE for Gaussian noise, and the asymptotically optimal shrinkage rules proposed in Shabalin and Nobel (2013); Nadakuditi (2014) and Gavish and Donoho (2017). In Section 4, we study the penalized log-likelihood criterion (7) used to select an active set of singular values. Its connection to the degrees of freedom of spectral estimators and rank estimation in matrix denoising is discussed. Various numerical experiments are finally proposed in Section 5 to illustrate the usefulness of the approach developed in this paper for low-rank denoising and to compare its performances with existing methods. The proofs of the main results of the paper are gathered in a technical Appendix A, and numerical implementation details are described in Appendix B.

2. SURE-like formulas in exponential families

For an introduction to exponential families, we refer to Brown (1986). The idea of unbiased risk estimation in exponential families dates back to Hudson (1978). More recently, generalized SURE formulas have been proposed for the estimation of the MSE risk, for denoising under various continuous and discrete distributions in Raphan and Simoncelli (2007), and for inverse problems whithin the continuous exponential families in Eldar (2009). In Deledalle (2017), SURE-like formula are derived for the estimation of the Kullback-Leibler risk that applies to both continuous and discrete exponential families. In what follows, we borrow some ideas and results from these works. We first treat the case of continuous exponential families, and then we focus on Poisson data in the discrete case.

2.1 Data sampled from a continuous exponential family

We recall that $\boldsymbol{Y}$ is an $n \times m$ matrix with independent and real entries $\boldsymbol{Y}_{ij}$. For each $1 \leq i \leq n$ and $1 \leq j \leq m$, we assume that the random variable $\boldsymbol{Y}_{ij}$ is sampled from a continuous exponential family, in the sense that each $\boldsymbol{Y}_{ij}$ admits a probability density function (pdf) $q(y; \boldsymbol{X}_{ij})$ with respect to the Lebesgue measure $\mathrm{d}y$ on the real line $\mathcal{Y} = \mathbb{R}$. The pdf $q(y; \boldsymbol{X}_{ij})$ of $\boldsymbol{Y}_{ij}$ can thus be written in the general form:

$$q(y; \boldsymbol{X}_{ij}) = h(y) \exp\left(\eta(\boldsymbol{X}_{ij}) y - A(\eta(\boldsymbol{X}_{ij}))\right), \; y \in \mathcal{Y}, \tag{14}$$

where η (the link function) is a one-to-one and smooth function, A (the log-partition function) is a twice differentiable mapping, h is a known function, and $\boldsymbol{X}_{ij}$ is an unknown parameter of interest belonging to some open subset $\mathcal{X}$ of $\mathbb{R}$. Throughout the paper, we will suppose that the following assumption holds:

Assumption 2 *The link function η and the log-partition function A are such that*

$$A'(\eta(x)) = x \text{ for all } x \in \mathcal{X},$$

where A' denotes the first derivative of A.

Since $\mathbb{E}[\boldsymbol{Y}_{ij}] = A'(\eta(\boldsymbol{X}_{ij}))$ for exponential families in the general form (14), Assumption 2 implies that $\mathbb{E}[\boldsymbol{Y}_{ij}] = \boldsymbol{X}_{ij}$, and thus the data matrix satisfies the relation $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ where $\boldsymbol{W}$ is a centered noise matrix, which is in agreement with model (1). Now, if we let $\Theta = \eta(\mathcal{X})$, it will be also convenient to consider the expression of the pdf of $\boldsymbol{Y}_{ij}$ in the canonical form:

$$p(y; \boldsymbol{\theta}_{ij}) = h(y) \exp\left(\boldsymbol{\theta}_{ij} y - A(\boldsymbol{\theta}_{ij})\right), \; y \in \mathcal{Y}, \tag{15}$$

where $\boldsymbol{\theta}_{ij} = \eta(\boldsymbol{X}_{ij}) \in \Theta$ is usually called the canonical parameter of the exponential family. Finally, we recall the relation $\mathrm{Var}(\boldsymbol{Y}_{ij}) = A''(\boldsymbol{\theta}_{ij}) = A''(\eta(\boldsymbol{X}_{ij}))$ where A'' denotes the second derivative of A. Then, we denote by $\boldsymbol{\theta}$ the $n \times m$ matrix whose entries are the $\boldsymbol{\theta}_{ij}$'s.

Examples of data satisfying model (14) are the following ones:

Gaussian noise with known variance τ^2:

$$q(y; \boldsymbol{X}_{ij}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y - \boldsymbol{X}_{ij})^2}{2\tau^2}\right), \; \mathbb{E}[\boldsymbol{Y}_{ij}] = \boldsymbol{X}_{ij}, \; \mathrm{Var}(\boldsymbol{Y}_{ij}) = \tau^2,$$

$$\mathcal{Y} = \mathbb{R}, \; \mathcal{X} = \mathbb{R}, \; \Theta = \mathbb{R}, \; h(y) = \frac{1}{\sqrt{2\pi\tau}} \exp\left(-\frac{y^2}{2\tau^2}\right), \; \eta(x) = \frac{x}{\tau^2}, \; A(\theta) = \tau^2 \frac{\theta^2}{2}.$$

Gamma-distributed measurements with known shape parameter $L > 0$:

$$q(y; \boldsymbol{X}_{ij}) = \frac{L^L y^{L-1}}{\Gamma(L) \boldsymbol{X}_{ij}^L} \exp\left(-L \frac{y}{\boldsymbol{X}_{ij}}\right) \mathbb{1}_{]0,+\infty[}(y), \; \mathbb{E}[\boldsymbol{Y}_{ij}] = \boldsymbol{X}_{ij}, \; \mathrm{Var}(\boldsymbol{Y}_{ij}) = \frac{\boldsymbol{X}_{ij}^2}{L},$$

$$\mathcal{Y} = \mathbb{R}, \; \mathcal{X} =]0, +\infty[, \; \Theta =]-\infty, 0[, \; h(y) = \frac{L^L y^{L-1}}{\Gamma(L)} \mathbb{1}_{]0,+\infty[}(y), \; \eta(x) = -\frac{L}{x}, \; A(\theta) = -L \log\left(-\frac{\theta}{L}\right).$$

The matrix $\boldsymbol{\theta} = \eta(X)$ can then be estimated via the $n \times m$ matrix $\hat{\boldsymbol{\theta}}^f = \hat{\boldsymbol{\theta}}^f(\boldsymbol{Y})$ whose entries are given by

$$\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}) = \eta\left(\hat{\boldsymbol{X}}^f_{ij}\right), \text{ for all } 1 \leq i \leq n,\ 1 \leq j \leq m, \tag{16}$$

where $\hat{\boldsymbol{X}}^f_{ij}$ is a spectral estimator as defined in eq. (3).

In the rest of this section, we follow the arguments in Eldar (2009) and Deledalle (2017) to derive SURE-like formulas under the exponential family for the estimators $\hat{\boldsymbol{\theta}}^f$ and $\hat{\boldsymbol{X}}^f$, using either the mean-squared error (MSE) risk or the Kullback-Leibler (KL) risk.

2.1.1 Unbiased estimation of the MSE risk

We consider the following MSE risk which provides a measure of discrepancy in the space Θ of natural parameters, and then indirectly in the space of interest $\mathcal{X}$.

Definition 3 *The squared error (SE) risk of* $\hat{\boldsymbol{\theta}}^f$ *is* $\mathrm{SE}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \|\hat{\boldsymbol{\theta}}^f - \boldsymbol{\theta}\|_F^2$*, and the mean-squared error (MSE) risk of* $\hat{\boldsymbol{\theta}}^f$ *is defined as* $\mathrm{MSE}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \mathbb{E}\left[\mathrm{SE}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta})\right] = \mathbb{E}\left[\|\hat{\boldsymbol{\theta}}^f - \boldsymbol{\theta}\|_F^2\right]$.

Using the above MSE risk to compare $\hat{\boldsymbol{\theta}}^f$ and $\boldsymbol{\theta}$ implies that the discrepancy between the estimator $\hat{\boldsymbol{X}}^f$ and the matrix of interest $\boldsymbol{X}$ is measured by the quantity $\mathrm{MSE}_\eta(\hat{\boldsymbol{X}}^f, \boldsymbol{X}) = \mathrm{MSE}(\eta(\hat{\boldsymbol{X}}^f), \eta(\boldsymbol{X}))$ which is different from $\mathrm{MSE}(\hat{\boldsymbol{X}}^f, \boldsymbol{X})$. For Gaussian noise, $\mathrm{MSE}_\eta(\hat{\boldsymbol{X}}^f, \boldsymbol{X}) = \frac{1}{\tau^2}\mathbb{E}\left[\|\hat{\boldsymbol{X}}^f - \boldsymbol{X}\|_F^2\right]$, while for Gamma distributed measurements with known shape parameter $L > 0$, it follows that

$$\mathrm{MSE}_\eta(\hat{\boldsymbol{X}}^f, \boldsymbol{X}) = L^2 \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\frac{\boldsymbol{X}_{ij} - \hat{\boldsymbol{X}}^f_{ij}}{\boldsymbol{X}_{ij}\hat{\boldsymbol{X}}^f_{ij}}\right)^2.$$

The following proposition gives a SURE formula for the MSE risk introduced in Definition 3.

Proposition 4 *Suppose that the data are sampled from a continuous exponential family. Assume that the function* h*, in the definition* (15) *of the exponential family, is twice continuously differentiable on* $\mathcal{Y} = \mathbb{R}$*. If the following condition holds*

$$\mathbb{E}\left[\left|\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})\right|\right] < +\infty, \text{ for all } 1 \leq i \leq n,\ 1 \leq j \leq m, \tag{17}$$

then, the quantity

$$\mathrm{GSURE}(\hat{\boldsymbol{\theta}}^f) = \|\hat{\boldsymbol{\theta}}^f(\boldsymbol{Y})\|^2 + \sum_{i=1}^{n}\sum_{j=1}^{m}\left(2\frac{h'(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})}\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}) + \frac{h''(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})}\right) + 2\,\mathrm{div}\,\hat{\boldsymbol{\theta}}^f(\boldsymbol{Y}), \tag{18}$$

where $\mathrm{div}\,\hat{\boldsymbol{\theta}}^f(\boldsymbol{Y}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\partial\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})}{\partial\boldsymbol{Y}_{ij}}$*, is an unbiased estimator of* $\mathrm{MSE}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta})$

Note that $\mathrm{GSURE}(\hat{\boldsymbol{\theta}}^f)$ is an unbiased estimator of $\mathrm{MSE}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta})$ and not of $\mathrm{MSE}(\hat{\boldsymbol{X}}^f, \boldsymbol{X})$. It is shown in Section 3.3 that the results of Proposition 17 coincide with the approach in Candès et al. (2013) on the derivation of a SURE formula in the case of Gaussian noise for smooth spectral estimators. In the case of Gamma noise, assuming $L > 2$ implies that the conditions on the function h in Proposition 4 is satisfied, hence assuming that conditions (17) holds as well, and using that

$$\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}) = -\frac{L}{f_{ij}(\boldsymbol{Y})} \quad \text{and} \quad \frac{\partial\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})}{\partial\boldsymbol{Y}_{ij}} = \frac{L}{|f_{ij}(\boldsymbol{Y})|^2}\frac{\partial f_{ij}(\boldsymbol{Y})}{\partial\boldsymbol{Y}_{ij}},$$

it follows that

$$\mathrm{GSURE}(\hat{\boldsymbol{\theta}}^f) = \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{L^2}{|f_{ij}(\boldsymbol{Y})|^2} - \frac{2L(L-1)}{\boldsymbol{Y}_{ij}f_{ij}(\boldsymbol{Y})} + \frac{2L}{|f_{ij}(\boldsymbol{Y})|^2}\frac{\partial f_{ij}(\boldsymbol{Y})}{\partial\boldsymbol{Y}_{ij}} - \frac{(L-1)(L-2)}{|\boldsymbol{Y}_{ij}|^2}. \tag{19}$$

2.1.2 Unbiased estimation of KL risks

Following the terminology in Deledalle (2017), let us now introduce two different notions of Kullback-Leibler risk, which arise from the non-symmetry of this discrepancy measure.

Definition 5 *Let* $f : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ *be a smooth spectral function. Consider the estimator* $\hat{\boldsymbol{\theta}}^f$ *defined by* (16)*, where* $\boldsymbol{Y}$ *is a matrix whose entries* $\boldsymbol{Y}_{ij}$ *are independent random variables sampled from the exponential family* (15) *in canonical form:*

- *the Kullback-Leibler synthesis (KLS) risk of* $\hat{\boldsymbol{\theta}}^f$ *is defined as*

$$\mathrm{KLS}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\int_{\mathbb{R}}\log\left(\frac{p(y;\hat{\boldsymbol{\theta}}^f_{ij})}{p(y;\boldsymbol{\theta}_{ij})}\right)p(y;\hat{\boldsymbol{\theta}}^f_{ij})\,\mathrm{d}y,$$

  and the mean KLS risk of $\hat{\boldsymbol{\theta}}^f$ *is defined as* $\mathrm{MKLS}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \mathbb{E}\left[\mathrm{KLS}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta})\right]$,

- *the Kullback-Leibler analysis (KLA) risk of* $\hat{\boldsymbol{\theta}}^f$ *is defined as*

$$\mathrm{KLA}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\int_{\mathbb{R}}\log\left(\frac{p(y;\boldsymbol{\theta}_{ij})}{p(y;\hat{\boldsymbol{\theta}}^f_{ij})}\right)p(y;\boldsymbol{\theta}_{ij})\,\mathrm{d}y,$$

  and the mean KLA risk of $\hat{\boldsymbol{\theta}}^f$ *is defined as* $\mathrm{MKLA}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \mathbb{E}\left[\mathrm{KLA}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta})\right]$.

A key advantage of the Kullback-Leibler risk is that it measures the discrepancy between the unknown distribution $p(y;\boldsymbol{\theta}_{ij})$ and its estimate $p(y;\hat{\boldsymbol{\theta}}^f_{ij})$. It is thus invariant with respect to the reparametrization $\hat{\boldsymbol{\theta}}^f = \eta(\hat{\boldsymbol{X}}^f)$ (unlike the MSE risk), and we may also write $\mathrm{MKLS}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \mathrm{MKLS}(\hat{\boldsymbol{X}}^f, \boldsymbol{X})$ and $\mathrm{MKLA}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \mathrm{MKLA}(\hat{\boldsymbol{X}}^f, \boldsymbol{X})$. As suggested

in Deledalle (2017), the MKLA risk represents how well the distribution $p(y;\hat{\boldsymbol{\theta}}^f_{ij})$ explain a random variable $\boldsymbol{Y}_{ij}$ sampled from the pdf $p(y;\boldsymbol{\theta}_{ij})$. The MKLA risk is a natural loss function in many statistical problems since it takes as a reference measure the true distribution of the data (see e.g. Hall (1987)). The MKLS risk represents how well one may generate an independent copy of $\boldsymbol{Y}_{ij}$ by sampling a random variable from the pdf $p(y;\hat{\boldsymbol{\theta}}^f_{ij})$. The MKLS risk has also been considered in various inference problems in statistics (Hannig and Lee, 2006; Yanagimoto, 1994).

By simple calculation, it follows that

$$\mathrm{MKLS}(\hat{\boldsymbol{\theta}}^f,\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}\left[\left(\hat{\boldsymbol{\theta}}^f_{ij}-\boldsymbol{\theta}_{ij}\right)A'(\hat{\boldsymbol{\theta}}^f_{ij})\right] + A(\boldsymbol{\theta}_{ij}) - \mathbb{E}\left[A(\hat{\boldsymbol{\theta}}^f_{ij})\right], \tag{20}$$

$$\text{and}\quad \mathrm{MKLA}(\hat{\boldsymbol{\theta}}^f,\boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}\left[\left(\boldsymbol{\theta}_{ij}-\hat{\boldsymbol{\theta}}^f_{ij}\right)A'(\boldsymbol{\theta}_{ij})\right] + \mathbb{E}\left[A(\hat{\boldsymbol{\theta}}^f_{ij})\right] - A(\boldsymbol{\theta}_{ij}). \tag{21}$$

Hence, in the case of Gaussian measurements with known variance τ^2, we easily retrieve that $\mathrm{MKLS}(\hat{\boldsymbol{\theta}}^f,\boldsymbol{\theta}) = \mathrm{MKLA}(\hat{\boldsymbol{\theta}}^f,\boldsymbol{\theta}) = \frac{\tau^2}{2}\mathrm{MSE}(\hat{\boldsymbol{\theta}}^f,\boldsymbol{\theta}) = \frac{1}{2\tau^2}\mathbb{E}\left[\|\hat{\boldsymbol{X}}^f-\boldsymbol{X}\|_F^2\right]$. In the case of Gamma distributed measurements with known shape parameter $L>0$, it follows that

$$\begin{aligned}
\mathrm{MKLS}(\hat{\boldsymbol{\theta}}^f,\boldsymbol{\theta}) &= L\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}\left[\frac{\hat{\boldsymbol{X}}^f_{ij}}{\boldsymbol{X}_{ij}} - \log\left(\frac{\hat{\boldsymbol{X}}^f_{ij}}{\boldsymbol{X}_{ij}}\right) - 1\right],\\
\mathrm{MKLA}(\hat{\boldsymbol{\theta}}^f,\boldsymbol{\theta}) &= L\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}\left[\frac{\boldsymbol{X}_{ij}}{\hat{\boldsymbol{X}}^f_{ij}} - \log\left(\frac{\boldsymbol{X}_{ij}}{\hat{\boldsymbol{X}}^f_{ij}}\right) - 1\right].
\end{aligned}$$

Below, we use some of the results in Deledalle (2017) whose main contributions are the derivation of new unbiased estimators of the MKLS and MKLA risks. For continuous exponential family, the risk estimate derived in Deledalle (2017) is unbiased for the MKLS risk, while it is only asymptotically unbiased for the MKLA risk with respect to the signal-to-noise ratio. For data sampled from a continuous exponential family, this makes simpler the use of the MKLS risk to derive data-driven shinkage in low rank matrix denoising, and we have therefore chosen to concentrate our study on this risk in this setting. The following proposition establishes a SURE formula to estimate the MKLS risk in the continuous case.

Proposition 6 *Suppose that the data are sampled from a continuous exponential family. Assume that the function h, in the definition* (15) *of the exponential family, is continuously differentiable on $\mathcal{Y}=\mathbb{R}$. Suppose that the function A, in the definition* (15) *of the exponential family, is twice continuously differentiable on Θ. If the following condition holds*

$$\mathbb{E}\left[\left|A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\right|\right] < +\infty, \text{ for all } 1\le i\le n,\ 1\le j\le m, \tag{22}$$

then, the quantity

$$\mathrm{SUKLS}(\hat{\boldsymbol{\theta}}^f) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\left(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}) + \frac{h'(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})}\right)A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})) - A(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\right) + \operatorname{div} f(\boldsymbol{Y}), \tag{23}$$

where $\operatorname{div} f(\boldsymbol{Y}) = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\partial f_{ij}(\boldsymbol{Y})}{\partial \boldsymbol{Y}_{ij}}$, *is an unbiased estimator of* $\mathrm{MKLS}(\hat{\boldsymbol{\theta}}^f,\boldsymbol{\theta}) - \sum_{i=1}^{n}\sum_{j=1}^{m}A(\boldsymbol{\theta}_{ij})$.

A key difference in the formula of unbiased estimates for the MSE and the KL risks is the computation of the divergence term in (18) and (23), when $\hat{\boldsymbol{X}}^f = \sum_{k=1}^{\min(n,m)} f_k(\tilde{\sigma}_k)\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t$ is a smooth spectral estimator in the sense where each function $f_k:\mathbb{R}_+\to\mathbb{R}_+$ is assumed to be (almost everywhere) differentiable for $1\le k\le\min(n,m)$. In this setting, the divergence term in the expression of $\mathrm{GSURE}(\hat{\boldsymbol{\theta}}^f)$ depends upon the matrix $\hat{\boldsymbol{\theta}}^f(\boldsymbol{Y}) = \eta(\hat{\boldsymbol{X}}^f)$. Therefore, when η is a nonlinear mapping, it is generally not possible to obtain a simpler expression for $\operatorname{div}\hat{\boldsymbol{\theta}}^f(\boldsymbol{Y})$. To the contrary, for $\mathrm{SUKLS}(\hat{\boldsymbol{\theta}}^f)$, the divergence term is $\operatorname{div} f(\boldsymbol{Y})$ which has the following closed-form expression for any smooth spectral estimators

$$\operatorname{div} f(\boldsymbol{Y}) = |m-n|\sum_{k=1}^{\min(n,m)}\frac{f_k(\tilde{\sigma}_k)}{\tilde{\sigma}_k} + \sum_{k=1}^{\min(n,m)}f'_k(\tilde{\sigma}_k) + 2\sum_{k=1}^{\min(n,m)}f_k(\tilde{\sigma}_k)\sum_{\ell=1;\ell\neq k}^{\min(n,m)}\frac{\tilde{\sigma}_k}{\tilde{\sigma}_k^2-\tilde{\sigma}_\ell^2}, \tag{24}$$

thanks to the results from Theorem IV.3 in Candès et al. (2013).

Note that $\mathrm{SUKLS}(\hat{\boldsymbol{X}}^r_w) = \frac{\tau^2}{2}\,\mathrm{SURE}(\hat{\boldsymbol{X}}^r_w)$ for Gaussian measurements, hence, the GSURE and SUKLS strategies match in this case. In the case of Gamma measurements, assuming that $L>2$ implies that the conditions on the function h in Proposition 6 is satisfied, and by assuming that condition (22) holds as well, it follows that

$$\mathrm{SUKLS}(\hat{\boldsymbol{\theta}}^f) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left((L-1)\frac{f_{ij}(\boldsymbol{Y})}{\boldsymbol{Y}_{ij}} - L\log\left(f_{ij}(\boldsymbol{Y})\right)\right) - Lmn + \operatorname{div} f(\boldsymbol{Y}),$$

where the expression of $\operatorname{div} f(\boldsymbol{Y})$ is given by (24).

Note that it is implicitly understood in the definition of $\operatorname{div} f(\boldsymbol{Y})$ that each mapping $f_{ij}:\mathbb{R}^{n\times m}\to\mathbb{R}$ is differentiable. The differentiability of the spectral function f (and thus of its components f_{ij}) is a consequence of the assumption that the functions $f_1,\dots,f_{\min(n,m)}$ (acting on the singular values) are supposed to be differentiable. For further details, on the differentiability of f and the f_{ij}'s, we refer to Section IV in Candès et al. (2013). From the arguments in Candès et al. (2013), it follows that formula (24) for the divergence of f is also valid under the assumption that each function f_k is differentiable on $\mathbb{R}_+$ except on a set of Lebesgue measure zero.

2.2 The case of Poisson data

For Poisson data, the key result to obtain unbiased estimate of a given risk is the following lemma which dates back to the work in Hudson (1978).

Lemma 7 *Let $f:\mathbb{Z}^{n\times m}\to\mathbb{R}^{n\times m}$ be a measurable mapping. Let $1\le i\le n$ and $1\le j\le m$, and denote by $f_{ij}:\mathbb{Z}^{n\times m}\to\mathbb{R}$ a measurable function. Let $\boldsymbol{Y}\in\mathbb{Z}^{n\times m}$ be a matrix whose entries are independently sampled from a Poisson distribution on $\mathbb{Z}$. Then,*

$$\mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{X}_{ij}f_{ij}(\boldsymbol{Y})\right] = \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{Y}_{ij}f_{ij}(\boldsymbol{Y}-\boldsymbol{e}_i\boldsymbol{e}_j^t)\right],$$

where, for each $1 \leq i \leq n$ and $1 \leq j \leq m$, $f_{ij}(\boldsymbol{Y})$ denotes the (i,j)-th entry of the matrix $f(\boldsymbol{Y})$, and $\boldsymbol{e}_i$ (resp. $\boldsymbol{e}_j$) denotes the vector of $\mathbb{Z}^n$ (resp. $\mathbb{Z}^m$) with the i-th entry (resp. j-th entry) equals to one and all others equal to zero.

Hudson's lemma provides a way to estimate (in an unbiased way) the expectation of the Frobenius inner product between the matrix $\boldsymbol{X}$ and the matrix $f(\boldsymbol{Y})$. To see the usefulness of this result, one may consider the following mean-squared error

$$\mathrm{MSE}(\hat{\boldsymbol{X}}^f, \boldsymbol{X}) = \mathbb{E}\left[\left\|\hat{\boldsymbol{X}}^f - \boldsymbol{X}\right\|_F^2\right] = \mathbb{E}\left[\left\|\hat{\boldsymbol{X}}^f\right\|_F^2 - 2\sum_{i=1}^n\sum_{j=1}^m \boldsymbol{X}_{ij}\hat{\boldsymbol{X}}^f_{ij}(\boldsymbol{Y}) + \|\boldsymbol{X}\|_F^2\right].$$

Therefore, by Lemma 7, one immediately obtains that

$$\mathrm{PURE}(\hat{\boldsymbol{\theta}}^f) = \left\|\hat{\boldsymbol{X}}^f\right\|_F^2 - 2\sum_{i=1}^n\sum_{j=1}^m \boldsymbol{Y}_{ij} f_{ij}(\boldsymbol{Y} - \boldsymbol{e}_i\boldsymbol{e}_j^t), \tag{25}$$

is an unbiased estimate for the quantity $\mathrm{MSE}(\hat{\boldsymbol{X}}^f, \boldsymbol{X}) - \|\boldsymbol{X}\|_F^2$.

For Poisson data, one may also define the following KL risks

$$\mathrm{MKLS}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \sum_{i=1}^n\sum_{j=1}^m \mathbb{E}\left[\boldsymbol{X}_{ij} - \hat{\boldsymbol{X}}^f_{ij} - \hat{\boldsymbol{X}}^f_{ij}\log\left(\frac{\boldsymbol{X}_{ij}}{\hat{\boldsymbol{X}}^f_{ij}}\right)\right], \tag{26}$$

$$\mathrm{MKLA}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \sum_{i=1}^n\sum_{j=1}^m \mathbb{E}\left[\hat{\boldsymbol{X}}^f_{ij} - \boldsymbol{X}_{ij} - \boldsymbol{X}_{ij}\log\left(\frac{\hat{\boldsymbol{X}}^f_{ij}}{\boldsymbol{X}_{ij}}\right)\right], \tag{27}$$

which are in agreement with Definition 5 of KL risks for data sampled from a Poisson distribution. From the arguments in Deledalle (2017), there does not currently exist an approach to derive a SURE formula for the MKLS risk in the Poisson case since they are no unbiased formula for $\hat{\boldsymbol{X}}^f_{ij}\log \boldsymbol{X}_{ij}$. Nevertheless, as shown in Deledalle (2017), Hudson's Lemma 7 provides an unbiased estimator for $\boldsymbol{X}_{ij}\log\hat{\boldsymbol{X}}^f_{ij}$, and then it is possible to unbiasedly estimate the MKLA risk as follows.

Proposition 8 *For data sampled from a Poisson distribution, the quantity*

$$\mathrm{PUKLA}(\hat{\boldsymbol{\theta}}^f) = \sum_{i=1}^n\sum_{j=1}^m \hat{\boldsymbol{X}}^f_{ij} - \boldsymbol{Y}_{ij}\log\left(f_{ij}(\boldsymbol{Y} - \boldsymbol{e}_i\boldsymbol{e}_j^t)\right), \tag{28}$$

is an unbiased estimator of $\mathrm{MKLA}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) + \sum_{i=1}^n\sum_{j=1}^m \boldsymbol{X}_{ij} - \boldsymbol{X}_{ij}\log(\boldsymbol{X}_{ij})$.

2.3 Data-driven shrinkage in low-rank matrix denoising

For a matrix $\boldsymbol{X}$ with entries $\boldsymbol{X}_{ij} \in \mathcal{X} = \mathbb{R}$, we consider shrinkage estimators of the form

$$\hat{\boldsymbol{X}}^s_w = f(\boldsymbol{Y}) = \sum_{k\in s} w_k\tilde{\sigma}_k\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t, \tag{29}$$

with $s \subseteq \mathcal{I} = \{1, 2, \ldots, \min(n,m)\}$ and $w_k \in [0,1]$, for all $k \in s$.

When the underlying matrix $\boldsymbol{X}$ is constrained to have positive entries, e.g. $\mathcal{X} =]0, +\infty[$ in the Gamma and Poisson cases, we consider instead estimators of the form

$$\hat{\boldsymbol{X}}^s_w = f(\boldsymbol{Y}) = \max\left[\sum_{k\in s} w_k\tilde{\sigma}_k\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t, \varepsilon\right], \tag{30}$$

where $\varepsilon > 0$ is an *a priori* lower bound on the smallest value of $\boldsymbol{X}_{ij}$, where for any matrix $\boldsymbol{X}$, $\max[\boldsymbol{X}, \varepsilon]_{ij} = \max[\boldsymbol{X}_{ij}, \varepsilon]$, for all $1 \leq i \leq m$ and $1 \leq j \leq n$.

The construction of the subset s is postponed to Section 4, and we focus here in selecting the weights in a data-driven way for a fixed given s. In the following, we denote by s^c the complementary set of s in $\mathcal{I}$, *i.e.*, $s^c = \mathcal{I} \setminus s$, and we let $\hat{\boldsymbol{\theta}}^s_w = \eta\left(\hat{\boldsymbol{X}}^s_w\right)$. When $\mathcal{X} =]0, +\infty[$, we have found that considering estimators of the form (30) is more appropriate than trying to find shrinking weights $(w_k)_{k\in s}$ such that all the entries of the matrix $\sum_{k\in s} w_k\tilde{\sigma}_k\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t$ are positive, for a given subset s.

Gaussian noise with known homoscedastic variance τ^2

By applying the GSURE formula (18) for Gaussian distributed measurements and thanks to the expression (24) for the divergence of smooth spectral estimators, we obtain for $\hat{\boldsymbol{X}}^s_w$, as defined in (29), the SURE expression given by

$$\mathrm{SURE}(\hat{\boldsymbol{X}}^s_w) = -mn\tau^2 + \sum_{k\in s}(w_k - 1)^2\tilde{\sigma}_k^2 + \sum_{k\in s^c}\tilde{\sigma}_k^2 + 2\tau^2\sum_{k=1}^s\left(1 + |m-n| + 2\sum_{\ell=1;\ell\neq k}^{\min(n,m)}\frac{\tilde{\sigma}_k^2}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2}\right)w_k$$

which unbiasedly estimate $\mathrm{MSE}(\hat{\boldsymbol{X}}^s_w, \boldsymbol{X})$. Hence, for each $k \in s$, by differentiating the above expression with respect to w_k, it follows that a data-driven weight for the k-th empirical singular value is given by

$$w_k(\boldsymbol{Y}) = \left(1 - \frac{\tau^2}{\tilde{\sigma}_k^2}\left(1 + |m-n| + 2\sum_{\ell=1;\ell\neq k}^{\min(n,m)}\frac{\tilde{\sigma}_k^2}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2}\right)\right)_+, \tag{31}$$

which fullfils the requirement that $w_k(\boldsymbol{Y}) \in [0,1]$. Note that as $\mathrm{SUKLS}(\hat{\boldsymbol{X}}^s_w) = \frac{\tau^2}{2}\mathrm{SURE}(\hat{\boldsymbol{X}}^s_w)$ for Gaussian measurements, the exact same data-driven weight would be obtained by minimizing an estimate of the $\mathrm{MKLS}(\hat{\boldsymbol{X}}^s_w, \boldsymbol{X})$.

The case of estimators with rank one. Consider the case of estimators with rank 1, *i.e.*, let $s = \{1\}$. It follows that $\hat{\boldsymbol{X}}^1_w = \hat{\boldsymbol{X}}^{\{1\}}_w = w_1\hat{\boldsymbol{X}}^1$ where $w_1 \in [0,1]$ is given by

$$w_1(\boldsymbol{Y}) = \left(1 - \frac{\tau^2}{\tilde{\sigma}_1^2}\left(1 + |m-n| + 2\sum_{\ell=1;\ell\neq 1}^{\min(n,m)}\frac{\tilde{\sigma}_1^2}{\tilde{\sigma}_1^2 - \tilde{\sigma}_\ell^2}\right)\right)_+.$$

Gamma and Poisson distributed measurements

In Gamma and Poisson cases, it is not possible to follow the same strategy as in the Gaussian case to derive optimal weights for (30) in a closed-form using the established SURE-like formulas. We shall investigate how data-driven shinkage can be approximated in Section 5 on numerical experiments using fast algorithms. Nevertheless, when the estimator is restricted to rank 1, optimizing KL risk estimators lead to closed-form expressions under the assumption that all the entries of the data matrix $\boldsymbol{Y}$ are strictly positive.

The case of estimators with rank one under Gamma noise. Consider again the case of estimators with rank 1, *i.e.*, let $s = \{1\}$, and let $\hat{\boldsymbol{X}}^1 = \tilde{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t$ denote the PCA approximation of rank 1 of $\boldsymbol{X}$. If all the entries of the matrix $\boldsymbol{Y}$ are strictly positive, by the Perron-Frobenius theorem, all the entries of the first singular vectors $\tilde{\boldsymbol{u}}_1$ and $\tilde{\boldsymbol{v}}_1$ are strictly positive. Therefore, all the entries of $\hat{\boldsymbol{X}}^1$ belong to the set $\mathcal{X} =]0, +\infty[$, and we can consider $\hat{\boldsymbol{X}}_w^1 = \hat{\boldsymbol{X}}_w^{\{1\}} = w_1 \tilde{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t$ as defined in (29) instead of (30). Assuming $L > 2$ for the SUKLS formula to hold, it follows by simple calculations that

$$\mathrm{SUKLS}(\hat{\boldsymbol{\theta}}_w^1) = \sum_{i=1}^{n}\sum_{j=1}^{m}(L-1)w_1\frac{\hat{\boldsymbol{X}}_{ij}^1}{\boldsymbol{Y}_{ij}} - mnL\log(w_1) - L\log\left(\frac{\hat{\boldsymbol{X}}_{ij}^1}{\boldsymbol{Y}_{ij}}\right) - Lmn + (1+|m-n|)w_1 + 2w_1\sum_{\ell=2}^{\min(n,m)}\frac{\tilde{\sigma}_1^2}{\tilde{\sigma}_1^2 - \tilde{\sigma}_\ell^2}.$$

Hence, by differentiating the above expression with respect to w_1 and as it is monotonic on both sides of its unique minimum, the optimal value of $w_1 \in [0,1]$ minimizing $\mathrm{SUKLS}(\hat{\boldsymbol{\theta}}_w^1)$ is given by

$$w_1(\boldsymbol{Y}) = \min\left[1, \left(\frac{L-1}{Lmn}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\hat{\boldsymbol{X}}_{ij}^1}{\boldsymbol{Y}_{ij}} + \frac{1}{Lmn}\left(1+|m-n|+2\sum_{\ell=2}^{\min(n,m)}\frac{\tilde{\sigma}_1^2}{\tilde{\sigma}_1^2-\tilde{\sigma}_\ell^2}\right)\right)^{-1}\right],$$

which yields the shrinking rule (11) stated in the introduction of this paper. Note that it is not possible to obtain, in a closed-form, the optimal value of the weight w_1 that minimizes the criterion $\mathrm{GSURE}(\hat{\boldsymbol{\theta}}_w^1)$.

The case of estimators with rank one under Poisson noise. Using again that all the assumption that the entries of $\boldsymbol{Y}$ are positive, we can consider (by the Perron-Frobenius theorem) $\hat{\boldsymbol{X}}_w^1 = \hat{\boldsymbol{X}}_w^{\{1\}} = w_1\tilde{\sigma}_1\tilde{\boldsymbol{u}}_1\tilde{\boldsymbol{v}}_1^t$ as defined in (29) instead of (30). Then, the PURE formula (25) and Proposition 8 apply to the estimator $\hat{\boldsymbol{\theta}}_w^1 = \log\left(\hat{\boldsymbol{X}}_w^1\right)$ which yield to

$$\mathrm{PURE}(\hat{\boldsymbol{\theta}}_w^1) = w_1^2\tilde{\sigma}_1^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{Y}_{ij}w_1\tilde{\sigma}_1^{(ij)}\tilde{\boldsymbol{u}}_{1,i}^{(ij)}\tilde{\boldsymbol{v}}_{1,j}^{(ij)},$$

$$\text{and}\quad \mathrm{PUKLA}(\hat{\boldsymbol{\theta}}_w^1) = \sum_{i=1}^{n}\sum_{j=1}^{m}w_1\hat{\boldsymbol{X}}_{ij}^1 - \boldsymbol{Y}_{ij}\left(\log(w_1) + \log\left(\tilde{\sigma}_1^{(ij)}\tilde{\boldsymbol{u}}_{1,i}^{(ij)}\tilde{\boldsymbol{v}}_{1,j}^{(ij)}\right)\right),$$

where $\hat{\boldsymbol{X}}_{ij}^1 = \tilde{\sigma}_1\tilde{\boldsymbol{u}}_{1,i}\tilde{\boldsymbol{v}}_{1,j}$, $\tilde{\sigma}_1^{(ij)}$ is the largest singular value of the matrix $\boldsymbol{Y} - \boldsymbol{e}_i\boldsymbol{e}_j^t$, and $\tilde{\boldsymbol{u}}_1^{(ij)}$ (resp. $\tilde{\boldsymbol{v}}_1^{(ij)}$) denotes its left (resp. right) singular vectors. Therefore, by differentiating the above expression with respect to w_1 and as it is monotonic on both sides of its unique minimum, an optimal value for $w_1 \in [0,1]$ which minimizes $\mathrm{PURE}(\hat{\boldsymbol{\theta}}_w^1)$ is given by

$$w_1(\boldsymbol{Y}) = \min\left[1, \frac{1}{\tilde{\sigma}_1^2}\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{Y}_{ij}\tilde{\sigma}_1^{(ij)}\tilde{\boldsymbol{u}}_{1,i}^{(ij)}\tilde{\boldsymbol{v}}_{1,j}^{(ij)}\right].$$

However, this optimal shrinking rule cannot be used in practice since evaluating the values of $\tilde{\sigma}_1^{(ij)}, \tilde{\boldsymbol{u}}_1^{(ij)}, \tilde{\boldsymbol{v}}_1^{(ij)}$ for all $1 \le i \le n$ and $1 \le j \le m$ is not feasible from a computational point of view for large values of n and m. Nevertheless, a fast algorithm to find a numerical approximation of the optimal value $w_1(\boldsymbol{Y})$ is proposed in Section 5.

To the contrary, using again that all the $\hat{\boldsymbol{X}}_{ij}^1$ are positive by the Perron-Frobenius theorem, the value of $w_1 \in [0,1]$ minimizing $\mathrm{PUKLA}(\hat{\boldsymbol{\theta}}_w^1)$ is

$$w_1(\boldsymbol{Y}) = \min\left[1, \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{Y}_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m}\hat{\boldsymbol{X}}_{ij}^1}\right],$$

which is straightforward to compute. This corresponds to the shrinkage rule (12) given in the introduction.

3. Gaussian spiked population model

In this section, we restrict our analysis to the Gaussian spiked population model and the asymptotic setting introduced in Definition 1.

3.1 Asymptotic location of empirical singular values

We summarize below the asymptotic behavior of the singular values of the data matrix $\boldsymbol{Y} = \sum_{k=1}^{\min(n,m)}\tilde{\sigma}_k\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t$ in the Gaussian spiked population model.

In the case where $\boldsymbol{X} = 0$, it is well known (Anderson et al., 2010; Bai and Silverstein, 2010) that the empirical distribution of the singular values of $\boldsymbol{Y} = \boldsymbol{W}$ (with $\tau = \frac{1}{\sqrt{m}}$) converges, as $n \to +\infty$, to the quarter circle distribution if $c = 1$ and to its generalized version if $c < 1$. This distribution is supported on the compact interval $[c_-, c_+]$ with

$$c_\pm = 1 \pm \sqrt{c}$$

where c_+ is the so-called bulk (right) edge.

When $\boldsymbol{X} \neq 0$ has a low rank structure, the asymptotic behavior of the singular values of $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ is also well understood (Benaych-Georges and Nadakuditi, 2012; Dozier and Silverstein, 2007; Shabalin and Nobel, 2013), and generalizations to noise matrix $\boldsymbol{W}$ whose distribution is orthogonally invariant have also been recently considered in Benaych-Georges and Nadakuditi (2012). Below, we recall some of these results that will be needed in this paper. To this end, let us introduce the real-valued function ρ defined by

$$\rho(\sigma) = \sqrt{\frac{(1+\sigma^2)(c+\sigma^2)}{\sigma^2}} \text{ for any } \sigma > 0.$$

Then, the following result holds (see e.g. Theorem 2.8 in Benaych-Georges and Nadakuditi (2012) and Proposition 9 in Shabalin and Nobel (2013)).

Proposition 9 *Assume that $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ is a random matrix sampled from the Gaussian spiked population model with $\tau = \frac{1}{\sqrt{m}}$ and $\boldsymbol{X} = \sum_{k=1}^{r^*} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^t$. Then, for any fixed $k \geq 1$, one has that, almost surely,*

$$\lim_{n \to +\infty} \tilde{\sigma}_k = \begin{cases} \rho(\sigma_k) & \text{if } k \leq r^* \text{ and } \sigma_k > c^{1/4}, \\ c_+ & \text{otherwise.} \end{cases}$$

Moreover,

$$\lim_{n \to +\infty} \tilde{\sigma}_{\min(n,m)} = c_-.$$

In what follows, we shall also use the relation

$$\frac{1}{\sigma^2} = \frac{\rho^2(\sigma) - (c+1) - \sqrt{(\rho^2(\sigma) - (c+1))^2 - 4c}}{2c} \text{ that holds for any } \sigma > c^{1/4}, \tag{32}$$

which is a consequence of e.g. the results in Section 3.1 in Benaych-Georges and Nadakuditi (2012).

3.2 Existing asymptotically optimal shrinkage rules

Below, we briefly summarize some results in Gavish and Donoho (2017) and Nadakuditi (2014) on the construction of asymptotically optimal spectral estimators. Let

$$\hat{\boldsymbol{X}}^f = f(\boldsymbol{Y}) = \sum_{k=1}^{\min(n,m)} f_k(\tilde{\sigma}_k) \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \tag{33}$$

be a given smooth spectral estimator, and consider the standard squared error $\mathrm{SE}(\hat{\boldsymbol{X}}^f, \boldsymbol{X}) = \|\hat{\boldsymbol{X}}^f - \boldsymbol{X}\|_F^2$ as a measure of risk. The set of spectral functions minimizing $\mathrm{SE}(\hat{\boldsymbol{X}}^f, \boldsymbol{X})$ is given by $f_k(\tilde{\sigma}_k) = \tilde{\boldsymbol{u}}_k^t \boldsymbol{X} \tilde{\boldsymbol{v}}_k$, for $1 \leq k \leq \min(n,m)$. However, it cannot be used in practice since $\boldsymbol{X}$ is obviously unknown. A first alternative suggested in Gavish and Donoho (2017) and Nadakuditi (2014) is to rather study the asymptotic risk

$$\mathrm{SE}_\infty(\hat{\boldsymbol{X}}^f) = \lim_{n \to \infty} \mathrm{SE}(\hat{\boldsymbol{X}}^f, \boldsymbol{X}) \text{ (in the almost sure sense)} \tag{34}$$

in the Gaussian spiked population model. Then, it is proposed in Gavish and Donoho (2017) and Nadakuditi (2014) to find an asymptotically optimal choice of f by minimizing $\mathrm{SE}_\infty(\hat{\boldsymbol{X}}^f)$ among a given class of smooth spectral functions. The results in Gavish and Donoho (2017) show that, among spectral estimators of the form $\hat{\boldsymbol{X}}^\eta = \sum_{k=1}^{\min(n,m)} \eta(\tilde{\sigma}_k) \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t$, where $\eta : \mathbb{R}_+ \to \mathbb{R}_+$ is a continuous shrinker such that $\eta(\sigma) = 0$ whenever $\sigma \leq c_+$, an asymptotically optimal shrinkage rule is given by the choice

$$\eta^*(\sigma) = \begin{cases} \frac{1}{\sigma}\sqrt{(\sigma^2 - (c+1))^2 - 4c} & \text{if } \sigma > c_+, \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

In Nadakuditi (2014), it is proposed to consider spectral estimators of the form $\hat{\boldsymbol{X}}^\delta = \sum_{k=1}^r \delta_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t$ where $\delta_1, \ldots, \delta_r$ are positive weights. By Theorem 2.1 in Nadakuditi (2014), it follows that, if $\sigma_k > c^{1/4}$ for all $1 \leq k \leq r$ with $r \leq r^*$, then the weights which minimize $\mathrm{SE}_\infty(\hat{\boldsymbol{X}}^\delta)$ over $\mathbb{R}_+^r$ are given by

$$\delta_k^* = \delta_k(\sigma_k) = \frac{\sigma_k^4 - c}{\sigma_k \sqrt{(1+\sigma_k^2)(c+\sigma_k^2)}}, \text{ for all } 1 \leq k \leq r. \tag{36}$$

In what follows, the shrinkage rules (35) and (36) are shown to be equivalent, and they will serve as a reference of asymptotic optimality. It should be stressed that the estimators in Gavish and Donoho (2017) and Nadakuditi (2014) are not equivalent. Indeed, the method in Nadakuditi (2014) requires an estimate of the rank, while the approach in Gavish and Donoho (2017) applies the same shrinker to all empirical singular values. Nevertheless, the shrinkage function that is applied to significant singular values (either above the bulk edge in Gavish and Donoho (2017) or up to a given rank in Nadakuditi (2014)) is the same.

3.3 Asymptotic behavior of data-driven estimators based on SURE

Following the principle of SURE, a second alternative to choose a smooth spectral estimator of the form (33) is to study the problem of selecting a set of functions $(f_k)_{1 \leq k \leq \min(n,m)}$ that minimize an unbiased estimate of $\mathrm{MSE}(\hat{\boldsymbol{X}}^f, \boldsymbol{X}) = \mathbb{E}\left[\|\hat{\boldsymbol{X}}^f - \boldsymbol{X}\|_F^2\right]$. For any $1 \leq i \leq m$ and $1 \leq j \leq n$, we recall that $f_{ij}(\boldsymbol{Y})$ denotes the (i,j)-th entry of the matrix $\hat{\boldsymbol{X}}^f = f(\boldsymbol{Y})$. Under the condition that

$$\mathbb{E}\left[|\boldsymbol{Y}_{ij} f_{ij}(\boldsymbol{Y})| + \left|\frac{\partial f_{ij}(\boldsymbol{Y})}{\partial \boldsymbol{Y}_{ij}}\right|\right] < +\infty, \text{ for all } 1 \leq i \leq n,\ 1 \leq j \leq m. \tag{37}$$

it follows from the results in Candès et al. (2013) (or equivalently from Proposition 4 for Gaussian noise with $\tau^2 = 1/m$) that

$$\mathrm{SURE}\left(\hat{\boldsymbol{X}}^f\right) = -n + \|f(\boldsymbol{Y}) - \boldsymbol{Y}\|_F^2 + \frac{2}{m} \operatorname{div} f(\boldsymbol{Y}), \tag{38}$$

is an unbiased estimate of $\mathrm{MSE}(\hat{\boldsymbol{X}}^f, \boldsymbol{X})$, where the divergence $\operatorname{div} f(\boldsymbol{Y})$ admits the closed-form expression (24). The SURE formula (38) has been used in Candès et al. (2013) to find a data-driven value for $\lambda = \lambda(\boldsymbol{Y})$ in the the case of singular values shrinkage by soft-thresholding which corresponds to the choice

$$f_k(\tilde{\sigma}_k) = (\tilde{\sigma}_k - \lambda)_+, \text{ for all } 1 \leq k \leq \min(n,m).$$

We study now the asymptotic behavior of the SURE formula (38). To this end, we shall use Proposition 9, but we will also need the following result (whose proof can be found in the Appendix) to study some of the terms in expression (24) of the divergence of $f(\boldsymbol{Y})$.

Proposition 10 *Assume that $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ is a random matrix sampled from the Gaussian spiked population model with $\tau = \frac{1}{\sqrt{m}}$ and $\boldsymbol{X} = \sum_{k=1}^{r^*} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^t$. Then, for any fixed $1 \leq k \leq r^*$ such that $\sigma_k > c^{1/4}$, one has that, almost surely,*

$$\lim_{n \to +\infty} \frac{1}{n} \sum_{\ell=1; \ell \neq k}^{n} \frac{\tilde{\sigma}_k}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2} = \frac{1}{\rho\left(\sigma_k\right)} \left(1 + \frac{1}{\sigma_k^2}\right).$$

In what follows, we restrict our analysis to the following class of spectral estimators (the terminology in the definition below is borrowed from Gavish and Donoho (2017)).

Definition 11 *Let $\hat{\boldsymbol{X}}^f = f(\boldsymbol{Y}) = \sum_{k=1}^{\min(n,m)} f_k(\tilde{\sigma}_k) \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t$ be a smooth spectral estimator. For a given $1 \leq r \leq \min(n,m)$, the estimator f is said to be a spectral shrinker of order r that collapses the bulk to 0 if*

$$\left\{ \begin{array}{ll} f_k(\sigma) = 0 & \text{whenever } \sigma \leq c_+ \text{ and } 1 \leq k \leq r, \\ f_k(\sigma) = 0 & \text{for all } \sigma \geq 0 \text{ and } k > r. \end{array} \right.$$

The reason for restricting the study to spectral estimators such that $f_k(\tilde{\sigma}_k) = 0$ whenever $\tilde{\sigma}_k < c_+$ is linked to the choice of the active set s^* (8) of singular values in the Gaussian case, as detailed in Section 4. Now, for a spectral shrinker $\hat{\boldsymbol{X}}^f$ of order r that collapses the bulk to 0, we study the asymptotic behavior of the terms in expression (38) that only depend on f, namely

$$\begin{aligned} \overline{\mathrm{SURE}}\left(\hat{\boldsymbol{X}}^f\right) &= \sum_{k=1}^{r} (f_k(\tilde{\sigma}_k) - \tilde{\sigma}_k)^2 + 2\left(1 - \frac{n}{m}\right) \sum_{k=1}^{r} \frac{f_k(\tilde{\sigma}_k)}{\tilde{\sigma}_k} + \frac{2}{m} \sum_{k=1}^{r} f_k'(\tilde{\sigma}_k) \\ &\quad + 4\frac{n}{m} \sum_{k=1}^{r} f_k(\tilde{\sigma}_k) \left(\frac{1}{n} \sum_{\ell=1; \ell \neq k}^{n} \frac{\tilde{\sigma}_k}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2} \right) \end{aligned} \tag{39}$$

The reason for studying $\overline{\mathrm{SURE}}\left(\hat{\boldsymbol{X}}^f\right)$ is that finding an optimal shrinkage rule that minimizes $\mathrm{SURE}\left(\hat{\boldsymbol{X}}^f\right)$ is equivalent to minimizing expression (39) over spectral shrinkers of order r that collapses the bulk to 0, since $\mathrm{SURE}\left(\hat{\boldsymbol{X}}^f\right) - \overline{\mathrm{SURE}}\left(\hat{\boldsymbol{X}}^f\right) = -n + \sum_{k=r+1}^{n} \tilde{\sigma}_k^2$ for such $\hat{\boldsymbol{X}}^f$.

Then, using Proposition 9, Proposition 10, and the assumption that the f_k's are continuously differentiable functions on $\mathbb{R}_+$, we immediately obtain the following result.

Lemma 12 *Assume that $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ is a random matrix sampled from the Gaussian spiked population model with $\tau = \frac{1}{\sqrt{m}}$ and $\boldsymbol{X} = \sum_{k=1}^{r^*} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^t$. Let $\hat{\boldsymbol{X}}^f$ be a spectral shrinker of order $r \leq r^*$ that collapses the bulk to 0, such that each function f_k, for $1 \leq k \leq r$, is continuously differentiable on $]c_+, +\infty[$. Moreover, assume that $\sigma_k > c^{1/4}$ for all $1 \leq k \leq r$. Then, one has that, almost surely,*

$$\lim_{n \to +\infty} \overline{\mathrm{SURE}}\left(\hat{\boldsymbol{X}}^f\right) = \sum_{k=1}^{r} (f_k(\rho(\sigma_k)) - \rho(\sigma_k))^2 + 2 f_k(\rho(\sigma_k)) \left(\frac{\sigma_k^2(1+c) + 2c}{\sigma_k^2 \rho(\sigma_k)} \right) \tag{40}$$

Asymptotically optimal shrinkage of singular values. Thanks to Lemma 12, one may determine an asymptotic optimal spectral shrinker as the one minimizing $\lim_{n \to +\infty} \overline{\mathrm{SURE}}\left(\hat{\boldsymbol{X}}^f\right)$. For this purpose, let us define the class of estimators

$$\hat{\boldsymbol{X}}_w^r = \sum_{k=1}^{r} w_k \tilde{\sigma}_k \mathbb{1}_{\{\tilde{\sigma}_k > c_+\}} \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t, \tag{41}$$

where $1 \leq r \leq r^*$ is a given integer, and the w_k's are positive weights. In practice, the estimator $\hat{\boldsymbol{X}}_w^r$ is computed by replacing the bulk edge c_+ by its approximation $c_+^{n,m} = 1 + \sqrt{\frac{n}{m}}$ in eq. (41). For moderate to large values of n and m, the quantities c_+ and $c_+^{n,m}$ are very close, and this replacement does not change the numerical performances of $\hat{\boldsymbol{X}}_w^r$.

Then, provided that $\sigma_k > c^{1/4}$ for all $1 \leq k \leq r$, it follows from Lemma 12 that

$$\lim_{n \to +\infty} \overline{\mathrm{SURE}}\left(\hat{\boldsymbol{X}}_w^r\right) = \sum_{k=1}^{r} \rho^2(\sigma_k)(w_k - 1)^2 + 2 w_k \left(\frac{\sigma_k^2(1+c) + 2c}{\sigma_k^2} \right).$$

Differentiating the above expression with respect to each weight w_k leads to the following choice of asymptotically optimal weights

$$w_k^* = 1 - \frac{\sigma_k^2(1+c) + 2c}{\sigma_k^2 \rho^2(\sigma_k)} \text{ for all } 1 \leq k \leq r. \tag{42}$$

Therefore, if the singular values of the matrix $\boldsymbol{X}$ to be estimated are sufficiently large (namely $\sigma_k > c^{1/4}$ for all $1 \leq k \leq r$), by using Proposition 9 and eq. (42), one has that an asymptotically optimal spectral shrinker of order $r \leq r^*$ is given by the choice of functions

$$f_k^*(\rho(\sigma_k)) = \left\{ \begin{array}{cc} \left(1 - \frac{\sigma_k^2(1+c)+2c}{\sigma_k^2 \rho^2(\sigma_k)}\right) \rho(\sigma_k) & \text{if } \rho(\sigma_k) > c_+, \\ 0 & \text{otherwise,} \end{array} \right. \quad \text{for all } 1 \leq k \leq r. \tag{43}$$

Using, the relation (32) one may also express the asymptotically optimal shrinking rule (43) either as a function of $\rho(\sigma_k)$ only,

$$f_k^*(\rho(\sigma_k)) = \left\{ \begin{array}{cc} \frac{1}{\rho(\sigma_k)} \sqrt{(\rho^2(\sigma_k) - (c+1))^2 - 4c} & \text{if } \rho(\sigma_k) > c_+, \\ 0 & \text{otherwise.} \end{array} \right. \tag{44}$$

or as function of σ_k only (using that $\rho(\sigma_k) > c_+$ is equivalent to $\sigma_k > c^{1/4}$),

$$f_k^*(\rho(\sigma_k)) = \left\{ \begin{array}{cc} \frac{\sigma_k^4 - c}{\sigma_k \sqrt{(1+\sigma_k^2)(c+\sigma_k^2)}} & \text{if } \sigma_k > c^{1/4}, \\ 0 & \text{otherwise.} \end{array} \right. \tag{45}$$

Therefore, for spectral shrinker of order r, we remark that the shrinkage rule (44) coincides with the rule (35) which has been obtained in Gavish and Donoho (2017). Similarly, when the quantity $f_k^*(\rho(\sigma_k))$ is expressed as a function of σ_k only in (45), then we retrieve the shrinking rule (36) derived in Nadakuditi (2014). Therefore, it appears that minimizing either the asymptotic behavior of the SURE, that is $\lim_{n \to +\infty} \overline{\mathrm{SURE}}\left(\hat{\boldsymbol{X}}^f\right)$, or the limit of

SE risk (34) leads to the same choice of an asymptotically optimal spectral estimator.

Data-driven shrinkage of empirical singular values. From the results in Section 2.3, the principle of SURE minimisation leads to the following data-driven choice of spectral shrinker of order r that collapses the bulk to 0

$$\hat{\boldsymbol{X}}_w^r = \sum_{k=1}^{r} f_k(\tilde{\sigma}_k)\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t, \tag{46}$$

where $f_k(\tilde{\sigma}_k) = w_k(\boldsymbol{Y})\tilde{\sigma}_k \mathbb{1}_{\{\tilde{\sigma}_k > c_+\}}$, for all $1 \leq k \leq r$, with $w_k(\boldsymbol{Y})$ given by (31). From Proposition 9 and Proposition 10 it follows that, if $\sigma_k > c^{1/4}$, then, almost surely,

$$\lim_{n\to+\infty} f_k(\tilde{\sigma}_k) = \left(1 - \frac{\sigma_k^2(1+c)+2c}{\sigma_k^2\rho^2(\sigma_k)}\right)\rho(\sigma_k), \text{ for all } 1 \leq k \leq r \leq r^*.$$

Therefore, the data-driven spectral estimator $\hat{\boldsymbol{X}}_w^r$ (46) asymptotically leads to the optimal shrinking rule of singular values given by (43) which has been obtained by minimizing the asymptotic behavior of the SURE.

Note that when $\tau \neq 1/\sqrt{m}$, it suffices to replace the condition $\tilde{\sigma}_k > c_+$ by $\tilde{\sigma}_k > \tau(\sqrt{m}+\sqrt{n})$ in the definition of $\hat{\boldsymbol{X}}_w^r$, which yields the shrinking rule (13) stated in the introduction of this paper.

4. Estimating active sets of singular values in exponential families

In this section, we propose to formulate a new Akaike information criterion (AIC) to select an appropriate set of singular values over which a shrinkage procedure might be applied. To this end, we shall consider the estimator $\tilde{\boldsymbol{X}}^s = \sum_{k\in s}\tilde{\sigma}_k\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t$ defined for a subset $s \subseteq \mathcal{I} = \{1, 2, \ldots, \min(n,m)\}$, and we address the problem of selecting an optimal subset $s^\star$ from the data $\boldsymbol{Y}$.

In the case of Gaussian measurements, the shrinkage estimators that we use in our numerical experiments are of the form $\hat{\boldsymbol{X}}^f = \sum_{k\in s^\star} f_k(\tilde{\sigma}_k)\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t$ where

$$s^\star = \{k \ ; \ \tilde{\sigma}_k > c_+^{n,m}\} \text{ with } c_+^{n,m} = 1 + \sqrt{\frac{n}{m}},$$

for some (possibly data-dependent) shrinkage functions f_k. The set $s^\star$ is based on the knowledge of an approximation $c_+^{n,m}$ of the bulk edge c_+. Thanks to Proposition 9, the bulk edge c_+ is interpreted as the threshold which allows to distinguish the locations of significant singular values in the data from those due to the presence of additive noise. Interestingly, the following result shows that the active set $s^\star$ may be interpreted through the prism of model selection using the minimisation of a penalized log-likelihood criterion.

Proposition 13 *Assume that $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ where the entries of $\boldsymbol{W}$ are iid Gaussian variables with zero mean and standard deviation $\tau = 1/\sqrt{m}$. Then, we have*

$$s^* = \arg\min_{s\subseteq\mathcal{I}} m\|\boldsymbol{Y} - \tilde{\boldsymbol{X}}^s\|_F^2 + 2|s|p_{n,m} \text{ with } p_{n,m} = \left(\frac{1}{2}\left(\sqrt{m}+\sqrt{n}\right)^2\right), \tag{47}$$

where $\tilde{\boldsymbol{X}}^s = \sum_{k\in s}\tilde{\sigma}_k\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t$ for $s \in \mathcal{I} = \{1, 2, \ldots, \min(n,m)\}$, and $|s|$ is the cardinal of s.

Proof We remark that $\boldsymbol{Y} - \tilde{\boldsymbol{X}}^s = \sum_{k\notin s}\tilde{\sigma}_k\tilde{\boldsymbol{u}}_k\tilde{\boldsymbol{v}}_k^t$. It results that

$$m\|\boldsymbol{Y} - \tilde{\boldsymbol{X}}^s\|_F^2 + 2|s|p_{n,m} = m\sum_{k\notin s}\tilde{\sigma}_k^2 + 2|s|p_{n,m} = \sum_{k=1}^{n}\begin{cases} m\tilde{\sigma}_k^2 & \text{if } k \notin s \\ 2p_{n,m} & \text{otherwise} \end{cases}. \tag{48}$$

Using that $\sqrt{2p_{n,m}/m} = c_+^{n,m}$, it follows that the set $s^\star = \{k \ ; \ \tilde{\sigma}_k > c_+^{n,m}\}$ is by definition such that $k \in s^\star$ if and only if $2p_{n,m} < m\tilde{\sigma}_k^2$. Therefore, by (48), the criterion $s \mapsto m\|\boldsymbol{Y} - \tilde{\boldsymbol{X}}^s\|_F^2 + 2|s|p_{n,m}$ is minimum at $s = s^\star$ which concludes the proof. ■

In the model $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$, where the entries of $\boldsymbol{W}$ are iid Gaussian variables with zero mean and variance τ^2, it is well known that the degrees of freedom (DOF) of a given estimator $\hat{\boldsymbol{X}}$ is defined as

$$\mathrm{DOF}(\hat{\boldsymbol{X}}) = \frac{1}{\tau^2}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathrm{Cov}(\hat{\boldsymbol{X}}_{ij}, \boldsymbol{Y}_{ij}) = \frac{1}{\tau^2}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}[\hat{\boldsymbol{X}}_{ij}\boldsymbol{W}_{ij}].$$

The DOF is widely used in statistics to define various criteria for model selection among a collection of estimators (see e.g. Bradley (2004)). In low rank matrix denoising, the following proposition shows that it is possible to derive the asymptotic behavior of the DOF of spectral estimators.

Proposition 14 *Assume that $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ is a random matrix sampled from the Gaussian spiked population model with $\tau = \frac{1}{\sqrt{m}}$ and $\boldsymbol{X} = \sum_{k=1}^{r^*}\sigma_k\boldsymbol{u}_k\boldsymbol{v}_k^t$. Let $\hat{\boldsymbol{X}}^f$ be a spectral shrinker of order $r \leq r^*$ that collapses the bulk to 0, such that each function f_k, for $1 \leq k \leq r$, is continuously differentiable on $]c_+, +\infty[$. Moreover, assume that $\sigma_k > c^{1/4}$ for all $1 \leq k \leq r$. Then, one has that, almost surely,*

$$\lim_{n\to+\infty}\frac{1}{m}\mathrm{DOF}(\hat{\boldsymbol{X}}^f) = \sum_{k=1}^{r}\frac{f_k(\rho(\sigma_k))}{\rho(\sigma_k)}\left(1 + c + \frac{2c}{\sigma_k^2}\right).$$

Proof Thanks to the derivation of the SURE in Stein (1981) and formula (24) on the divergence of spectral estimators, one has that

$$\mathrm{DOF}(\hat{\boldsymbol{X}}^f) = \mathbb{E}\left[\mathrm{div}\,\hat{\boldsymbol{X}}^f\right] = \mathbb{E}\left[|m-n|\sum_{k=1}^{r}\frac{f_k(\tilde{\sigma}_k)}{\tilde{\sigma}_k} + \sum_{k=1}^{r}f_k'(\tilde{\sigma}_k) + 2\sum_{k=1}^{r}f_k(\tilde{\sigma}_k)\sum_{\ell=1;\ell\neq k}^{n}\frac{\tilde{\sigma}_k}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2}\right].$$

By Proposition 9, Proposition 10, and our assumptions on the f_k's, one has that, almost surely,

$$\lim_{n\to+\infty}\frac{1}{m}\,\mathrm{div}\,\hat{\boldsymbol{X}}^f = \sum_{k=1}^{r}\frac{f_k(\rho(\sigma_k))}{\rho(\sigma_k)}\left(1 + c + \frac{2c}{\sigma_k^2}\right).$$

which completes the proof. ■

Hence, in the Gaussian spiked population model, by Proposition 14 and using that $\sigma_k^2 > \sqrt{c}$ for all $1 \leq k \leq r$, it follows that if $s \subseteq \{1, \dots, r\}$ then

$$\lim_{n \to +\infty} \frac{1}{m} \mathrm{DOF}(\tilde{\boldsymbol{X}}^s) = |s| \left(1 + c + \frac{2c}{\sigma_k^2}\right) \leq |s| \left(1 + \sqrt{c}\right)^2 = |s| c_+^2. \tag{49}$$

Hence, the quantity $2|s| \left(\frac{1}{2}\left(\sqrt{m} + \sqrt{n}\right)^2\right)$ is asymptotically an upper bound of $\mathrm{DOF}(\tilde{\boldsymbol{X}}^s)$ (when normalized by $1/m$) for any given set $s \subseteq \{1, \dots, r\}$.

Let us now consider the more general case where the entries of $\boldsymbol{Y}$ are sampled from an exponential family. To the best of our knowledge, extending the notion of the bulk edge to non-Gaussian data sampled from an exponential family has not been considered so far in the literature on random matrices and low rank perturbation model. Therefore, except in the Gaussian case, it is far from being trivial to find an appropriate threshold value $\bar{c}$ to define an active set in the form $\bar{s} = \{k \,;\, \tilde{\sigma}_k > \bar{c}\}$.

Nevertheless, to select an appropriate active set of singular values, we introduce the following criterion that is inspired by the previous results on the DOF of the estimator $\tilde{\boldsymbol{X}}^s$ in the Gaussian case and the statistical literature on the well known AIC for model selection (Akaike, 1974).

Definition 15 *The AIC associated to* $\tilde{\boldsymbol{X}}^s = \sum_{k \in s} \tilde{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t$ *is*

$$\mathrm{AIC}(\tilde{\boldsymbol{X}}^s) = -2 \log q(\boldsymbol{Y}; \tilde{\boldsymbol{X}}^s) + 2|s| p_{n,m} \quad \textit{with} \quad p_{n,m} = \frac{1}{2}\left(\sqrt{m} + \sqrt{n}\right)^2 . \tag{50}$$

where $|s|$ *is the cardinal of* s*, and* $q(\boldsymbol{Y}; \tilde{\boldsymbol{X}}^s) = \prod_{i=1}^n \prod_{j=1}^m q(\boldsymbol{Y}_{ij}; \tilde{\boldsymbol{X}}^s_{ij})$ *is the likelihood of the data in the general form* (14) *at the estimated parameters* $\boldsymbol{X}_{ij} = \tilde{\boldsymbol{X}}^s_{ij}$*.*

In the above definition of $\mathrm{AIC}(\tilde{\boldsymbol{X}}^s)$, the quantity $2|s|p_{n,m}$ is an approximation of the degree of freedom of $\tilde{\boldsymbol{X}}^s$, i.e., of the numbers of its free parameters as it is justified by Proposition 14 in the case of Gaussian measurements. The AIC allows us to define an optimal subset of active variables as

$$s^* = \operatorname*{arg\,min}_{s \subseteq \mathcal{I}} \mathrm{AIC}(\tilde{\boldsymbol{X}}^s).$$

For Gaussian measurements, Proposition 13 gives the value of the optimal set s^* in a closed-form.

Following the arguments in Section 2.3, for Gamma or Poisson measurements and for a given subset s, we consider the estimator

$$\tilde{\boldsymbol{X}}^s_\epsilon = \max\left[\sum_{k \in s} \tilde{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t, \varepsilon\right], \tag{51}$$

when $\epsilon > 0$ is an *a priori* value to satisfy the positivity constraint on the entries of an estimator in this setting. However, contrary to the case of Gaussian noise, the search of an optimal subset $s^\star \subset \operatorname*{arg\,min}_{s \subseteq \mathcal{I}} \mathrm{AIC}(\tilde{\boldsymbol{X}}^s_\epsilon)$ becomes a combinatorial problem in this context. In

our numerical experiments, we thus choose to construct an approximation $\tilde{s}$ of $s^\star$ with a greedy search strategy that reads as follows

$$\tilde{s} = \mathcal{I} \setminus \left\{k \in \mathcal{I} \,;\, \mathrm{AIC}(\tilde{\boldsymbol{X}}_\epsilon^{\mathcal{I}\setminus\{k\}}) \leq \mathrm{AIC}(\tilde{\boldsymbol{X}}_\epsilon^{\mathcal{I}})\right\}. \tag{52}$$

For Gaussian measurements, $\tilde{s} = s^\star$ since the optimisation problem (52) becomes separable. In our numerical experiments, we have found that $\tilde{s}$ selects a relevant set of active singular values which separates well the structural content of $\boldsymbol{X}$ while removing most of the noise component. Further details are given in Section 5 below.

For Gaussian noise, the computation of the active set s^* of singular values may also be interpreted as a way to estimate the unknown rank r^* of the signal matrix $\boldsymbol{X}$. In this setting, one has that $s^\star = \{k \,;\, \tilde{\sigma}_k > c_+^{n,m}\}$ which suggests the choice

$$\hat{r} = \max\{k \,;\, \tilde{\sigma}_k > c_+^{n,m}\}, \tag{53}$$

as an estimator of r^*.

There exists an abundant literature of the problem of estimating the rank of an empirical covariance matrix for the purpose of selecting the appropriate number of significant components to be kept in PCA or factor analysis. It is much beyond the scope of this paper to give an overview of this topic. We point to the review in Jolliffe (2002) for a summary of existing methods to determine the number of components in PCA that are grouped into three categories: subjective methods, distribution-based test tools, and computational procedures. For recent contributions in the matrix denoising model (1) with Gaussian noise, we refer to the works in Choi et al. (2014); Gavish and Donoho (2014) and references therein. For example for Gaussian data with know variance $\tau^2 = 1/m$, Eq. (11) in Gavish and Donoho (2014) on optimal hard thresholding of singular values suggest to take

$$\hat{r} = \max\{k \,;\, \tilde{\sigma}_k > \lambda(c)\}, \text{ with } \lambda(c) = \sqrt{2(c+1) + \frac{8c}{(c+1) + \sqrt{c^2 + 14c + 1}}}, \tag{54}$$

as a simple method to estimate the rank. It should be remarked that the problem of estimating the true rank r^* of $\boldsymbol{X}$ in model (1) is somewhat ill-posed as, in the Gaussian spiked population model, Proposition 9 implies that one may only expect to estimate the so-called effective rank $r_{\mathrm{eff}} = \max\{k \,;\, \sigma_k > c^{1/4}\}$ (see e.g. Section II.D in Nadakuditi (2014)).

In our numerical experiments, we shall compare different choices for the active set of singular values of the form $\hat{s} = \{1, \dots, \hat{r}\}$ where $\hat{r}$ is either given by (53), (54), or by the "oracle choices" $\hat{r} = r^*$ and $\hat{r} = r_{\mathrm{eff}}$.

Other methods based on hypothesis testing (Choi et al., 2014) could be used for rank estimation in the Gaussian model (1), but it is beyond the purpose of this paper to give a detailed comparison.

For Poisson or Gamma noise, it is more difficult to interpret the computation of s^* as a way to estimate the rank of $\boldsymbol{X}$ since, in our numerical experiments, we have found that the cardinality of s^* is generally not equal to $\max\{k \,; k \in s^*\}$. Moreover, to the best of our knowledge, there is not so much work on the estimation of the true rank of a noisy matrix beyond the Gaussian case. Therefore, we have not included a numerical comparison with other methods for the choice of the active set of singular values in these two cases.

5. Numerical experiments

In this section, we assess of the performance of data-driven srhinkage rules under various numerical experiments involving Gaussian, Gamma and Poisson measurements.

5.1 The case of a signal matrix of rank one

We consider the simple setting where the rank r^* of the matrix $\boldsymbol{X}$ is known and equal to one meaning that

$$\boldsymbol{X} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^t,$$

where $\boldsymbol{u}_1 \in \mathbb{R}^n$ and $\boldsymbol{v}_1 \in \mathbb{R}^m$ are vectors with unit norm that are fixed in this numerical experiment, and σ_1 is a positive real that we will let varying. We also choose to fix $n = m = 100$, and so to take $c = \frac{n}{m} = 1$ and $c_+ = 2$. For the purpose of sampling data from Gamma and Poisson distribution, we took singular vectors $\boldsymbol{u}_1$ and $\boldsymbol{v}_1$ with positive entries. The i-th entry (resp. j-th entry) of $\boldsymbol{u}_1$ (resp. $\boldsymbol{v}_1$) is chosen to be proportional to $1 - (i/n - 1/2)^2$ (resp. $1 - (j/m - 1/2)^2$). Let $\boldsymbol{Y} = \sum_{k=1}^{\min(n,m)} \tilde{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t$ be an $n \times m$ matrix whose entries are sampled from model (14) and then satisfying $\mathbb{E}[\boldsymbol{Y}] = \boldsymbol{X}$.

Gaussian measurements

We first consider the case of Gaussian measurements, where $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ with $\mathbb{E}[\boldsymbol{W}_{ij}] = 0$, $\mathrm{Var}(\boldsymbol{W}_{ij}) = \tau^2$ with $\tau = \frac{1}{\sqrt{m}}$. In this context, we compare the following spectral shrinkage estimators:

- Rank-1 PCA shrinkage

$$\hat{\boldsymbol{X}}^1 = \tilde{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t,$$

- Rank-1 SURE-driven soft-thresholding

$$\hat{\boldsymbol{X}}^1_{\mathrm{soft}} = \hat{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t \quad \text{with} \quad \hat{\sigma}_1 = (\tilde{\sigma}_1 - \lambda(\boldsymbol{Y}))_+,$$

- Rank-1 asymptotically optimal shrinkage proposed in Nadakuditi (2014) and Gavish and Donoho (2017)

$$\hat{\boldsymbol{X}}^1_* = \hat{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t \quad \text{with} \quad \hat{\sigma}_1 = \sqrt{\tilde{\sigma}_1^2 - 4}\, \mathbb{1}_{\{\tilde{\sigma}_1 > 2\}},$$

- Rank-1 SURE-driven weighted estimator that we have derived in Section 2.3

$$\hat{\boldsymbol{X}}^1_w = \hat{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t \quad \text{with} \quad \hat{\sigma}_1 = \left(1 - \frac{1}{\tilde{\sigma}_1^2}\left(\frac{1}{m} + \frac{2}{m}\sum_{\ell=2}^{n} \frac{\tilde{\sigma}_1^2}{\tilde{\sigma}_1^2 - \tilde{\sigma}_\ell^2}\right)\right)_+ \tilde{\sigma}_1 \mathbb{1}_{\{\tilde{\sigma}_1 > 2\}},$$

where the above formula follows from the results in Section 3.3 using that $c = 1$ and $c_+ = 2$ in these numerical experiments, and where, for the soft-thresholding, the value $\lambda(\boldsymbol{Y}) > 0$ is obtained by a numerical solver in order to minimize the SURE. As a benchmark, we will also consider the oracle estimator $\boldsymbol{X}^1_*$ that performs shrinkage by using the knowledge of the true singular-value σ_1 defined as

$$\boldsymbol{X}^1_* = \hat{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t \quad \text{with} \quad \hat{\sigma}_1 = \sqrt{\rho(\sigma_1)^2 - 4}\, \mathbb{1}_{\{\rho(\sigma_1) > 2\}}$$

which corresponds to the asymptotically optimal shrinking rule (44) as a function of $\rho(\sigma_k)$ in the setting $c = 1$ and $c_+ = 2$. Note that form the formula above $\hat{w}_1 = \hat{\sigma}_1/\tilde{\sigma}_1$ is necessary in the range $[0, 1]$ for all considered estimators.

In Figure 1, we compare the estimated singular-values $\hat{\sigma}_1$ and the estimated weights $\hat{w}_1 = \hat{\sigma}_1/\tilde{\sigma}_1$ as functions of σ_1 for the four aforementionned estimators. Because all estimators are subject to noise variance, we display, for all estimators, the median values and the 80% confidence intervals obtained from $M = 100$ noise realizations. It can be seen that the median curves for the eigenvelues and the weights of $\hat{\boldsymbol{X}}^1_w$ and $\hat{\boldsymbol{X}}^1_*$ coincide (up to variations that are slightly larger for the former) which is in agreement with the asymptotic analysis of shrinkage rules that has been carried out in Section 3.3. Spectral estimator obtained by SURE-driven soft-thresholding also leads to an optimal shrinkage rule.

In Figure 2, for each of the four spectral estimators above, we display for $M = 100$ noise realizations, as functions of σ_1, the following normalized MSE

$$\mathrm{NMSE}(\hat{\boldsymbol{X}}) = \frac{\|\hat{\boldsymbol{X}} - \boldsymbol{X}\|_F^2}{\|\boldsymbol{X}\|_F^2}.$$

The normalized MSE of the estimators $\hat{\boldsymbol{X}}^1_{\mathrm{soft}}, \hat{\boldsymbol{X}}^1_*$ and $\hat{\boldsymbol{X}}^1_w$ are the same for values of σ_1 larger than $c^{1/4} = 1$, and they only differ for values of σ_1 close or below the threshold $c^{1/4} = 1$ (corresponding to values of $\rho(\sigma_1)$ below the bulk edge $c_+ = 2$). More remarkably, above $c^{1/4} = 1$, they offer similar NMSE values to the oracle shrinkage estimator $\boldsymbol{X}^1_*$, not only in terms of median but also in terms of variability, as assessed by the confidence intervals. The performances of the estimator $\hat{\boldsymbol{X}}^1$ (standard PCA) are clearly poorer. These numerical experiments also illustrate that, for finite-dimensional low rank matrix denoising with $r^* = 1$, data-driven spectral estimators obtained by minimizing a SURE criterion achieve performances that are similar to asymptotically optimal shrinkage rules.

Gamma and Poisson distributed measurements

Let us now consider the case where the entries of $\boldsymbol{Y}_{ij} \geq 0$ of the data matrix $\boldsymbol{Y}$ are independently sampled from a Gamma or Poisson distribution with mean $\boldsymbol{X}_{ij} > 0$. To satisfy the constraint that the estimators must be matrices with positive entries, we consider estimators of the form (30). In this context, we compare the following spectral shrinkage estimators, set for $\varepsilon = 10^{-6}$, as:

- Rank-1 PCA shrinkage

$$\hat{\boldsymbol{X}}^1 = \max\left[\tilde{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t, \varepsilon\right],$$

- Rank-1 GSURE/SUKLS/PURE/PUKLA-driven soft-thresholding

$$\hat{\boldsymbol{X}}^1_{\mathrm{soft}} = \max\left[\hat{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t, \varepsilon\right] \quad \text{with} \quad \hat{\sigma}_1 = (\tilde{\sigma}_1 - \lambda(\boldsymbol{Y}))_+,$$

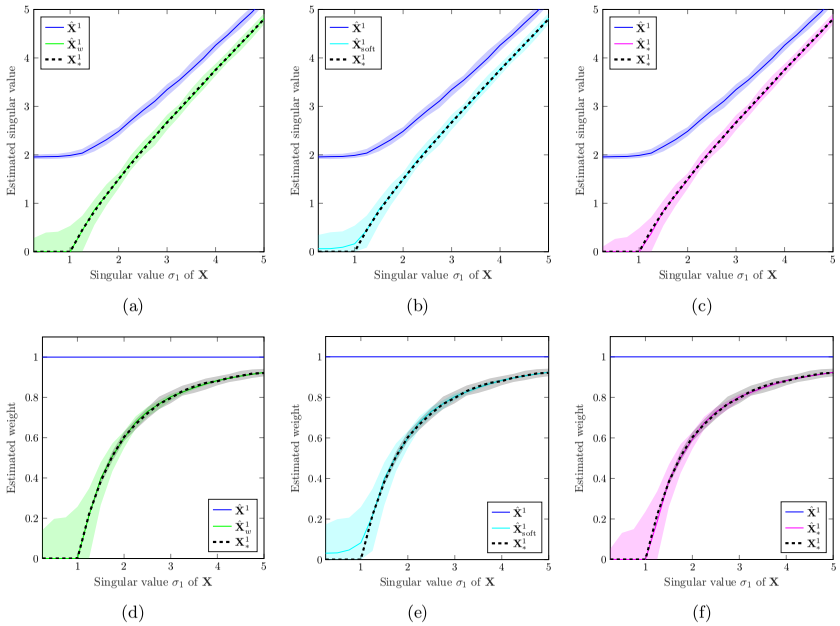


Figure 1: The case of Gaussian measurements with $m = n = 100$. Estimated first singular value $\hat{\sigma}_1$ as a function of the true underlying one σ_1, for (a) our proposed estimator $\hat{\boldsymbol{X}}^1_w$, (b) the soft-thresholding $\hat{\boldsymbol{X}}^1_{\text{soft}}$ and (c) the asymptotical one $\hat{\boldsymbol{X}}^1_*$. All of them are compared to the first singular value $\tilde{\sigma}_1$ of $\boldsymbol{Y}^1$ and the one of the oracle asymptotical estimator $\boldsymbol{X}^1_*$. (c,d,e) Same but for the corresponding weight $\hat{w}_1 = \hat{\sigma}_1/\tilde{\sigma}_1$. Curves have been computed on $M = 100$ noise realizations, only the median and an 80% confidence interval are represented respectively by a stroke and a shadded area of the same color.

- Rank-1 GSURE/SUKLS/PURE/PUKLA-driven weighted estimator

$$\hat{\boldsymbol{X}}^1_w = \max\left[\hat{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t, \varepsilon\right] \quad \text{with} \quad \hat{\sigma}_1 = w_1(\boldsymbol{Y})\tilde{\sigma}_1 \mathbb{1}_{\{1 \in \tilde{s}\}},$$

where $\tilde{s}$ is the approximated active subset as defined in Section 4. For the soft-thresholding, the value $\lambda(\boldsymbol{Y}) > 0$ is obtained by a numerical solver in order to minimize either the GSURE or the SUKLS criterion (in the Gamma case) and either the PURE or the PUKLA criterion (in the Poisson case). The weight $w_1(\boldsymbol{Y}) \in [0, 1]$ is obtained by a numerical solver in order to minimize the GSURE and the PURE, as described in Section B. According to Section 2.3, the weight $w_1(\boldsymbol{Y}) \in [0, 1]$, minimizing the SUKLS criterion, has the following closed-form

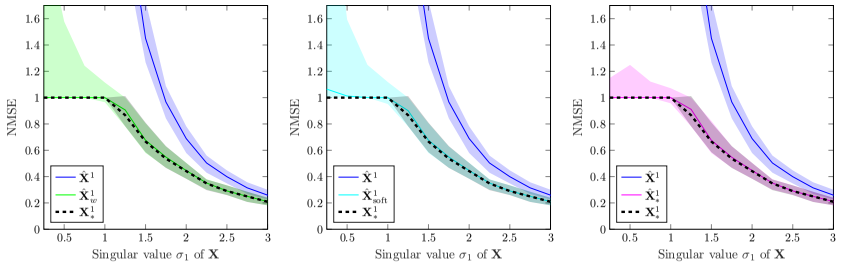


Figure 2: Same as Fig. 1 but for the normalized MSE of the corresponding estimators.

formula

$$w_1(\boldsymbol{Y}) = \min\left[1, \left(\frac{L-1}{Lmn}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\hat{\boldsymbol{X}}^1_{ij}}{\boldsymbol{Y}_{ij}} + \frac{1}{Lmn}\left(1 + 2\sum_{\ell=2}^{n}\frac{\tilde{\sigma}_1^2}{\tilde{\sigma}_1^2 - \tilde{\sigma}_\ell^2}\right)\right)^{-1}\right],$$

and for the PUKLA criterion, we have

$$w_1(\boldsymbol{Y}) = \min\left[1, \frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{Y}_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m}\hat{\boldsymbol{X}}^1_{ij}}\right].$$

To evaluate the performances of these estimators, we perform again a study involving $M = 100$ noise realizations.

In the Gamma case with shape parameter $L = 3$, results are reported in Figure 3 where σ_1 ranges from 0.1 to 5. In the Poisson case, results are reported in Figure 4. To generate data from a Poisson distribution with mean value $\boldsymbol{X} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^t$, we took σ_1 ranging from 25 to 400. In this context, the entries $\boldsymbol{X}_{i,j}$ are in average ranging from 0.25 to 4. When $\sigma_1 = 25$, about 78% of the entries of $\boldsymbol{Y}$ are 0 and 20% are equals to 1 which correspond to an extreme level of noise, while when $\sigma_1 = 400$, the entries of $\boldsymbol{Y}$ concentrate around 4 with a standard deviation of 2 which correspond to a simpler noisy setting.

In these experiments, it can be seen that all the data-dependent spectral estimators achieve comparable results with really small errors in terms of MSE and MKL risks. Their performances are similar to $\hat{\boldsymbol{X}}^1 = \tilde{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t$ meaning that optimizing either SURE-like criteria leads to a spectral estimator closed to correspond to matrix denoising by ordinary PCA. However, unlike the Gamma case, it might be observed in the Poisson case that when reaching a stronger noise level, *i.e*, for small value of σ_1, the NMSE of all estimator increases as the denoising problem becomes more challenging. Nevertheless, only the weight of $\hat{\boldsymbol{X}}^1_w$ driven by PUKLA does not present a drop wich allows reaching a slightly smaller MKLA. In the Gamma case, the noise level being proportional to the signal level, the NMSE/MKLS remain constant for all σ_1.

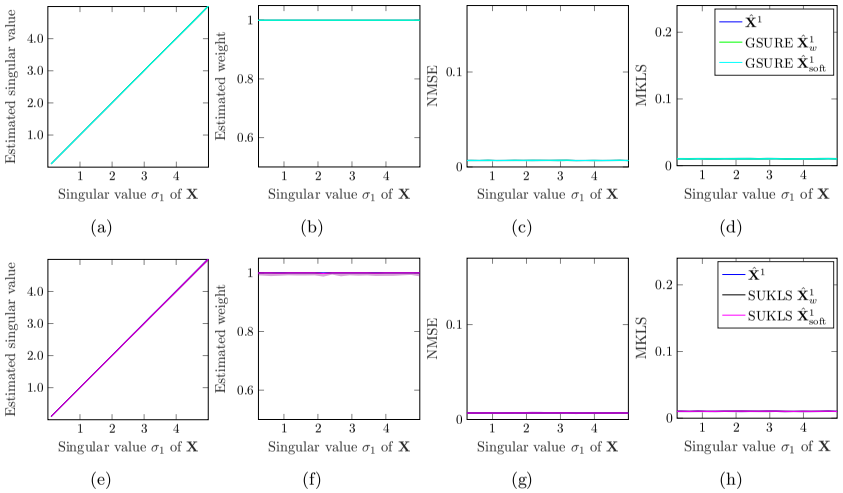


Figure 3: The case of Gamma measurements with $m = n = 100$. (a) Estimated first eigenvalue $\hat{\sigma}_1$ as a function of the true underlying one σ_1 for our proposed estimator $\hat{\boldsymbol{X}}^1_w$ and the soft-thresholding $\hat{\boldsymbol{X}}^1_{\text{soft}}$ when both are guided by the GSURE. Both of them are compared to the first singular value $\tilde{\sigma}_1$ of $\boldsymbol{Y}^1$. Same but for (b) the corresponding weights $\hat{w}_1 = \hat{\sigma}_1/\tilde{\sigma}_1$, (c) the NMSE risk and (d) the MKLS risk. (e-h) Exact same esperiments but when our proposed estimator and the soft-thresholding are both guided by SUKLS. Curves have been computed on $M = 100$ noise realizations, only the median and an 80% confidence interval are represented respectively by a stroke and a shadded area of the same color.

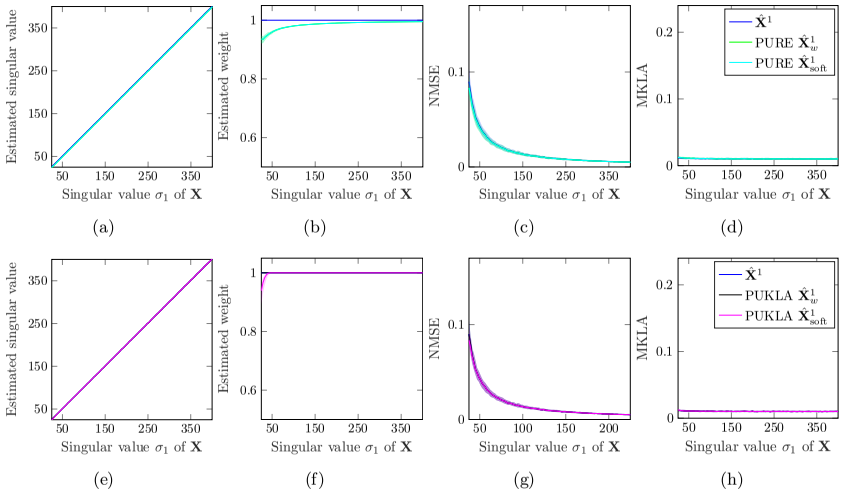


Figure 4: The case of Poisson measurements with $m = n = 100$. (a) Estimated first eigenvalue $\hat{\sigma}_1$ as a function of the true underlying one σ_1 for our proposed estimator $\hat{\boldsymbol{X}}^1_w$ and the soft-thresholding $\hat{\boldsymbol{X}}^1_{\text{soft}}$ when both are guided by the PURE. Both of them are compared to the first singular value $\tilde{\sigma}_1$ of $\boldsymbol{Y}^1$. Same but for (b) the corresponding weights $\hat{w}_1 = \hat{\sigma}_1/\tilde{\sigma}_1$, (c) the NMSE risk and (d) the MKLA risk. (e-h) Exact same esperiments but when our proposed estimator and the soft-thresholding are both guided by PUKLA. Curves have been computed on $M = 100$ noise realizations, only the median and an 80% confidence interval are represented respectively by a stroke and a shadded area of the same color.

Finally, as mentionned by Gavish and Donoho (2017), to use the estimator $\hat{\boldsymbol{X}}^1_*$ in a Gaussian model with homoscedastic variance $\tau^2 \neq \frac{1}{m}$, one may take the estimator $\hat{\boldsymbol{X}}^1_* = \sqrt{m}\tau f^*_1(\tilde{\sigma}_1/(\sqrt{m}\tau))\tilde{\boldsymbol{u}}_1\tilde{\boldsymbol{v}}^t_1$. Hence, provided the variance of the entries of the data matrix $\boldsymbol{Y}$ is known, it is always possible to use a scaled version of the shrinkage rule from Gavish and Donoho (2017) when $\tau^2 \neq \frac{1}{m}$. However, in the setting of Gamma or Poisson noise, the variance of the additive noise varies from one entry to another and depends on the entries of the unknown signal matrix $\boldsymbol{X}$ to recover. For this reason, it is not possible to use a scaled version of the shrinkage rule from Gavish and Donoho (2017) as this would require to use scaling factors depending on the unknown values of the entries of $\boldsymbol{X}$. Therefore, a comparison between our approach and the asymptotically optimal shrinkage proposed in Nadakuditi (2014) and Gavish and Donoho (2017) (for Gaussian noise) is not possible in the case of Gamma or Poisson measurements. Note that for Gamma measurements one has

that $\text{Var}(\boldsymbol{Y}_{ij}) = \boldsymbol{X}^2_{ij}/L$, and, in our numerical experiments, it is assumed that the constant L is known.

A typical example where this assumption is reasonable, is the one of the statistical models of speckle used in coherent imagery, such as, Synthetic Aperture Radar (SAR) and SOund Navigation And Ranging (SONAR) imagery. In such imaging systems, the observed irradiance $\boldsymbol{Y}_{ij}$ of a pixel with indices (i, j) is obtained as the square modulus of a complex signal modeled as being zero-mean circular complex Gaussian distributed (consequence of the Central Limit Theorem), see e.g. Goodman (1976). It follows that $\boldsymbol{Y}_{ij}$ has an exponential distribution[1] with mean $\boldsymbol{X}_{ij}$ corresponding to the underlying irradiance to be estimated. In order to improve the contrast of such images (namely, the signal to noise ratio), an average of L independent and identically distributed images is often performed, and the resulting pixel value becomes Gamma distributed with parameter L (Ulaby and

1. The exponential distribution is a particular instance of the Gamma distribution with parameter $L = 1$

Dobson, 1989). Because the number L of images to be averaged is chosen by the practitioner, the parameter L is absolutely known without uncertainties, and for this reason it does not require to be estimated. Nevertheless the variance $\mathrm{Var}(\boldsymbol{Y}_{ij}) = \boldsymbol{X}_{ij}^2/L$ remains unknown.

While all estimators behave similarly in the rank 1 setting, we will see in the next section that they can significantly differ when the rank is let to be larger than 2.

5.2 The case of a signal matrix of rank larger than two

We now consider the more complex an realistic setting where the rank r^* of the matrix $\boldsymbol{X}$ is unknown and potentially larger than two, i.e.,

$$\boldsymbol{X} = \sum_{k=1}^{r^*} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^t,$$

where $\boldsymbol{u}_k \in \mathbb{R}^n$ and $\boldsymbol{v}_k \in \mathbb{R}^m$ are vectors with unit norm that are fixed in this numerical experiment, and σ_k are positive real values also fixed in this experiment. We also choose to fix $n = 100$ and $m = 200$, while the true rank is $r^* = 9$ as shown by the red curve in Figure 5(i). Again, let $\boldsymbol{Y} = \sum_{k=1}^{\min(n,m)} \tilde{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t$ be an $n \times m$ matrix whose entries are sampled from model (14) and then satisfying $\mathbb{E}[\boldsymbol{Y}] = \boldsymbol{X}$.

Gaussian distributed measurements

We first consider the case of Gaussian measurements, where $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ with $\mathbb{E}[\boldsymbol{W}_{ij}] = 0$, $\mathrm{Var}(\boldsymbol{W}_{ij}) = \tau^2$ with $\tau = 80$. In the following numerical experiments, we study the behavior of the spectral estimator:

- PCA shrinkage

$$\hat{\boldsymbol{X}}^r = \sum_{k=1}^{r} \tilde{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \, \mathbb{1}_{\{k \le \hat{r}\}},$$

- SURE-driven soft-thresholding

$$\hat{\boldsymbol{X}}_{\mathrm{soft}} = \sum_{k=1}^{\min(m,n)} \hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \quad \text{with} \quad \hat{\sigma}_k = (\tilde{\sigma}_k - \lambda(\boldsymbol{Y}))_+,$$

- Asymptotically optimal shrinkage proposed in Nadakuditi (2014) and Gavish and Donoho (2017)

$$\hat{\boldsymbol{X}}_*^r = \sum_{k=1}^{r} \hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \quad \text{with} \quad \hat{\sigma}_k = \frac{1}{\tilde{\sigma}_k} \sqrt{\left(\tilde{\sigma}_k^2 - (c+1)\right)^2 - 4c} \, \mathbb{1}_{\{k \le \hat{r}\}},$$

- SURE-driven weighted estimator that we have derived in Section 2.3

$$\hat{\boldsymbol{X}}_w^r = \sum_{k=1}^{r} \hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \quad \text{with} \quad \hat{\sigma}_k = \left(1 - \frac{1}{\tilde{\sigma}_k^2}\left(\frac{k}{m} + \frac{2}{m}\sum_{\ell=2}^{n} \frac{\tilde{\sigma}_k^2}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2}\right)\right)_+ \tilde{\sigma}_k \mathbb{1}_{\{k \le \hat{r}\}},$$

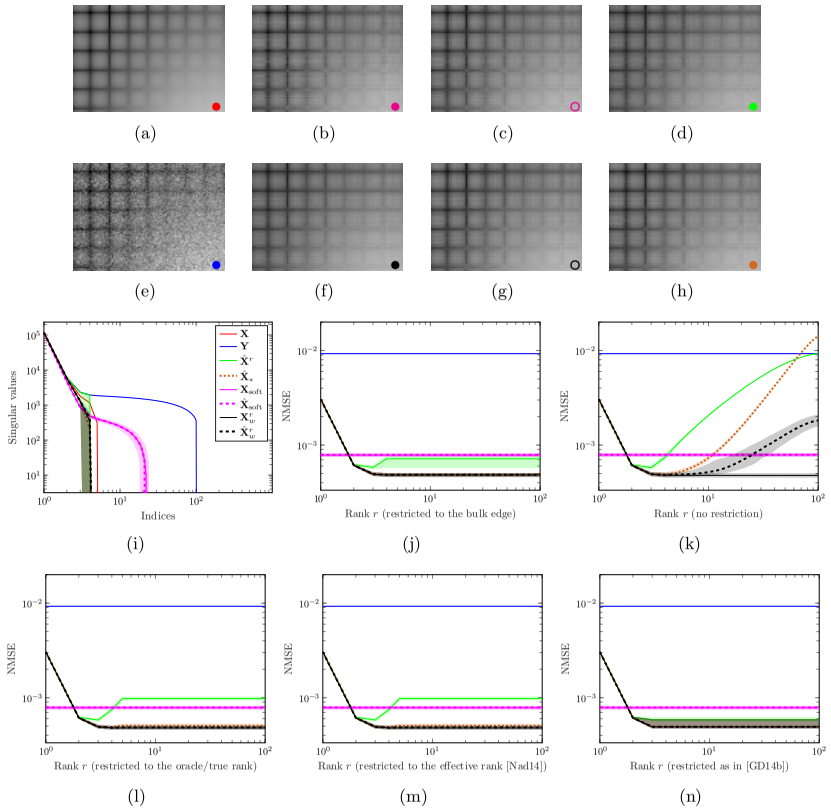


Figure 5: (a) Zoom on a 100×200 noise-free matrix and (e) a single realization of corrupted version by Gaussian noise ($\tau = 80$). (b,c) Oracle soft-thresholding $\boldsymbol{X}_{\mathrm{soft}}$ and data-driven soft-thresholding $\hat{\boldsymbol{X}}_{\mathrm{soft}}$. (d) PCA full rank $\hat{\boldsymbol{X}}^{r_{\max}}$, i.e., $r_{\max} = \min(n, m)$. (f,g,h) Oracle full rank approximation $\boldsymbol{X}_w^{r_{\max}}$, and data-driven full rank estimation $\hat{\boldsymbol{X}}_w^{r_{\max}}$ and $\hat{\boldsymbol{X}}_*^{r_{\max}}$. (i) Their corresponding singular values. (j) NMSE of the various approximations as a function of the rank r. (k) Same but without knowledge the bulk edge, namely $c_+ = 0$. (l,m,n) Same when the active set of singular values is of the form $\hat{s} = \{1, \ldots, \hat{r}\}$ where $\hat{r}$ is either given by $\hat{r} = r^*$ (oracle/true rank), $\hat{r} = r_{\mathrm{eff}}$ (effective rank) or by (54). In all the figures, the solid curves correspond to oracle estimators and the dashed curves correspond to data-driven estimators, obtained over $M = 1,000$ noise realizatrions. The grey areas represent a 80% confidence interval.

where $r \in [1, \min(n,m)]$, and for the soft-thresholding, the value $\lambda(\boldsymbol{Y}) > 0$ is obtained by a numerical solver in order to minimize the SURE. Otherwise specified, we consider $\hat{r} = \max\{k \; ; \; \tilde{\sigma}_k > c_+^{n,m}\}$, *i.e.*, an estimator of the rank using knowledge of the bulk edge $c_+ \approx c_+^{n,m}$, hence, $\mathbb{1}_{\{k \leq \hat{r}\}} = \mathbb{1}_{\{\tilde{\sigma}_k > c_+^{n,m}\}}$. As discussed in Section 4, we compare, in these experiments, the influence of rank estimation by analyzing the performances of the same estimators when either $\hat{r} = r_{\max} = \min(n,m)$ (i.e. without knowledge the bulk edge, namely $c_+ = 0$), $\hat{r} = r^*$ (oracle/true rank), $\hat{r} = r_{\text{eff}}$ (effective rank Nadakuditi (2014)) or by (54) (from hard-thresholding of singular values in Gavish and Donoho (2014)).

In order to assess the quality of SURE as an estimator of the MSE, we also compare the aforementioned approach with their oracle counterparts given by

$$\boldsymbol{X}_{\text{soft}} = \sum_{k=1}^{\min(m,n)} \hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \quad \text{with} \quad \hat{\sigma}_k = (\tilde{\sigma}_k - \lambda^{\text{oracle}}(\boldsymbol{Y}))_+, \quad \text{and}$$

$$\boldsymbol{X}_w^r = \sum_{k=1}^{r} \hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t \quad \text{with} \quad \hat{\sigma}_k = \tilde{\boldsymbol{v}}_k^t \boldsymbol{X} \tilde{\boldsymbol{u}}_k,$$

where $\lambda^{\text{oracle}}(\boldsymbol{Y})$ minimizes the squared error SE (non-expected risk) over the sets and soft-thresholding approximations respectively. Note that $\boldsymbol{X}_w^r$ and $\boldsymbol{X}_{\text{soft}}$ are ideal approximations of $\boldsymbol{X}$ that cannot be used in practice but serve as benchmarks to evaluate the performances of the data-driven estimators $\hat{\boldsymbol{X}}^r$, $\hat{\boldsymbol{X}}_{\text{soft}}$, $\hat{\boldsymbol{X}}_*^r$ and $\hat{\boldsymbol{X}}_w^r$. In order to shed some light on the variance of these estimators, and indirectly on the variance of the SURE, we perform this experiments over $M = 1000$ independent realizations of $\boldsymbol{Y}$.

The results are reported on Figure 5. For an estimator of the rank given either by $\hat{r} = \max\{k \; ; \; \tilde{\sigma}_k > c_+^{n,m}\}$ (knowledge of the bulk edege), $\hat{r} = r^*$ (oracle/true rank), $\hat{r} = r_{\text{eff}}$ (effective rank) or by (54), it can be observed that $\hat{\boldsymbol{X}}_w^r$, $\hat{\boldsymbol{X}}_*^r$ and $\boldsymbol{X}_w^r$ achieve comparable performances for all $r \in [1, \min(m,n)]$ even though the two first do not rely on the unknown matrix $\boldsymbol{X}$. Similarly $\hat{\boldsymbol{X}}_{\text{soft}}$ and $\boldsymbol{X}_{\text{soft}}$ achieve also comparable performances showing again that the SURE accurately estimates the MSE. In terms of error bands for the NMSE, $\hat{\boldsymbol{X}}_w^r$, $\hat{\boldsymbol{X}}_*^r$ and $\boldsymbol{X}_w^r$ outperform $\hat{\boldsymbol{X}}_{\text{soft}}$ and $\boldsymbol{X}_{\text{soft}}$ provided that r is large enough. Moreover, the performance of $\hat{\boldsymbol{X}}_w^r$ plateaus to its optimum when the rank r becomes large. This allows us to choose $r = \min(n,m)$ when we do not have *a priori* on the true or effective rank.

Interestingly, Fig. 5.(k) shows that when the above estimators are used without the knowledge of the bulk edge (i.e. by taking $c_+^{n,m} = 0$ in their computation instead of $c_+^{n,m} = 1 + \sqrt{\frac{n}{m}}$, which corresponds to the choice $\hat{r} = r_{\max} = \min(n,m)$), the performance of $\hat{\boldsymbol{X}}_w^r$ actually decreases when the rank r becomes too large. Indeed, it is clear from Fig. 5.(k), that the the error band of the NMSE of $\hat{\boldsymbol{X}}_w^r$ becomes much larger as the rank r increases. This illustrates that the SURE suffers from estimation variance in the case of over parametrization when r becomes too large, and thus it cannot be used to estimate jointly a too large number of weights. Therefore, the knowledge of an appropriate estimator $\hat{r}$ of the rank (e.g. using the bulk edge) seems to provide a relevant upper bound on the number of weights that can be jointly and robustly estimated with the SURE.

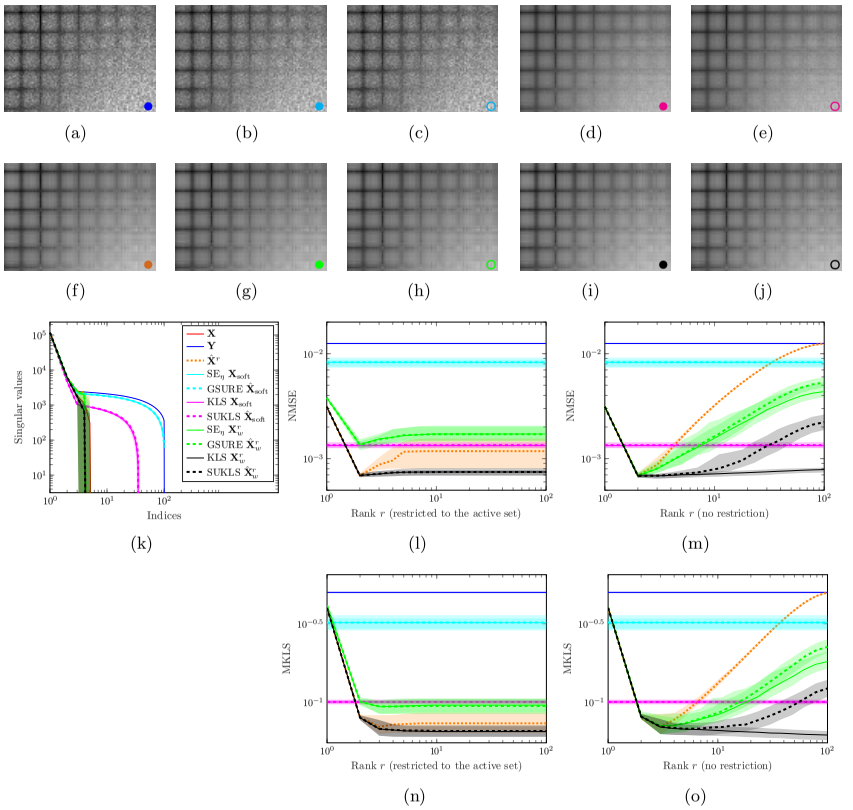


Figure 6: (a) A single realization of corrupted version by Gamma noise ($L = 80$) with zoom on a 100×200 matrix. (b,c,d,e) Oracle soft-thresholding $\boldsymbol{X}_{\text{soft}}$ and data-driven soft-thresholding $\hat{\boldsymbol{X}}_{\text{soft}}$ respectively for SE_η, GSURE, KLS and SUKLS. (f) PCA $\hat{\boldsymbol{X}}^{r_{\max}}$ with full rank approximation i.e. $r_{\max} = \min(n,m)$. (g,h,i,j) Oracle full rank approximation $\boldsymbol{X}_w^{r_{\max}}$, and data-driven full rank estimation $\hat{\boldsymbol{X}}_w^{r_{\max}}$ respectively for SE_η, GSURE, KLS and SUKLS. (k) Their corresponding singular values averaged over $M = 100$ noise realizations. (l,m) NMSE averaged over $M = 100$ noise realizations as a function of the rank r with and without using the active set. (n,o) Same but with respect to MKLS.

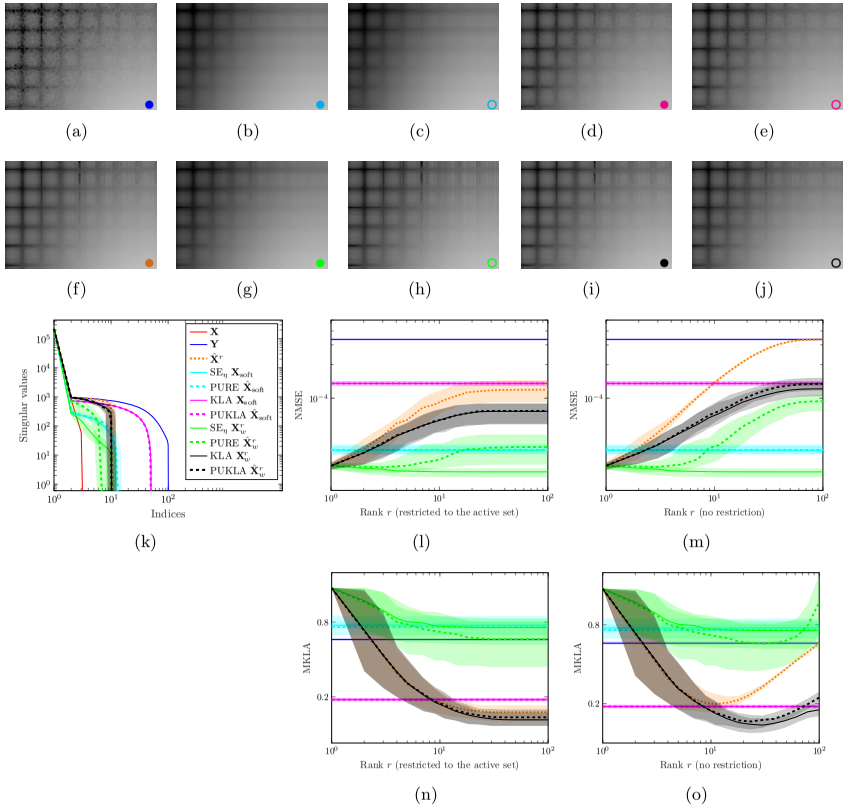


Figure 7: (a) A single realization of corrupted version by Poisson noise with zoom on a 100×200 noise-free matrix (b,c,d,e) Oracle soft-thresholding $\boldsymbol{X}_{\text{soft}}$ and data-driven soft-thresholding $\hat{\boldsymbol{X}}_{\text{soft}}$ respectively for SE, PURE, KLA and PUKLA. (f) PCA $\hat{\boldsymbol{X}}^{r_{\max}}$ with full rank approximation i.e. $r_{\max} = \min(n, m)$. (g,h,i,j) Oracle full rank approximation $\boldsymbol{X}_w^{r_{\max}}$, and data-driven full rank estimation $\hat{\boldsymbol{X}}_w^{r_{\max}}$ respectively for SE, PURE, KLA and PUKLA. (k) Their corresponding singular values averaged over 200 noise realizations. (l,m) NMSE averaged over 200 noise realizations as a function of the rank r with and without using the active set. (n,o) Same but with respect to MKLA. (Matrix entries are displayed in log-scale for better visual assessment.)

Gamma and Poisson measurements

Let us now consider the case where the entries of $\boldsymbol{Y}_{ij} > 0$ of the data matrix $\boldsymbol{Y}$ are independently sampled from a Gamma or Poisson distribution with mean $\boldsymbol{X}_{ij} > 0$. We again consider estimators of the form (30). In this context, we compare the following spectral shrinkage estimators, set for $\varepsilon = 10^{-6}$, as:

- PCA shrinkage

$$\hat{\boldsymbol{X}}^r = \sum_{k=1}^{r} \max\left[\tilde{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t, \varepsilon\right] \quad \text{with} \quad \hat{\sigma}_k = \tilde{\sigma}_k \mathbb{1}_{\{k \in \tilde{s}\}},$$

- GSURE/SUKLS/PURE/SUKLA driven soft-thresholding

$$\hat{\boldsymbol{X}}_{\text{soft}} = \sum_{k=1}^{\min(m,n)} \max\left[\hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t, \varepsilon\right] \quad \text{with} \quad \hat{\sigma}_k = (\tilde{\sigma}_k - \lambda(\boldsymbol{Y}))_+,$$

- GSURE/SUKLS/PURE/SUKLA driven weighted estimator

$$\hat{\boldsymbol{X}}_w^r = \sum_{k=1}^{r} \max\left[\hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t, \varepsilon\right] \quad \text{with} \quad \hat{\sigma}_k = w_k(\boldsymbol{Y}) \tilde{\sigma}_k \mathbb{1}_{\{k \in \tilde{s}\}},$$

where $r \in [1, \min(n, m)]$, and $\tilde{s}$ is the approximated active subset as defined in Section 4. For the soft-thresholding, the value $\lambda(\boldsymbol{Y}) > 0$ is obtained by a numerical solver in order to minimize either the GSURE or the SUKLS criterion (in the Gamma case) and either the PURE or the PUKLA criterion (in the Poisson case). As shown in Section 2.3, in the case of Gamma (resp. Poisson) measurements, the value of $w_k(\boldsymbol{Y})$ for $k \in \tilde{s}$ which minimizes the GSURE (resp. PURE) or the SUKLS (resp. PUKLA), cannot be obtained in closed form. As an alternative, we adopt a greedy one-dimensional optimization strategy starting from the matrix $\tilde{\sigma}_1 \tilde{\boldsymbol{u}}_1 \tilde{\boldsymbol{v}}_1^t$ and next updating the weights w_ℓ sequentially by starting $\ell = 1$ to $\ell = \min(n, m)$, with the constraint that, for all $\ell \notin \tilde{s}$, the weight w_ℓ is set to zero.

To this end, we resort to one-dimensional optimization techniques in the interval $[0, 1]$ using Matlab's command `fminbnd`. This strategy is used for GSURE, SUKLS, PURE and PUKLA by evaluating them as described in Section B. As in the Gaussian setting, we compare this spectral estimators with their oracle counterparts given by

$$\boldsymbol{X}_{\text{soft}} = \sum_{k=1}^{\min(m,n)} \max\left[\hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t, \varepsilon\right] \quad \text{with} \quad \hat{\sigma}_k = (\tilde{\sigma}_k - \lambda^{\text{oracle}}(\boldsymbol{Y}))_+, \quad \text{and}$$

$$\boldsymbol{X}_w^r = \sum_{k=1}^{r} \max\left[\hat{\sigma}_k \tilde{\boldsymbol{u}}_k \tilde{\boldsymbol{v}}_k^t, \varepsilon\right] \quad \text{with} \quad \hat{\sigma}_k = w_k^{\text{oracle}}(\boldsymbol{Y}) \tilde{\sigma}_k \mathbb{1}_{\{k \in \tilde{s}\}}.$$

where $w_k^{\text{oracle}}(\boldsymbol{X})$ and $\lambda^{\text{oracle}}(\boldsymbol{Y})$ minimizes one of the objective SE_η, KLS, SE or KLA (non-expected risks) over the set of matrices sharing with $\boldsymbol{Y}$ the same r first left and right singular vectors, and soft-thresholding approximations respectively. Note again that $\boldsymbol{X}_w^r$ and $\boldsymbol{X}_{\text{soft}}$

are ideal approximations of $\boldsymbol{X}$ that cannot be used in practice but serve as benchmarks to evaluate the performances of the data-driven estimators $\hat{\boldsymbol{X}}_w^r$ and $\hat{\boldsymbol{X}}_{\text{soft}}$.

The results for the Gamma noise are reported on Figure 6. As in the Gaussian setting, it can be observed that $\hat{\boldsymbol{X}}_w^r$ and $\boldsymbol{X}_w^r$ achieve comparable performances, as well as $\hat{\boldsymbol{X}}_{\text{soft}}$ and $\boldsymbol{X}_{\text{soft}}$ showing that the GSURE (resp. SUKLS) accurately estimates the MSE_η (resp. KLS). Visual inspection of the restored matrices tends to show that the estimators driven by MSE_η or GSURE produce less relevant results compared to KLS or SUKLS, as confirmed by the curves of NMSE and MKLS. Performance in terms of NMSE also illustrates that minimizers of SE_η do not coincides with those of SE. As in the Gaussian setting, $\hat{\boldsymbol{X}}_w^r$ and $\boldsymbol{X}^r$ outperform $\hat{\boldsymbol{X}}_{\text{soft}}$, $\boldsymbol{X}_{\text{soft}}$ and standard PCA $\hat{\boldsymbol{X}}^r$ provided that r is large enough. Moreover, the performance of $\hat{\boldsymbol{X}}_w^r$ obtained with KL objectives plateaus to its optimum when the rank r becomes large. Again, this allows us to choose $r = \min(n, m)$ when we do not have *a priori* on the true rank $r^\star$.

The results for the Poisson noise are reported on Figure 7. The conclusions are similar to the Gaussian and Gamma cases. Obviously, the NMSE is smaller for approximations that minimizes SE (or PURE) than for those minimizing KLA (or PUKLA). However, visual inspection of the obtained matrices tends to demonstrate that minimizing such objectives might be less relevant than minimizing KL objectives. In this setting, the performance of $\hat{\boldsymbol{X}}_w^r$ is on a par with the one of $\hat{\boldsymbol{X}}_{\text{soft}}$ based on PUKLA. In fact, for other choices of matrices $\boldsymbol{X}$, $\hat{\boldsymbol{X}}_w^r$ based on PUKLA might improve, in terms of MKLS, much more on $\hat{\boldsymbol{X}}_{\text{soft}}$, and might improve not as much on $\hat{\boldsymbol{X}}_w^r$ based on PURE. Nevertheless, whatever $\boldsymbol{X}$, we observed that $\hat{\boldsymbol{X}}_w^r$ driven by PUKLA always reaches at least as good performance in terms of MKLS as the best of $\hat{\boldsymbol{X}}_w^r$ driven by SE and $\hat{\boldsymbol{X}}_{\text{soft}}$.

Fig. 6.(m), Fig. 6.(o), Fig. 7.(m) and Fig. 7.(o) show that when the above estimators are used without the active set (i.e., by choosing $\tilde{s} = [1, \min(n, m])$, the performance of $\hat{\boldsymbol{X}}_w^r$ actually decreases when the rank r becomes too large. As in the Gaussian setting, this can be explained by the fact that the GSURE, SUKLS, PURE and PUKLA suffer from estimation variance in the case of over parametrization, hence, they cannot be used to estimate jointly a too large number of weights. The active set $\tilde{s}$ (in the same manner as the bulk edge) seems to provide a relevant selection of the weights that can be jointly and robustly estimated in a data driven way.

5.3 Signal matrix with equal singular values and increasing rank

We propose now to highlight potential limitations of our approach in the situation where the rank r^* of the matrix $\boldsymbol{X} = \sum_{k=1}^{r^*} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^t$ is let growing and all positive singular values σ_k of $\boldsymbol{X}$ are equal, namely

$$\boldsymbol{Y} = \sum_{k=1}^{r^*} \sigma_k \boldsymbol{u}_k \boldsymbol{v}_k^t + \boldsymbol{W} \quad \text{with} \quad \sigma_k = \gamma c_{n,m}^{1/4} \text{ for all } 1 \leq k \leq r^*, \tag{55}$$

where $\boldsymbol{u}_k \in \mathbb{R}^n$ and $\boldsymbol{v}_k \in \mathbb{R}^m$ are vectors with unit norm that are fixed, $c_{n,m} = \frac{n}{m}$ and $\boldsymbol{W}$ is centered random matrix whose entries are iid Gaussian variables with variance $\tau^2 = 1/m$. We again choose to fix $n = 100$ and $m = 200$, while the true rank is r^* let growing from 1 to $\min(n, m)$ in the following numerical experiments. The constant γ is chosen to be larger than 1. Hence, eq. (55) corresponds to the Gaussian spiked population model in the setting

where all positive singular values are equal and larger than the threshold $c_{n,m}^{1/4}$. The choice $\sigma_k = \gamma c_{n,m}^{1/4}$ with $\gamma > 1$ is motivated by the results from Proposition 9.

For a given value of the true rank r^*, we performed experiments involving $M = 1000$ realizations from model (55) to compare the NMSE of the estimators by oracle soft-thresholding $\boldsymbol{X}_{\text{soft}}$, data-driven soft-thresholding $\hat{\boldsymbol{X}}_{\text{soft}}$, PCA full rank $\hat{\boldsymbol{X}}^{r_{\max}}$ i.e. $r_{\max} = \min(n, m)$, oracle full rank approximation $\boldsymbol{X}_w^{r_{\max}}$, and data-driven full rank estimation $\hat{\boldsymbol{X}}_w^{r_{\max}}$ and $\hat{\boldsymbol{X}}_*^{r_{\max}}$. All these estimators have been introduced in Section 5.2.

In Figure 8, we report the results of numerical experiments by displaying errors bars of the NMSE of these estimators as functions of the true rank r^*. For low values of the true rank ($r^* \leq 20$), the data-driven estimators $\hat{\boldsymbol{X}}_w^{r_{\max}}$ (our approach) and $\hat{\boldsymbol{X}}_*^{r_{\max}}$ (shrinkage rule from Gavish and Donoho (2017)) achieve the best performances that are similar in term of median value of the NMSE. However, our approach has some limitations with respect to the performances of the estimator from Gavish and Donoho (2017) or data-driven soft-thresholding (Candès et al., 2013) in the setting where the signal matrix has equal positive singular values and when its rank is increasing. Moreover, the error bands of the NMSE for our approach becomes significantly larger than those of the other data-driven estimators when the true rank r^* increases. This illustrates that SURE minimization may lead to estimators with a high variance in the case of over parametrization, that is, when there exists a large number of significant and close singular values in the signal matrix.

5.4 Influence of the dimension and the signal-to-noise ratio

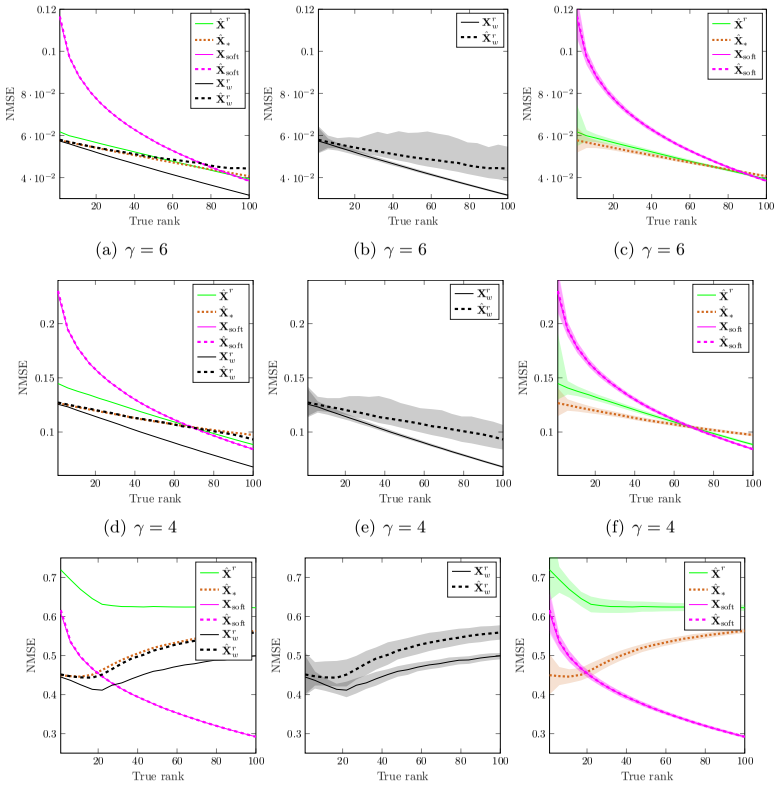
In Section 5.3, we used simulated data consisting of a signal matrix with equal positive singular values and an increasing rank. In such a setting , it is likely that the empirical weights $w_k(\boldsymbol{Y})$, used in our approach, will have a high variance due to the term $\sum_{\ell=1;\ell\neq k}^{\min(n,m)} \frac{\tilde{\sigma}_k^2}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2}$ in their expression (13). However, the numerical experiments carried out in Section 5.3 correspond to a very specific configuration of the signal matrix (with many equal singular values and a high rank) which is not likely to be encountered with real data.

To conclude these numerical experiments, we finally analyze the influence of the dimension of the data and the signal-to-noise ratio on the performances of our approach and the estimator from Gavish and Donoho (2017) in a more realistic setting (with Gaussian noise). These two estimators are the ones giving the best results, and it is thus of interest to compare them with further experiments.

We use real and square signal matrices $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ having a relatively fast decay of their singular values, see Figure 9 and Figure 10. We choose to re-size them to let n varying from 20 to 250, and we define the root of the signal-to-noise ratio (RSNR) as

$$\text{RSNR} = \frac{\sqrt{\frac{1}{n^2}\sum_{i,j=1}^n (\boldsymbol{X}_{ij} - \bar{\boldsymbol{X}})^2}}{\tau} \quad \text{with} \quad \bar{\boldsymbol{X}} = \frac{1}{n^2}\sum_{i,j=1}^n \boldsymbol{X}_{ij}.$$

For each value of n and RSNR (ranging from 5 to 10), we performed experiments involving $M = 400$ realizations from model (55) to compare the NMSE of the estimators by data-driven full rank estimation $\hat{\boldsymbol{X}}_w^{r_{\max}}$ (our approach) and $\hat{\boldsymbol{X}}_*^{r_{\max}}$ (shrinkage rule from Gavish and Donoho (2017)) with $r_{\max} = n$. In Figure 9 and Figure 10, we report the results of these numerical experiments by displaying errors bars of the NMSE of these estimators as

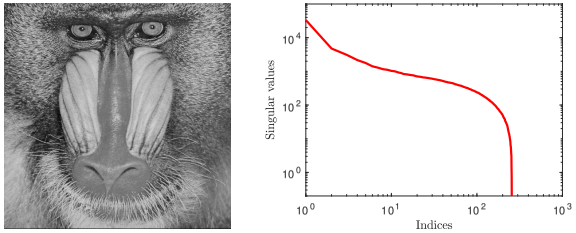


(a) $\gamma = 6$ (b) $\gamma = 6$ (c) $\gamma = 6$

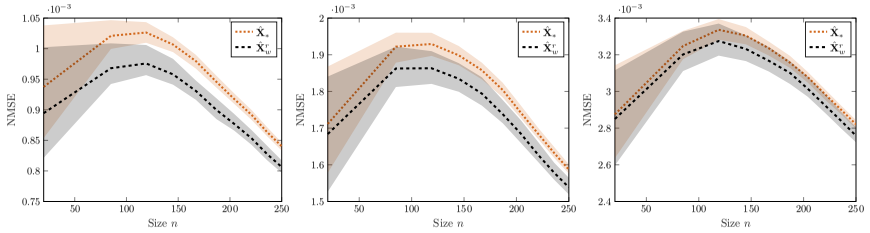
(d) $\gamma = 4$ (e) $\gamma = 4$ (f) $\gamma = 4$

(g) $\gamma = 2$ (h) $\gamma = 2$ (i) $\gamma = 2$

Figure 8: Comparison of NMSE as a function of the true rank r^* in model (55) for different values of γ for the estimator by oracle soft-thresholding $\boldsymbol{X}_{\text{soft}}$, data-driven soft-thresholding $\hat{\boldsymbol{X}}_{\text{soft}}$, PCA full rank $\hat{\boldsymbol{X}}^{r_{\max}}$ i.e. $r_{\max} = \min(n, m)$, oracle full rank approximation $\boldsymbol{X}_w^{r_{\max}}$, and data-driven full rank estimation $\hat{\boldsymbol{X}}_w^{r_{\max}}$ and $\hat{\boldsymbol{X}}_*^{r_{\max}}$. The active set set of singular values is of the form $\hat{s} = \{1, \dots, \hat{r}\}$ where $\hat{r} = \max\{k \ ; \ \tilde{\sigma}_k > c_+^{n,m}\}$ is an estimator of the rank using knowledge of the bulk edge $c_+ \approx c_+^{n,m}$ (a), (d), (g) Median value of the NMSE of the various estimators over $M = 1000$ Gaussian noise realizations in model (55) as a function of the true rank r^*. (b), (c), (e), (f), (h), (i) The grey areas represent error bands of the NMSE of data-driven and oracle estimators.



(a) Signal matrix $\boldsymbol{X}$ (b) Singular values of $\boldsymbol{X}$



(c) RSNR = 10 (d) RSNR = 7 (e) RSNR = 5

Figure 9: Comparison of NMSE as a function of the dimension n in model (55) with a square matrix $\boldsymbol{X}$ for different values of RSNR for the estimator $\hat{\boldsymbol{X}}_w^{r_{\max}}$ and $\hat{\boldsymbol{X}}_*^{r_{\max}}$ with $r_{\max} = n$. The active set set of singular values is of the form $\hat{s} = \{1, \dots, \hat{r}\}$ where $\hat{r} = \max\{k \ ; \ \tilde{\sigma}_k > c_+^{n,m}\}$ is an estimator of the rank using knowledge of the bulk edge $c_+ \approx c_+^{n,m}$. (a) Signal matrix of size 250×250, (b) Decay of the singular values of $\boldsymbol{X}$ in log-log scale, (c), (d), (e) Median value of the NMSE of $\hat{\boldsymbol{X}}_w^{r_{\max}}$ and $\hat{\boldsymbol{X}}_*^{r_{\max}}$ over $M = 400$ Gaussian noise realizations in model (55) as a function of the dimension n. The orange and grey areas represent error bands of the NMSE of these two estimators.

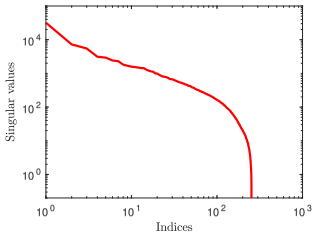
functions of the dimension n. It can be seen that our approach dominates numerically the estimator from Gavish and Donoho (2017) (for all values of n and RSNR) in settings that are more likely to be encountered in practice than the simulated data used in Section 5.3.

Acknowledgments

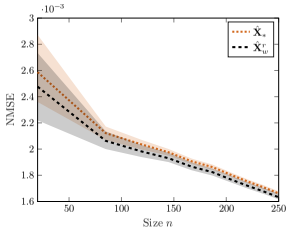
We are very much indebted to the referees and the Associate Editor for their constructive criticism, comments and remarks that resulted in a major revision of the original manuscript.



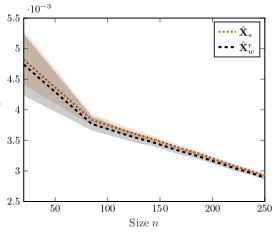
(a) Signal matrix $\boldsymbol{X}$



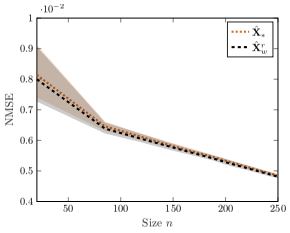
(b) Singular values of $\boldsymbol{X}$



(c) RSNR = 10



(d) RSNR = 7



(e) RSNR = 5

Figure 10: Same as Fig. 9 with another signal matrix $\boldsymbol{X}$.

Appendix A. Proof of the main results

A.1 Proof of Proposition 10

Let us first introduce some notation and definitions to be used in the proof. For all $1 \leq \ell \leq n$, let $\tilde{\lambda}_\ell$ be the eigenvalues of $\boldsymbol{Y}\boldsymbol{Y}^t$ namely $\tilde{\lambda}_\ell = \tilde{\sigma}_\ell^2$. For a fixed $1 \leq k \leq r^*$ such that $\sigma_k > c^{1/4}$, let us introduce the complex-valued function g_k defined by

$$g_k(z) = \frac{1}{n} \sum_{\ell=1;\ell\neq k}^{n} \frac{1}{z - \tilde{\lambda}_\ell} \quad \text{for } z \in \mathbb{C} \setminus \mathrm{supp}(\mu_k),$$

where $\mathrm{supp}(\mu_k) = \left\{\tilde{\lambda}_\ell; 1 \leq \ell \leq n,\ \ell \neq k\right\}$ is the support of the random measure $\mu_k = \frac{1}{n}\sum_{\ell=1;\ell\neq k}^{n} \delta_{\tilde{\lambda}_\ell}$ on $\mathbb{R}_+$, where δ_λ denotes the Dirac measure at λ. It is clear that

$$g_k(z) = \int \frac{1}{z-\lambda} d\mu_k(\lambda).$$

The main difficulty in the proof is to show that, almost surely,

$$\lim_{n\to+\infty} g_k(\tilde{\sigma}_k^2) = \frac{1}{\rho^2(\sigma_k)}\left(1 + \frac{1}{\sigma_k^2}\right),$$

which is the purpose of what follows.

For a matrix $A \in \mathbb{R}^{n\times m}$ (with $n \leq m$), we denote its singular values by $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_n(A) \geq 0$. Hence, one has that $\tilde{\sigma}_\ell = \sigma_\ell(\boldsymbol{Y})$ for all $1 \leq \ell \leq n$. Now, we recall that $\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{W}$ where $\boldsymbol{X}$ is a fixed matrix of rank r^* and $\boldsymbol{W}$ is a random matrix with iid entries sampled from a Gaussian distribution with zero mean and variance $\frac{1}{m}$. The first step in the proof is to show that the random measure μ_k behaves asymptotically as the almost sure limit of the empirical spectral measure $\mu_{\boldsymbol{W}\boldsymbol{W}^t}$ of the Wishart matrix $\boldsymbol{W}\boldsymbol{W}^t$. By definition, the eigenvalues of $\boldsymbol{W}\boldsymbol{W}^t$ are $\lambda_\ell(\boldsymbol{W}) = \sigma_\ell^2(\boldsymbol{W})$ for all $1 \leq \ell \leq n$ and $\mu_{\boldsymbol{W}\boldsymbol{W}^t}$ is thus defined as

$$\mu_{\boldsymbol{W}\boldsymbol{W}^t} = \frac{1}{n}\sum_{\ell=1}^{n} \delta_{\lambda_\ell(\boldsymbol{W})}.$$

It is well know (see e.g. Theorem 3.6 in Bai and Silverstein (2010)) that, once $m = m_n \geq n$ and $\lim_{n\to+\infty} \frac{n}{m} = c$ with $0 < c \leq 1$, then, almost surely, the empirical spectral measure $\mu_{\boldsymbol{W}\boldsymbol{W}^t}$ converges weakly to the so-called Marchenko-Pastur distribution μ_{MP} which is deterministic and has the following density $\frac{d\mu_{MP}(\lambda)}{d\lambda} = \frac{1}{2\pi c\lambda}\sqrt{(c_+^2 - \lambda)(\lambda - c_-^2)}\ \mathbf{1}_{[c_-^2, c_+^2]}(\lambda)$. We recall that such a convergence can also be characterized through the so-called Cauchy or Stieltjes transform which is defined for any probability measure μ on $\mathbb{R}$ as

$$\forall z \in \mathbb{C} \text{ outside the support of } \mu, \quad g_\mu(z) = \int \frac{1}{z-\lambda} d\mu(\lambda).$$

By eq. (3.3.2) in Bai and Silverstein (2010), one obtains that, almost surely,

$$\lim_{n\to\infty} \int \frac{1}{z-\lambda} d\mu_{\boldsymbol{W}\boldsymbol{W}^t}(\lambda) = g_{MP}(z) \text{ for any } z \in \mathbb{C} \setminus \mathbb{R}, \tag{56}$$

where g_{MP} is the Cauchy transform of μ_{MP} and

$$g_{MP}(z) = \int \frac{1}{z-\lambda} d\mu_{MP}(\lambda) = \frac{z - (1-c) - \sqrt{(z-(c+1))^2 - 4c}}{2cz} \quad \text{for all } z \in \mathbb{C} \setminus [c_-^2, c_+^2].$$

Moreover, by Proposition 6 in Pastur and Lejay (2003), the convergence (56) is uniform over any compact subset of $\mathbb{C} \setminus \mathbb{R}$.

Then, it follows from the so-called Weyl's interlacing inequalities (see e.g. Theorem 3.1.2 in Horn and Johnson (1991)) that for all $1 \leq \ell \leq n$

$$\sigma_{\ell+r^*}(\boldsymbol{W}) \leq \sigma_\ell(\boldsymbol{Y}) \leq \sigma_{\ell-r^*}(\boldsymbol{W}), \tag{57}$$

with the convention that $\sigma_k(\boldsymbol{W}) = -\infty$ if $k > n$ and $\sigma_k(\boldsymbol{W}) = +\infty$ if $k \leq 0$. Thanks to the results that have been recalled above on the asymptotic properties of $\mu_{\boldsymbol{W}\boldsymbol{W}^t}$, one may use inequalities (57) to prove that, almost surely, the random measure μ_k converges weakly to the Marchenko-Pastur distribution μ_{MP}. Under the assumptions of Proposition 10 and using Proposition 9, it can be shown that there exists $\eta_k > 0$ such that, almost surely and for all sufficiently large n

$$\tilde{\lambda}_\ell \notin K_k := [\rho^2(\sigma_k) - \eta_k, \rho^2(\sigma_k) + \eta_k]$$

for any $1 \leq \ell \leq n$ with $\ell \neq k$. Now, recall that the support $\mathrm{supp}(\mu_k)$ of the random measure μ_k is $\left\{\tilde{\lambda}_\ell; 1 \leq \ell \leq n,\ \ell \neq k\right\}$, and that $\mathrm{supp}(\mu_{\mathrm{MP}}) = [\mathrm{c}_-^2, \mathrm{c}_+^2]$. Hence, for all sufficiently large n, one has that

$$\mathrm{supp}(\mu_k) \cap \mathrm{K_k} = \emptyset \quad \text{and} \quad \mathrm{supp}(\mu_{\mathrm{MP}}) \cap \mathrm{K_k} = \emptyset.$$

Therefore, thanks to the weak convergence of μ_k to μ_{MP} and using Ascoli's Theorem, one may prove that

$$\lim_{n\to\infty} \sup_{z\in K_k} |g_k(z) - g_{MP}(z)| = 0 \text{ almost surely.} \tag{58}$$

Thanks to our assumptions, one has that, almost surely, $\lim_{n\to+\infty} \tilde{\sigma}_k^2 = \rho^2(\sigma_k)$ by Proposition 9. Hence, almost surely and for all sufficiently large n, one has that $\tilde{\sigma}_k^2 \in K_k$ and so

$$|g_k(\tilde{\sigma}_k^2) - g_{MP}(\rho^2(\sigma_k))| \leq \sup_{z\in K_k} |g_k(z) - g_{MP}(z)| + |g_{MP}(\tilde{\sigma}_k^2) - g_{MP}(\rho^2(\sigma_k))|.$$

Therefore, using the uniform convergence (58) of g_k to g_{MP} and the continuity of g_{MP} at $z = \rho^2(\sigma_k)$, one obtains that, almost surely,

$$\lim_{n\to+\infty} g_k(\tilde{\sigma}_k^2) = g_{MP}(\rho^2(\sigma_k)) = \frac{1}{\rho^2(\sigma_k)} \times \frac{\rho^2(\sigma_k) - 1 + c - \sqrt{(\rho^2(\sigma_k) - (c+1))^2 - 4c}}{2c}.$$

Since $g_k(\tilde{\sigma}_k^2) = \frac{1}{n}\sum_{\ell=1;\ell\neq k}^n \frac{1}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2}$, using the above equation and relation (32), it follows immediately that $g_{MP}(\rho^2(\sigma_k)) = \frac{1}{\rho^2(\sigma_k)}\left(1 + \frac{1}{\sigma_k^2}\right)$ so that, almost surely,

$$\lim_{n\to+\infty} \frac{1}{n} \sum_{\ell=1;\ell\neq k}^n \frac{\tilde{\sigma}_k}{\tilde{\sigma}_k^2 - \tilde{\sigma}_\ell^2} = \lim_{n\to+\infty} \tilde{\sigma}_k g_k(\tilde{\sigma}_k^2) = \rho(\sigma_k)\, g_{MP}(\rho^2(\sigma_k)) = \frac{1}{\rho(\sigma_k)}\left(1 + \frac{1}{\sigma_k^2}\right),$$

which completes the proof.

A.2 A technical result to prove SURE-like formulas

We recall the key lemma needed to prove the SURE-like formulas in an exponential family in the continuous case. Similar results have already been formulated in different papers in the literature, see e.g. the review proposed in Deledalle (2017).

Lemma 16 *Let $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$ be a random matrix whose entries $\boldsymbol{Y}_{ij}$ are independently sampled from the continuous exponential family* (15) *in canonical form (that is the distribution of $\boldsymbol{Y}_{ij}$ is absolutely continuous with respect to the Lebesgue measure dy on $\mathbb{R}$). Suppose that the function h is continuously differentiable on $\mathcal{Y} = \mathbb{R}$. Let $1 \leq i \leq n$ and $1 \leq j \leq m$, and denote by $F_{ij} : \mathbb{R}^{n\times m} \to \mathbb{R}$ a continuously differentiable function such that*

$$\mathbb{E}\left[|F_{ij}(\boldsymbol{Y})|\right] < +\infty. \tag{59}$$

Then, the following relation holds

$$\mathbb{E}\left[\boldsymbol{\theta}_{ij} F_{ij}(\boldsymbol{Y})\right] = -\mathbb{E}\left[\frac{h'(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})} F_{ij}(\boldsymbol{Y}) + \frac{\partial F_{ij}(\boldsymbol{Y})}{\partial \boldsymbol{Y}_{ij}}\right].$$

Proof Using the expression (15) of the pdf of the random varibles $\boldsymbol{Y}_{ij}$, one has that

$$\mathbb{E}\left[\boldsymbol{\theta}_{ij} F_{ij}(\boldsymbol{Y})\right] = \int_{\mathbb{R}^{n\times m}} F_{ij}(Y) h(y_{ij}) \boldsymbol{\theta}_{ij} \exp\left(\boldsymbol{\theta}_{ij} y_{ij} - A(\boldsymbol{\theta}_{ij})\right)\, \mathrm{d}y_{ij} \prod_{\substack{1\leq k\leq n\\ 1\leq \ell\leq m\\ (k,\ell)\neq(i,j)}}^{n} p(y_{k\ell}; \boldsymbol{\theta}_{k\ell})\, \mathrm{d}y_{k\ell}.$$

where $Y = (y_{k\ell})_{1\leq k\leq n, 1\leq \ell\leq m}$. Thanks to condition (59), it follows that

$$\int_{\mathbb{R}^{n\times m}} F_{ij}(Y) h(y_{ij}) \exp\left(\boldsymbol{\theta}_{ij} y_{ij} - A(\boldsymbol{\theta}_{ij})\right)\, \mathrm{d}y_{ij} \prod_{\substack{1\leq k\leq n\\ 1\leq \ell\leq m\\ (k,\ell)\neq(i,j)}}^{n} p(y_{k\ell}; \boldsymbol{\theta}_{k\ell})\, \mathrm{d}y_{k\ell} < +\infty. \tag{60}$$

Therefore, given that $\boldsymbol{\theta}_{ij} \exp\left(\boldsymbol{\theta}_{ij} y_{ij} - A(\boldsymbol{\theta}_{ij})\right) = \frac{\partial \exp(\boldsymbol{\theta}_{ij} y_{ij} - A(\boldsymbol{\theta}_{ij}))}{\partial y_{ij}}$, an integration by part and eq. (60) imply that

$$\mathbb{E}\left[\boldsymbol{\theta}_{ij} F_{ij}(\boldsymbol{Y})\right] = -\int_{\mathbb{R}^{n\times m}} \frac{\partial F_{ij}(Y) h(y_{ij})}{\partial y_{ij}} \exp\left(\boldsymbol{\theta}_{ij} y_{ij} - A(\boldsymbol{\theta}_{ij})\right)\, \mathrm{d}y_{ij} \prod_{\substack{1\leq k\leq n\\ 1\leq \ell\leq m\\ (k,\ell)\neq(i,j)}}^{n} p(y_{k\ell}; \boldsymbol{\theta}_{k\ell})\, \mathrm{d}y_{k\ell}.$$

Now, since $\frac{\partial F_{ij}(Y) h(y_{ij})}{\partial y_{ij}} = h'(y_{ij}) F_{ij}(Y) + \frac{\partial F_{ij}(Y)}{\partial y_{ij}} h(y_{ij})$, we finally obtain that

$$\mathbb{E}\left[\boldsymbol{\theta}_{ij} F_{ij}(\boldsymbol{Y})\right] = -\mathbb{E}\left[\frac{h'(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})} F_{ij}(\boldsymbol{Y}) + \frac{\partial F_{ij}(\boldsymbol{Y})}{\partial \boldsymbol{Y}_{ij}}\right],$$

which completes the proof. ∎

A.3 Proof of Proposition 4

We remark that

$$\mathrm{MSE}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \sum_{i=1}^n \sum_{j=1}^m \left(\mathbb{E}\left[|\hat{\boldsymbol{\theta}}_{ij}^f(\boldsymbol{Y})|^2 - 2\boldsymbol{\theta}_{ij}\hat{\boldsymbol{\theta}}_{ij}^f(\boldsymbol{Y})\right] + \boldsymbol{\theta}_{ij}^2\right). \tag{61}$$

Using Lemma 16 with $F_{ij}(\boldsymbol{Y}) = \hat{\boldsymbol{\theta}}_{ij}^f(\boldsymbol{Y})$ and condition (17), it follows that

$$\mathbb{E}\left[\boldsymbol{\theta}_{ij}\hat{\boldsymbol{\theta}}_{ij}^f(\boldsymbol{Y})\right] = \mathbb{E}\left[\frac{h'(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})}\hat{\boldsymbol{\theta}}_{ij}^f(\boldsymbol{Y})\right] + \mathbb{E}\left[\frac{\partial \hat{\boldsymbol{\theta}}_{ij}^f(\boldsymbol{Y})}{\partial \boldsymbol{Y}_{ij}}\right]. \tag{62}$$

Then, by definition (15) of the exponential family, we remark that

$$\mathbb{E}\left[\frac{h''(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})}\right] = \int_{\mathbb{R}} h''(y_{ij}) \exp\left(\boldsymbol{\theta}_{ij} y_{ij} - A(\boldsymbol{\theta}_{ij})\right)\, \mathrm{d}y_{ij}.$$

Hence, using an integration by parts twice, we arrive at

$$\mathbb{E}\left[\frac{h''(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})}\right] = \boldsymbol{\theta}_{ij}^2 \int_{\mathbb{R}} h(y_{ij}) \exp\left(\boldsymbol{\theta}_{ij} y_{ij} - A(\boldsymbol{\theta}_{ij})\right)\, \mathrm{d}y_{ij} = \boldsymbol{\theta}_{ij}^2. \tag{63}$$

To complete the proof, it suffices to insert equalities (62) and (63) into (61).

A.4 Proof of Proposition 6

Thanks to eq. (21), one has that

$$\mathrm{MKLS}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbb{E}\left[\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})) - \boldsymbol{\theta}_{ij}A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})) - A(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\right] + A(\boldsymbol{\theta}_{ij}). \quad (64)$$

Using Lemma 16 with $F_{ij}(\boldsymbol{Y}) = A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))$ and condition (22), it follows that

$$\mathbb{E}\left[\boldsymbol{\theta}_{ij}A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\right] = -\mathbb{E}\left[\frac{h'(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})}A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\right] - \mathbb{E}\left[\frac{\partial\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})}{\partial\boldsymbol{Y}_{ij}}A''(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\right]. \quad (65)$$

Thus, inserting equality (65) into (64) implies that

$$\mathrm{SUKLS}(\hat{\boldsymbol{\theta}}^f) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\left(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}) + \frac{h'(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})}\right)A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})) - A(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\right) + \sum_{i=1}^{n}\sum_{j=1}^{m}A''(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\frac{\partial\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})}{\partial\boldsymbol{Y}_{ij}}$$

is an unbiased estimator of $\mathrm{MKLS}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) - \sum_{i=1}^{n}\sum_{j=1}^{m}A(\boldsymbol{\theta}_{ij})$. Now recall that $f_{ij}(\boldsymbol{Y}) = \eta^{-1}\left(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})\right)$ and that $A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})) = \eta^{-1}\left(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})\right)$ by Assumption 2. Therefore, $\frac{\partial f_{ij}(\boldsymbol{Y})}{\partial\boldsymbol{Y}_{ij}} = A''(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\frac{\partial\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})}{\partial\boldsymbol{Y}_{ij}}$, and thus

$$\mathrm{SUKLS}(\hat{\boldsymbol{\theta}}^f) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\left(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}) + \frac{h'(\boldsymbol{Y}_{ij})}{h(\boldsymbol{Y}_{ij})}\right)A'(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y})) - A(\hat{\boldsymbol{\theta}}^f_{ij}(\boldsymbol{Y}))\right) + \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{\partial f_{ij}(\boldsymbol{Y})}{\partial\boldsymbol{Y}_{ij}},$$

which completes the proof.

A.5 Proof of Proposition 8

Thanks to the expression (27) of the MKLA risk for data sampled from a Poisson distribution, it follows that

$$\mathrm{MKLA}(\hat{\boldsymbol{\theta}}^f, \boldsymbol{\theta}) + \sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{X}_{ij} - \boldsymbol{X}_{ij}\log(\boldsymbol{X}_{ij}) = \sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{E}\left[\hat{\boldsymbol{X}}^f_{ij} - \boldsymbol{X}_{ij}\log\left(\hat{\boldsymbol{X}}^f_{ij}\right)\right]$$

In the case of Poisson data, one has that $\exp(\boldsymbol{\theta}_{ij}) = \boldsymbol{X}_{ij}$ and $\frac{h(\boldsymbol{Y}_{ij}-1)}{h(\boldsymbol{Y}_{ij})} = \boldsymbol{Y}_{ij}$. Therefore, by applying Hudson's Lemma 7 with $F_{ij}(\boldsymbol{Y}) = \log\left(\hat{\boldsymbol{X}}^f_{ij}\right)$, it follows that

$$\mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{X}_{ij}\log\left(\hat{\boldsymbol{X}}^f_{ij}\right)\right] = \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{m}\boldsymbol{Y}_{ij}\log\left(f_{ij}(\boldsymbol{Y} - \boldsymbol{e}_i\boldsymbol{e}_j^t)\right)\right],$$

which completes the proof.

Appendix B. Implementation details

We discuss below an algorithmic approach to find data-driven spectral estimators.

First, we discuss on how to compute data-driven spectral estimators from the expression of risk estimators. For SUKLS in continuous exponential families, and for SURE in the Gaussian case only, eq. (38) and (23) provide respectively a closed-form solution that can be evaluated in linear time $O(nm)$. On the contrary, the computations of GSURE (beyond the Gaussian case), PURE and PUKLA, given respectively in eq. (19), (25) and (28), cannot be evaluated in reasonable time. They rely respectively on the computation of the divergence $\operatorname{div}\hat{\boldsymbol{\theta}}^f(\boldsymbol{Y})$, $\sum\sum\boldsymbol{Y}_{ij}f_{ij}(\boldsymbol{Y} - \boldsymbol{e}_i\boldsymbol{e}_j^t)$ and $\sum\sum\boldsymbol{Y}_{ij}\log\left(f_{ij}(\boldsymbol{Y} - \boldsymbol{e}_i\boldsymbol{e}_j^t)\right)$. Without further assumptions, such quantities requires $O(n^2m^2)$ operations in general. A standard approach for the computation of the divergence, suggested in Girard (1989); Ramani et al. (2008), is to unbiasedly estimate it with Monte-Carlo simulations by sampling the following relation

$$\operatorname{div}\hat{\boldsymbol{\theta}}^f(\boldsymbol{Y}) = \mathbb{E}_{\boldsymbol{\delta}}\left[\operatorname{tr}\left(\boldsymbol{\delta}^t\frac{\partial\hat{\boldsymbol{\theta}}^f(\boldsymbol{Y})}{\partial\boldsymbol{Y}}\boldsymbol{\delta}\right)\right]$$

at random directions $\boldsymbol{\delta} \in \mathbb{R}^{n\times m}$ satisfying $\mathbb{E}[\boldsymbol{\delta}] = 0$, $\mathbb{E}[\boldsymbol{\delta}_i\boldsymbol{\delta}_i] = 1$ and $\mathbb{E}[\boldsymbol{\delta}_i\boldsymbol{\delta}_j] = 0$. Following Deledalle (2017), a similar first order approximation can be used for the other two quantities as

$$\sum\sum\boldsymbol{Y}_{ij}f_{ij}(\boldsymbol{Y} - \boldsymbol{e}_i\boldsymbol{e}_j^t) \approx \sum\sum\boldsymbol{Y}_{ij}\left[f_{ij}(\boldsymbol{Y}) - \boldsymbol{\delta}_{i,j}\left(\frac{\partial f(\boldsymbol{Y})}{\partial\boldsymbol{Y}}\boldsymbol{\delta}\right)_{i,j}\right], \quad \text{and}$$

$$\sum\sum\boldsymbol{Y}_{ij}\log\left(f_{ij}(\boldsymbol{Y} - \boldsymbol{e}_i\boldsymbol{e}_j^t)\right) \approx \sum\sum\boldsymbol{Y}_{ij}\log\left[f_{ij}(\boldsymbol{Y}) - \boldsymbol{\delta}_{i,j}\left(\frac{\partial f(\boldsymbol{Y})}{\partial\boldsymbol{Y}}\boldsymbol{\delta}\right)_{i,j}\right]$$

where the entries of $\boldsymbol{\delta}$ should be chosen Bernoulli distributed with parameter $p = 0.5$. The advantage of these three approximations is that they can be computed in linear time $O(nm)$ by making use of the results of Lewis and Sendov (2001); Sun and Sun (2003); Edelman (2005); Candès et al. (2013); Deledalle et al. (2012) that provide an expression for the directional derivative given by

$$\frac{\partial f(\boldsymbol{Y})}{\partial\boldsymbol{Y}}\boldsymbol{\delta} = \tilde{\boldsymbol{U}}(\mathbf{D} + \boldsymbol{S} + \boldsymbol{A})\tilde{\boldsymbol{V}}^t \quad (66)$$

where $\tilde{\boldsymbol{U}}$ and $\tilde{\boldsymbol{V}}$ are the matrices whose columns are $\tilde{\boldsymbol{u}}_k$ and $\tilde{\boldsymbol{v}}_k$, and $\mathbf{D}$, $\boldsymbol{S}$ and $\boldsymbol{A}$ are $n\times m$ matrices defined, for all $1 \le i \le n$ and $1 \le j \le m$, as

$$\mathbf{D}_{i,j} = \bar{\boldsymbol{\delta}}_{i,j} \times \begin{cases} f_i'(\tilde{\sigma}_i) & \text{if} \quad i = j \\ 0 & \text{otherwise,} \end{cases}$$

$$\boldsymbol{S}_{i,j} = \frac{\bar{\boldsymbol{\delta}}_{i,j} + \bar{\boldsymbol{\delta}}_{j,i}}{2} \times \begin{cases} 0 & \text{if} \quad i = j \\ \frac{f_i(\tilde{\sigma}_i) - f_j(\tilde{\sigma}_j)}{\tilde{\sigma}_i - \tilde{\sigma}_j} & \text{otherwise,} \end{cases}$$

$$\boldsymbol{A}_{i,j} = \frac{\bar{\boldsymbol{\delta}}_{i,j} - \bar{\boldsymbol{\delta}}_{j,i}}{2} \times \begin{cases} 0 & \text{if} \quad i = j \\ \frac{f_i(\tilde{\sigma}_i) + f_j(\tilde{\sigma}_j)}{\tilde{\sigma}_i + \tilde{\sigma}_j} & \text{otherwise,} \end{cases}$$

where $\tilde{\sigma}_k$ and $f_k(\tilde{\sigma}_k)$ are extended to 0 for $k > \min(n, m)$ and $\bar{\boldsymbol{\delta}} = \tilde{\boldsymbol{U}}^t\boldsymbol{\delta}\tilde{\boldsymbol{V}} \in \mathbb{R}^{n\times m}$.

References

H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

O. Alter, P. O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences (PNAS)*, 97(18), august 2000.

G. W. Anderson, A. Guionnet, and O. Zeitouni. *An introduction to random matrices*, volume 118 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2010.

Z. Bai and J. W. Silverstein. *Spectral analysis of large dimensional random matrices*. Springer Series in Statistics. Springer, New York, second edition, 2010.

J. Baik and J. W. Silverstein. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis*, 97(6):1382 – 1408, 2006.

J. A. Bazerque, Gonzalo M., and G. B. Giannakis. *Inference of Poisson count processes using low-rank tensor data*, pages 5989–5993. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, 10 2013.

F. Benaych-Georges and R. R. Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *J. Multivariate Analysis*, 111:120–135, 2012.

E. Bradley. The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association*, pages 99–467, 2004.

L. D. Brown. *Fundamentals of statistical exponential families: with applications in statistical decision theory*. Institute of Mathematical Statistics, 1986.

M. Bydder and J. Du. Noise reduction in multiple-echo data sets using singular value decomposition. *Magn Reson Imaging*, 24(7):849–56, 2006.

D. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

E. J. Candès, C. A. Sing-Long, and J. D. Trzasko. Unbiased risk estimates for singular value thresholding and spectral estimators. *IEEE Trans. Signal Process.*, 61(19):4643–4657, 2013.

Y. Cao and Y. Xie. Poisson matrix recovery and completion. *IEEE Trans. Signal Processing*, 64(6):1609–1620, 2016.

Y. Choi, J. Taylor, and R. Tibshirani. Selecting the number of principal components: Estimation of the true rank of a noisy matrix. *Preprint arXiv:1405.7511*, 2014.

C.-A. Deledalle. Estimation of kullback-leibler losses for noisy recovery problems within the exponential family. *Electronic Journal of Statistics*, 11(2):3141–3164, 2017.

C.-A. Deledalle, S. Vaiter, G. Peyré, J. Fadili, and C. Dossal. Risk estimation for matrix recovery with spectral regularization. In *arXiv:1205.1482*, 2012. Presented at ICML'2012 workshop on Sparsity, Dictionaries and Projections in Machine Learning and Signal Processing, Edinburgh, United Kingdom, 2012.

D. Donoho and M. Gavish. Minimax risk of matrix denoising by singular value thresholding. *Ann. Statist.*, 42(6):2413–2440, 12 2014.

R. B. Dozier and J. W. Silverstein. On the empirical distribution of eigenvalues of large dimensional information-plus-noise-type matrices. *J. Multivariate Anal.*, 98(4):678–694, 2007.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

A. Edelman. Matrix jacobians with wedge products. *MIT Handout*, 2005.

B. Efron and C. Morris. Empirical Bayes on Vector Observations: An Extension of Stein's Method. *Biometrika*, 59(2):335–347, 1972.

B. Efron and C. Morris. Multivariate empirical bayes and estimation of covariance matrices. *Ann. Statist.*, 4(1):22–32, 01 1976.

Y. C. Eldar. Generalized sure for exponential families: Applications to regularization. *IEEE Transactions on Signal Processing*, 57(2):471–481, 2009.

M. Gavish and D. L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Trans. Information Theory*, 60(8):5040–5053, 2014.

M. Gavish and D. L. Donoho. Optimal shrinkage of singular values. *IEEE Trans. Information Theory*, 63(4):2137–2152, 2017.

A Girard. A fast monte-carlo cross-validation procedure for large least squares problems with noisy data. *Numerische Mathematik*, 56(1):1–23, 1989.

J. W. Goodman. Some fundamental properties of speckle. *JOSA*, 66(11):1145–1150, 1976.

S. Haipeng and H. Jianhua. Analysis of call centre arrival data using singular value decomposition: Research articles. *Appl. Stoch. Model. Bus. Ind.*, 21(3):251–263, May 2005.

P. Hall. On Kullback-Leibler loss and density estimation. *Ann. Statist.*, 15(4):1491–1519, 1987.

J. Hannig and T. C. M. Lee. On Poisson signal estimation under Kullback-Leibler discrepancy and squared risk. *J. Statist. Plann. Inference*, 136(3):882–908, 2006.

R. A. Horn and C. R. Johnson. *Topics in matrix analysis*. Cambridge University Press, Cambridge, New York, Melbourne, 1991.

H. M. Hudson. A natural identity for exponential families with applications in multiparameter estimation. *Ann. Statist.*, 6(3):473–484, 05 1978.

I. T. Jolliffe. *Principal component analysis*. Springer Series in Statistics. Springer-Verlag, New York, second edition, 2002.

J. Josse and S. Sardy. Adaptive shrinkage of singular values. *Statistics and Computing*, pages 1–10, 2015.

J. Lafond. Low rank matrix completion with exponential family noise. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 1224–1243, 2015.

F. Lam, S. D. Babacan, J. P. Haldar, N Schuff, and Z.-P. Liang. Denoising diffusion-weighted MR magnitude image sequences using low rank and edge constraints. In *ISBI*, pages 1401–1404. IEEE, 2012.

O. Ledoit and M. Wolf. Nonlinear shrinkage estimation of large-dimensional covariance matrices. *Ann. Statist.*, 40(2):1024–1060, 04 2012.

A.S. Lewis and H.S. Sendov. Twice differentiable spectral functions. *SIAM Journal on Matrix Analysis on Matrix Analysis and Applications*, 23:368–386, 2001.

R. R. Nadakuditi. OptShrink: an algorithm for improved low-rank signal matrix denoising by optimal, data-driven singular value shrinkage. *IEEE Trans. Inform. Theory*, 60(5): 3002–3018, 2014.

H. M. Nguyen, X. Peng, M. N. Do, and Z.-P. Liang. Spatiotemporal denoising of mr spectroscopic imaging data by low-rank approximations. In *ISBI*, pages 857–860. IEEE, 2011.

L. Pastur and A. Lejay. Matrices aléatoires: statistique asymptotique des valeurs propres. In *Séminaire de Probabilités, XXXVI*, volume 1801 of *Lecture Notes in Math.*, pages 135–164. Springer, Berlin, 2003.

S. Ramani, T. Blu, and M. Unser. Monte-Carlo SURE: a black-box optimization of regularization parameters for general denoising algorithms. *IEEE Trans. on Image Processing*, 17(9):1540–1554, 2008.

M. Raphan and E. P. Simoncelli. Learning to be Bayesian without supervision. In *Advances in Neural Inf. Process. Syst. (NIPS)*, volume 19, pages 1145–1152. MIT Press, 2007.

A. A. Shabalin and A. B. Nobel. Reconstruction of a low-rank matrix in the presence of Gaussian noise. *J. Multivariate Anal.*, 118:67–76, 2013.

C. M. Stein. Estimation of the mean of a multivariate normal distribution. *Ann. Statist.*, 9(6):1135–1151, 1981.

D. Sun and J. Sun. Nonsmooth matrix valued functions defined by singular values. Technical report, Department of Decision Sciences, National University of Singapore, 2003.

M. Udell, C. Horn, R. Zadeh, and S. Boyd. Generalized low rank models. *Foundations and Trends in Machine Learning*, 9(1):1–118, 2016.

F. T. Ulaby and M. C. Dobson. *Handbook of Radar Scattering Statistics for Terrain*. Norwood, MA: Artech House, 1989.

M.E. Wall, P.A. Dyck, and T.S. Brettin. Svdman-singular value decomposition analysis of microarray data. *Bioinformatics*, 17(6):566–568, 2001.

T. Yanagimoto. The Kullback-Leibler risk of the Stein estimator and the conditional MLE. *Ann. Inst. Statist. Math.*, 46(1):29–41, 1994.

Journal of Machine Learning Research 18 (2017) 1-35 Submitted 5/16; Revised 10/17; Published 11/17

Dimension Estimation Using Random Connection Models

Paulo Serra* P.J.SERRA@TUE.NL
Department of Mathematics and Computer Science
Groene Loper 5
MetaForum Building
Eindhoven University of Technology
5612 AZ Eindhoven, the Netherlands

Michel Mandjes M.R.H.MANDJES@UVA.NL
Korteweg-de Vries Institute for Mathematics
Science Park 105–107
University of Amsterdam
1098 XG Amsterdam, the Netherlands

Editor: Françqis Caron

Abstract

Information about intrinsic dimension is crucial to perform dimensionality reduction, compress information, design efficient algorithms, and do statistical adaptation. In this paper we propose an estimator for the intrinsic dimension of a data set. The estimator is based on binary neighbourhood information about the observations in the form of two adjacency matrices, and does not require any explicit distance information. The underlying graph is modelled according to a subset of a specific random connection model, sometimes referred to as the Poisson blob model. Computationally the estimator scales like $n \log n$, and we specify its asymptotic distribution and rate of convergence. A simulation study on both real and simulated data shows that our approach compares favourably with some competing methods from the literature, including approaches that rely on distance information.

Keywords: adaptation, dimensionality reduction, intrinsic dimension, random connection model, random graph

1. Introduction

In machine learning and computational geometry we often want to discover, or sometimes impose, structure on observations, and dimension plays a crucial role in this task. The dimension of a data set can perhaps be best interpreted as the number of variables needed to describe it. However, there is often a gap between the ambient dimension of a data set – the number of variables *used* to describe it – and its intrinsic dimension – the number of variables *needed* to describe it (eventually up to a certain level of precision). For instance, high-dimensional data sets often live in lower dimensional spaces; infinite-dimensional parameters of non-parametric models can often be accurately estimated using just a few parameters; complex data can potentially be highly compressible. However, this is usually not evident by just looking at the data, so in this paper we propose an algorithm to estimate the intrinsic dimension of a data set, and study its behaviour.

There are plenty of reasons to be interested in intrinsic dimensions. Perhaps the most straightforward one is to perform dimensionality reduction. Dimensionality reduction arises from a need to be able to extract meaningful conclusions from high-dimensional observations. There is an extensive literature on this subject using multidimensional scaling, manifold learning, and projection techniques like principal component analysis and projection pursuit; for a general overview cf. (Fodor, 2002; Burges, 2010; Lee and Verleysen, 2007). For specific techniques see for example (Kohonen, 1990; Cox and Cox, 2000; Tenenbaum et al., 2000; Roweis and Saul, 2000; Donoho and Grimes, 2003; Huo and Chen, 2002; Giné and Koltchinskii, 2006), and the references therein. In order to be used to their full extent, these approaches require a priori knowledge about the intrinsic dimension of the data set. Knowledge about intrinsic dimension is also important in independent component analysis; cf. (Hyviirinen et al., 2001).

From a statistical perspective, the intrinsic dimension provides information about the difficulty of making inference. Non-parametric estimators usually rely on approximation properties of certain function spaces; the dimension of the support of these functions influences these approximation properties. Statistical adaptation often focuses on smoothness, but dimension actually has a more substantial impact on rates of convergence. Knowledge about dimension is also important to avoid (if possible) the curse of dimensionality. Dimension plays an important role in classification problems as well, where performance is greatly compromised in high dimensions; cf. (Bickel and Levina, 2004; Fan and Fan, 2008). There are also connections to search, and to outlier detection; cf. (Amsaleg et al., 2015) and the references therein.

From a computational perspective, the dimension of a data set impacts the amount of space needed to store data (compressibility). The speed of algorithms is also commonly affected by the dimension of input data. Learning the underlying dimension is also important to design algorithms that require less data (meaning, make better use of available data) when data happen to live on a low dimensional space. Because of this, knowledge about dimension is crucial in many fields such as biomedicine, economics, engineering, astronomy, remote sensing, and computer vision, with important applications in mass spectrometry, genetics, networking, image processing, automatic text analysis, among others; for some concrete applications see (Verleysen et al., 1999; Lähdesmäki et al., 2005; Abrahao and Kleinberg, 2008; Carter et al., 2010) and references therein.

Early work on dimension estimation dates back to (Shepard, 1962a,b; Kruskal, 1964a,b; Bennett, 1969) on multidimensional scaling. The idea is that one has measurements of similarities (or dissimilarities) between data points, and would like to find points in a potentially high-dimensional vector space that are consistent with the observed similarities (resp. dissimilarities). Other approaches such as that of (Fukunaga and Olsen, 1971; Fukunaga, 1982) are based on Karhunen–Loève expansions. There, and much in the spirit of principal component analysis, the idea is that minimal representations that capture most of the variation in the data carry information about dimensionality; see also an approach based on testing by (Trunk, 1968). Information criteria such as AIC by (Akaike, 1974) and BIC by (Schwarz, 1978) can also be used to estimate dimensions within a model.

∗. This research took place while this author was a postdoctoral researcher at the Korteweg-de Vries Institute for Mathematics, of the University of Amsterdam, in Amsterdam, the Netherlands.

©2017 Paulo Serra and Michel Mandjes.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-232.html.

More recent work of (Camastra and Vinciarelli, 2002; Kégl, 2002; Hein and Audibert, 2005; Sricharan et al., 2010) is based on the notion of correlation integral; cf. (Grassberger and Procaccia, 2004). The correlation dimension is a limit associated with this correlation integral. This notion of dimension, which we also use in this paper, is computationally attractive when compared with other notions of dimension such as that of box counting dimension, for example. There are also the techniques of (Costa and Hero, 2004; Farahmand et al., 2007; Leonenko et al., 2008; Kleindessner and von Luxburg, 2015) based on different types of graphs where edges represent some form of proximity, like k-nearest neighbour graphs, and geodesic minimal spanning trees. Their work relies on the fact that certain quantities associated with these graphs scale monotonically with dimension. By exploring this, one can extract information about dimensionality by inverting these relations. A similar idea is used in (Amsaleg et al., 2015) by recurring to notions from extreme value theory. See also (Eriksson and Crovella, 2012) for a clustering-based approach, and the work of (Levina and Bickel, 2004) for an estimator based on an approximation of the likelihood.

There is some room for improvement in the approaches mentioned above. Some of them rely on rather extensive knowledge about distances between all possible pairs of observations, sometimes also of perturbations thereof, or on certain hierarchical constructs like dendrograms; because execution times tend to scale quadratically with the sample size and linearly with dimension, these quickly become computationally costly as the number of observations or the dimension of the measurements is high – exactly the situation where dimension estimation is most important. (In genetics and computer vision applications, for instance, the number of observed dimensions can easily reach hundreds of thousands.) Therefore, either due to the volume or to complexity of the data, we may be computationally limited to work with very crude information, such as knowing only whether each pair of observations is close or not. Because of this, it is of interest to develop methods to estimate dimension that rely on as little information as possible.

Another aspect that is often overlooked in the literature is that the intrinsic dimension of a data set is usually scale-dependent: the dimension of the data set depends on the scale at which we analyse it; cf. (Burges, 2010). Say we sample points uniformly at random on a manifold with noise; if we look at the data set on a fine scale we only pick up on the noise, whereas at a larger scale the features of the manifold will dominate. The manifold itself can have different dimensions depending on which scale we look at it, and the noise may have arbitrary dimension. It is therefore not clear what "the dimension of a data set" is, unless we specify a scale to go with it (or if the support of the distribution of the data is homogeneous or unstructured). However, even then the dimension is very dependent on the specific data (e.g., structure of the manifold, distribution of the noise). Approaches based on regressing the logarithm of the correlation integral on the logarithm of its argument fail to capture this and instead return something akin to an average dimension across scales for the data set. Approaches based on k-nearest-neighbour graphs also have limitations with regards to this; the distance from a datapoint to its k-nearest neighbour scales in a non-trivial way with dimension and is quite dependent on the distribution of the observations. This makes it difficult to estimate dimension by inversion without specific knowledge on the distribution of the data.

In this paper we resolve the limitations identified above. We estimate the intrinsic dimension of a data set at a user-prescribed scale based solely on binary neighbourhood

relations between observations. More specifically, we assume that certain undirected graphs (or their adjacency matrices) can be observed. In this graph, each vertex corresponds to an observation that lives in some high- (possibly infinite-) dimensional space. An edge is present between two vertices if the corresponding observations are *close*. What we consider to be close determines the scale at which we analyse the data. The goal is to estimate the intrinsic dimension of the data set based on the adjacency matrix of the graph only, i.e., without explicit access to distance information. We model such a graph according to a subset of a random connection model, a model from continuum percolation; cf. (Penrose, 1991), and (Meester and Roy, 1996) for an overview on the subject. More specifically, we model it according to a subgraph of a graph sampled from the so called Poisson blob model; cf. (Grimmett, 1999).

We propose an estimator based on the doubling property of the Lebesgue measure and on the notion of correlation integral. The estimator does not rely on distance information about the observations and has computation time that scales like $n \log n$, which is particularly important when dealing with large, high-dimensional data sets. Since only (sparse) adjacency matrices have to be stored, our approach also leads to a reduction of the required storage space. Under an identifiability condition, we show that the estimator is consistent and asymptotically Gaussian, and we compute its rate of convergence. To the best of our knowledge such results are not yet available in the literature. The estimator strongly concentrates around its expectation but in general the constants involved in the rate scale exponentially with the intrinsic dimension; the bias plays an important role as it is the main bottleneck in the procedure leading to a logarithmic rate. We propose a bias corrected estimator that follows the same (optimal) asymptotics, but which performs much better according to our numerical experiments.

Minimax rates are unknown for the type of data that we consider, but in the (easier) case where one has access to the actual observations, these can be found in (Koltchinskii, 2000) and are logarithmic. (For the noiseless case see Kim et al., 2016.) This means that our procedure is essentially optimal and that its computational efficacy is not obtained at the expense of precision. Furthermore, we are capable of producing estimates of the spread of the estimator, without a need for resampling, and these quantify the uncertainty in the estimate fairly well. This is particularly important given the slow convergence rate, and is a major improvement over competing approaches, which tend to overly concentrate around biased estimates. We also run some numerical simulations that show that our estimator compares favourably with competing estimators (particularly when it comes to recuperate an integer dimension), including estimators that rely on distance information.

This paper is structured as follows. In Section 2 we formally define our model. The interpretation of the effect of scale in the model is given in Section 3. Section 4 has a description of our estimator. Section 5 contains consistency results for a relevant probability in the model. Section 6 has our main result about the consistency of our estimator for the intrinsic dimension, and a comparison with related work from the literature. In Section 7 we present some numerical illustrations for our method, and we propose our bias corrected estimator. We close with some conclusions in Section 8. The proofs of our main results are collected in the Appendix.

2. Sampling, Model, Notation, and Problem Formulation

Consider the following model. Sample *design points* $X_1, \ldots, X_n \in \mathbb{R}^D$, independently, from a distribution F, where $D \in \mathbb{N}$ is some *ambient dimension*. Given the design points $\boldsymbol{X} = (X_1, \ldots, X_n)^T$, construct a random undirected graph by placing an edge between two vertices $i < j \in \{1, \ldots, n\}$ if $r(X_i, X_j) \leq \epsilon$, $\epsilon > 0$, where r is a metric[1] on $\mathbb{R}^D$. We denote the adjacency matrix of the resulting random graph as $\boldsymbol{A}_\epsilon$. If we disregard that our design points are typically concentrated (since they are sampled from F), this is a subset of a graph sampled from what is usually called the Poisson blob model, a model from continuum percolation. In this model we allow $\epsilon = \epsilon_n$ to converge to 0 as $n \to \infty$, if need be; we discuss the role of ϵ in more detail in Section 3. Most of the quantities that we define in the following depend on n but this is omitted from the notation except when n plays a role.

We assume that the observations in our data set actually live (potentially in approximate form) in a lower (potentially fractional) dimensional space. For instance, the design points can have the form $X_i = \varphi(\tilde{X}_i) + \sigma \cdot \epsilon_i$, $\sigma \geq 0$, where $\varphi : \mathbb{R}^d \mapsto \mathbb{R}^D$, $d \leq D$, is some embedding. The observations can therefore be highly structured; they can be concentrated around, say, a manifold. The number d is called the *intrinsic dimension* of the data set $\boldsymbol{X}$, and it is our object of interest.

Our statistical problem is the following: for a data set with n observations we have access to a symmetric, binary matrix $\boldsymbol{\mathcal{A}}$ where $\mathcal{A}_{i,j} = \mathcal{A}_{j,i} = 1$ if, and only if, the i-th and j-th observations are "close"; the data points (or distances between data points) are not actually observed. We assume that our notion of "close" is reasonable, in the sense that we can model $\boldsymbol{\mathcal{A}}$ according to a random connection model: $\boldsymbol{\mathcal{A}} = \boldsymbol{A}_\epsilon$ for some ϵ and some metric r (which are not necessarily known to us). Given access to such adjacency matrices[2] we would like to estimate the intrinsic dimension d. The point is that although the support of F may be high-dimensional, most of the mass of F might be concentrated on a (lower) dimensional sub-space or manifold (eventually as $n \to \infty$, or $\sigma \to 0$), such that one can find a parsimonious representation for that data that still preserves its main features.

We denote by $\boldsymbol{B}_\epsilon$ the degrees of the vertices in the graph such that $\boldsymbol{B}_\epsilon = (B_{\epsilon,1}, \ldots, B_{\epsilon,n})$,

$$B_{\epsilon,i} = \sum_{j=1}^{n} A_{\epsilon,i,j}, \quad i = 1, \ldots, n, \tag{1}$$

where the (binary) entries $A_{\epsilon,i,j}$ of the adjacency matrix $\boldsymbol{A}_\epsilon$ satisfy

$$A_{\epsilon,i,j} = A_{\epsilon,j,i} = \mathbf{1}_{\{r(X_i,X_j)\leq\epsilon\}}, \qquad \text{and} \qquad A_{\epsilon,i,i} = 0, \quad i,j = 1, \ldots, n,\ i \neq j. \tag{2}$$

By construction, the distribution of the random matrix $\boldsymbol{A}_\epsilon$ is invariant under any permutation of its rows and columns so that the $B_{\epsilon,i}$ are identically distributed but not independent. Define the two functions

$$p_\epsilon(x) = \mathbb{P}\{r(X, x) \leq \epsilon\}, \qquad \text{and} \qquad p_\epsilon(x, y) = \mathbb{P}\{r(X, x) \leq \epsilon, r(X, y) \leq \epsilon\}, \tag{3}$$

1. All of the assumptions on the metric will be implicit.
2. In fact we consider two adjacency matrices as it will become clear from the definition of our estimator in (7). We argue in Section 4, that this is inevitable since ϵ trades off with the standard deviation of F.

where the probability is taken with respect to $X \sim F$. This is the (local) connection probability for a design point at site x, and the probability of two design points at sites x and y sharing a neighbour. With this notation,

$$B_{\epsilon,i}|X_i \sim \text{Bin}\{n-1,\, p_\epsilon(X_i)\}. \tag{4}$$

From this we see that if p were constant, then the model would reduce to the Erdős–Rényi model of (Erdős and Rényi, 1959). If p_ϵ (which depends exclusively on F and ϵ) is not constant, then this leads to some inhomogeneity for the resulting random graph.

In what follows we denote, for i, j, k mutually different,

$$p_{\epsilon,1} = \mathbb{E}A_{\epsilon,i,j}, \qquad \text{and} \qquad p_{\epsilon,2} = \mathbb{E}A_{\epsilon,i,k}A_{\epsilon,k,j}. \tag{5}$$

These two numbers (or sequences, if $\epsilon \to 0$) are the probability that two vertices connect and the probability that two vertices have a common neighbour, respectively. Note that if X, Y, Z are independent and distributed according to F, then

$$p_{\epsilon,1} = \mathbb{P}\{r(X, Y) \leq \epsilon\} = \mathbb{E}\mathbb{P}\{r(X, Y) \leq \epsilon|X\} = \mathbb{E}p_\epsilon(X),$$

and in the same way,

$$p_{\epsilon,2} = \mathbb{E}\mathbb{P}\{r(X, Z) \leq \epsilon, r(Z, Y) \leq \epsilon|X, Y\} = \mathbb{E}p_\epsilon(X, Y).$$

By definition, $p_\epsilon(X, X) = p_\epsilon(X)$. Also, by independence and Jensen's inequality,

$$p_{\epsilon,2} = \mathbb{E}\mathbb{P}\{r(X, Z) \leq \epsilon, r(Z, Y) \leq \epsilon|Z\} = \mathbb{E}\{p_\epsilon(Z)^2\} \geq \mathbb{E}\{p_\epsilon(Z)\}^2 = p_{\epsilon,1}^2.$$

In fact, the (non-negative) difference $p_{\epsilon,2} - p_{\epsilon,1}^2$ is the variance of the connection probability function $p_\epsilon(x)$ which will play an important role later in the paper.

Both $p_{\epsilon,1}$ and $p_{\epsilon,2}$ depend on ϵ (and F), but also on the dimensionality of the data. For example, it is clear that $p_{\epsilon,1}$ and $p_{\epsilon,2}$ decrease as ϵ decreases. In fact, most of what follows is based on this dependence. Before we give the intuition behind our estimator, we discuss the role of ϵ in our approach.

3. Role of ϵ in the Model

The parameter $1/\epsilon$ can be seen as a resolution level that determines at which distance we distinguish between design points. This parameter plays a crucial role in our approach as is explained in this section. In Figure 1 we exemplify the effect of the size of ϵ. We sampled points uniformly at random on a manifold, then added some (3-dimensional) Gaussian noise; these points are the design points $\boldsymbol{X}$ and are embedded on a 3-dimensional space. We then took one of the design points, and coloured red all points that fall within a given Euclidean distance ϵ of the selected design point; the three plots correspond to different choices of ϵ.

If ϵ is so small that no red dots would be present, then the detected dimension is 0. If ϵ is large enough to capture just a few nearest neighbours (leftmost plot), then we capture only the effect of the noise – 3-dimensional in our case – but arbitrary in general. Increasing ϵ (central plot), the intersection is now a 2-dimensional section of the surface of the manifold. Further increasing ϵ (rightmost plot) changes the shape of the intersection which is now the

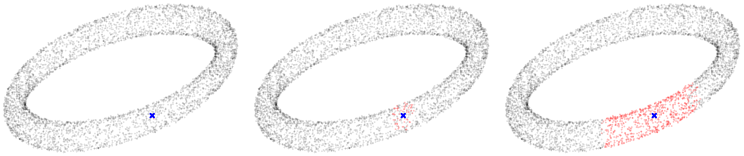


Figure 1: Design points sampled uniformly on a torus, with noise. Design points within an ϵ distance (increasing from left to right) of a fixed design point are coloured red.

(1-dimensional) surface of a tube. In either case, with much larger choices for ϵ we would capture all design points and the volume of the intersection would vanish with respect to ϵ; this would again lead the trivial case where dimension 0 is detected.

Of course the dimension can also be fractal, and we could also be interested in the dimension of just a region of the manifold in which case the choice of ϵ (and the design point that defines the neighbourhood) plays an important role again. The bottom line is that ϵ should be considered a parameter of the model (as opposed to a parameter of the estimator) in that the target intrinsic dimension should be seen as a function of ϵ; cf. (Burges, 2010) for a similar discussion. In other words, the resolution level should be chosen in line with the goals of the analysis; to learn the structure of the noise one would pick relatively small values of ϵ, while to learn the dimension of the manifold we would have to pick larger values. This is related to the scale of the observations, and it should be taken into consideration when choosing ϵ. Another issue is that one should also account for the sample size in the form of a finite sample correction. We return to this point in Section 7 where we use some numerical experiments to illustrate this point.

Next we define and give the intuition for our estimator of the intrinsic dimension d.

4. Estimation of the Intrinsic Dimension

We start by providing some heuristic motivation for our estimator of d. Consider, for $x \in \mathbb{R}^m$, $m \in \mathbb{N}$ the balls $V_\epsilon(x,m) = \{y \in \mathbb{R}^m : r(x,y) \leq \epsilon\}$ for a homogeneous, translation-invariant metric r, and denote $V_\epsilon(m) = V_\epsilon(0,m)$. Assume, without loss of generality, that $0 \in \mathcal{X} \subseteq \mathbb{R}^D$, where $\mathcal{X}$ is an appropriate high probability set in the support of F. If ϵ is small (or if $\epsilon \to 0$) and if F admits a continuous density f with respect to the Lebesgue measure μ, then we expect

$$p_\epsilon(x) \approx \int_{\mathcal{X}} \mathbf{1}_{V_\epsilon(x,D)}(y)\, f(y)\, d\mu(y) \approx f(x) \cdot \int_{\mathcal{X}} \mathbf{1}_{V_\epsilon(D)}(y)\, d\mu(y) \triangleq f(x) \cdot v_\epsilon.$$

The assumption that the intrinsic dimension of the data set is d, corresponds to assuming, with mild abuse of the notation, that (for all appropriately small ϵ)

$$v_\epsilon \triangleq \mu\{V_\epsilon(D) \cap \mathcal{X}\} \approx \mu\{V_\epsilon(d)\}.$$

(In fact, this relation can be seen as a definition of the ϵ-scale intrinsic dimension.) This has the interpretation that at the ϵ-scale the data looks d-dimensional. The point is that $p_\epsilon(x)$

should not be sensitive to the dimension D of the data points, but instead to the intrinsic dimension d of the data set and an ϵ-scale.

Since $p_{\epsilon,1} = \mathbb{E}\{p_\epsilon(X)\}$, we can approximate

$$p_{\epsilon,1} \approx \mathbb{E}f(X) \cdot \mu\{V_\epsilon(d)\}.$$

One can estimate d by replacing $p_{\epsilon,1}$ by an estimator and inverting the relation above. However, this would only be feasible with knowledge of the distribution F via the constant $\mathbb{E}f(X)$ and of the parameter ϵ, which in general we do not have access to.

Arguably, the most reasonable way to get rid of the dependence on F and ϵ is to examine the data at two different scales simultaneously. For appropriately small ϵ,

$$\frac{p_{2\epsilon,1}}{p_{\epsilon,1}} \approx \frac{\mu\{V_{2\epsilon}(d)\}}{\mu\{V_\epsilon(d)\}}. \tag{6}$$

This is a natural idea. Looking back at Figure 1, the (hyper-)volume of the intersections (which can be inferred from the number of points in the intersection) does not give us any information about dimension; it is how this quantity *scales* with ϵ that is informative.

With this approximation in mind, we define an estimator for d: for an arbitrary function $g_\epsilon(d)$ – ideally $p_{2\epsilon,1}/p_{\epsilon,1}$, but in general any reasonable approximation of $\mu\{V_{2\epsilon}(d)\}/\mu\{V_\epsilon(d)\}$ – the estimator is defined as (any) implicit solution $\hat{d}_n$ to the equality

$$\frac{\hat{p}_{n,2\epsilon,1}}{\hat{p}_{n,\epsilon,1}} = g_\epsilon\big(\hat{d}_n\,\big), \tag{7}$$

where $\hat{p}_{n,\epsilon,1}$ is any estimator for $p_{\epsilon,1}$, $\epsilon > 0$. If d is an integer, then we can consider an estimator $[\hat{d}_n\,]$, where $[\,\cdot\,]$ represents the argument rounded to the closest integer. Note that the function g_ϵ is allowed to depend on n.

The need to look at the data at two different scales simultaneously should not be a surprise. The probability $p_{\epsilon,1}$ itself does not carry any information about dimension if F is unknown; it is instead how $p_{\epsilon,1}$ scales as a function of ϵ that provides information about dimension. This notion of scaling is in fact connected with the notion of expansion dimension of (Karger and Ruhl, 2002).

For a given metric r one can numerically approximate the function $\mu\{V_{2\epsilon}(d)\}/\mu\{V_\epsilon(d)\}$, but in analogy to the doubling property of the Lebesgue measure this function should be constant over ϵ, at least if ϵ is appropriately small. So a canonical choice (independent of ϵ) would be $g_\epsilon(d) = 2^d$, in which case one has an explicit estimator for d:

$$\hat{d}_n = \frac{\log \hat{p}_{n,2\epsilon,1} - \log \hat{p}_{n,\epsilon,1}}{\log 2}. \tag{8}$$

This is just an example of a possible form that the estimator can take. However, it does suggest that at least for certain models one can expect to have explicit estimators for d that do not require knowledge of ϵ, F, or r and are therefore completely parameter-free.

Although ϵ should be known (or indeed picked, as pointed out in the discussion in Section 3), one can define reasonable estimators for d in the case where ϵ is unknown. In fact, it is is not possible to estimate ϵ consistently from adjacency matrices $\boldsymbol{A}_\epsilon$ without

knowledge of the distribution F. This follows from the trade-off between the standard deviation of the distribution F and the radius ϵ. Let $\sigma > 0$, and say $\boldsymbol{A}_{\epsilon/\sigma}$ is associated with design points $X \sim F$ and $\boldsymbol{A}'_\epsilon$ is associated with design points $X' \sim F(\cdot/\sigma)$; then

$$A_{\epsilon/\sigma,i,j} = \mathbf{1}_{\{r(X_i,X_j)\le\epsilon/\sigma\}} = \mathbf{1}_{\{r(\sigma X_i,\sigma X_j)\le\epsilon\}} \sim A'_{\epsilon,i,j},$$

so that information about ϵ cannot be retrieved from the adjacency matrix without knowledge of the distribution F. However, if ϵ is known, one may try to flesh out lower order terms in the approximation above to reduce the bias of the estimates.

We can more precisely approximate the local connection probability as

$$p_\epsilon(x) \approx f(x)\cdot v_\epsilon + \int_{\mathcal{X}} \mathbf{1}_{V_\epsilon(x,D)}(y)\,(y-x)^T\nabla f(x)\,d\mu(y),$$

where ∇f denotes the gradient of the density f, so that by taking expectation,

$$p_{\epsilon,1} \approx \mathbb{E}f(X)\cdot v_\epsilon\cdot\big(1+\Delta_\epsilon\big), \qquad \Delta_\epsilon = \frac{\mathbb{E}\int_{\mathcal{X}}\mathbf{1}_{V_\epsilon(X,D)}(y)\,(y-X)^T\nabla f(X)\,d\mu(y),}{\mathbb{E}f(X)\cdot v_\epsilon}.$$

For our canonical choice $g_\epsilon(d) = 2^d$ we thus obtain the approximation

$$\frac{\log p_{2\epsilon,1} - \log p_{\epsilon,1}}{\log 2} \approx \frac{\log v_{2\epsilon} - \log v_\epsilon + \log\big(1+\Delta_{2\epsilon}\big) - \log\big(1+\Delta_\epsilon\big)}{\log 2} \approx d + \frac{\Delta_{2\epsilon}-\Delta_\epsilon}{\log 2}.$$

By the Cauchy-Schwarz inequality to the inner product $(y-x)^T\nabla f(x)$, we conclude that

$$\frac{|\Delta_{2\epsilon}-\Delta_\epsilon|}{\log 2} \le \frac{3}{\log 2}\cdot\frac{\mathbb{E}\|\nabla f(X)\|}{\mathbb{E}f(X)}\cdot\epsilon.$$

Although the multiplier above is unknown, it depends only on F and is therefore fixed. Furthermore, it is reasonable to expect the multiplier to be of order d, since the gradient of the density should only be non-trivial along d independent directions. This means that certain choices for the function $g_\epsilon(d)$, like for example choices that are independent of the scale ϵ, should result in a bias of order $O(d\cdot\epsilon)$. We return to this discussion in Section 6 after we have specified the asymptotics of $\hat{d}_n$ for arbitrary g_ϵ.

Remark 1 *A similar reasoning to the one that was applied to $p_{\epsilon,1}$ above can be applied to other probabilities associated with the model, like for example $p_{\epsilon,2}$, to motivate alternative estimators for the intrinsic dimension. Although not reported here, we did not find any noticeable difference between the $\hat{d}_n$ estimator defined in* (7) *and a $p_{\epsilon,2}$ based estimator.*

From the discussion above, it is clear that the consistency of the estimators defined in (7) depends on three factors: the consistency of the estimates of $p_{\epsilon,1}$ that are used, the quality of the approximation in (6), and the slope of $g_\epsilon(d)$ around the underlying intrinsic dimension d. First we address the estimation of $p_{\epsilon,1}$.

5. Estimates of the Connection Probability and their Asymptotics

An estimator for $p_{\epsilon,1}$ is obtained by averaging off-diagonal entries of the matrix $\boldsymbol{A}_\epsilon$. For any $m_n \le n$ (m_n is for now left unspecified, and is a parameter of the estimator),

$$\hat{p}_{n,\epsilon,1} = \frac{1}{m_n}\sum_{i=1}^{m_n}\frac{B_{\epsilon,i}}{n-1} = \frac{2}{m_n(n-1)}\sum_{i=1}^{m_n}\sum_{j=i+1}^{n} A_{\epsilon,i,j}, \tag{9}$$

using the symmetry of $\boldsymbol{A}_\epsilon$. Since $\mathbb{E}B_{\epsilon,i}/(n-1) = p_{\epsilon,1}$, the estimator $\hat{p}_{n,\epsilon,1}$ is unbiased. This estimator can be evaluated in $O(n\,m_n)$ instructions. If we set $m_n = n$ then the execution time may be prohibitive if n is large so the parameter m_n offers some flexibility. However, as we will see below and in Sections 6 and 7, the role of the sequence m_n goes beyond just controlling the computational complexity of the estimator. In Section 7.2, in particular, we discuss what constitutes a "good" choice for m_n.

The following theorem provides the asymptotics for the estimator in (9).

Theorem 2 *Let $m_n \le n$ and $m_n \to \infty$ as $n \to \infty$. If $m_n = o(n)$, and $p_{\epsilon,2} > p_{\epsilon,1}^2$, then*

$$S_{n,\epsilon,1}^{-1/2}\cdot\left\{\frac{\hat{p}_{n,\epsilon,1}}{p_{\epsilon,1}} - 1\right\} \xrightarrow{d} N(0,1), \qquad \textit{where} \qquad S_{n,\epsilon,1} = \frac{p_{\epsilon,2}-p_{\epsilon,1}^2}{m_n\,p_{\epsilon,1}^2}. \tag{10}$$

If $m_n = n$ then the previous display also holds if we further assume that $n(p_{\epsilon,2}-p_{\epsilon,1}^2)^2 \to \infty$. (This assumption always holds if ϵ is fixed.)

■

The proof of this theorem can be found in the Appendix. This result is valid irrespectively of the distribution F, and holds even if $\epsilon \to 0$, as $n \to \infty$, so that $p_{\epsilon,1}, p_{\epsilon,2} \to 0$. The difference $p_{\epsilon,2} - p_{\epsilon,1}^2 = \mathbb{E}\{p_\epsilon(X)^2\} - \{\mathbb{E}p_\epsilon(X)\}^2$ is the variance of the function p_ϵ. From this we see that p_ϵ being more variable has a negative impact on the estimation of $p_{\epsilon,1}$, which is not surprising. If ϵ is fixed, then $p_{\epsilon,1}$ can be estimated with rate $m_n^{-1/2}$. However, if $\epsilon \to 0$ the rates may be different depending on how the probabilities involved scale with ϵ, which in turn depends on the specific distribution F and the metric r at hand.

Next we give conditions under which the estimators (7) are consistent for d.

6. Consistency of Estimates for the Intrinsic Dimension

Based on the asymptotics of $\hat{p}_{n,\epsilon,1}$ from the previous section, whether the procedure outlined in Section 4 delivers consistent estimates for d or not, now depends on the specific model in question and on $g_\epsilon(d)$.

Theorem 3 *Consider the implicit estimators* (7). *Assume that the conditions of Theorem 2 required for the convergence of $\hat{p}_{n,\epsilon,1}$ and $\hat{p}_{n,2\epsilon,1}$ with rate $m_n^{-1/2}$ hold. For that ϵ, d, and m_n, assume that, as $n \to \infty$,*

$$p_{2\epsilon,1} = p_{\epsilon,1}\cdot g_\epsilon\Big\{d + o(m_n^{-1/2})\Big\}. \tag{B}$$

Assume also that the derivative (with respect to d) of $g_\epsilon(d)$ exists, is continuous and non-zero at d. If $p_{\epsilon,1}\cdot(1-p_{2\epsilon,1}) = o\big[n\{p_{\epsilon,2\epsilon,2} - p_{\epsilon,1}\cdot p_{2\epsilon,1}\}\big]$ and $m_n = o(n)$, then as $n\to\infty$,

$$S_{n,\epsilon}^{-1/2}\cdot\Big\{\hat{d}_n - d\Big\} \xrightarrow{d} N(0,1), \qquad \text{where} \qquad S_{n,\epsilon} = \left\{\frac{\partial \log g_\epsilon(d)}{\partial d}\right\}^2\cdot\frac{\mathcal{V}_\epsilon}{m_n}$$

where for $p_{\epsilon_1,\epsilon_2,2} = \mathbb{P}\{r(X,Z)\le\epsilon_1, r(Y,Z)\le\epsilon_2\}$,

$$\mathcal{V}_\epsilon = \frac{p_{\epsilon,1}^2\cdot p_{2\epsilon,2} - 2\cdot p_{\epsilon,2\epsilon,2}\cdot p_{\epsilon,1}\cdot p_{2\epsilon,1} + p_{2\epsilon,1}^2\cdot p_{\epsilon,2}}{p_{\epsilon,1}^2\cdot p_{2\epsilon,1}^2}$$

■

Remark 4 *In Theorem 3 we consider the case $m_n = o(n)$, which is the most relevant case in practice. The general expression for $\mathcal{V}_\epsilon$ (meaning for any sequence $m_n \le n$) can be found in* (16)*, in the Appendix.*

Remark 5 *Condition* (B) *controls the asymptotic bias of the estimator for d. Note that this condition should be interpreted as a condition on g_ϵ and on the sequence m_n, and not a condition on ϵ, since ϵ is a modelling parameter set by the user. If condition* (B) *does not hold, then the statement of the previous theorem is still valid if we centre $\hat{d}_n$ with $\mathbb{E}(\hat{d}_n)$ instead of d, but in that case the estimator might be asymptotically biased. Alternatively, if* (B) *holds with $r_n = o(m_n)$, instead of m_n, then we conclude that $r_n^{-1/2}\big(\hat{d}_n - d\big) = o_P(1)$. (Note that the condition becomes more restrictive for faster rates.)*

For the explicit estimator in (8) the $g_\epsilon(d)$-dependent scaling in the variance is $\log(2)^{-2}$. In this case, the bias condition (B) reduces to

$$p_{2\epsilon,1} = p_{\epsilon,1}\cdot 2^{d+o\left(m_n^{-1/2}\right)}.$$

This requires the connection probability $p_{\epsilon,1}$ to approximately have a doubling property: if the distance at which vertices connect doubles, then the probability of connection goes up by a factor 2^d, approximately. (Note that also ϵ should in general depend on n, but we return to this point in Section 7.1.) How much leverage we have in terms of the approximation depends mostly on the sequence m_n and is, in a sense, the price to pay for the computational speed-up. However, the choice of m_n goes well beyond this.

Remark 6 *Since our estimator for d only requires access to m_n rows of $\boldsymbol{A}_\epsilon$ and $\boldsymbol{A}_{2\epsilon}$, one can obtain several (correlated) estimates $\hat{d}_n^{(1)}$, $\hat{d}_n^{(2)}$, ..., of d based on disjoint sets of rows if $m_n = o(n)$. From these one can estimate the variance of $\hat{d}_n$.*

Say that we consider the following estimator for the variance

$$\hat{\sigma}_n^2 = \widehat{\mathbb{V}\hat{d}_n} = \frac{1}{t_n - 1}\sum_{i=1}^{t_n-1}\Big\{\hat{d}_n^{(i)} - \bar{d}_n\Big\}^2, \qquad \text{with} \qquad \bar{d}_n = \frac{1}{t_n}\sum_{i=1}^{t_n}\hat{d}_n^{(i)},$$

where $t_n\in\mathbb{N}$ is at most n/m_n. It is straightforward to check that

$$\mathbb{E}\hat{\sigma}_n^2 = \sigma^2\Big[1 - \rho\big\{\hat{d}_n^{(1)}, \hat{d}_n^{(2)}\big\}\Big],$$

where $\rho(X,Y)$ represents the correlation between X and Y. Making use of the fact that $\hat{p}_{n,\epsilon,1}^{(i)}$ and $\hat{p}_{n,2\epsilon,1}^{(i)}$ are independent for $i\in\mathbb{N}$, we have

$$\mathbb{V}\big\{\hat{d}_n^{(1)}, \hat{d}_n^{(2)}\big\} = \frac{1}{\big\{\log(2)^2\big\}}\Big[\mathbb{V}\big\{\log\hat{p}_{n,\epsilon,1}^{(1)}, \log\hat{p}_{n,\epsilon,1}^{(2)}\big\} + \mathbb{V}\big\{\log\hat{p}_{n,2\epsilon,1}^{(1)}, \log\hat{p}_{n,2\epsilon,1}^{(2)}\big\}\Big].$$

Using the approximation[3] $\mathbb{V}(\log X, \log Y)\approx\mathbb{V}(X,Y)/(\mathbb{E}X\cdot\mathbb{E}Y)$*, it is enough to look at $\mathbb{V}\big\{\hat{p}_{n,\epsilon,1}^{(1)}, \hat{p}_{n,\epsilon,1}^{(2)}\big\}$. If we denote the range of rows associated with these two estimators as respectively I_1 and I_2, then writing the covariance as a four-fold sum, we get*

$$\begin{aligned}\mathbb{V}\big\{\hat{p}_{n,\epsilon,1}^{(1)}, \hat{p}_{n,\epsilon,1}^{(2)}\big\} &= \left\{\frac{2}{m_n(n-1)}\right\}^2\sum_{i_1\in I_1}\sum_{i_2\in I_2}\sum_{j_1=i_1+1}^{n}\sum_{j_2=i_2+1}^{n}\mathbb{V}\big(A_{\epsilon,i_1,j_1}, A_{\epsilon,i_2,j_2}\big)\\ &\le \frac{4}{n-1}\big(p_{\epsilon,2} - p_{\epsilon,1}^2\big),\end{aligned}$$

where we use the fact that since $I_1\cap I_2 = \emptyset$, then $\mathbb{V}\big(A_{\epsilon,i_1,j_1}, A_{\epsilon,i_2,j_2}\big) = p_{\epsilon,2} - p_{\epsilon,1}^2$ if $j_1 = j_2$, and $\mathbb{V}\big(A_{\epsilon,i_1,j_1}, A_{\epsilon,i_2,j_2}\big) = 0$, if $j_1\neq j_2$. Finally, under the conditions of Theorems 2 and 3, we can put everything together to bound

$$\big|\mathbb{E}\hat{\sigma}_n^2 - \sigma^2\big| \lesssim \frac{m_n}{n}\cdot\frac{S_{n,\epsilon,1}}{S_{n,\epsilon}};$$

this upper bound converges to zero for appropriate ϵ, if $m_n = o(n)$.

Note that when the metric r is induced by the Euclidean norm $\|\cdot\|_2$, and $m_n = n$, the estimator $\hat{p}_{n,\epsilon,1}$ (as a function of ϵ) coincides with a realisation of the so-called correlation integral; cf. (Camastra and Vinciarelli, 2002). This is defined in the following way. With $x_1,\ldots,x_n$ denoting points on a manifold whose dimension we would like to measure, let

$$C_n(\epsilon) = \frac{2}{n(n-1)}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\mathbf{1}_{\{\|x_i - x_j\|_2\le\epsilon\}}; \tag{11}$$

the correlation integral $C(\epsilon)$ is the limit, when $n\to\infty$, of $C_n(\epsilon)$. The underlying idea behind the intrinsic dimension being d is that $C(\epsilon)$ should scale like ϵ^d so that the limit as $\epsilon\to 0$ of $\log\{C(\epsilon)\}/\log(\epsilon)$ is d; this is then called the correlation dimension, which is a type of fractal dimension. Following up on Remark 1, basing our estimator on $p_{\epsilon,2}$ would lead to a variation on the correlation integral above. We found no advantage in using the $p_{\epsilon,2}$ based estimator over the $p_{\epsilon,1}$ based one.

Some estimators for intrinsic dimension like that of (Grassberger and Procaccia, 1983) are based on the idea of regressing $\log\{C_n(\epsilon)\}$ on $\log(\epsilon)$, and estimating the (correlation) dimension from the slope of the fit. However, based on the discussion from Section 3, this slope actually corresponds to some average dimension over different scales, which is not what we are interested in; cf. Fig. 3.3 of Lee and Verleysen, 2007, for an example of the scale dependence of the correlation integral. Further, this also makes extensive use of

3. Here we use the fact that $\mathbb{V}(\log X, \log Y) = \mathbb{V}\{\log(X/\mathbb{E}X), \log(Y/\mathbb{E}Y)\}\approx\mathbb{V}(X/\mathbb{E}X - 1, Y/\mathbb{E}Y - 1) = \mathbb{V}(X,Y)/(\mathbb{E}X\cdot\mathbb{E}Y)$, since $\log(1+x)\approx x$, for all appropriately small x.

distances between the observations, while we rely only on the neighbourhood information provided by adjacency matrices.

As we argue in the previous section, one has to examine the data at (at least) two different scales to derive meaningful information about dimension, and this has been noted in the literature before. (Kégl, 2002) proposed a scale dependent notion of correlation dimension based on

$$D_n(\epsilon_1, \epsilon_2) = \frac{\log\{C_n(\epsilon_2)\} - \log\{C_n(\epsilon_1)\}}{\log(\epsilon_2) - \log(\epsilon_1)} \tag{12}$$

This improves upon the idea of regressing the logarithms of the empirical version of the correlation integral (cf. Grassberger and Procaccia, 1983; Pettis et al., 1979) on the logarithm of ϵ by allowing one to focus on a specific range of scales. Indeed, this is closely reflected in our estimator $\hat{d}_n$, although we distinguish ourselves from other approaches by deriving the asymptotic distribution of our estimator within a flexible framework; to the best of our knowledge the central limit theorem that we derive is new to the literature.

Other approaches to intrinsic dimension estimation follow similar ideas but make use of other notions of dimension like for example Hausdorff dimension, information dimension, box counting dimension, (generalised) expansion dimension, packing dimension, and also local versions of these concepts. The success of such approaches depends mostly on how computationally tractable computing the estimate of the dimension is, and how adequate the particular notion of dimension at hand is for the model under consideration.

Another approach corresponds to the maximum likelihood estimator of (Bickel and Levina, 2004). This estimator is based on maximising the likelihood obtained by assuming that the observations come from a homogeneous Poisson process. The estimator is then based on distances to the k-the nearest neighbour of each point, with k interpreted as a "bandwidth" parameter of the estimator. Another estimator based on k-nearest-neighbours is that of (Kleindessner and von Luxburg, 2015). In both cases the connection between k and the scale at which we estimate dimension has not yet been explored in detail, however. The disadvantage of k-nearest-neighbour approaches seems to be that the way in which the distance of an observation to its k nearest neighbour scales with dimensions may heavily depend on the underlying distribution of the data. Although this does open the door to sharper estimates of the intrinsic dimension, this is done at the expense of needing more information about the distribution of the data. Besides this, relating k to the scale at which the intrinsic dimensions is being estimated also seems to be difficult.

The approaches mentioned above, as well as our approach, are examples of so called geometric methods. A different class of methods are eigenvalue (or projection) methods. These stem from the work of (Fukunaga and Olsen, 1971); see also (Bruske and Sommer, 1998). These methods are typically based on principal component analysis (PCA) and estimate the dimension based on how many eigenvalues are above certain (small) threshold. They seem to be less useful for estimating intrinsic dimension because of the difficulty of determining what constitutes an appropriate threshold; cf. (Verveer and Duin, 1995).

In the next section we present some numerical results to illustrate our approach. These results guide us in our choice of the sequence m_n.

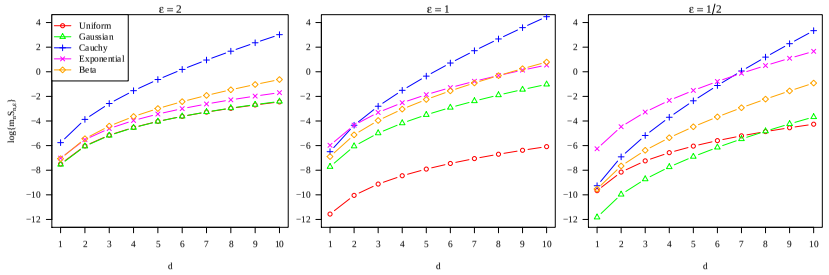


Figure 2: Effect of the distribution of the design points, d, and ϵ on the logarithm of the asymptotic variance of our estimator for d. The three plots correspond to $\epsilon \in \{2, 1, 1/2\}$, left to right. In each plot, each line corresponds to a different distribution for the design points.

7. Numerical Results

In this section we present some numerical results. We start by exemplifying in Section 7.1 how the probabilities $p_{\epsilon,1}$ and $p_{\epsilon,2}$ determine the bias and variance of our estimator for different distributions for the observations $\boldsymbol{X}$. Section 7.2 is about the consequences of the choice of the sequence m_n in our estimator. In Section 7.3 the performance of our estimator is evaluated for different combinations of dimension d and resolution $1/\epsilon$. The main goal of these first three subsections is to understand what constitutes a good choice for the sequence m_n that features in the definition (9). Section 7.4 concerns a non-trivial choice for the function g_ϵ (meaning a choice other than 2^d), as well as other types of bias correction. In Section 7.5 we illustrate the effect of noise on our estimator. Finally, in Section 7.6, we apply our estimator to a batch of data sets of both simulated, and real data. To simplify the exposition, in all cases the metric r is the Euclidean distance.

7.1 Scaling of $p_{\epsilon,1}$ and $p_{\epsilon,2}$, and their Influence on the Bias and Variance of $\hat{d}_n$

The probabilities $p_{\epsilon,1}$ and $p_{\epsilon,2}$ play an important role in our approach. The quantity $\{p_{\epsilon,2} - p_{\epsilon,1}^2\}/p_{\epsilon,1}^2$ is the variance of the function $p_\epsilon(x)/p_{\epsilon,1}$, which is the relative connection probability at each site x. We see, for example, that if $m_n = o(n)$, then the scaling $m_n \cdot S_{n,\epsilon}$ that features in the asymptotics for our estimator for d is up to a constant factor the variance of $p_\epsilon(X)/p_{\epsilon,1} - p_{2\epsilon}(X)/p_{2\epsilon,1}$. However, this quantity still depends on ϵ and d, so it is interesting to see how it behaves for different distributions for the design points.

In Figure 2 we plot the logarithm of $m_n \cdot S_{n,\epsilon}$, as a function of the intrinsic dimension d for several different choices for the distribution of the design points, for three choices of ϵ. If ϵ is fixed, then $m_n S_{n,\epsilon}$ is just the constant that features in the rate of convergence of the estimator of $\hat{d}_n$. (The curves were computed by numerical integration, but are otherwise exact.) The coordinates of the design points X_i were sampled independently from the indicated distributions (uniform, Gaussian, exponential, and beta$\{2, 5\}$, all scaled standard deviation 1; Cauchy with scale parameter 1). The main message is that for appropriately

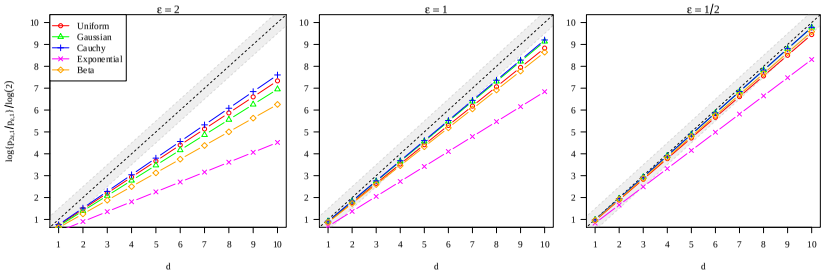


Figure 3: Effect of the distribution of the design points, d, and ϵ on $\{\log p_{2\epsilon,1} - \log p_{\epsilon,1}\}/\log(2)$. Left to right, the plots correspond to $\epsilon \in \{2, 1, 1/2\}$. The shaded area corresponds to $d \pm 1/2$.

large d (about $d \geq 5$) the lines increase roughly linearly, which would mean that the constants in the asymptotic statement in Theorem 3 increase exponentially with d. In general, d is fixed, but these plots give an indication that if the intrinsic dimension is large, then in order to attain a given level of precision, one should need a fairly large number of observations. This should give a notion of when the asymptotics described in Section 6 *kick in*. The effect of ϵ, on the other hand, does not seem too pronounced and affects mostly how the lines behave when the intrinsic dimension d is relatively small.

While the variance of the estimator is, up to the scaling $\{\partial \log g_\epsilon(d)/\partial d\}^2$, only model dependent, the bias depends greatly on the function g_ϵ used in the definition of the estimator. In particular it depends on how well $g_\epsilon(d)$ approximates $p_{2\epsilon,1}/p_{\epsilon,1}$ as prescribed by the bias condition (B). As discussed in Section 4, for appropriately small ϵ, it should hold that $p_{2\epsilon,1}/p_{\epsilon,1} \approx 2^d$, making $g_\epsilon(d) = 2^d$ our canonical choice for g_ϵ. For this choice of g and for the same distributions as before, in Figure 3 we plot $d \mapsto \{\log p_{2\epsilon,1} - \log p_{\epsilon,1}\}/\log(2)$, for different ϵ; we compare it with the identity $d \mapsto d$.

In the plots above, the black dotted line along the diagonal is the identity, and the grey shaded area encompasses $d \pm 1/2$ (for reference). As before, the remaining lines correspond to different distributions for the design points. As expected from the discussion in Section 4, as ϵ gets smaller, these lines mimic the doubling property of the Lebesgue measure more closely so that $d \mapsto p_{2\epsilon,1}/p_{\epsilon,1}$ indeed gets closer to $d \mapsto 2^d$. The plots also suggest that considering $g_\epsilon(d) = 2^d$ should lead to the dimension being systematically underestimated, and that one may want to consider a multiplicative correction based on ϵ. In a sense, this is the price to be paid for having a parameter-free estimator.

Part of the bias is a consequence of the fact that, in general, we do not have access to a good (model dependent) function g. For example, if g is constant over ϵ, as discussed in Section 4, the bias should, in fact, be of order ϵ. One should therefore consider to take ϵ small to reduce the bias of the estimates. However, this is somewhat at odds with the notion put forward in Section 3, where we explain that ϵ determines the scale at which we examine the data and is therefore a (fixed) modelling parameter.

As it turns out, there is a justification for taking $\epsilon = \epsilon_n$ converging to zero. The rationale is the following. Typically, there will be noise in our data so that the support of F is an enlarged version of the manifold whose dimension we would like to estimate. If the design points are relatively concentrated, in the sense that $\mathbb{P}\{r(0, X) > x\} \lesssim \exp(-x^2)$, $x > 0$, say, then, by the union bound,

$$\mathbb{P}\Big\{ \max_{i=1,\ldots,n} r(0, X_i) > \sqrt{\delta \log n} \Big\} \leq \sum_{i=1}^{n} \mathbb{P}\Big\{ r(0, X_i) > \sqrt{\delta \log n} \Big\} \lesssim n^{1-\delta}, \qquad \delta > 1.$$

In other words, if ϵ_1 is to express the distance at which we would like to analyse the data, then if we observe only n points and take $\epsilon \equiv \epsilon_1$, we are overestimating the typical distances between points by roughly a factor $\sqrt{\log n}$. In a sense, this growing spread can be thought of as arising from noise, so that we should therefore establish connections at a slightly smaller distance, say for instance $\epsilon_n = \epsilon_1/\sqrt{\log(1+n)}$, $\epsilon_1 > 0$. This can also be seen as a finite sample correction for the estimator; cf. (Grassberger, 1988). Another reason to consider ϵ of this kind would be to ensure that the adjacency matrices that we work with remain relatively sparse. This means that we avoid storage problems even when the sample size n is large. This can also be motivated from the point of view of discriminability; cf. (Beyer et al., 1999; Weber et al., 1998; Houle, 2013). If the dimensionality of the data is high, then distance values are less discriminative, in the sense that they tend to concentrate more around the mean of their distribution. Because of this, it makes sense to increase the strictness with which new connections are accepted as the sample size grows.

This has three important consequences. The first is that for the parameter-free estimator (8), with the finite sample correction described above should have squared bias $O(1/\log n)$. This means that the sequence m_n should be set to $O(\log n)$ to balance variance and squared bias. The proverbial *less is more* comes to mind: picking m_n large and averaging over many vertices leads to deceptive results, since the variance of the estimate is reduced, while the bias remains unchanged. This is an inherent feature of estimators obtained via inversion, but it is something that is invariably missed in the literature – estimates are strongly concentrated around biased estimates. This is undesirable from the point of view of uncertainty quantification; see also the next section. By doing this our estimator attains the minimax rate for this problem which is known to be logarithmic; cf. (Koltchinskii, 2000).

The second consequence follows from the first: setting $m_n = O(\log n)$ leads to an algorithm with complexity $O(n \log n)$. This is a great advantage over competing algorithms whose execution time typically scales like $O(n^2)$, sometimes like $O(D\, n^2)$; cf. Table 1 in (Eriksson and Crovella, 2012). Finally, since m_n is rather small compared with n, this means that we are estimating d based on the degrees of only a few vertices. By repeating the estimation for disjoint sets of vertices we can estimate the standard deviation of the estimator without a need for resampling. The conclusion is that picking small m_n is better, both from a theoretical and practical perspective.

In the next subsection we perform some numerical experiments to investigate more closely the consequences of different choices for m_n.

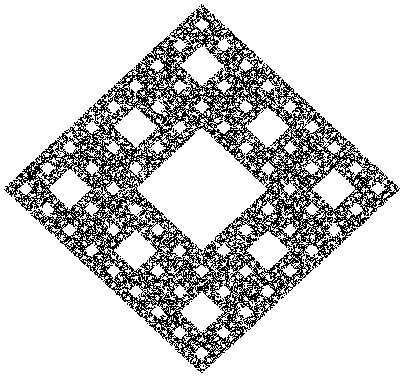


Figure 4: Example of $5 \cdot 10^4$ points sampled uniformly at random on a Sierpinski carpet. Points like these are used as design points in the simulations in this section.

7.2 Different Choices of m_n

In this section we look more closely at the choice of m_n by exemplifying the effect that the choice of this sequences has on: a) the bias, b) the variance, and c) the execution time. To have a nontrivial dimension we consider design points sampled uniformly at random on a Sierpinski carpet. This can be done in the following way. Consider

$$P_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{C} = \begin{bmatrix} 0 & 1/2 & 1 & 1/2 & 0 & -1/2 & -1 & -1/2 \\ 1 & 1/2 & 0 & -1/2 & -1 & -1/2 & 0 & 1/2 \end{bmatrix}.$$

Let $e_i = [0 \cdots 0\, 1\, 0 \cdots 0]^T$, $i = 1, \ldots, 8$, be unit vectors that have a 1 in the i-th position, and let $i_j \sim U\{1, \ldots, 8\}$, $j = 1, 2, \ldots$, be a sequences of independent, discrete uniform random variables taking values on $\{1, \ldots, 8\}$. A point can be drawn uniformly at random on a Sierpinski carpet as

$$P = P_0 + \boldsymbol{C} \sum_{j=1}^{\infty} 3^{-j} \cdot e_{i_j}.$$

Figure 4 depicts $5 \cdot 10^4$ points drawn according to this procedure. In practice we truncate the sum at 100 terms. (Note that this is accurate enough to get the neighbourhood matrices $\boldsymbol{A}_\epsilon$ exactly.) The correlation dimension of the Sierpinski carpet is, to the best of our knowledge, unknown, but its Hausdorff dimension is $\log 8 / \log 3 \approx 1.89$, which should provide a good indication to what the intrinsic dimension should be.

In this section we ran our algorithm for all combinations of $n = 10 \cdot 2^\gamma$, $\gamma \in \{0, 1, \ldots, 8\}$, and m_n either $\log(n)$ or n^γ, $\gamma \in \{1/4, 1/2, 3/4, 1\}$. The parameter ϵ was set to $\epsilon_n = \text{sd}\{\text{vect}(\boldsymbol{X})\}/\sqrt{\log(n+1)}$, where vect($\boldsymbol{X}$) represents the entries for $\boldsymbol{X}$ written as a vector; the standard deviation of the coordinates of the design points, although in practice unknown,

		m_n				
		$\log(n)$	$n^{1/4}$	$n^{1/2}$	$n^{3/4}$	n
n	10	1.1780 (0.78)	1.1243 (0.55)	1.0945 (0.39)	1.0877 (0.32)	1.0861 (0.26)
	20	1.2678 (0.54)	1.2528 (0.43)	1.2483 (0.33)	1.2409 (0.24)	1.2367 (0.18)
	30	1.3784 (0.49)	1.3718 (0.40)	1.3673 (0.26)	1.3631 (0.17)	1.3602 (0.12)
	40	1.4852 (0.44)	1.4781 (0.35)	1.4668 (0.20)	1.4660 (0.12)	1.4636 (0.08)
	160	1.5491 (0.32)	1.5441 (0.27)	1.5434 (0.15)	1.5398 (0.08)	1.5395 (0.05)
	320	1.6037 (0.28)	1.5963 (0.22)	1.5925 (0.12)	1.5930 (0.06)	1.5923 (0.03)
	640	1.6395 (0.27)	1.6376 (0.19)	1.6332 (0.09)	1.6320 (0.04)	1.6320 (0.02)
	1280	1.6595 (0.22)	1.6622 (0.18)	1.6596 (0.07)	1.6595 (0.03)	1.6597 (0.01)
	2560	1.6790 (0.22)	1.6770 (0.15)	1.6778 (0.06)	1.6768 (0.02)	1.6772 (0.01)

Table 1: Results for the estimation of the intrinsic dimension d for random Sierpinski carpet design points, for different combinations of n and m_n. For each combination, $\hat{d}_n$ was estimated 10^5 times; we display the mean estimate, and in parenthesis the standard deviation among the estimates.

was used here to ensure that ϵ is on the right scale. These simulations were repeated 10^5 times and the results are averaged.

Table 1 summarises the average and standard deviation of the estimates that were obtained for each combination of n and m_n. Irrespectively of the sequence m_n, it is clear that as n grows the estimates stabilise. This is in tune with our consistency result, also in that the reduction of the bias is rather slow. It is also clear from the results that the sequence m_n does not seem to have much influence on the quality of the estimate, particularly as n grows. This is also in tune with our results: larger m_n does increase the precision of the estimates of the probabilities $p_{\epsilon,1}$; what mainly determines the precision of the estimate of d is the bias introduced by the function g_ϵ, though. The effect of m_n on the standard deviation is also as expected: increasing either n or m_n generally leads to a decreased of the variability of the estimate. From this it might seem reasonable to set m_n to a large value (after all, it does reduce the variance of the estimator without reducing precision). There are however two good reasons to keep the growth of m_n slow.

The first reason is execution time. Figure 5 shows the evolution of the average execution time of the algorithm as a function of n, when $m_n = \log(n)$. For comparison, Table 2 shows the average execution time as a multiplier of the execution time for $m_n = \log(n)$, for the same value of n. In words: the numbers on the table indicate how much slower it is to set m_n to each choice, compared to just setting it to $\log(n)$.

The conclusion from Figure 5 is that the execution time for $m_n = \log(n)$ grows roughly linearly with n. On the other hand, from Table 2, for other choices of m_n the execution time quickly becomes prohibitive, particularly for faster-growing sequences m_n.

There is a second reason to set $m_n = O\{\log(n)\}$. Table 3 shows the percentage of runs in which d falls within 2 standard deviations from $\hat{d}_n$. The effect of m_n is clear. The confidence interval contains the true value of d, only when m_n grows appropriately slowly.

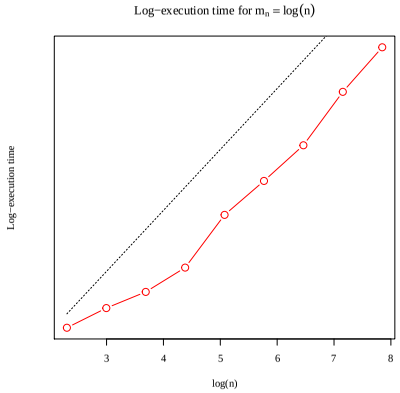


Figure 5: The evolution of the average execution time of our algorithm as a function of n when $m_n = \log(n)$. The plot is on a log-log scale. For reference, the dashed line represents $\log(n)$ plotted against $\log(n) - 10$. The execution time grows roughly linearly with n.

		m_n			
		$n^{1/4}$	$n^{1/2}$	$n^{3/4}$	n
n	10	1.13	1.65	2.14	3.20
	20	1.29	1.63	2.94	5.98
	40	1.34	2.63	5.59	12.72
	80	1.31	3.51	9.75	28.43
	160	1.20	3.57	12.10	42.46
	320	1.52	5.24	21.72	90.87
	640	1.91	8.05	39.19	194.22
	1280	1.53	9.03	53.86	325.12
	2560	1.99	12.68	89.80	638.54

Table 2: Average execution times for the algorithm. For each combination of n and sequence m_n, the respective entry in the table specifies how many times larger the execution time of the algorithm is compared to using $m_n = \log(n)$. For example, if we set $m_n = n$, then, when n is 2560, we have to wait more than 638 times longer for the algorithm to terminate than if we had used $m_n = \log(n)$.

Since the estimates are biased, if the sequence m_n grows too quickly, then the variance of the estimates is too small. The estimate $\hat{d}_n$ becomes overly concentrated around its

		m_n				
		$\log(n)$	$n^{1/4}$	$n^{1/2}$	$n^{3/4}$	n
n	10	84.75	71.68	47.59	30.78	15.16
	20	80.44	69.40	52.01	24.14	5.62
	40	83.08	76.07	48.46	14.38	0.60
	80	86.30	79.55	43.72	6.03	0.00
	160	82.07	76.13	36.11	1.45	0.00
	320	83.70	73.95	27.07	0.12	0.00
	640	85.73	73.30	17.81	0.00	0.00
	1280	83.21	75.57	10.24	0.00	0.00
	2560	84.95	71.52	4.52	0.00	0.00

Table 3: Percentage of the 10^5 runs where the true value of d is within 2 standard deviations of $\hat{d}_n$.

biased mean. In effect, because the bias and the variance are out of balance, the standard deviation fails to properly quantify the uncertainty in the estimate. Also remember that if m_n is small, then we can produce several estimates of d, from which we can estimate the standard deviation without a need for resampling.

To conclude, the sequence m_n affects the variance of the estimator $\hat{d}_n$, but it does not affect the bias of the estimate, which comes mostly from the function g_ϵ. Faster growing m_n therefore leads to estimates that are overly concentrated around their (biased) mean, so that their variability gives misleading information about the uncertainty in the estimate. Such choices of m_n also lead to a large computations cost. Therefore, setting $m_n = O\{\log(n)\}$ is arguably the correct choice to make.

7.3 Different Combinations of d and n

In this section we show how the estimator $\hat{d}_n$ behaves for different combinations of d and n. We set the distribution of the design points $X \sim N_d(0, \boldsymbol{I})$, for $d \in \{1, 2, 3, 4, 5, 10\}$, and chose $n \in \{10^3, 10^4, 10^5, 10^6, 10^7\}$; irrespectively of the dimension we always set $\epsilon = \epsilon_n = 4/(\log n)^{1/2}$. Based on the discussion from the previous section, the parameter m_n was set to $\max(1, \log n)$.

Table 4 below contains the results of estimating the intrinsic dimension d 10 times: for each combination of n and d we sampled an adjacency matrix, estimated d 10 times from 10 disjoint subsets of m_n vertices (chosen at random, without replacement); the average and (in brackets) the standard deviation of the 10 estimates make up the entries of the table. Note that since we only average over a relatively small number of vertices in the graph, producing Table 4 does not actually require any resampling; the same data set can be used (for each combination of n and d). Note also that for the choice of m_n above, the execution time of the algorithm is $O(n \log n)$, meaning that it is considerable faster than any alternative approach.

		n				
		10^3	10^4	10^5	10^6	10^7
d	1	0.49 (0.15)	0.58 (0.11)	0.54 (0.12)	0.63 (0.07)	0.71 (0.15)
	2	0.91 (0.18)	1.18 (0.19)	1.31 (0.26)	1.41 (0.25)	1.53 (0.17)
	3	1.76 (0.24)	1.99 (0.29)	2.23 (0.36)	2.28 (0.34)	2.51 (0.20)
	4	2.55 (0.50)	2.94 (0.30)	3.23 (0.42)	3.05 (0.44)	3.42 (0.32)
	5	3.24 (0.37)	3.61 (0.38)	3.96 (0.43)	4.31 (0.18)	4.39 (0.44)
	10	5.95 (0.61)	7.46 (0.53)	8.59 (0.70)	8.81 (0.70)	8.97 (0.75)

Table 4: Results for the estimation of d for Gaussian design points for different combinations of d and n. For each combination $\hat{d}_n$ was estimated 10 times; we display the mean estimate, and in parentheses the standard deviation among the estimates.

A few things are clear from the results in Table 4. As hinted in Section 7.1, the estimator tends to underestimate the true intrinsic dimension, especially if the dimension is large. However, as far as recuperating the integer dimension, the estimator performs well, especially considering that the data is entirely comprised of random fluctuations. The standard deviation also does a good job at quantifying the precision of the estimate.

We emphasise that the estimates are parameter free – one can improve the results with extra knowledge about the distribution of the data. We do this in the following subsection.

7.4 Non-canonical Choice of $g_\epsilon(d)$, and Bias Corrections

In this section we propose some modifications of our estimator aimed at removing (or at least mitigating) its bias. Since we know that the estimator $\hat{d}_n$ systematically underestimates d, a simple way of obtaining a "bias corrected" estimator is by scaling $\hat{d}_n$ up. We consider three different ways of doing this.

Following the discussion at the end of Section 4, where we justified that the bias should be of order $O(d \cdot \epsilon)$, we consider

$$\tilde{d}_n = \hat{d}_n \cdot \left\{1 + \frac{2}{\log(2)} \cdot \epsilon\right\}. \tag{13}$$

The motivation for considering this form is indeed the heuristic upper bound on the bias from Section 4. We replaced the factor $3/\log(2)$ with a $2/\log(2)$ to be conservative, since we do not expect the upper bound on the bias to be tight. We also use the fact that the ratio of expectations in the bound can reasonably be expected to be of order d since, on average, the gradient of the density should only be non-trivial along d independent directions. A multiplicative correction of this form seems to be appropriate not just for the Gaussian case, where the expectations that feature in the bound on the bias can be computed. Indeed, this form of the estimator performs quite well without any adjustments on subsequent experiments; see Table 8. With more knowledge about F one can certainly make more informed choices about the specific constants in (13). Table 5 shows the values that the bias corrected estimates (13) take.

		n				
		10^3	10^4	10^5	10^6	10^7
d	1	0.55 (0.17)	0.64 (0.12)	0.59 (0.13)	0.69 (0.08)	0.78 (0.16)
	2	1.03 (0.20)	1.32 (0.21)	1.44 (0.29)	1.54 (0.28)	1.66 (0.18)
	3	2.00 (0.27)	2.22 (0.32)	2.46 (0.40)	2.49 (0.37)	2.72 (0.22)
	4	2.89 (0.57)	3.28 (0.34)	3.55 (0.46)	3.34 (0.48)	3.71 (0.35)
	5	3.67 (0.42)	4.02 (0.43)	4.37 (0.48)	4.71 (0.19)	4.77 (0.48)
	10	6.74 (0.69)	8.31 (0.59)	9.47 (0.77)	9.63 (0.77)	9.74 (0.81)

Table 5: This table contains the results of repeating the experiment from the previous section, but we compensate for the bias by considering a multiplicative correction.

Comparing these results with those of Table 4, we see that indeed this correction seems to substantially improve the estimates.

One can also consider other choices for the function g_ϵ to reduce the bias. The idea is to use knowledge of the distribution of the design points to select a better suited candidate for this function. This also leads to more precise estimates of the intrinsic dimension. If $X \sim N_d(0, \boldsymbol{I})$, independent of $Y \sim N_d(0, \boldsymbol{I})$ then $Z = X - Y \sim N_d(0, 2\boldsymbol{I})$, so that, if we abbreviate $\mathcal{Z}_\epsilon = \{z \in \mathbb{R}^d : \|z\| \leq \epsilon\}$, then

$$p_{\epsilon,1} = \mathbb{P}(\|X - Y\| \leq \epsilon) = \mathbb{P}(\|Z\| \leq \epsilon) = \frac{1}{(4\pi)^{d/2}} \int_{\mathcal{Z}_\epsilon} e^{-\frac{1}{4}\|z\|^2}\, dz = \frac{v_\epsilon}{(4\pi)^{d/2}} \int_0^1 e^{-\frac{1}{4}u^2\epsilon^2}\, du,$$

where v_ϵ represents the volume of a d-dimensional Euclidean ball of radius ϵ. The integral above can be expressed in terms of the Gauss error function erf, so that

$$p_{\epsilon,1} = (4\pi)^{-d/2} \cdot v_\epsilon \cdot \frac{\sqrt{\pi}}{\epsilon} \cdot \mathrm{erf}(\epsilon/2), \qquad \text{whence} \qquad \frac{p_{2\epsilon,1}}{p_{\epsilon,1}} = 2^d \cdot \frac{\mathrm{erf}(\epsilon)/2}{\mathrm{erf}(\epsilon/2)}.$$

What is arguably the ideal choice for the function g_ϵ is then

$$g_\epsilon(d) = 2^d \cdot \frac{\mathrm{erf}(\epsilon)/2}{\mathrm{erf}(\epsilon/2)}, \qquad \text{leading to} \qquad \bar{d}_n = \hat{d}_n + \frac{\log\{\mathrm{erf}(\epsilon/2)\} - \log\{\mathrm{erf}(\epsilon)/2\}}{\log 2},$$

where $\hat{d}_n$ is the canonical estimator from (8). Note that using the exact function $g_\epsilon(d)$ does not entirely remove the bias of the estimate since $g_\epsilon(d)$ is not linear in d, and since $\hat{p}_{n,2\epsilon,1}/\hat{p}_{n,\epsilon,1}$ is not an unbiased estimator for $p_{2\epsilon,1}/p_{\epsilon,1}$. Note also that the correction factor depends only on ϵ, but not on d. To understand the effect of this new estimator based on the more precise choice of $g_\epsilon(d)$, we repeat the numerical experiment of the previous section now for the estimator $\bar{d}_n$ from the previous display; Table 6 summarises these results. Comparing Tables 4 and 6, it is clear that, as one would expect, a more informed choice for mapping g_ϵ considerably improves the estimates. The improvement provided by this additive correction typically performs somewhat worse than the multiplicative correction, but the difference

		n				
		10^3	10^4	10^5	10^6	10^7
d	1	1.04 (0.25)	1.04 (0.11)	1.06 (0.14)	1.00 (0.11)	0.99 (0.15)
	2	1.63 (0.22)	1.76 (0.33)	1.69 (0.24)	1.80 (0.27)	1.84 (0.17)
	3	2.42 (0.37)	2.45 (0.48)	2.54 (0.23)	2.48 (0.33)	2.73 (0.33)
	4	2.99 (0.41)	3.23 (0.35)	3.71 (0.38)	3.64 (0.32)	3.50 (0.28)
	5	3.98 (0.54)	4.37 (0.47)	4.49 (0.44)	4.36 (0.34)	4.30 (0.33)
	10	6.56 (0.67)	7.87 (0.60)	8.47 (0.79)	9.63 (0.82)	9.39 (0.71)

Table 6: This table contains the results of repeating the experiment from the previous section, but instead using the true underlying function $g_\epsilon(d)$ that maps d to the ratios $p_{2\epsilon,1}/p_{\epsilon,1}$.

		n				
		10^3	10^4	10^5	10^6	10^7
d	1	0.79	0.80	0.78	0.77	1.01
	2	1.27	1.56	1.83	1.91	1.87
	3	2.24	2.57	2.95	2.96	2.91
	4	3.55	3.54	4.07	3.93	4.06
	5	3.98	4.37	4.82	4.67	5.27
	10	7.17	8.52	9.99	10.21	10.47

Table 7: This table contains the results of repeating the experiment from the previous section, but we compensate for the bias by adding two standard deviations to the estimate.

seems to be small. This is most likely due to the fact that the multiplicative correction is better at compensating for the bias induced by the bias of our moment estimator of $p_{2\epsilon,1}/p_{\epsilon,1}$.

As a third and final alternative, one can also shift the estimates up by a factor depending on the standard deviations of the estimates. This is motivated by the fact that our choice of $m_n = O\{\log(n)\}$ balances squared bias and variance. (Again, the standard deviation is estimated using the same data as $\hat{d}_n$ without any need from resampling.) Table 7 presents the results of adding (since we otherwise under-estimate) two standard deviations to the corresponding estimate $\hat{d}_n$.

This correction performs well, particularly considering that it requires no extra information about the distribution of the data. Overall this seems to present the best correction for exactly this reason.

Although asymptotically any of the three corrections performs equally well, for finite samples, and with extra knowledge of the distribution of the design points, the multiplicative correction provides the best results. Without extra knowledge though, the variance based

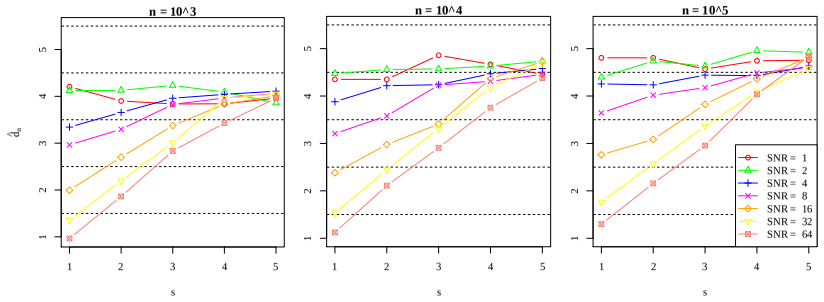


Figure 6: The estimator's sensitivity to noise. Each plot corresponds to a different sample size, from left to right: $n \in \{10^3, 10^4, 10^5\}$. The lines correspond to different estimates of the intrinsic dimension for different values of s, and the different colours correspond to different SNR.

correction still improves the canonical estimate $\hat{d}_n$ considerably. However, this does not change the fact that one will always need a large sample size n when the intrinsic dimension d is large.

7.5 Effect of the Noise

An important aspect to be taken into consideration is the robustness of the estimation procedure to the presence of noise. Irrespectively of the nature of the observations, if they are corrupted with enough noise, the estimation procedure will only detect the noise. In this respect we would like to see how sensitive the estimator is to the presence of noise. To see if this, we sampled design points $\boldsymbol{X}^{(s)}$ in $\mathbb{R}^5$ according to

$$X_i^{(s)} \overset{i.i.d.}{\sim} N_5\Big\{0,\ \mathrm{diag}\big(\overbrace{\sigma^2_{\text{signal}},\ldots,\sigma^2_{\text{signal}}}^{s},\overbrace{\sigma^2_{\text{noise}},\ldots,\sigma^2_{\text{noise}}}^{5-s}\big)\Big\},\quad i=1,\ldots,n,\ s=1,\ldots,5.$$

We set $\epsilon = \epsilon_n = \sigma_{\text{signal}}(2\log n)^{-1/2}$, and ran our estimation procedure on the adjacency matrix obtained from $\boldsymbol{X}^{(s)}$ for $s = 1,\ldots,5$. Each plot in Figure 6 corresponds to a different sample size $n \in \{10^3, 10^4, 10^5\}$. The coloured lines in each plot correspond to the 5 estimates $\hat{d}_n$, averaged over 10 disjoint sets of vertices. Different colours correspond to different signal to noise ratio (SNR); namely, we fixed $\sigma_{\text{signal}} = 1$ and chose σ_{noise} such that SNR $= \sigma^2_{\text{signal}}/\sigma^2_{\text{noise}} \in \{1, 2, 4, 8, 16, 32, 64\}$.

The estimator performs as intended. Consider first the rightmost plot in Figure 6, where n is the largest. When the SNR is 1, the intrinsic dimension is 5; in this case, we detect intrinsic dimension 5 for $s = 1,\ldots,5$, and the corresponding line is roughly the constant function $s \mapsto 5$. When the SNR is high, the mass of the design points is mostly concentrated on an s-dimensional subspace so that the intrinsic dimension is s; the resulting estimates are then close to the identity $s \mapsto s$. Between these extreme cases it is not clear exactly what the intrinsic dimension is. However, the lines corresponding to the estimates behave roughly monotonically. This means that the estimator is correctly picking up on the fact

	n	d	D	Data set	$\tilde{d}_n$
1	1000	1	3	Uniform on a helix	1.25 (0.16)
2	1000	2	3	Swiss roll	2.44 (0.59)
3	1000	5	5	Independent Gaussian	5.26 (1.15)
4	1000	7	8	Uniform on a sphere	6.66 (0.16)
5	5000	7	8	Uniform on a sphere	7.10 (0.08)
6	1000	12	12	Uniform on $[0,1]^{12}$	8.70 (0.61)
7	5000	12	12	Uniform on $[0,1]^{12}$	9.65 (0.35)
8	698	–	64×64	Isomap faces	4.22 (0.70)
9	481	–	512×480	Hands	2.14 (0.35)
10	7141	–	28×28	MNIST "3"	15.62 (0.11)
11	6824	–	28×28	MNIST "4"	15.65 (0.16)
12	6313	–	28×28	MNIST "5"	15.45 (0.07)

Table 8: Numerical experiments from Section 4.2 of (Kleindessner and von Luxburg, 2015) for different simulated and real data sets. For each data set we indicate the sample size, intrinsic dimension d, and ambient dimension D. The first seven data sets are simulated, while the last five are real. The intrinsic dimension of the real data sets is unknown.

that the mass of the distribution is concentrating on a lower dimensional subspace, and gradually changing to reflect this. For smaller sample sizes, as per the discussion from Section 7.1, the rightmost points in the plot are not reliable estimates.

7.6 Comparison with Other Estimators

We compare our estimator with competing approaches from the literature. To do this we repeat the numerical simulations of Section 4.2 of (Kleindessner and von Luxburg, 2015), which is conceptually close to our estimator. Our results can then be compared directly with the results from their Table 1. Note that we base our estimates on two (symmetric) adjacency matrices $\boldsymbol{A}_\epsilon$ and $\boldsymbol{A}_{2\epsilon}$, while the estimators from (Kleindessner and von Luxburg, 2015) are based on a (directed) k-nearest neighbour graph, so that both approaches require the same number of measurements. We consider twelve data sets; seven consist of simulated data, and five of real data. Table 8 contains the results.

The simulated data sets are self explanatory. The *Isomap faces* data set[4] contains 698 images ($D = 64 \times 64$ pixels) of a rendered face of a sculpture taken from different angles, under different lighting conditions. The *Hands* data set[5] contains 481 frames ($D = 512 \times 480$ pixels) from a video of a hand holding a rice bowl and revolving it while moving from right

4. http://isomap.stanford.edu/datasets.html
5. http://vasc.ri.cmu.edu//idb/html/motion/hand/index.html

to left. The *MNIST* data sets[6] contain 7141, 6824, and 6313 images ($D = 28 \times 28$ pixels) of handwritten digits "3", "4", and "5", respectively.

About the choice of ϵ for the experiments. For every synthetic data set we set $\epsilon = 4/(\log n)^{1/2}$ as before. For the real data sets this turned out to be inappropriate since the observations are on completely different scales. We scaled ϵ up so that the resulting adjacency matrices $\boldsymbol{A}_\epsilon$ and $\boldsymbol{A}_{2\epsilon}$ were neither complete graphs nor empty graphs, since this would lead to trivial estimates. This turn out to give us ϵ equal to 14, 17, 1600, 1600, and 1600, for data sets 8–12, respectively. In practice this can be achieved based on some notion of the scale of the observations, or based on a preliminary analysis, simulated data sets, or subsamples. In all experiments we set $m_n = 2 \log n$.

Our estimator compares quite well with the competing approaches, particularly as far as recuperating the integer dimension is concerned; the results are perhaps closer to the ones obtained by the estimator of (Levina and Bickel, 2004), but contrary to their estimator we do not require any knowledge about how the distance from a given point to its k-nearest-neighbour scales, or any distance data, or for any parameters to be set in the estimator. Also, the computational complexity of our estimator scales like $O(n \log n)$, considerably smaller than the typical $O(n^2)$. Furthermore, the results in each row of Table 8 were obtained from a single data set, without repeated sampling. Because of this we can provide standard deviations for our estimates without a need to resample the data, including for the real data sets. These standard deviations are much more realistic than the ones associated with competing approaches. We suspect that this is because for those it is difficult to properly balance the variance and the squared bias of the estimator, resulting in estimates that are overly concentrated around their biased means.

For the simulated data sets the true dimension is recuperated with good accuracy, with the exception of the sixth and seventh data set where the intrinsic dimension is relatively high. Indeed, as discussed in Section 7.1, in order to recuperate the intrinsic dimension consistently, the sample size should be quite large compared to the intrinsic dimension, since the minimax rates for the problem are logarithmic in n. This also comes from the fact that the support of the data set is rather unstructured, unlike for example data set 2, 4, and 5, or even 10, 11, and 12.

Less can be said about the results for the real data sets since the true intrinsic dimension is unknown. However, our results are comparable to the ones obtained by competing approaches. In particular, in all cases the intrinsic dimension is substantially smaller than the ambient dimension. For the *Isomap faces* data set we estimate the intrinsic dimension as 4; although the statue is 3-dimensional, the different lighting conditions may explain the fact that we detect an extra dimension in the data set. For the *Hands* data set we estimate the dimension as 2.14. One would probably expect the dimension to be 3, but given the symmetries in the hand and bowl, and that the images actually make up a smooth animation may explain the lower estimate. As for the *MNIST* data set, it seems reasonable that the estimates are not too different for the three digits. Also, if one were to parametrise the digits in terms of lengths, relative angles, and curvatures of the line segments, the estimate seems rather natural.

6. http://yann.lecun.com/exdb/mnist/

8. Discussion

In this paper we propose a method to estimate the intrinsic dimension of high-dimensional data sets. The approach combines the notion of correlation dimension with the doubling property of the Lebesgue measure to provide a computationally tractable estimator for data sets with (potentially) scale-dependent dimension. The approach does not require any parameters to be chosen, other than the scale at which one would like to estimate the dimension. This is particularly useful for data that live on manifolds whose dimension may be different at different scales, data sets corrupted with noise, or whenever not much is known about the distribution of the data. We compute the estimator's asymptotic distribution and rate of convergence. The rate that we obtain matches the logarithmic minimax rate for the (easier) problem where one has access to the observations – not just whether each pair of observations is close or not – and is therefore optimal. The estimator can be quickly evaluated in $O(n \log n)$ steps which is also an advantage over competing approaches, whose execution time typically scales like $O(n^2)$. Also in terms of storage there are advantages because the adjacency matrix that we base our estimator on will typically be sparse (since the underlying graph is embedded in a Euclidean space). Our results provide important information for algorithms commonly used to perform dimensionality reduction, learn manifolds, compress information, do statistical adaptation, and design efficient algorithms.

Distance-based estimators usually require rather (distribution specific) knowledge since one needs to know quite precisely how distances between perturbed observations scale. These also usually require certain bandwidth parameters to be defined without an automatic, or data driven way of picking them. Rather than assuming that we have access to the observations (or distances between them), we simply assume that we observe a graph encoding whether observations are close or not at the scale we are interested in. This is particularly relevant when dealing with large data sets. Modelling the resulting graph as a random connection model allows us to provide bounds on the probability of recuperating the correct intrinsic dimension of the data set, under a mild identifiability condition.

Our numerical experiments show that the intrinsic dimension can be well recuperated even without access to any distance information between the observations. Furthermore, our estimator properly picks up on the uncertainty of the estimate (the standard deviation of the estimator can be estimated without need for resampling), which can be used to avoid the estimator to be overly concentrated around its biased mean. Distance-based estimators tend to be much more costly, computationally. The estimator is parameter-free, but it can easily be improved by using any knowledge one may have about the distribution of the data. This is done by incorporating this knowledge into the choice of the function g_ϵ that features in the definition of the estimator.

Similar distance-free estimators such as those based on k-nearest-neighbours, seem to somewhat underperform in comparison. Although based on a similar idea, it seems like the number of k-nearest-neighbours to a fixed observations does not scale in a simple way with the dimension, making it more difficult to recuperate the dimension accurately from such kind of information alone. It is also not clear how to control exactly at which scale the dimension is being recuperated by choice of k. This suggests that in data sets with different dimensions at different scales, or data sets corrupted with noise, those estimators may return some form of "average dimension" across scales.

Another point to make is that one can also use our work for estimation of local dimension; cf. (Amsaleg et al., 2015). These are cases where the dimension of the underlying manifold of interest is not equal everywhere, and the space is instead some heterogeneous manifold. Our estimator is based on looking at just m_n vertices in the underlying neighbourhood graph. If we focus on vertices corresponding to the region of the manifold where we would like to estimate the local dimension (rather than picking these m_n vertices arbitrarily), then our estimator promptly delivers an estimate of this local dimension.

Acknowledgments

We would like to thank the Action Editor and the two anonymous Referees for pointing out inconsistencies in the notation of our initial submission, for suggesting references to other closely related work, and for their useful comments and suggestions.

Appendix

In this appendix we collect the proofs to our results.

Proof of Theorem 2: By the symmetry of $\boldsymbol{A}_\epsilon$ it suffices to control the sum

$$\sum_{i=1}^{m_n}\sum_{j=i+1}^{n}\{A_{\epsilon,i,j}-p_{\epsilon,1}\}=\sum_{i=1}^{m_n}\sum_{j=i+1}^{m_n}\{A_{\epsilon,i,j}-p_{\epsilon,1}\}+\sum_{i=1}^{m_n}\sum_{j=m_n+1}^{n}\{A_{\epsilon,i,j}-p_{\epsilon,1}\}=(A)+(B).$$

The proof now proceeds in two different ways: if $m_n = n$ then $(B) = 0$ and use the martingale central limit theorem to show that a properly rescaled version of (A) is asymptotically standard Gaussian[7]; if $m_n = o(n)$ then (B) dominates (A), so we show instead that (B) is asymptotically Gaussian.

Let $m_n = n$. By the symmetry of $\boldsymbol{A}_\epsilon$, and since $A_{\epsilon,i,i} = 0$, (A) can be rewritten as

$$\sum_{i=2}^{n}\left[\sum_{j=1}^{i-1}\{A_{\epsilon,i,j}-p_\epsilon(X_j)\}+(n-i)\{p_\epsilon(X_i)-p_{\epsilon,1}\}\right]+(n-1)\{p_\epsilon(X_1)-p_{\epsilon,1}\}.$$

(Note that we are just adding and subtracting $p_\epsilon(X_j)$ inside the double sum and reordering terms.) Denote the term in square brackets as $\eta_{n,i}$. The two terms in the previous display are uncorrelated; the variance of the second term is $(n-1)^2(p_{\epsilon,2}-p_{\epsilon,1}^2)$. Further, $\eta_{n,i}$ has expectation 0, and is measurable with respect to $\mathcal{F}_{n,i} = \mathcal{F}_i = \sigma(X_1,\ldots,X_i)$. Since $j < i$,

$$\mathbb{E}[\eta_{n,i}\mid\mathcal{F}_{i-1}]=\sum_{j=1}^{i-1}\{\mathbb{E}[A_{\epsilon,i,j}\mid X_j]-p_\epsilon(X_j)\}+(n-i)\{\mathbb{E}p_\epsilon(X_i)-p_{\epsilon,1}\}=0,\quad i=2,\ldots,n,$$

so that $\eta_{n,i}$ is a martingale increment. This means that if we define $S_{n,k}=\sum_{i=2}^{k}\eta_{n,i}$, then $\{S_{n,k},\mathcal{F}_{n,k},2\le k\le n,n\ge 2\}$ is a martingale array.

7. The case where $O(n) \le m_n < n$ is controlled in the same way.

We show that a scaled version of $S_{n,n}$ is asymptotically standard Gaussian, as $n \to \infty$. With the convention that $p_\epsilon(X_i, X_i) = p_\epsilon(X_i)$, by expanding the square and taking conditional expectations term-wise it holds that

$$\begin{aligned}\mathbb{E}\big[\eta_{n,i}^2 \mid \mathcal{F}_{i-1}\big] = &\sum_{j_1=1}^{i-1}\sum_{j_2=1}^{i-1}\big\{p_\epsilon(X_{j_1}, X_{j_2}) - p_\epsilon(X_{j_1})p_\epsilon(X_{j_2})\big\} + \\ &+ 2(n-i)\sum_{j=1}^{i-1}\big\{\mathbb{E}\big[A_{\epsilon,i,j}\, p_\epsilon(X_i) \mid X_j\big] - p_{\epsilon,1}\, p_\epsilon(X_j)\big\} + (n-i)^2(p_{\epsilon,2} - p_{\epsilon,1}^2),\end{aligned} \tag{14}$$

where we use the definitions of $p_\epsilon(X)$ and $p_\epsilon(X, Y)$, the fact that $p_\epsilon(X_j) \in \mathcal{F}_{n,i-1}$, for $j \le i-1$, and that X_i – or indeed $p_\epsilon(X_i)$ – is independent of $\mathcal{F}_{n,i-1}$, $i = 2, \dots, n$.

The expectation of $\eta_{n,i}^2$ is obtained by taking expectation of the previous display and noting that for any $j_1, j_2, j \ne i$,

$$\begin{aligned}\mathbb{E}\big\{p_\epsilon(X_{j_1}, X_{j_2}) - p_\epsilon(X_{j_1})p_\epsilon(X_{j_2})\big\} &= \{p_{\epsilon,1} - \mathbb{V}P_\epsilon(X)\}\, 1_{\{j_1=j_2\}} + p_{\epsilon,2} 1_{\{j_1\ne j_2\}} - p_{\epsilon,1}^2, \\ \mathbb{E}\big\{\mathbb{E}\big[A_{\epsilon,i,j}\, p_\epsilon(X_i) \mid X_j\big] - p_{\epsilon,1}\, p_\epsilon(X_j)\big\} &= p_{\epsilon,2} - p_{\epsilon,1}^2.\end{aligned}$$

Note that $\sum_{i=2}^n (i-1)$ is $n^2\{1/2 + o(1)\}$, and the sums $\sum_{i=2}^n (n-i)^2$, $2\sum_{i=2}^n (n-i)(i-1)$, and $\sum_{i=2}^n (i-1)(i-2)$ are all $n^3\{1/3 + o(1)\}$. We assume[8] that $p_{\epsilon,1} - p_{\epsilon,2} = o\{n(p_{\epsilon,2} - p_{\epsilon,1}^2)\}$, as $n \to \infty$ so that the variance of $S_{n,n}$ becomes

$$\mathbb{V}S_{n,n} = \sum_{i=2}^n \mathbb{E}\eta_{n,i}^2 = \left\{\frac{n^2}{2}(p_{\epsilon,1} - p_{\epsilon,1}^2) + n^3(p_{\epsilon,2} - p_{\epsilon,1}^2)\right\}\{1 + o(1)\} = n^3(p_{\epsilon,2} - p_{\epsilon,1}^2)\{1 + o(1)\}.$$

We therefore define $Z_{n,i} = S_{n,i}/\{\mathbb{V}S_{n,n}\}^{1/2}$, with increments $\xi_{n,i} = \eta_{n,i}/\{\mathbb{V}S_{n,n}\}^{1/2}$.

Based on the preceding, $\{Z_{n,i}, \mathcal{F}_{n,i}, 2 \le i \le n, n \ge 2\}$ is a (zero-mean, unit variance) martingale array. Since $\mathcal{F}_{n,i} = \mathcal{F}_i$, the σ-fields satisfy $\mathcal{F}_{n,i} \subseteq \mathcal{F}_{n+1,i}$, $2 \le i \le n$, $n \ge 2$, so that they are nested. We check the conditions of Corollary 3.1 of (Hall and Heyde, 2014):

$$\sum_{i=2}^n \mathbb{E}\big[\xi_{n,i}^2 \mathbf{1}_{\{|\xi_{n,i}| > \delta\}} \mid \mathcal{F}_{i-1}\big] \xrightarrow{P} 0, \quad \delta > 0, \qquad \text{and} \qquad \sum_{i=2}^n \mathbb{E}\big[\xi_{n,i}^2 \mid \mathcal{F}_{i-1}\big] \xrightarrow{P} 1. \tag{15}$$

To check the first condition we use $\mathbb{E}\big[|Z|^2 \mathbf{1}_{\{|Z|>\delta\}}\big] \le \mathbb{E}Z^4/\delta^2$ (by the Cauchy-Schwarz inequality, and Markov's inequality), and $(a+b)^4 \le 2^3(a^4 + b^4)$, $a, b \in \mathbb{R}$ (by Young's inequality). Therefore, checking (15) reduces to showing that as $n \to \infty$,

$$\sum_{i=2}^n \frac{\mathbb{E}\big[\big(\sum_{j=1}^{i-1}\{A_{\epsilon,i,j} - p_\epsilon(X_j)\}\big)^4 \mid \mathcal{F}_{i-1}\big]}{n^6(p_{\epsilon,2} - p_{\epsilon,1}^2)^2} \xrightarrow{P} 0, \qquad \sum_{i=2}^n \frac{(n-i)^4 \mathbb{E}\big[\big(p_\epsilon(X_i) - p_{\epsilon,1}\big)^4\big]}{n^6(p_{\epsilon,2} - p_{\epsilon,1}^2)^2} \longrightarrow 0.$$

If ϵ is fixed, then $p_{\epsilon,1}$ and $p_{\epsilon,2}$ are fixed, the numerators are of order n^5 and the conditions are met. Assume therefore that $\epsilon \to 0$ so that $p_\epsilon(x) \to 0$. Since the summands are positive, both conditions follow if we assume that $n(p_{\epsilon,2} - p_{\epsilon,1}^2)^2 \to \infty$.

8. This assumption is weaker than the assumption $n(p_{\epsilon,2} - p_{\epsilon,1}^2)^2 \to \infty$ which will be imposed later.

To show the second condition in (15) holds, it suffices to show that the sum over $i = 2, \dots, n$ of each of the three terms in (14) divided by $n^3(p_{\epsilon,2} - p_{\epsilon,1}^2)$ converges to 1/3, in probability; this is obvious for the third term, for the second term it follows easily from Chebyshev's inequality (since the terms in the sum are independent), so that only the convergence of the first term requires justification.

By Chebyshev's inequality and symmetry, it suffices to show that

$$\frac{\mathbb{E}\left(\sum_{i=2}^n \sum_{j_1=1}^{i-1}\sum_{j_2=1}^{i-1}\{p_\epsilon(X_{j_1}, X_{j_2}) - p_\epsilon(X_{j_1})p_\epsilon(X_{j_2}) - (p_{\epsilon,2} - p_{\epsilon,1}^2)\}\right)^2}{n^6(p_{\epsilon,2} - p_{\epsilon,1}^2)^2}$$

converges to zero. By Jensen's inequality, we can upper bound the previous display by

$$(n-1)\sum_{i=2}^n \frac{\mathbb{E}\left(\sum_{j_1=1}^{i-1}\sum_{j_2=1}^{i-1}\{p_\epsilon(X_{j_1}, X_{j_2}) - p_\epsilon(X_{j_1})p_\epsilon(X_{j_2}) - (p_{\epsilon,2} - p_{\epsilon,1}^2)\}\right)^2}{n^6(p_{\epsilon,2} - p_{\epsilon,1}^2)^2} \lesssim \frac{1}{n(p_{\epsilon,2} - p_{\epsilon,1}^2)^2}.$$

The last equality follows from the fact that of the $(i-1)^4$ involved in the square above, $(i-1)^2(i-3)^2$ of them have zero mean; the remaining terms are $O(i^3)$. The previous display is $o(1)$ since we assume $n(p_{\epsilon,2} - p_{\epsilon,1}^2)^2 \to \infty$.

To complete the proof, we consider the case $m_n = o(n)$. Adding and subtracting $p_\epsilon(X_i)$ inside the double sum, and then interchanging the two summations we can rewrite (B) as

$$\sum_{j=m_n+1}^n \left[\sum_{i=1}^{m_n}\big\{A_{\epsilon,i,j} - p_\epsilon(X_i)\big\}\right] + (n - m_n)\sum_{i=1}^{m_n}\big\{p_\epsilon(X_i) - p_{\epsilon,1}\big\} = (C) + (D).$$

Since the terms in (D) are independent, it is easy to see that the variance of (D) is $m_n(n - m_n)^2(p_{\epsilon,2} - p_{\epsilon,1}^2)$, so that it dominates the terms in (A), and is asymptotically Gaussian. It remains to show that the variance of (C) is dominated by that of (D).

Define the sum in square brackets in the previous display as ζ_j. These terms are uncorrelated: for indices $j_1 \ne j_2$, with $j_1, j_2 > i_1, i_2$,

$$\begin{aligned}\mathbb{E}\zeta_{j_1}\zeta_{j_2} = &\sum_{i=1}^{m_n}\mathbb{E}\big\{A_{\epsilon,i,j_1} - p(X_i)\big\}\big\{A_{\epsilon,i,j_2} - p(X_i)\big\} \\ &+ \sum_{i_1=1}^{m_n}\sum_{\substack{i_2=1 \\ i_2\ne i_1}}^{m_n}\mathbb{E}\big\{A_{\epsilon,i_1,j_1} - p_\epsilon(X_{i_1})\big\}\big\{A_{\epsilon,i_2,j_2} - p_\epsilon(X_{i_2})\big\},\end{aligned}$$

which is zero. From this we conclude that the variance of (C) is $O\{nm_n^2(p_{\epsilon,2} - p_{\epsilon,1}^2)\}$, and is therefore dominated by that of (D). This concludes the proof.

Proof of Theorem 3: We apply the delta method. For this we need to know the joint distribution of $\{\hat{p}_{n,\epsilon,1}, \hat{p}_{n,2\epsilon,1}\}$. This is established using the Cramér-Wold device by showing that for each $\alpha, \beta \in \mathbb{R}$, $\alpha\,\hat{p}_{n,\epsilon,1} + \beta\,\hat{p}_{n,2\epsilon,1}$ is asymptotically Gaussian. As in the proof of Theorem 2 we control

$$\sum_{i=1}^{m_n}\sum_{j=i+1}^n \big\{\alpha A_{\epsilon,i,j} + \beta A_{2\epsilon,i,j} - \alpha p_{\epsilon,1} - \beta p_{2\epsilon,1}\big\} = \sum_{i=1}^{m_n}\sum_{j=i+1}^n \big\{A_{\epsilon,i,j}^* - p_{\epsilon,1}^*\big\}.$$

Since $A^*_{\epsilon,i,j}$ is measurable with respect to X_i and X_j, and bounded, and since the previous display has mean zero, we can follow exactly the same steps as in Theorem 2 to show that the previous display is asymptotically Gaussian.

To fully specify the asymptotic distribution of $\{\hat{p}_{n,\epsilon,1}, \hat{p}_{n,2\epsilon,1}\}$ it remains to compute the covariance of the estimates $\hat{p}_{n,\epsilon,1}$ and $\hat{p}_{n,2\epsilon,1}$. Define $p_{\epsilon_1,\epsilon_2,2} = \mathbb{P}\{r(X,Z) \leq \epsilon_1, r(Z,Y) \leq \epsilon_2\}$, and assume that $p_{\epsilon,1} - p_{\epsilon,1} \cdot p_{2\epsilon,1} = o\big[n\{p_{\epsilon,2\epsilon,2} - p_{\epsilon,1} \cdot p_{2\epsilon,1}\}\big]$. Simple computations then give

$$\mathbb{V}\big\{\hat{p}_{n,\epsilon,1},\, \hat{p}_{n,2\epsilon,1}\big\} = \frac{(n+3m_n)\{p_{\epsilon,2\epsilon,2} - p_{\epsilon,1}p_{2\epsilon,1}\}}{n \cdot m_n}\{1 + o(1)\}.$$

Write $A_\epsilon = p_{\epsilon,2} - p_{\epsilon,1}^2$, $B_\epsilon = p_{2\epsilon,2} - p_{2\epsilon,1}^2$, and $C_\epsilon = (n+3m_n)\{p_{\epsilon,2\epsilon,2} - p_{\epsilon,1} \cdot p_{2\epsilon,1}\}/n$. Conclude that under the assumptions of Theorem 2 (verified also with ϵ replaced with 2ϵ),w

$$m_n^{1/2}\left\{\begin{bmatrix}\hat{p}_{n,\epsilon,1}\\ \hat{p}_{n,2\epsilon,1}\end{bmatrix} - \begin{bmatrix}p_{\epsilon,1}\\ p_{2\epsilon,1}\end{bmatrix}\right\} \xrightarrow{d} N\left\{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}A_\epsilon & C_\epsilon\\ C_\epsilon & B_\epsilon\end{bmatrix}\right\}.$$

In the general case of the implicit estimates in (7), consider $d \mapsto g_\epsilon(d)$; let g_ϵ^{-1} represent the inverse of g_ϵ (which exists, at least in a neighbourhood of d). The asymptotic distribution of the estimator $\hat{d}_n$ is obtained by applying the delta method to the previous display using the function $(\alpha,\beta) \mapsto g_\epsilon^{-1}(\beta/\alpha)$, whose gradient is $\{\partial \log g_\epsilon(d)/\partial d\}^{-1}[-1/\alpha, 1/\beta]^T$. This delivers the asymptotic distribution of

$$g_\epsilon^{-1}(\hat{p}_{n,\epsilon,1}/\hat{p}_{n,2\epsilon,1}) - g_\epsilon^{-1}(p_{\epsilon,1}/p_{2\epsilon,1}) = \hat{d}_n - d + d - g_\epsilon^{-1}(p_{\epsilon,1}/p_{2\epsilon,1}) = \hat{d}_n - d + o(m_n^{-1/2}),$$

where the last equality follows from the bias condition (B). We conclude that

$$m_n^{1/2}\left\{\hat{d}_n - d\right\} \xrightarrow{d} N\left\{0,\, \left\{\frac{\partial \log g_\epsilon(d)}{\partial d}\right\}^{-2}\left[\frac{A_\epsilon}{p_{\epsilon,1}^2} + \frac{B_\epsilon}{p_{2\epsilon,1}^2} - \frac{2C_\epsilon}{p_{\epsilon,1}\, p_{2\epsilon,1}}\right]\right\}.$$

Replacing A_ϵ, B_ϵ, and C_ϵ, the component of the variance in square brackets is

$$6\frac{m_n}{n} + \frac{p_{\epsilon,1}^2 p_{2\epsilon,2} + p_{2\epsilon,1}^2 p_{\epsilon,2}}{p_{\epsilon,1}^2 p_{2\epsilon,1}^2} - 2\left(1 + 3\frac{m_n}{n}\right)\frac{p_{\epsilon,2\epsilon,2}}{p_{\epsilon,1}p_{2\epsilon,1}}. \tag{16}$$

For the explicit estimator from (8), the delta method is used with the function $(\alpha,\beta) \mapsto \log(\beta/\alpha)/\log(2) = g_\epsilon^{-1}(\beta/\alpha)$, such that the scaling in the variance becomes $\log(2)^{-2}$.

References

Bruno Abrahao and Robert Kleinberg. On the internet delay space dimensionality. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, pages 157–168. ACM, 2008.

Hirotugu Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E Houle, Kenichi Kawarabayashi, and Michael Nett. Estimating local intrinsic dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 29–38. ACM, 2015.

Robert S Bennett. The intrinsic dimensionality of signal collections. *Information Theory, IEEE Transactions on*, 15(5):517–525, 1969.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer, 1999.

Peter J Bickel and Elizaveta Levina. Some theory for Fisher's linear discriminant function,'naive Bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, pages 989–1010, 2004.

Jörg Bruske and Gerald Sommer. Intrinsic dimensionality estimation with optimally topology preserving maps. *IEEE Transactions on pattern analysis and machine intelligence*, 20(5):572–575, 1998.

Christopher JC Burges. *Dimension reduction: A guided tour*. Now Publishers Inc, 2010.

Francesco Camastra and Alessandro Vinciarelli. Estimating the intrinsic dimension of data with a fractal-based method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(10):1404–1407, 2002.

Kevin M Carter, Raviv Raich, and Alfred O Hero III. On local intrinsic dimension estimation and its applications. *Signal Processing, IEEE Transactions on*, 58(2):650–663, 2010.

Jose A Costa and Alfred O Hero. Learning intrinsic dimension and intrinsic entropy of high-dimensional datasets. In *Signal Processing Conference, 2004 12th European*, pages 369–372. IEEE, 2004.

Trevor F Cox and Michael AA Cox. *Multidimensional scaling*. CRC press, 2000.

David L Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100 (10):5591–5596, 2003.

Paul Erdős and Alfréd Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

Brian Eriksson and Mark Crovella. Estimating intrinsic dimension via clustering. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 760–763. IEEE, 2012.

Jianqing Fan and Yingying Fan. High dimensional classification using features annealed independence rules. *Annals of Statistics*, 36(6):2605, 2008.

Amir M Farahmand, Csaba Szepesvári, and Jean-Yves Audibert. Manifold-adaptive dimension estimation. In *Proceedings of the 24th International Conference on Machine Learning (ICML-07)*, pages 265–272, 2007.

Imola K Fodor. A survey of dimension reduction techniques, 2002.

Keinosuke Fukunaga. Intrinsic dimensionality extraction. *Handbook of Statistics*, 2:347–360, 1982.

Keinosuke Fukunaga and David R Olsen. An algorithm for finding intrinsic dimensionality of data. *Computers, IEEE Transactions on*, 100(2):176–183, 1971.

Evarist Giné and Vladimir I Koltchinskii. Empirical graph laplacian approximation of laplace–beltrami operators: Large sample results. In *High dimensional probability*, pages 238–259. Institute of Mathematical Statistics, 2006.

Peter Grassberger. Finite sample corrections to entropy and dimension estimates. *Physics Letters A*, 128(6):369–373, 1988.

Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. *Physica D: Nonlinear Phenomena*, 9(1-2):189–208, 1983.

Peter Grassberger and Itamar Procaccia. Measuring the strangeness of strange attractors. In *The Theory of Chaotic Attractors*, pages 170–189. Springer, 2004.

Geoffrey R Grimmett. *Percolation*, volume 321 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1999.

Peter Hall and Christopher C Heyde. *Martingale limit theory and its application*. Academic press, 2014.

Matthias Hein and Jean-Yves Audibert. Intrinsic dimensionality estimation of submanifolds in r d. In *Proceedings of the 22nd international conference on Machine learning*, pages 289–296. ACM, 2005.

Michael E Houle. Dimensionality, discriminability, density and distance distributions. In *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*, pages 468–473. IEEE, 2013.

Xiaoming Huo and Jihong Chen. Local linear projection (llp). In *Proc. of First Workshop on Genomic Signal Processing and Statistics (GENSIPS)*, 2002.

Aapo Hyviirinen, Juha Karhunen, and Erki Oja. Independent component analysis. *Wileyand Sons*, 2001.

David R. Karger and Matthias Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pages 741–750, New York, NY, USA, 2002. ACM.

Balázs Kégl. Intrinsic dimension estimation using packing numbers. In *Advances in neural information processing systems*, pages 681–688, 2002.

Jisu Kim, Alessandro Rinaldo, and Larry Wasserman. Minimax rates for estimating the dimension of a manifold. *arXiv:1605.01011 [math.ST]*, 2016.

Matthäus Kleindessner and Ulrike von Luxburg. Dimensionality estimation without distances. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 471–479, 2015.

Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

Vladimir I Koltchinskii. Empirical geometry of multivariate data: a deconvolution approach. *Annals of Statistics*, pages 591–629, 2000.

Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964a.

Joseph B Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129, 1964b.

Harri Lähdesmäki, Olli Yli-Harja, Wei Zhang, and Ilya Shmulevich. Intrinsic dimensionality in gene expression analysis. *Proc. GENSIPS*, pages 1–2, 2005.

John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

Nikolai Leonenko, Luc Pronzato, Vippal Savani, et al. A class of Rényi information estimators for multidimensional densities. *The Annals of Statistics*, 36(5):2153–2182, 2008.

Elizaveta Levina and Peter J Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in neural information processing systems*, pages 777–784, 2004.

Ronald Meester and Rahul Roy. *Continuum Percolation*, volume 119. Cambridge University Press, 1996.

Mathew D Penrose. On a continuum percolation model. *Advances in applied probability*, pages 536–556, 1991.

Karl W Pettis, Thomas A Bailey, Anil K Jain, and Richard C Dubes. An intrinsic dimensionality estimator from near-neighbor information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):25–37, 1979.

Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461–464, 1978.

Roger N Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. i. *Psychometrika*, 27(2):125–140, 1962a.

Roger N Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. ii. *Psychometrika*, 27(3):219–246, 1962b.

Kumar Sricharan, Raviv Raich, and Alfred O Hero III. Optimized intrinsic dimension estimator using nearest neighbor graphs. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5418–5421. IEEE, 2010.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Gerard V Trunk. Statistical estimation of the intrinsic dimensionality of data collections. *Information and Control*, 12(5):508–525, 1968.

Michel Verleysen, Eric de Bodt, and Amaury Lendasse. Forecasting financial time series through intrinsic dimension estimation and non-linear data projection. In *Engineering Applications of Bio-Inspired Artificial Neural Networks*, pages 596–605. Springer, 1999.

Peter J Verveer and Robert P W Duin. An evaluation of intrinsic dimensionality estimators. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):81–86, 1995.

Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *24th International Conference on 24th International Conference on Very Large Databases*, volume 98, pages 194–205, 1998.

Journal of Machine Learning Research 18 (2017) 1-58 Submitted 9/15; Revised 3/17; Published 12/17

Bayesian Inference for Spatio-temporal Spike-and-Slab Priors

Michael Riis Andersen* michael.andersen@aalto.fi
Aki Vehtari Aki.Vehtari@aalto.fi
Helsinki Institute for Information Technology HIIT
Department of Computer Science, Aalto University
P.O. Box 15400, FI-00076, Finland

Ole Winther olwi@dtu.dk
Lars Kai Hansen lkai@dtu.dk
Department of Applied Mathematics and Computer Science
Technical University of Denmark
DK-2800 Kgs. Lyngby, Denmark

Editor: Lawrence Carin

Abstract

In this work, we address the problem of solving a series of underdetermined linear inverse problemblems subject to a sparsity constraint. We generalize the spike-and-slab prior distribution to encode a priori correlation of the support of the solution in both space and time by imposing a transformed Gaussian process on the spike-and-slab probabilities. An expectation propagation (EP) algorithm for posterior inference under the proposed model is derived. For large scale problems, the standard EP algorithm can be prohibitively slow. We therefore introduce three different approximation schemes to reduce the computational complexity. Finally, we demonstrate the proposed model using numerical experiments based on both synthetic and real data sets.

Keywords: Linear inverse problems, bayesian inference, expectation propagation, sparsity-promoting priors, spike-and-slab priors

1. Introduction

Many problems of practical interest in machine learning involve a high dimensional feature space and a relatively small number of observations. Inference is in general difficult for such underdetermined problems due to high variance and therefore regularization is often the key to extracting meaningful information from such problems (Tibshirani, 1994). The classical approach is Tikhonov regularization (also known as ℓ_2 regularization), but during the last few decades sparsity has been an increasingly popular choice of regularization for many problems, giving rise to methods such as the LASSO (Tibshirani, 1994), Sparse Bayesian Learning (Tipping, 2001) and sparsity promoting priors (Mitchell and Beauchamp, 1988).

In this work, we address the problem of finding sparse solutions to linear inverse problems of the form

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^D$ is the desired solution, $\boldsymbol{y} \in \mathbb{R}^N$ is an observed measurement vector, $\boldsymbol{A} \in \mathbb{R}^{N\times D}$ is a known forward model and $\boldsymbol{e} \in \mathbb{R}^N$ is additive measurement noise. We are mainly interested in the underdetermined regime, where the number of observations is smaller than the number of unknowns, that is $N < D$. In the sparse recovery literature, it has been shown that the sparsity constraint is crucial for recovering $\boldsymbol{x}$ from a small set of linear measurements (Candès et al., 2006). Furthermore, the ratio between the number non-zero coefficients $K = \|\boldsymbol{x}\|_0$ and the dimension D dictates the required number of measurements N for robust reconstruction of $\boldsymbol{x}$ and this relationship has given rise to so-called *phase transition curves* (Donoho and Tanner, 2010). A large body of research has been dedicated to improve these phase transition curves and these endeavors have lead to the concepts of *multiple measurement vectors* (Cotter et al., 2005) and *structured sparsity* (Huang et al., 2009).

The multiple measurement vector problem (MMV) is a natural extension of eq. (1), where multiple measurements $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T$ are observed and assumed to be generated from a series of signals $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T$, which share a common sparsity pattern. In matrix notation, we can write the problem as

$$\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E}, \tag{2}$$

where the desired solution is now a matrix $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \ldots & \boldsymbol{x}_T\end{bmatrix} \in \mathbb{R}^{D\times T}$ and similarly for the measurement matrix $\boldsymbol{Y} \in \mathbb{R}^{N\times T}$ and the noise term $\boldsymbol{E} \in \mathbb{R}^{N\times T}$. The assumption of *joint sparsity* allows one to recover $\boldsymbol{X}$ with significantly fewer observations compared to solving each of the T inverse problems in eq. (1) separately (Cotter et al., 2005). The MMV approach has also been generalized to problems, where the sparsity pattern is evolving slowly in time (Ziniel and Schniter, 2013a). Structured sparsity, on the other hand, is a generalization of simple sparsity and seeks to exploit the fact that the sparsity patterns of many natural signals contain a richer structure than simple sparsity, for example, *group sparsity* (Jacob et al., 2009b) or *cluster structured sparsity* (Yu et al., 2012).

In this paper, we combine these two approaches and focus on problems, where the sparsity pattern of $\boldsymbol{X}$ exhibits a spatio-temporal structure. In particular, we assume that the row and column indices of $\boldsymbol{X}$ can be associated with a set of spatial and temporal coordinates, respectively. This can equivalently be interpreted as a sparse linear regression problem, where the support of the regressors is correlated in both space and time. Applications of such a model include dynamic compressed sensing (Ziniel and Schniter, 2013a), background subtraction in computer vision (Cevher et al., 2009) and EEG source localization problem (Baillet et al., 2001).

We take a Bayesian approach to modeling this structure since it provides a natural way of incorporating such prior knowledge in a model. In particular, we propose a hierarchical probabilistic model for $\boldsymbol{X}$ based on the so-called spike-and-slab prior (Mitchell and Beauchamp,

∗. Work done mainly while at Department of Applied Mathematics and Computer Science, Technical University of Denmark

©2017 Michael Riis Andersen, Aki Vehtari, Ole Winther and Lars Kai Hansen.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/15-464.html.

1988). We introduce a smooth latent variable controlling the spatio-temporal structure of the support of $\boldsymbol{X}$ by extending the work by Andersen et al. (2014). We aim for full Bayesian inference under the proposed probabilistic model, but inference w.r.t. the exact posterior distribution of interest is intractable. Instead we resort to approximate inference using Expectation Propagation (Minka, 2001; Opper and Winther, 2000), which has been shown to provide accurate inference for spike-and-slab priors (Hernández-Lobato et al., 2013; Hernandez-Lobato et al., 2010; Jylänki et al., 2014; Peltola et al., 2014). Our model formulation is generic and generalizes easily to other types of observations. In particular, we also combine the proposed prior with a probit observation model to model binary observations in a sparse linear classification setting.

The contribution of this paper is three-fold. First we extend the structured spike-and-slab prior and the associated EP inference scheme to incorporate both spatial and temporal smoothness of the support. However, the computational complexity of the resulting EP algorithm is prohibitively slow for problems of even moderate sizes of signal dimension D and length T. To alleviate the computational bottleneck of the EP algorithm we propose three different approximation schemes. Finally, we discuss several approaches for learning the hyperparameters and evaluate them based on synthetic and real data sets.

1.1 Related Work

In this section, we briefly review some of the most common approaches to simple sparsity and their generalization to structured sparsity. The classical approach to sparsity is the LASSO (Tibshirani, 1994), which operates by optimizing a least squares cost function augmented with an ℓ_1 penalty on the regression weights. Several extensions have been proposed in the literature to generalize the LASSO to the structured sparsity setting, examples include group and graph LASSO (Jacob et al., 2009b). From a probabilistic perspective sparsity can be encouraged through the use of *sparsity-promoting priors*. A non-exhaustive list of sparsity-promoting priors includes the Laplace prior (Park and Casella, 2008), Automatic Relevance Determination prior (Neal, 1996), the horseshoe prior (Carvalho et al., 2009) and the spike-and-slab prior (Mitchell and Beauchamp, 1988). All of these were originally designed to enforce simply sparsity, but they have all been generalized to the structured sparsity setting. The general strategy is to extend univariate densities to correlated multivariate densities by augmenting the models with a latent multivariate variable, where the correlation structure can be controlled explicitly, for example, using Markov Random Fields (Cevher et al., 2009; Hernandez-Lobato et al., 2011) or multivariate Gaussian distributions (Engelhardt and Adams, 2014). Here we limit ourselves to consider the latter.

From a probabilistic perspective, optimizing with an ℓ_1 regularization term can be interpreted as maximum a posteriori (MAP) inference under an i.i.d. Laplace prior distribution on the regression weights (Park and Casella, 2008). The univariate Laplace prior has been generalized to the multivariate Laplace (MVL) distribution, which couples the prior variance of the regression weights through a scale mixture formulation (Gerven et al., 2009).

Another approach is Automatic Relevance Determination (ARD) (Neal, 1996), which works

by imposing independent zero mean Gaussian priors with individual precision parameters on the regression weights. These precision parameters are then optimized using a maximum likelihood type II and the idea is then that the precision parameters of irrelevant features will approach infinity and thereby forcing the weights of the irrelevant features to zero. Wu et al. (2014b) extend the ARD framework to promote spatial sparsity by introducing a latent multivariate Gaussian distribution to impose spatial structure onto the precision parameters of ARD giving rise to *dependent relevance determination priors*.

The horseshoe prior is defined as a scale mixture of Gaussians, where a half-Cauchy distribution is used as prior for the standard deviation of the Gaussian density (Carvalho et al., 2009). The resulting density has two very appealing properties for promoting sparsity, namely heavy tails and an infinitely large spike at zero. A generalization to the multivariate case has been proposed by Hernández-Lobato and Hernández-Lobato (2013).

The spike-and-slab prior is an increasingly popular choice of sparsity promoting prior and is given by a binary mixture of two components: a Dirac delta distribution (spike) at zero and Gaussian distribution (slab) (Mitchell and Beauchamp, 1988; Carbonetto and Stephens, 2012). The spike-and-slab prior has been generalized to the group setting by Hernández-Lobato et al. (2013), to clustered sparsity setting by Yu et al. (2012) and spatial structures by Andersen et al. (2014), Nathoo et al. (2014), and Engelhardt and Adams (2014). Nathoo et al. (2014) induce the spatial structure using basis functions and Andersen et al. (2014) impose the structure using a multivariate Gaussian density. The latter is the starting point of this work.

Our work is closely related to the work on the multivariate Laplace prior (MVL) (Gerven et al., 2009) as mentioned above and the work on the network-based sparse Bayesian classification algorithm (NBSBC) (Hernandez-Lobato et al., 2011). The former also uses EP for approximating the posterior distribution of a Gaussian linear model with the MVL prior, where the structure of the support is encoded into the model using a sparse precision matrix. The NBSBC method also uses EP to approximate the posterior distribution of linear model with coupled spike-and-slab priors, but the structure of the support is encoded in a network using a Markov Random Field (MRF) prior. In contrast, we can inject a priori knowledge of the structure into the model using generic covariance functions rather than clique potentials as in the MRF-based models, which makes it easier to interpret interesting quantities like the characteristic lengthscale etc.

1.2 Structure of Paper

This paper is organized as follows. In Section 2 we review the structured spike-and-slab prior and in Section 3 we discuss different ways of extending the model to include the temporal structure as well. After introducing the models we propose an algorithm for approximate inference based on the expectation propagation (EP) framework. We review the basics of EP and describe the proposed algorithm in Section 4. In Section 5 we introduce three simple approximation schemes to speed of the inference process and discuss their properties.

Finally, in Section 7 we demonstrate the proposed method using synthetic and real data sets.

1.3 Notation

We use bold uppercase letters to denote matrices and bold lowercase letters to denote vectors. Unless stated otherwise, all vectors are column vectors. Furthermore, we use the notation $\boldsymbol{a}_{n,\cdot} \in \mathbb{R}^{1\times D}$ and $\boldsymbol{a}_{\cdot,i} \in \mathbb{R}^{N\times 1}$ for the n'th row and i'th column in the matrix $\boldsymbol{A} \in \mathbb{R}^{N\times D}$, respectively. $[K]$ denotes the set of integers from 1 to K, that is $[K] = \{1, 2, .., K\}$. We use the notation $\boldsymbol{a} \circ \boldsymbol{b}$ to denote the element-wise Hadamard product of $\boldsymbol{a}$ and $\boldsymbol{b}$ and $\boldsymbol{A} \otimes \boldsymbol{B} \in \mathbb{R}^{MN\times MN}$ for the Kronecker product of matrices $\boldsymbol{A} \in \mathbf{R}^{M\times M}$ and $\boldsymbol{B} \in \mathbf{R}^{N\times N}$. We use $\mathcal{N}(\boldsymbol{x}|\boldsymbol{m}, \boldsymbol{V})$ to denote a multivariate Gaussian density over $\boldsymbol{x}$ with mean vector $\boldsymbol{m}$ and covariance matrix $\boldsymbol{V}$ and $\mathrm{Ber}(z|p)$ denotes a Bernoulli distribution on z with probability of $p(z = 1) = p$.

2. The Structured Spike-and-Slab Prior

The purpose of this section is to describe the *structured spike-and-slab prior* (Andersen et al., 2014), but first we briefly review the conventional spike-and-slab prior (Mitchell and Beauchamp, 1988). For $\boldsymbol{x} \in \mathbb{R}^D$, the spike-and-slab prior distribution is given by

$$p(\boldsymbol{x}|p_0, \rho_0, \tau_0) = \prod_{i=1}^{D} \left[(1 - p_0)\delta(x_i) + p_0\mathcal{N}(x_i|\rho_0, \tau_0)\right], \tag{3}$$

where $\delta(x)$ is the Dirac delta function and p_0, ρ_0 and τ_0 are hyperparameters. In particular, p_0 is the prior probability of a given variable being active, that is $p(x_i \neq 0) = p_0$, and ρ_0, τ_0 are the prior mean and variance, respectively, of the active variables. The spike-and-slab prior in eq. (3) is also known as the Bernoulli-Gaussian prior since the prior can decomposed as

$$p(\boldsymbol{x}|p_0, \rho_0, \tau_0) = \sum_{\boldsymbol{z}} \prod_{i=1}^{D} \left[(1 - z_i)\delta(x_i) + z_i\mathcal{N}(x_i|\rho_0, \tau_0)\right] \prod_{i=1}^{D} \mathrm{Ber}(z_i|p_0), \tag{4}$$

where the sum is over all the binary variables z_i for $i \in [D]$. Thus, the latent binary variable $z_i \in \{0, 1\}$ can interpreted as an indicator variable for the event $x_i \neq 0$. We will refer to $\boldsymbol{z}$ as the *sparsity pattern* or the *support* of $\boldsymbol{x}$. In eq. (3) and (4) we condition explicitly on the hyperparameters p_0, ρ_0, τ_0, but to ease the notation we will omit this in the remainder of this paper.

The variables x_i and x_j are assumed to be independent for $i \neq j$ as seen in eq. (3) and (4). This implies that the number of active variables follows a binomial distribution and hence, the marginal probability of x_i and x_j being jointly active, is given by $p(x_i \neq 0, x_j \neq 0) = p_0^2$ for all $i \neq j$. However, in many applications the variables $\{x_k\}_{k=1}^D$ might a priori have an underlying topographic relationship such as a spatial or temporal structure. Without loss of generality we will assume a spatial relationship, where $\boldsymbol{d}_i$ denotes the spatial coordinates of x_i. For such models, it is often a reasonable assumption that $p(x_i \neq 0, x_j \neq 0)$ should

depend on $\|\boldsymbol{d}_i - \boldsymbol{d}_j\|$. For instance, neighboring voxels in functional magnetic resonance imaging (fMRI) analysis (Penny et al., 2005) are often more likely to be active simultaneously compared to two voxels far apart. Such a priori knowledge is neglected by the conventional spike-and-slab prior in eq. (3).

The structured spike-and-slab model is capable of modeling such structure and is given in terms of a hierarchical model

$$p(\boldsymbol{x}|\boldsymbol{z}) = \prod_{i=1}^{D} \left[(1 - z_i)\,\delta(x_i) + z_i\mathcal{N}\left(x_i\middle|\rho_0, \tau_0\right)\right], \tag{5}$$

$$p(\boldsymbol{z}|\boldsymbol{\gamma}) = \prod_{i=1}^{D} \mathrm{Ber}\left(z_i\middle|\phi(\gamma_i)\right), \qquad \phi : \mathbb{R} \to (0, 1), \tag{6}$$

$$p(\boldsymbol{\gamma}) = \mathcal{N}\left(\boldsymbol{\gamma}\middle|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right), \tag{7}$$

where $\boldsymbol{\gamma}$ is a latent variable controlling the structure of the sparsity pattern. Using this model prior knowledge of the structure of the sparsity pattern can be encoded using $\boldsymbol{\mu}_0$ and $\boldsymbol{\Sigma}_0$. The mean value $\boldsymbol{\mu}_0$ controls the expected degree of sparsity and the covariance matrix $\boldsymbol{\Sigma}_0$ determines the prior correlation of the support. The map $\phi : \mathbb{R} \to (0, 1)$ serves the purpose of squeezing γ_i into the unit interval and thereby $\phi(\gamma_i)$ represents the probability of $z_i = 1$. Here we choose ϕ to be the standard normal cumulative distribution function (CDF), but other choices, such as the logistic function, are also possible.

Using this formulation, the marginal prior probability of the i'th variable being active is given by

$$p(z_i = 1) = \int p(z_i = 1|\gamma_i)p(\gamma_i)\mathrm{d}\gamma_i = \int \phi(\gamma_i)\mathcal{N}\left(\gamma_i\middle|\mu_i, \Sigma_{0,ii}\right)\mathrm{d}\gamma_i = \phi\left(\frac{\mu_i}{\sqrt{1 + \Sigma_{0,ii}}}\right). \tag{8}$$

From this expression it is seen that when $\mu_i = 0$, the prior belief of z_i is unbiased and $p(z_i = 1) = 0.5$, but when $\mu_i < 0$ the variable z_i is biased toward zero and vice versa. If a subset of features $\{x_j|j \in \mathcal{J} \subset [D]\}$ is a priori more likely to explain the observed data $\boldsymbol{y}$, then this information can be encoded in the prior distribution by assigning the prior mean of $\boldsymbol{\gamma}$ such that $\mu_j > \mu_i$ for all $j \in \mathcal{J}$ and for all $i \notin \mathcal{J}$. However, in the remainder of this paper we will assume that the prior mean is constant, that is $\mu_i = \nu_0$ for some $\nu_0 \in \mathbb{R}$. For more details on the prior distribution, see Appendix D.

The prior probability of two variables, x_i and x_j, being jointly active is

$$p(z_i = 1, z_j = 1) = \int \phi(\gamma_i)\phi(\gamma_j)\,\mathcal{N}\left(\boldsymbol{\gamma}\middle|\boldsymbol{\mu}, \boldsymbol{\Sigma}_0\right)\mathrm{d}\boldsymbol{\gamma}. \tag{9}$$

If $\boldsymbol{\Sigma}_0$ is a diagonal matrix, γ_i and γ_j become independent and we recover the conventional spike-and-slab prior. On the other hand, if we choose $\boldsymbol{\Sigma}_0$ to be a covariance matrix of the form $\Sigma_{0,ij} = g(\|\boldsymbol{d}_i - \boldsymbol{d}_j\|)$, we see that the joint activation probabilities indeed depend on the spatial distance as desired. Finally, we emphasize that this parametrization it not limited to nearest neighbors-type structures. In fact, this parametrization supports general structures that can be modeled using generic covariance functions.

3. The Spatio-temporal Spike-and-Slab Prior

In the following we will extend the structured spike-and-slab prior distribution to model temporal smoothness of the sparsity pattern as well. Let $t \in [T]$ be the time index, then $\boldsymbol{x}_t$, $\boldsymbol{z}_t$ and $\boldsymbol{\gamma}_t$ are the signal coefficients, the sparsity patterns and the latent structure variable at time t. Furthermore, we define the corresponding matrix quantities $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_T\end{bmatrix}$, $\boldsymbol{Z} = \begin{bmatrix}\boldsymbol{z}_1 & \boldsymbol{z}_2 & \boldsymbol{z}_T\end{bmatrix}$ and $\boldsymbol{\Gamma} = \begin{bmatrix}\boldsymbol{\gamma}_1 & \boldsymbol{\gamma}_2 & \dots & \boldsymbol{\gamma}_T\end{bmatrix}$.

There are several natural temporal extensions of the model. The simplest extension is to assume that $\{\boldsymbol{\gamma}_t\}_{t=1}^T$ is independent in time, so that $p(\boldsymbol{Z}, \boldsymbol{\Gamma}) = \prod_{t=1}^T p(\boldsymbol{z}_t|\boldsymbol{\gamma}_t) \prod_{t=1}^T p(\boldsymbol{\gamma}_t)$, which is equivalent to solving each of the T regressions problems in eq. (1) independently. Another simple extension is to use the so-called *joint sparsity* assumption (Cotter et al., 2005; Zhang and Rao, 2011; Ziniel and Schniter, 2013b) and assume that the sparsity pattern is static across time, and thus all $\{\boldsymbol{x}_t\}_{t=1}^T$ vectors share a common binary support vector $\boldsymbol{z}$, and $p(\boldsymbol{X}|\boldsymbol{z}) = \prod_{t=1}^T \prod_{i=1}^D \left[(1-z_i)\,\delta\left(x_{i,t}\right) + z_i \mathcal{N}\left(x_{i,t}|\rho_0, \tau_0\right)\right]$. A more interesting and flexible model is to assume that the support is slowly changing in time, by modelling the temporal evolution of $\boldsymbol{\gamma}_t$ using a first order Gauss-Markov process of the form $p\left(\boldsymbol{\gamma}_t|\boldsymbol{\gamma}_{t-1}\right) = \mathcal{N}\left(\boldsymbol{\gamma}_t|\,(1-\alpha)\,\boldsymbol{\mu}_0 + \alpha\boldsymbol{\gamma}_{t-1}, \beta\boldsymbol{\Sigma}_0\right)$, where the hyperparameters $\alpha \in [0,1]$ and $\beta > 0$ control the temporal correlation and the "innovation" of the process, respectively.

The first order model has the advantage that it factorizes across time, which makes the resulting inference problem much easier. On the other hand, first order Markovian dynamics is often not sufficient for capturing long range correlations. Imposing a Gaussian process distribution on $\boldsymbol{\Gamma}$ with arbitrary covariance structure would facilitate modeling of long range correlations in both time and space. Therefore, the hierarchical prior distribution for $\boldsymbol{X}$ becomes

$$p(\boldsymbol{X}|\boldsymbol{Z}) = \prod_{t=1}^T \prod_{i=1}^D \left[(1-z_{i,t})\,\delta\left(x_{i,t}\right) + z_{i,t}\mathcal{N}\left(x_{i,t}|\rho_0, \tau_0\right)\right], \tag{10}$$

$$p(\boldsymbol{Z}|\boldsymbol{\Gamma}) = \prod_{t=1}^T \mathrm{Ber}\left(\boldsymbol{z}_t|\phi\left(\boldsymbol{\gamma}_t\right)\right), \tag{11}$$

$$p(\boldsymbol{\Gamma}) = \mathcal{N}\left(\boldsymbol{\Gamma}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right), \tag{12}$$

where the mean $\boldsymbol{\mu}_0 \in \mathbb{R}^{TD\times 1}$ and covariance matrix $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{TD\times TD}$ are now defined for the full $\boldsymbol{\Gamma}$-space. This model is more expressive, but the resulting inference problem becomes infeasible for even moderate sizes of D and T. But if we assume that the underlying spatio-temporal grid can be written in Cartesian product form, then covariance matrix simplifies to a Kronecker product

$$p(\boldsymbol{\Gamma}) = \mathcal{N}\left(\boldsymbol{\Gamma}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_{\text{temporal}} \otimes \boldsymbol{\Sigma}_{\text{spatial}}\right), \tag{13}$$

where $\boldsymbol{\Sigma}_{\text{temporal}} \in \mathbb{R}^{T\times T}$ and $\boldsymbol{\Sigma}_{\text{spatial}} \in \mathbb{R}^{D\times D}$. This decomposition leads to more efficient inference schemes as we will discuss in Section 5. In the remainder of the paper, we will focus on the model with Kronecker structure, but we refer to (Andersen et al., 2015) for more details on the first order model and joint sparsity model.

The coefficients $\{x_{i,t}\}$ are conditionally independent given the support $\{z_{i,t}\}$. For some applications it could be desirable to impose either spatial smoothness, temporal smoothness or both on the non-zero coefficients themselves (Wu et al., 2014a; Ziniel and Schniter, 2013a), but in this work we only assume a priori knowledge of the structure of the support. Although temporal smoothness of $x_{i,t}$ could easily be incorporated into the models described above.

4. Inference Using Spatio-temporal Priors

In the previous sections we have described the structured spike-and-slab prior and how to extend it to model temporal smoothness as well. We now turn our attention on how to perform inference using these models. We focus our discussion on the most general formulation using as given in eq. (10)-(12). Let $\boldsymbol{Y} = \begin{bmatrix}\boldsymbol{y}_1 & \boldsymbol{y}_2 & \dots & \boldsymbol{y}_T\end{bmatrix}$ be an observation matrix, where $\boldsymbol{y}_t \in \mathbb{R}^N$ is an observation vector for time t. We assume that the distribution on $\boldsymbol{Y}$ factors over time and is given by

$$p(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{t=1}^T p(\boldsymbol{y}_t|\boldsymbol{x}_t). \tag{14}$$

We consider two different noise models: an isotropic Gaussian noise model and a probit noise model. The Gaussian noise model $p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{y}_t|\boldsymbol{A}\boldsymbol{x}_t, \sigma_2\boldsymbol{I}\right)$ is suitable for linear inverse problems with forward model $\boldsymbol{A} \in \mathbb{R}^{N\times D}$ or equivalently sparse linear regression problems with design matrix $\boldsymbol{A} \in \mathbb{R}^{N\times D}$. On the other hand, the probit model is suitable for modeling binary observations, with $y_{t,n} \in \{-1, 1\}$, and is given by $p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \prod_{n=1}^N \phi\left(y_{t,n}\boldsymbol{a}_{n,\cdot}\boldsymbol{x}_t\right)$, where $\boldsymbol{a}_{n,\cdot}$ is the n'th row of $\boldsymbol{A}$. For both models we further assume that the matrix $\boldsymbol{A}$ is constant across time. However, this assumption can be easily relaxed to have $\boldsymbol{A}$ depend on t.

For both noise models the resulting joint distribution becomes

$$\begin{aligned} p\left(\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}\right) &= p(\boldsymbol{Y}|\boldsymbol{X})p(\boldsymbol{X}|\boldsymbol{Z})p(\boldsymbol{Z}|\boldsymbol{\Gamma})p(\boldsymbol{\Gamma}) &(15)\\ &= \prod_{t=1}^T p(\boldsymbol{y}_t|\boldsymbol{x}_t) \prod_{t=1}^T \left[(1-\boldsymbol{z}_t)\circ\delta\left(\boldsymbol{x}_t\right) + \boldsymbol{z}_t \circ \mathcal{N}\left(\boldsymbol{x}_t|0, \tau\boldsymbol{I}\right)\right] \\ &\quad \prod_{t=1}^T \mathrm{Ber}\left(\boldsymbol{z}_t|\phi\left(\boldsymbol{\gamma}_t\right)\right)\mathcal{N}\left(\boldsymbol{\Gamma}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right). &(16)\end{aligned}$$

We seek the posterior distribution of the parameters $\boldsymbol{X}, \boldsymbol{Z}$ and $\boldsymbol{\Gamma}$ conditioned on the observations $\boldsymbol{Y}$, which is obtained by applying Bayes's Theorem to the joint distribution in eq. (15)

$$\begin{aligned} p\left(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}|\boldsymbol{Y}\right) &= \frac{1}{Z}\prod_{t=1}^T p(\boldsymbol{y}_t|\boldsymbol{x}_t) \prod_{t=1}^T \left[(1-\boldsymbol{z}_t)\circ\delta\left(\boldsymbol{x}_t\right) + \boldsymbol{z}_t \circ \mathcal{N}\left(\boldsymbol{x}_t|0, \tau\boldsymbol{I}\right)\right] \\ &\quad \prod_{t=1}^T \mathrm{Ber}\left(\boldsymbol{z}_t|\phi\left(\boldsymbol{\gamma}_t\right)\right)\mathcal{N}\left(\boldsymbol{\Gamma}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right), &(17)\end{aligned}$$

where $Z = p(\boldsymbol{Y})$ is the marginal likelihood of $\boldsymbol{Y}$. Due to the product of mixtures in the distribution $p(\boldsymbol{X}|\boldsymbol{Z})$, the expression for the marginal likelihood Z involves a sum over 2^{DT} terms. This renders the computation of the normalization constant Z intractable for even small D and T. Hence, the desired posterior distribution is also intractable and we have to resort to approximate inference.

In the literature researchers have applied a whole spectrum of approximate inference methods for spike-and-slab priors, for example, Monte Carlo-methods (Mitchell and Beauchamp, 1988), mean-field variational inference (Titsias and Lazaro-Gredilla, 2011), approximate message passing (Vila and Schniter, 2013) and expectation propagation (Hernández-Lobato et al., 2013; Andersen et al., 2014). We use the latter since expectation propagation has been shown to have good performance for linear models with spike-and-slab priors (Hernández-Lobato et al., 2015) and it has been shown to provide a much better approximation of the first and second moment posterior moment for spike-and-slab models (Peltola et al., 2014).

4.1 The Expectation Propagation Framework

In this section, we briefly review expectation propagation for completeness. Expectation propagation (EP) (Minka, 2001; Opper and Winther, 2000) is a deterministic framework for approximating probability distributions. Consider a probability distribution over the variable $\boldsymbol{x} \in \mathbb{R}^D$ that factorizes into N components

$$f(\boldsymbol{x}) = \prod_{i=1}^{N} f_i(\boldsymbol{x}_i), \tag{18}$$

where $\boldsymbol{x}_i$ is taken to be a subvector of $\boldsymbol{x}$. EP takes advantage of this factorization and approximates f with a distribution Q that shares the same factorization

$$Q(\boldsymbol{x}) = \prod_{i=1}^{N} \tilde{f}_i(\boldsymbol{x}_i). \tag{19}$$

EP approximates each *site term* f_i with a (scaled) distribution $\tilde{f}_i$ from the exponential family. Since the exponential family is closed under products, the *global approximation* Q will also be in the exponential family. Consider the product of all $\tilde{f}_i$ terms except the j'th term

$$Q^{\backslash j}(\boldsymbol{x}) = \prod_{i \neq j} \tilde{f}_i(\boldsymbol{x}_i) = \frac{Q(\boldsymbol{x})}{\tilde{f}_j(\boldsymbol{x}_j)}. \tag{20}$$

The core of the EP framework is to choose $\tilde{f}_j$ such that $\tilde{f}_j(\boldsymbol{x}_j)Q^{\backslash j}(\boldsymbol{x}_j) \approx f_j(\boldsymbol{x}_j)\, Q^{\backslash j}(\boldsymbol{x}_j)$. By approximating f_j with $\tilde{f}_j$ in the context of $Q^{\backslash j}$, we ensure that the approximation is most accurate in the region of high density according to the *cavity distribution* $Q^{\backslash j}$. This scheme is implemented by iteratively minimizing the Kullbach-Leibler divergence $\mathrm{KL}\left(f_j(\boldsymbol{x}_j)\, Q^{\backslash j}(\boldsymbol{x}) \middle|\middle| \tilde{f}_j(\boldsymbol{x}_j)\, Q^{\backslash j}(\boldsymbol{x})\right)$. Since $\tilde{f}_j(\boldsymbol{x}_j)\, Q^{\backslash j}(\boldsymbol{x})$ belongs to the exponential family, the unique solution is obtained by matching the expected sufficient statistics (Bishop, 2006).

Once the solution,

$$Q^* = \underset{q}{\operatorname{argmin}}\ \mathrm{KL}\Big(f_j(\boldsymbol{x}_j)\, Q^{\backslash j}(\boldsymbol{x}) \Big|\Big| q\Big), \tag{21}$$

is obtained, the j'th site approximation is updated as

$$\tilde{f}_j^*(\boldsymbol{x}_j) \propto \frac{Q^*(\boldsymbol{x})}{Q^{\backslash j}(\boldsymbol{x})}. \tag{22}$$

The steps in eq. (20), (21) and (22) are repeated sequentially for all $j \in [D]$ until convergence is achieved.

4.2 The Expectation Propagation Approximation

The EP framework provides flexibility in the choice of the approximating factors. This choice is a trade-off between analytical tractability and sufficient flexibility for capturing the important characteristics of the true density. Consider the desired posterior density of interest

$$\begin{aligned} p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}|\boldsymbol{Y}) \propto & \underbrace{\prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{x}_t)}_{f_1(\boldsymbol{X})} \underbrace{\prod_{t=1}^{T} \left[(1 - \boldsymbol{z}_t) \circ \delta(\boldsymbol{x}_t) + \boldsymbol{z}_t \circ \mathcal{N}\left(\boldsymbol{x}_t|0, \tau \boldsymbol{I}\right)\right]}_{f_2(\boldsymbol{X}, \boldsymbol{Z})} \\ & \underbrace{\prod_{t=1}^{T} \mathrm{Ber}\left(\boldsymbol{z}_t|\phi(\boldsymbol{\gamma}_t)\right)}_{f_3(\boldsymbol{Z}, \boldsymbol{\Gamma})} \underbrace{\mathcal{N}\left(\boldsymbol{\Gamma}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right)}_{f_4(\boldsymbol{\Gamma})}. \end{aligned} \tag{23}$$

This posterior density is decomposed into four terms f_i for $i = 1, .., 4$, where the first three terms can be further decomposed. The term $f_1(\boldsymbol{X})$ is decomposed into T terms of the form $f_{1,t}(\boldsymbol{x}_t) = p(\boldsymbol{y}_t|\boldsymbol{x}_t)$, whereas the terms f_2 and f_3 are further decomposed as follows

$$f_1(\boldsymbol{X}) = \prod_{t=1}^{T} \tilde{f}_{1,}(\boldsymbol{x}_t) = \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{x}_t), \tag{24}$$

$$f_2(\boldsymbol{X}, \boldsymbol{Z}) = \prod_{t=1}^{T} \prod_{i=1}^{D} f_{2,i,t}(x_{i,t}, z_{i,t}) = \prod_{t=1}^{T} \prod_{i=1}^{D} \left[(1 - z_{i,t}) \circ \delta(x_{i,t}) + z_{i,t} \circ \mathcal{N}\left(x_{i,t}|\rho, \tau\right)\right], \tag{25}$$

$$f_3(\boldsymbol{Z}, \boldsymbol{\Gamma}) = \prod_{t=1}^{T} \prod_{i=1}^{D} f_{3,i,t}(z_{i,t}, \gamma_{i,t}) = \prod_{t=1}^{T} \prod_{i=1}^{D} \mathrm{Ber}\left(z_{i,t}|\phi(\gamma_{i,t})\right). \tag{26}$$

Each $f_{1,t}$ term only depends on $\boldsymbol{x}_t$, $f_{2,i,t}$ only depends on $x_{i,t}$ and $z_{i,t}$ and $f_{3,j,t}$ only depends on $z_{i,t}$ and $\gamma_{i,t}$. Furthermore, the terms $f_{2,i,t}$ couple the variables $x_{i,t}$ and $z_{i,t}$, while $f_{3,i,t}$ couple the variables $z_{i,t}$ and $\gamma_{i,t}$. Based on these observations, we choose $\tilde{f_{1,t}}$, $\tilde{f}_{2,i,t}$ and $\tilde{f}_{3,j,t}$ to have the following forms

$$\tilde{f}_{1,t}(\boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{x}_t|\hat{\boldsymbol{m}}_{1,t}, \hat{\boldsymbol{V}}_{1,t}\right), \tag{27}$$

$$\tilde{f}_{2,i,t}(x_{i,t}, z_{i,t}) = \mathcal{N}\left(x_{i,t}|\hat{m}_{2,i,t}, \hat{v}_{2,i,t}\right) \mathrm{Ber}\left(z_{i,t}|\phi(\hat{\gamma}_{2,i,t})\right), \tag{28}$$

$$\tilde{f}_{3,i,t}(z_{i,t}, \gamma_{i,t}) = \mathcal{N}\left(\gamma_{i,t}|\hat{\mu}_{3,j,t}, \hat{\sigma}_{3,i,t}\right) \mathrm{Ber}\left(z_{i,t}|\phi(\hat{\gamma}_{3,j,t})\right). \tag{29}$$

The exact term f_1 is a distribution wrt. $\boldsymbol{y}$ conditioned on $\boldsymbol{x}$, whereas the approximate term $\tilde{f}_1$ is a function of $\boldsymbol{x}$ that depends on the data $\boldsymbol{y}$ through $\hat{\boldsymbol{m}}_1$ and $\hat{\boldsymbol{V}}_1$ etc. Finally, f_4 already belongs to the exponential family and does therefore not have to be approximated by EP. That is, $\tilde{f}_4(\boldsymbol{\Gamma}) = f_4(\boldsymbol{\Gamma}) = \mathcal{N}(\boldsymbol{\Gamma}|\mu_0, \boldsymbol{\Sigma}_0)$.

Define $\hat{\boldsymbol{m}}_{2,t} = \begin{bmatrix}\hat{m}_{2,t,1} & \hat{m}_{2,t,2} & \dots & \hat{m}_{2,t,D}\end{bmatrix}^T$, $\hat{\boldsymbol{V}}_{2,t} = \operatorname{diag}\begin{pmatrix}\hat{v}_{2,t,1} & \hat{v}_{2,t,2} & \dots & \hat{v}_{2,t,D}\end{pmatrix}^T$ and $\hat{\boldsymbol{\gamma}}_{2,t} = \begin{bmatrix}\hat{\gamma}_{2,t,1} & \hat{\gamma}_{2,t,2} & \dots & \hat{\gamma}_{2,t,D}\end{bmatrix}$ and similarly for $\hat{\boldsymbol{\mu}}_{3,t}$, $\hat{\boldsymbol{\Sigma}}_{3,t}$ and $\hat{\gamma}_{3,t}$, then the resulting global approximation becomes

$$
\begin{aligned}
Q(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}) &\propto \prod_{t=1}^{T} \underbrace{\mathcal{N}\left(\boldsymbol{x}_t|\hat{\boldsymbol{m}}_{1,t}, \hat{\boldsymbol{V}}_{1,t}\right)}_{\tilde{f}_{1,t}} \prod_{t=1}^{T} \underbrace{\mathcal{N}\left(\boldsymbol{x}_t|\hat{\boldsymbol{m}}_{2,t}, \hat{\boldsymbol{V}}_{2,t}\right) \operatorname{Ber}\left(\boldsymbol{z}_t|\phi\left(\hat{\boldsymbol{\gamma}}_{2,t}\right)\right)}_{\tilde{f}_{2,t}} \\
&\quad \prod_{t=1}^{T} \underbrace{\mathcal{N}\left(\boldsymbol{\gamma}_t|\hat{\boldsymbol{\mu}}_{3,t}, \hat{\boldsymbol{\Sigma}}_{3,t}\right) \operatorname{Ber}\left(\boldsymbol{z}_t|\phi\left(\hat{\boldsymbol{\gamma}}_{3,t}\right)\right)}_{\tilde{f}_{3,t}} \underbrace{\mathcal{N}\left(\boldsymbol{\Gamma}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right)}_{\tilde{f}_4} \\
&\propto \prod_{t=1}^{T} \mathcal{N}\left(\boldsymbol{x}_t|\hat{\boldsymbol{m}}_t, \hat{\boldsymbol{V}}_t\right) \prod_{t=1}^{T} \operatorname{Ber}\left(\boldsymbol{z}_t|\phi\left(\hat{\boldsymbol{\gamma}}_t\right)\right) \mathcal{N}\left(\boldsymbol{\Gamma}|\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}\right), \qquad (30)
\end{aligned}
$$

where the parameters of the global approximation are obtained by summing the natural parameters. In terms of mean and variance, we get

$$
\hat{\boldsymbol{V}}_t = \left[\hat{\boldsymbol{V}}_{1,t}^{-1} + \hat{\boldsymbol{V}}_{2,t}^{-1}\right]^{-1}, \qquad (31)
$$

$$
\hat{\boldsymbol{m}}_t = \hat{\boldsymbol{V}}_t\left[\hat{\boldsymbol{V}}_{1,t}^{-1}\hat{\boldsymbol{m}}_{1,t} + \hat{\boldsymbol{V}}_{2,t}^{-1}\hat{\boldsymbol{m}}_{2,t}\right], \qquad (32)
$$

$$
\hat{\boldsymbol{\Sigma}} = \left[\boldsymbol{\Sigma}_0^{-1} + \hat{\boldsymbol{\Sigma}}_3^{-1}\right]^{-1}, \qquad (33)
$$

$$
\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\Sigma}}\left[\boldsymbol{\Sigma}_0^{-1}\boldsymbol{\mu}_0 + \hat{\boldsymbol{\Sigma}}_3^{-1}\hat{\boldsymbol{\mu}}_3\right], \qquad (34)
$$

$$
\phi\left(\hat{\gamma}_{i,t}\right) = \frac{\phi\left(\hat{\gamma}_{2,i,t}\right)\phi\left(\hat{\gamma}_{3,i,t}\right)}{\left(1-\phi\left(\hat{\gamma}_{2,i,t}\right)\right)\left(1-\phi\left(\hat{\gamma}_{3,j,t}\right)\right) + \phi\left(\hat{\gamma}_{2,i,t}\right)\phi\left(\hat{\gamma}_{3,i,t}\right)}, \qquad (35)
$$

where $\hat{\boldsymbol{\Sigma}}_3 \in \mathbb{R}^{TD\times TD}$ is a diagonal matrix, whose the diagonal is obtained by stacking the site variances $\hat{\boldsymbol{\Sigma}}_{3,t}$ for each time point and $\hat{\boldsymbol{\mu}}_3 \in \mathbb{R}^{TD}$ is a vector obtained by stacking the site means $\hat{\boldsymbol{\mu}}_{3,t}$ for each time point. To compute the global approximation, we need to estimate the parameters $\hat{\boldsymbol{m}}_{1,t}$, $\hat{\boldsymbol{V}}_{1,t}$, $\hat{\boldsymbol{m}}_{2,t}$, $\hat{\boldsymbol{V}}_{2,t}$, $\hat{\boldsymbol{\mu}}_{3,t}$, $\hat{\boldsymbol{\Sigma}}_{3,t}$, $\hat{\boldsymbol{\gamma}}_{2,t}$ and $\hat{\boldsymbol{\gamma}}_{3,t}$ for all $t \in [T]$ using EP. The estimation procedure of $\hat{\boldsymbol{m}}_{1,t}$ and $\hat{\boldsymbol{V}}_{1,t}$ depends on the observation model being used, whereas the estimation procedure of the remaining parameters are independent on the choice of observation model.

In principle, we could choose the approximate posterior distribution of $\boldsymbol{\Gamma}$ in eq. (30) from a family of distributions that factorizes across space, time or both to reduce the computational complexity. This choice would indeed reduce the computational burden, but in contrast to classical variational inference schemes, the correlation structure of the prior would be ignored in the EP scheme and thus, the resulting posterior approximation would

be meaningless for this specific model.

In the conventional EP algorithm, the site approximations are updated in a sequential manner meaning that the global approximation is updated every time a single site approximation (Minka, 2001) is refined. In this work, we use the parallel update scheme to reduce the computational complexity of the algorithm. That is, we first update all the site approximations of the form $\tilde{f}_{2,i,t}$ for $i \in [D]$, $t \in [T]$, and then we update the global approximation w.r.t. $\boldsymbol{x}_t$ and similarly for the $\tilde{f}_{3,i,t}$ and the global approximation w.r.t. $\boldsymbol{\gamma}_t$. From a message passing perspective this can be interpreted as a particular scheduling of messages (Minka, 2005). The proposed algorithm is summarized in Algorithm 1.

- Initialize approximation terms $\tilde{f}_a$ for $a = 1, 2, 3, 4$ and Q
- Repeat until stopping criteria
  - For each $\tilde{f}_{1,n,t}$ (*For non-Gaussian likelihoods only*):
    - Compute cavity distribution: $Q^{\setminus 1,n,t} \propto \frac{Q}{\tilde{f}_{1,n,t}}$
    - Minimize: $\mathrm{KL}(f_{1,n,t}Q^{\setminus 1,n,t}||Q^{1,t,\mathrm{new}})$ w.r.t. Q^{new}
    - Compute: $\tilde{f}_{1,n,t} \propto \frac{Q^{1,t,\mathrm{new}}}{Q^{\setminus 1,n,t}}$ to update parameters $\hat{m}_{1,n,t}, \hat{v}_{1,n,t}$ and $\hat{\gamma}_{1,n,t}$.
  - For each $\tilde{f}_{2,i,t}$:
    - Compute cavity distribution: $Q^{\setminus 2,i,t} \propto \frac{Q}{\tilde{f}_{2,i,t}}$
    - Minimize: $\mathrm{KL}(f_{2,i,t}Q^{\setminus 2,i,t}||Q^{2,t,\mathrm{new}})$ w.r.t. Q^{new}
    - Compute: $\tilde{f}_{2,i,t} \propto \frac{Q^{2,t,\mathrm{new}}}{Q^{\setminus 2,i,t}}$ to update parameters $\hat{m}_{2,i,t}, \hat{v}_{2,i,t}$ and $\hat{\gamma}_{2,i,t}$.
  - Update joint approximation parameters: $\hat{\boldsymbol{m}}, \hat{\boldsymbol{V}}$ and $\hat{\boldsymbol{\gamma}}$
  - For each $\tilde{f}_{3,i,t}$:
    - Compute cavity distribution: $Q^{\setminus 3,i,t} \propto \frac{Q}{\tilde{f}_{3,i,t}}$
    - Minimize: $\mathrm{KL}(f_{3,i,t}Q^{\setminus 3,i,t}||Q^{3,t,\mathrm{new}})$ w.r.t. $Q^{3,t,\mathrm{new}}$
    - Compute: $\tilde{f}_{3,i,t} \propto \frac{Q^{3,t,\mathrm{new}}}{Q^{\setminus 3,i,t}}$ to update parameters $\hat{\mu}_{3,i,t}, \hat{\sigma}_{3,i,t}$ and $\hat{\gamma}_{3,i,t}$
  - Update joint approximation parameters: $\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\gamma}}$
- Compute marginal likelihood approximation

Algorithm 1: Proposed algorithm for approximating the joint posterior distribution over $\boldsymbol{X}$, $\boldsymbol{Z}$ and $\boldsymbol{\Gamma}$ conditioned on $\boldsymbol{Y}$ using parallel EP.

4.3 Estimating Parameters for $\tilde{f}_{1,t}$

The estimation procedure for $\tilde{f}_{1,t}$ depends on the choice of observation model. Here we consider two different observation models, namely the isotropic Gaussian and the probit models. Both of these models lead to closed form update rules, but this is not true for all choices of $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$. In general if $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ factorizes over n and each term only depends on $\boldsymbol{x}_t$ through $\boldsymbol{A}\boldsymbol{x}_t$, then the resulting moment integrals are 1-dimensional and can be solved relatively fast using numerical integration procedures (Jylänki et al., 2011) if no closed form

solution exists.

Under the Gaussian noise model, we have

$$f_{1,t}\left(\boldsymbol{x}_t\right) = p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{y}_t|\boldsymbol{A}\boldsymbol{x}_t, \sigma^2\boldsymbol{I}\right). \tag{36}$$

Thus, $f_{1,t}$ is already in the exponential family for all $t \in [T]$ and does therefore not have to be approximated using EP. In particular, the parameters for $\tilde{f}_{1,t}$ are determined by the relations $\hat{\boldsymbol{V}}_{1,t}^{-1} = \frac{1}{\sigma^2}\boldsymbol{A}^T\boldsymbol{A}$ and $\hat{\boldsymbol{V}}_{1,t}^{-1}\hat{\boldsymbol{m}}_{1,t} = \frac{1}{\sigma^2}\boldsymbol{A}^T\boldsymbol{y}_t$. For simplicity we also assume that the noise variance is constant for all t.

Under the probit likelihood the term $f_{1,t}$ decompose to $f_{1,t} = \prod_{n=1}^{N} f_{1,t,n}$. In this case, the update of each site approximation $\tilde{f}_{1,t,n}$ resembles the updates for Gaussian process classification using EP, see appendix C for details.

4.4 Estimating Parameters for $\tilde{f}_{2,t}$

The terms $\tilde{f}_{2,t} = \prod_{i=1}^{D} \tilde{f}_{2,i,t}\left(x_{i,t}, z_{i,t}\right)$ factor over i, which implies that we only need the marginal cavity distributions of each pair of $x_{i,t}$ and $z_{i,t}$. Consider the update of the j'th term at time t, that is $\tilde{f}_{2,j,t}\left(x_{j,t}, z_{j,t}\right)$. The first step is to compute the marginal cavity distributions by removing the contribution of $\tilde{f}_{2,j,t}\left(x_{j,t}, z_{j,t}\right)$ from the marginal of the global approximation Q using eq. (20)

$$Q^{\setminus 2,j,t}\left(x_{j,t}, z_{j,t}\right) = \frac{Q^{\setminus 2,j,t}\left(x_{j,t}, z_{j,t}\right)}{\tilde{f}_{2,j,t}\left(x_{j,t}, z_{j,t}\right)} \propto \mathcal{N}\left(x_{j,t}|\mu^{\setminus 2,j,t}, \Sigma^{\setminus 2,j,t}\right) \mathrm{Ber}\left(z_{j,t}|\phi\left(\gamma^{\setminus 2,j,t}\right)\right). \tag{37}$$

When the approximate distribution belongs to the exponential family, the cavity distribution is simply obtained by computing the differences in natural parameters. Expressed in terms of mean and variance, we get

$$\hat{v}^{\setminus 2,j,t} = \left[\hat{V}_{t,jj}^{-1} - \hat{v}_{2,j,t}^{-1}\right]^{-1}, \tag{38}$$

$$\hat{m}^{\setminus 2,j,t} = \hat{v}^{\setminus 2,j,t}\left[\hat{V}_{t,jj}^{-1}\hat{m}_{j,t} - \hat{v}_{2,j,t}^{-1}\hat{m}_{2,j,t}\right], \tag{39}$$

$$\hat{\gamma}^{\setminus 2,j,t} = \hat{\gamma}_{3,j,t}. \tag{40}$$

The cavity parameter for $\gamma_{j,t}$ in $f_{2,j,t}$ is simply equal to $\hat{\gamma}_{3,j,t}$ (and vice versa) since $\hat{\gamma}_{2,j,t}$ and $\hat{\gamma}_{3,j,t}$ are the only two terms contributing to the distribution over $\boldsymbol{z}_{j,t}$. Next, we form the *tilted* distribution $f_{2,j,t}Q^{\setminus 2,j,t}$ and compute the solution to the KL minimization problem in eq. (21) by matching the expected sufficient statistics. This amounts to computing the zeroth, first and second moments w.r.t. $x_{j,t}$

$$X_m = \sum_{z_{j,t}} \int x_{j,t}^m \cdot f_{2,j,t}\left(x_{j,t}, z_{j,t}\right) Q^{\setminus 2,j,t}\left(x_{j,t}, z_{j,t}\right) \mathrm{d}x_{j,t} \quad \text{for} \quad m = 0, 1, 2, \tag{41}$$

and the first moment of $z_{j,t}$

$$Z_1 = \sum_{z_{j,t}} \int z_{j,t} \cdot f_{2,j,t}\left(x_{j,t}, z_{j,t}\right) Q^{\setminus 2,j,t}\left(x_{j,t}, z_{j,t}\right) \mathrm{d}x_{j,t}. \tag{42}$$

For notational convenience we have dropped the dependencies of X_m and Z_1 on the indices t and j. Alternatively, the moments could be obtained by computing the partial derivatives of the log normalizer of the tilted distribution.

The central moments of Q^* in eq.(21) are given by

$$E\left[x_{j,t}\right] = \frac{X_1}{X_0}, \qquad V\left[x_{j,t}\right] = \frac{X_2}{X_0} - \frac{X_1^2}{X_0^2}, \qquad E\left[z_{j,t}\right] = \frac{Z_1}{X_0}. \tag{43}$$

Refer to Appendix A for analytical expressions for these moments. Once Q^* has been obtained, we can compute the new update site approximation for $\tilde{f}_{2,j,t}$ using eq. (22) as follows

$$\tilde{f}_{2,j,t}^*\left(x_{j,t}, z_{j,t}\right) = \frac{Q^*\left(x_{j,t}, z_{j,t}\right)}{Q^{\setminus 2,j,t}\left(x_{j,t}, z_{j,t}\right)} \propto \mathcal{N}\left(x_{j,t}|\hat{m}_{2,j,t}^*, \hat{v}_{2,j,t}^*\right) \mathrm{Ber}\left(z_{j,t}|\phi\left(\hat{\gamma}_{2,j,t}^*\right)\right), \tag{44}$$

where the new site parameters $\hat{m}_{2,j,t}^*$ and $\hat{v}_{2,j,t}^*$, are obtained by computing differences in natural parameters in the same manner as for the cavity parameters in eq. (38) - (40)

$$\hat{v}_{2,j,t}^* = \left[V\left[x_{j,t}\right]^{-1} - \left(\hat{v}^{\setminus 2,j,t}\right)^{-1}\right]^{-1}, \tag{45}$$

$$\hat{m}_{2,j,t}^* = \hat{v}_{2,j,t}^*\left[V\left[x_{j,t}\right]^{-1} E\left[x_{j,t}\right] - \left(\hat{v}^{\setminus 2,j,t}\right)^{-1}\hat{m}^{\setminus 2,j,t}\right]. \tag{46}$$

The new site parameters for $z_{j,t}$ are obtained as (see Appendix A for details)

$$\phi\left(\hat{\gamma}_{2,j,t}^*\right) \overset{(a)}{=} \frac{\frac{E[z_{j,t}]}{\phi\left(\hat{\gamma}^{\setminus 2,j,t}\right)}}{\frac{1-E[z_{j,t}]}{1-\phi\left(\hat{\gamma}^{\setminus 2,j,t}\right)} + \frac{E[z_{j,t}]}{\phi\left(\hat{\gamma}^{\setminus 2,j,t}\right)}} \overset{(b)}{=} \frac{\mathcal{N}\left(0|\hat{m}^{\setminus 2,i} - \rho_0, \hat{V}^{\setminus 2,j,t} + \tau_0\right)}{\mathcal{N}\left(0|\hat{m}^{\setminus 2,i}, \hat{V}^{\setminus 2,i}\right) + \mathcal{N}\left(0|\hat{m}^{\setminus 2,i} - \rho_0, \hat{V}^{\setminus 2,j,t} + \tau_0\right)}, \tag{47}$$

where (a) follows from forming the quotient of the two Bernoulli distributions and (b) follows from straightforward algebraic reduction after substituting in the expression for the expectation of $z_{j,t}$.

4.5 Estimating Parameters for $\tilde{f}_{3,t}$

The procedure for updating $\tilde{f}_{3,t} = \prod_{i=1}^{D} \tilde{f}_{3,j,t}$ is completely analogously to the procedure for $\tilde{f}_{2,t}$. Consider the update for the j'th term at time t, that is $\tilde{f}_{3,j,t}$. After computing the cavity distribution in the same manner as in eq. (38)-(40), we now compute the moments w.r.t. $\gamma_{j,t}$ and $z_{j,t}$ of the (unnormalized) tilted distribution

$$G_m = \sum_{z_{j,t}} \int \gamma_{j,t}^m \cdot f_{3,j,t}\left(z_{j,t}, \gamma_{j,t}\right) Q^{\setminus 3,j,t}\left(z_{j,t}, \gamma_{j,t}\right) \mathrm{d}\gamma_{j,t} \quad \text{for} \quad m = 0, 1, 2, \tag{48}$$

$$Z_1 = \sum_{z_{j,t}} \int z_{j,t} \cdot f_{3,j,t}\left(z_{j,t}, \gamma_{j,t}\right) Q^{\setminus 3,j,t}\left(z_{j,t}, \gamma_{j,t}\right) \mathrm{d}\gamma_{j,t}. \tag{49}$$

Given these moments, we can obtain the central moments for Q^* in eq. (21)

$$E\left[\gamma_{j,t}\right] = \frac{G_1}{G_0}, \qquad V\left[\gamma_{j,t}\right] = \frac{G_2}{G_0} - \frac{G_1^2}{G_0^2}, \qquad E\left[z_{j,t}\right] = \frac{Z_1}{G_0}. \tag{50}$$

Refer to Appendix B for analytical expression of the moments. These moments completely determine Q^* and the j'th site update at the t is computed analogous to $\tilde{f}_{2,j,t}$ in eq. (44) using eq. (45), (46) and (47).

4.6 The Computational Details

In the previous sections, we have described how to use EP for approximate inference for the proposed model, and in this section, we discuss some of the computational details of the resulting EP algorithm.

4.6.1 Updating the Global Covariance Matrices

Given a set of updated site approximations, $\tilde{f}_{2,t} = \prod_j \tilde{f}_{2,j,t}$, we can compute the parameters for the global approximate distribution of $\boldsymbol{x}_t$ using eq. (31) and (32). Direct evaluation of eq. (31) results in a computational complexity of $\mathcal{O}\left(D^3\right)$. Recall, that N is assumed to be smaller than D. This implies that $\hat{\boldsymbol{V}}_{1,t}^{-1} = \frac{1}{\sigma_0^2}\boldsymbol{A}^T\boldsymbol{A}$ has low rank. Furthermore, the matrix $\hat{\boldsymbol{V}_{2,t}}$ is diagonal, and therefore we can apply the matrix inversion lemma as follows

$$\hat{\boldsymbol{V}}_t = \hat{\boldsymbol{V}}_{2,t} - \hat{\boldsymbol{V}}_{2,t}\boldsymbol{A}^T\left(\sigma_0^2\boldsymbol{I} + \boldsymbol{A}\hat{\boldsymbol{V}}_{2,t}\boldsymbol{A}^T\right)^{-1}\boldsymbol{A}\hat{\boldsymbol{V}}_{2,t}. \tag{51}$$

The inverse of $\sigma_0^2\boldsymbol{I} + \boldsymbol{A}\hat{\boldsymbol{V}}_{2,t}\boldsymbol{A}^T = \boldsymbol{L}_t\boldsymbol{L}_t^T$ can be computed in $\mathcal{O}\left(N^3\right)$ using a Cholesky decomposition. Thus, for $N < D$ eq. (51) scales as $\mathcal{O}\left(ND^2\right)$. Moreover, eq. (38) shows that we only require the diagonal elements of $\tilde{\boldsymbol{V}}_t$ in order to update the site approximation parameters for $\tilde{f}_{2,t}$. Hence, we can further reduce the computational complexity by only computing the diagonal of $\hat{\boldsymbol{V}}_t$ as follows

$$\begin{aligned}\operatorname{diag}\left[\hat{\boldsymbol{V}}_t\right] &= \operatorname{diag}\left[\hat{\boldsymbol{V}}_{2,t}\right] - \operatorname{diag}\left[\hat{\boldsymbol{V}}_{2,t}\boldsymbol{A}^T\boldsymbol{L}_t^{-T}\boldsymbol{L}_t^{-1}\boldsymbol{A}\hat{\boldsymbol{V}}_{2,t}\right] \\ &= \operatorname{diag}\left[\hat{\boldsymbol{V}}_{2,t}\right] - \operatorname{diag}\left[\hat{\boldsymbol{V}}_{2,t}^2\right] \circ \left(\boldsymbol{1}^T\left(\boldsymbol{R}_t \circ \boldsymbol{R}_t\right)\right),\end{aligned} \tag{52}$$

where $\boldsymbol{R}_t \in \mathbb{R}^{N\times D}$ is defined as $\boldsymbol{R}_t = \boldsymbol{L}_t^{-1}\boldsymbol{A}$ and $\boldsymbol{1}$ is a column vector of ones. The resulting computational cost is $\mathcal{O}\left(N^2D\right)$. Similarly, the mean of the global approximate distribution of $\boldsymbol{x}_t$, can be efficiently evaluated as

$$\hat{\boldsymbol{m}}_t = \hat{\boldsymbol{V}}_{2,t}\boldsymbol{\eta}_t - \hat{\boldsymbol{V}}_{2,t}\boldsymbol{R}_t^T\boldsymbol{R}_t\hat{\boldsymbol{V}}_{2,t}\boldsymbol{\eta}_t, \tag{53}$$

where $\boldsymbol{\eta}_t = \hat{\boldsymbol{V}}_{1,t}^{-1}\hat{\boldsymbol{m}}_{1,t} + \hat{\boldsymbol{V}}_{2,t}^{-1}\hat{\boldsymbol{m}}_{2,t}$. The total cost of updating the posterior distribution for $\boldsymbol{x}_t$ for all $t \in [T]$ is therefore $\mathcal{O}\left(TN^2D\right)$.

Unfortunately, we cannot get the same speed up for the refinement of the global approximation of $\boldsymbol{\Gamma}$ since the prior covariance matrix $\boldsymbol{\Sigma}_0$ in general is full rank. However,

we still only require the diagonal elements of the approximate covariance matrix $\hat{\boldsymbol{\Sigma}}$. We implement the update as advocated by Rasmussen and Williams (2006), that is,

$$\begin{aligned}\hat{\boldsymbol{\Sigma}} &= \left[\boldsymbol{\Sigma}_0^{-1} + \hat{\boldsymbol{\Sigma}}_3^{-1}\right]^{-1} \\ &= \boldsymbol{\Sigma}_0 - \boldsymbol{\Sigma}_0\hat{\boldsymbol{\Sigma}}_3^{-\frac{1}{2}}\left(\hat{\boldsymbol{\Sigma}}_3^{-\frac{1}{2}}\boldsymbol{\Sigma}_0\hat{\boldsymbol{\Sigma}}_3^{-\frac{1}{2}} + \boldsymbol{I}\right)^{-1}\hat{\boldsymbol{\Sigma}}_3^{-\frac{1}{2}}\boldsymbol{\Sigma}_0,\end{aligned} \tag{54}$$

where the second equality follows from the matrix inverse lemma. Again, we compute the required inverse matrix using the Cholesky decomposition, so that the total cost is $\mathcal{O}\left(D^3T^3\right)$.

4.6.2 Initialization, Convergence and Negative Variances

We initialize all the site terms to be rather uninformative, that is $\hat{m}_{2,i,t} = 0$, $\hat{v}_{2,i,t} = 10^4$, $\hat{\gamma}_{2,i,t} = 0$, $\hat{\mu}_{3,i,t} = 0$, $\hat{\sigma}_{3,i,t} = 10^4$, $\hat{\gamma}_{3,i,t} = 0$ for all $i \in [D]$ and $t \in [T]$ assuming standard scaling of the forward model $\boldsymbol{A}$.

There are in general no convergence guarantees for EP and the parallel version in particular can suffer from convergence problems (Seeger, 2005). The standard procedure to overcome this problem is to use "damping" when updating the site parameters

$$\tilde{f}^* = \tilde{f}_{\text{old}}^{1-\alpha}\,\tilde{f}_{\text{new}}^{\alpha}, \tag{55}$$

where $\alpha \in [0, 1]$ is the damping parameter and $\tilde{f}_{\text{old}}$ is the site approximation at the previous iteration. Since both $\tilde{f}_{\text{old}}$ and $\tilde{f}_{\text{new}}$ belongs to the exponential family, the update in eq. (55) corresponds to taking a convex combination of the previous and the new natural parameters of the site approximation.

Negative variances occur "naturally" in EP (Bishop, 2006) when updating the site approximations. However, this can lead to instabilities of the algorithm, non-positive semi-definitiveness of the posterior covariance matrices and convergence problems. We therefore take measures to prevent negative site variances. One way to circumvent this is to change a negative variance to $+\infty$, which corresponds to minimizing the KL divergence in eq. (21) with the site variance constrained to be positive (Hernández-Lobato et al., 2013). In practice, when encountering a negative variance after updating a given site we use $v_\infty = 10^2$ and $\sigma_\infty = 10^6$ for $\tilde{f}_{2,i,t}$ and $\tilde{f}_{3,i,t}$, respectively.

5. Further Approximations

As mentioned earlier, the updates of the global parameters for $\boldsymbol{x}_t$ and $\boldsymbol{\Gamma}$ are the dominating operations scaling as $\mathcal{O}\left(TN^2D\right)$ and $\mathcal{O}\left(D^3T^3\right)$, respectively. The latter term becomes prohibitive for moderate sizes of D and T and calls for further approximations. In this section, we introduce three simple approximations to reduce the computational complexity of the refinement of the posterior distribution for $\boldsymbol{\Gamma}$. The approximations and their computational complexities are summarized in table 1.

Approximation		Complexity	Storage
Full EP	(EP)	$\mathcal{O}\left(T^3D^3\right)$	$\mathcal{O}\left(T^2D^2\right)$
Low rank	(LR)	$\mathcal{O}\left(K^2TD\right)$	$\mathcal{O}\left(KTD\right)$
Common precision	(CP)	$\mathcal{O}\left(TD^2 + DT^2\right)$	$\mathcal{O}\left(D^2 + T^2\right)$
Group	(G)	$\mathcal{O}\left(T_g^3 D_g^3\right)$	$\mathcal{O}\left(T_g^2 D_g^2\right)$

Table 1: Summary of approximation schemes for updating the global parameters for $\boldsymbol{\Gamma}$.

5.1 The Low Rank Approximation

The eigenvalue spectrum of many prior covariance structures of interest, for example simple neighborhoods, decay relatively fast. Therefore, we can approximate $\boldsymbol{\Sigma}_0$ with a low rank approximation plus a diagonal matrix $\boldsymbol{\Sigma}_0 \approx \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^T + \boldsymbol{\Lambda}$, where $\boldsymbol{S} \in \mathbb{R}^{K\times K}$ is a diagonal matrix containing K largest eigenvalues, and $\boldsymbol{U} \in \mathbb{R}^{DT\times K}$ is a matrix containing the corresponding eigenvectors (Riihimäki et al., 2014). The diagonal matrix $\boldsymbol{\Lambda}$ is chosen such that the diagonal in the exact prior covariance matrix $\boldsymbol{\Sigma}_0$ is preserved. This allows us to apply the matrix inversion lemma to compute the update of the posterior covariance matrix for $\boldsymbol{\Gamma}$ (see Section 4.6.1).

Computing the eigendecomposition of $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{DT\times DT}$ scales in general as $\mathcal{O}\left(D^3T^3\right)$. However, when the prior covariance has Kronecker structure, the eigendecompositions of $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_t \otimes \boldsymbol{\Sigma}_s$ can be efficiently obtained from the eigendecompositions of $\boldsymbol{\Sigma}_t \in \mathbb{R}^{T\times T}$ and $\boldsymbol{\Sigma}_s \in \mathbb{R}^{D\times D}$. In this case, the eigendecomposition of $\boldsymbol{\Sigma}_0$ can be obtained in $\mathcal{O}\left(D^3 + T^3\right)$.

Using a K-rank approximation, the computational cost of refining the covariance matrix for $\boldsymbol{\Gamma}$ becomes $\mathcal{O}\left(K^2DT\right)$ and the memory footprint is $\mathcal{O}\left(TDK\right)$. For a fixed value of K this scales linearly in both D and T. However, to maintain a sufficiently good approximation K can scale with both D and T.

5.2 The Common Precision Approximation

Rather than approximating the prior covariance matrix as done in the low rank approximation, we now approximate the EP approximation scheme itself. If the prior covariance matrix for $\boldsymbol{\Gamma}$ can be written in terms of Kronecker products, we can significantly speed up the computation of the posterior covariance matrix of $\boldsymbol{\Gamma}$ by approximating the site precisions with a single common parameter. Let $\tilde{\boldsymbol{\theta}}_3 \in \mathbb{R}^{DT\times 1}$ be a vector containing the site precisions (inverse variances) for the site approximations $\{f_{3,i,t}\}$ for all $i \in [D]$ and for all $t \in [T]$, then we make the following approximation

$$\tilde{\boldsymbol{\Sigma}}_3 \approx \bar{\theta}^{-1}\boldsymbol{I}, \tag{56}$$

where $\bar{\theta}$ is the mean of value of $\tilde{\boldsymbol{\theta}}_3$. Assume the prior covariance matrix for $\boldsymbol{\Gamma}$ can be decomposed into a temporal part and a spatial part as follows $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_t \otimes \boldsymbol{\Sigma}_s$. Let $\boldsymbol{U}_t$, $\boldsymbol{U}_s$ and $\boldsymbol{S}_t$, $\boldsymbol{S}_s$ be eigenvectors and eigenvalues for $\boldsymbol{\Sigma}_t \in \mathbb{R}^{T\times T}$ and $\boldsymbol{\Sigma}_s \in \mathbb{R}^{D\times D}$, respectively. The global covariance matrix is updated as $\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_0\left(\boldsymbol{\Sigma}_0 + \tilde{\boldsymbol{\Sigma}}_3\right)^{-1}\tilde{\boldsymbol{\Sigma}}_3$. We now use the

properties of eigendecompositions for Kronecker products to compute the inverse matrix

$$\begin{aligned}\left(\boldsymbol{\Sigma}_t \otimes \boldsymbol{\Sigma}_s + \tilde{\boldsymbol{\Sigma}}_3\right)^{-1} &\approx \left(\boldsymbol{\Sigma}_t \otimes \boldsymbol{\Sigma}_s + \bar{\Sigma}_3\boldsymbol{I}\right)^{-1} \\ &= \left[\left(\boldsymbol{U}_t \otimes \boldsymbol{U}_s\right)\left(\boldsymbol{S}_t \otimes \boldsymbol{S}_s\right)\left(\boldsymbol{U}_t^T \otimes \boldsymbol{U}_s^T\right) + \bar{\Sigma}_3\boldsymbol{I}\right]^{-1} \\ &= \left(\boldsymbol{U}_t \otimes \boldsymbol{U}_s\right)\left(\boldsymbol{S}_t \otimes \boldsymbol{S}_s + \bar{\Sigma}_3\boldsymbol{I}\right)^{-1}\left(\boldsymbol{U}_t^T \otimes \boldsymbol{U}_s^T\right), \end{aligned} \tag{57}$$

where $\left(\boldsymbol{S}_t \otimes \boldsymbol{S}_s + \bar{\Sigma}_3\boldsymbol{I}\right)$ is diagonal and therefore fast to invert. The *common precision* approximation $\hat{\boldsymbol{\Sigma}}_{CP}$ is then obtained as

$$\begin{aligned}\hat{\boldsymbol{\Sigma}}_{CP} &= \left(\boldsymbol{\Sigma}_t \otimes \boldsymbol{\Sigma}_s\right)\left(\boldsymbol{\Sigma}_t \otimes \boldsymbol{\Sigma}_s + \bar{\Sigma}_3\boldsymbol{I}\right)^{-1}\bar{\Sigma}_3\boldsymbol{I} \\ &= \left(\boldsymbol{U}_t \otimes \boldsymbol{U}_s\right)\left(\boldsymbol{S}_t \otimes \boldsymbol{S}_s\right)\left(\boldsymbol{S}_t \otimes \boldsymbol{S}_s + \bar{\Sigma}_3\boldsymbol{I}\right)^{-1}\left(\boldsymbol{U}_t^T \otimes \boldsymbol{U}_s^T\right)\bar{\Sigma}_3. \end{aligned} \tag{58}$$

Let $\boldsymbol{M} \in \mathbb{R}^{TD\times 1}$ denote the diagonal of $\left(\boldsymbol{S}_t \otimes \boldsymbol{S}_s\right)\left(\boldsymbol{S}_t \otimes \boldsymbol{S}_s + \bar{\Sigma}_3\boldsymbol{I}\right)^{-1}$, then we can compute the diagonal of $\hat{\boldsymbol{\Sigma}}_{CP}$ as follows

$$\begin{aligned}\operatorname{diag}\left[\hat{\boldsymbol{\Sigma}}_{CP}\right]_i &= \bar{\Sigma}_3\sum_k \left(\boldsymbol{U}_t \otimes \boldsymbol{U}_s\right)_{ik} M_k \left(\boldsymbol{U}_t^T \otimes \boldsymbol{U}_s^T\right)_{ki} \\ &= \bar{\Sigma}_3\sum_k \left(\boldsymbol{U}_t \otimes \boldsymbol{U}_s\right)_{ik}^2 M_k \\ \Rightarrow \quad \operatorname{diag}\left[\hat{\boldsymbol{\Sigma}}_{CP}\right] &= \bar{\Sigma}_3\left(\boldsymbol{U}_t \circ \boldsymbol{U}_t \otimes \boldsymbol{U}_s \circ \boldsymbol{U}_s\right)\boldsymbol{M}, \end{aligned} \tag{59}$$

where $\circ$ is the Hadamard-product. We now see that the desired diagonal can be obtained by multiplying a Kronecker product with a vector and this can be computed efficiently using the identity

$$\operatorname{vec}\left[\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}\right] = \left(\boldsymbol{C}^T \otimes \boldsymbol{A}\right)\operatorname{vec}\left[\boldsymbol{B}\right]. \tag{60}$$

Therefore,

$$\operatorname{diag}\left[\hat{\Sigma}_{CP}\right] = \bar{\Sigma}_3 \cdot \operatorname{vec}\left[\left(\boldsymbol{U}_s \circ \boldsymbol{U}_s\right)\operatorname{vec}^{-1}\left[\boldsymbol{M}\right]\left(\boldsymbol{U}_t \circ \boldsymbol{U}_t\right)^T\right]. \tag{61}$$

Since the Hadamard products can be precomputed, this scales as $\mathcal{O}\left(D^2T + T^2D\right)$. During the EP iterations we only need to store $\boldsymbol{U}_s \in \mathbb{R}^{D\times D}$ and $\boldsymbol{U}_t \in \mathbb{R}^{T\times T}$, so the resulting memory footprint is $\mathcal{O}\left(D^2 + T^2\right)$. The posterior mean vector can also be computed efficiently by iteratively applying the result from eq. (60)

$$\hat{\boldsymbol{\Sigma}}_{CP}\boldsymbol{\eta} = \left(\boldsymbol{U}_t \otimes \boldsymbol{U}_s\right)\operatorname{diag}\left[M\right]\left(\boldsymbol{U}_t^T \otimes \boldsymbol{U}_s^T\right)\boldsymbol{\eta}, \tag{62}$$

where $\boldsymbol{\eta} = \hat{\boldsymbol{\Sigma}}_3^{-1}\hat{\boldsymbol{\mu}}_3 + \hat{\boldsymbol{\Sigma}}_0^{-1}\hat{\boldsymbol{\mu}}_0$.

The proposed approximation reduces the cost from $\mathcal{O}\left(D^3T^3\right)$ to $\mathcal{O}\left(D^2T + T^2D\right)$. If the spatial covariance matrix is a Kronecker product itself, for example, $\boldsymbol{\Sigma}_s = \boldsymbol{\Sigma}_x \otimes \boldsymbol{\Sigma}_{\boldsymbol{y}}$ or $\boldsymbol{\Sigma}_s = \boldsymbol{\Sigma}_x \otimes \boldsymbol{\Sigma}_y \otimes \boldsymbol{\Sigma}_z$, the computational complexity can be further reduced. Such covariance structures could occur in image application or in analysis of fMRI data.

This common precision approximation is closely related to the recently proposed *Stochastic Expectation Propagation* (SEP) (Li et al., 2015), where both the means and variances of the site approximation terms have been tied together. Tying both means and variances is reasonable when the site terms are approximating likelihood terms and $N \gg D$. In case of the present model, we expect positive values of $\Gamma_{i,t}$ for $z_{i,t} = 1$ and negative values of $\Gamma_{i,t}$ for $z_{i,t} = 0$, and thus enforcing a common mean for the site approximation terms $\tilde{f}_{3,i,t}$ would not make sense.

From experiments we have observed that this common precision approach significantly increases the number of iterations until convergence. However, this problem can be mitigated by repeating the updates for the site approximations $\tilde{f}_{3,i,t}$ and the global approximation for $\boldsymbol{\Gamma}$ a few times before moving on to update the site approximations for $\tilde{f}_{2,i,t}$. Specifically, within each EP iteration we repeat the updates for posterior distribution of $\boldsymbol{\Gamma}$ 5 times. The added computational workload is still negligible compared to full EP. Furthermore, for some problem instances CP-EP can oscillate. The oscillation can be alleviated heuristically by decreasing the damping parameter α by 10% if the approximate log likelihood decreases from one iteration to another after the first 100 iterations.

5.3 Grouping the Latent Structure Variables

Consider a problem, where the spatial coordinates $\boldsymbol{d}_i$ for each $x_{i,\cdot}$ form a uniformly spaced grid. Assume the characteristic length-scale of the sparsity pattern is large relative to the grid size, then support variables $\{z_i\}$ in a neighborhood could "share" the same γ-variable with a little loss of accuracy (Jacob et al., 2009a; Hernández-Lobato et al., 2013). This *grouping* of the latent variables could either be in the spatial, temporal or both dimensions. Let G be the number of groups and $g : [D] \times [T] \to [G]$ be a grouping function that maps from a spatial and temporal index to a group index, then the grouped version of the prior is given by

$$p(\boldsymbol{Z}|\boldsymbol{\gamma}) = \prod_{t=1}^{T}\prod_{i=1}^{D} \mathrm{Ber}\left(z_{t,i}|\phi\left(\gamma_{g(i,t)}\right)\right), \tag{63}$$

$$p(\boldsymbol{\gamma}) = \mathcal{N}\left(\boldsymbol{\gamma}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0\right), \tag{64}$$

where $\boldsymbol{\mu}_0 \in \mathbb{R}^G$ and $\boldsymbol{\Sigma}_0 \in \mathbb{R}^{G\times G}$ are the prior mean and covariance for the new grouped model. The resulting computational complexity is indeed determined by the size of the groups. For example, assume that the support variable for a given problem have been grouped in groups of 2 in both the spatial dimension and temporal dimension, then the total number of groups becomes $G = \frac{1}{2}D\frac{1}{2}T = \frac{1}{4}DT$ and the resulting computational cost is reduced to a fraction of $\left(\frac{1}{4}\right)^3$ of the cost of the full EP scheme. Furthermore, if necessary both the low rank and the common precision approximation can be applied on top of this approximation.

6. The Marginal Likelihood Approximation and Model Selection

The model contains several hyperparameters $\boldsymbol{\Omega} \in \mathbb{R}^L$, which include, for example, the hyperparameters of the kernel for $\boldsymbol{\Gamma}$. In a fully Bayesian setting, the natural approach to handle hyperparameters is to impose prior distributions and marginalize over the hyperparameters. The exact, but generally intractable marginalization integral is given by

$$p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}|\boldsymbol{Y}) = \int p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}|\boldsymbol{Y}, \boldsymbol{\Omega}) p\left(\boldsymbol{\Omega}\middle|\boldsymbol{Y}\right) \mathrm{d}\boldsymbol{\Omega}, \tag{65}$$

where $p\left(\boldsymbol{\Omega}\middle|\boldsymbol{Y}\right) \propto p(\boldsymbol{Y}|\boldsymbol{\Omega})p(\boldsymbol{\Omega})$ for some prior distribution $p(\boldsymbol{\Omega})$. The true marginal likelihood $p(\boldsymbol{Y}|\boldsymbol{\Omega})$ is given by the following marginalization

$$p(\boldsymbol{Y}|\boldsymbol{\Omega}) = \int f_1\left(\boldsymbol{X}|\boldsymbol{\Omega}\right) \sum_{\boldsymbol{Z}} f_2\left(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\Omega}\right) \mathrm{d}\boldsymbol{X} \int f_3\left(\boldsymbol{Z}, \boldsymbol{\Gamma}|\boldsymbol{\Omega}\right) f_4\left(\boldsymbol{\Gamma}|\boldsymbol{\Omega}\right) \mathrm{d}\boldsymbol{\Gamma}. \tag{66}$$

The exact quantity is intractable, but the EP framework provides an approximation to the marginal likelihood conditioned on the hyperparameters, $p(\boldsymbol{Y}|\boldsymbol{\Omega}) \approx Q(\boldsymbol{Y}|\boldsymbol{\Omega})$. The approximation $Q(\boldsymbol{Y}|\boldsymbol{\Omega})$ is obtained by substituting the exact site terms, for example, $f_{2,i,t}$, with a scaled version of the corresponding site approximation, for example, $s_{2,i,t}\tilde{f}_{2,i,t}$, and then carrying out the marginalization analytically. The scaling constants, for example, $s_{2,i,t}$, are chosen such that

$$\mathbb{E}_{Q^{\setminus 2,i,t}}\left[f_{2,i,t}\left(x_{i,t}, z_{i,t}\right)\right] = s_{2,i,t}\mathbb{E}_{Q^{\setminus 2,i,t}}\left[\tilde{f}_{2,i,t}\left(x_{i,t}, z_{i,t}\right)\right] \tag{67}$$

and similarly for all the site terms $f_{a,i,t}$ for $a \in [4]$, $i \in [D]$, $t \in [T]$. In the following, we will describe three different approximation strategies based on the marginal likelihood approximation.

6.1 Maximum Likelihood and MAP Estimation

This simplest and most crude approximation is to use a point estimate of $\boldsymbol{\Omega}$ instead of integrating over the uncertainty. Specifically, we aim to locate the maximum a posteriori (MAP) value by maximizing $\ln Q\left(\boldsymbol{\Omega}|\boldsymbol{Y}\right) = \ln Q\left(\boldsymbol{Y}\middle|\boldsymbol{\Omega}\right) + \ln p\left(\boldsymbol{\Omega}\right) + \text{constant}$ using gradient-based methods. A maximum likelihood type II estimate is obtained by choosing an (improper) flat prior $p\left(\boldsymbol{\Omega}\right) \propto 1$. For severely ill-posed problems, the marginal likelihood approximation can be completely non-informative with regard to one or more hyperparameters and thus, the maximum likelihood estimate can lead to suboptimal and unstable results for some problems. For some problem instances, it can also happen that the marginal likelihood solution with regard to the prior mean and variance of $\boldsymbol{\Gamma}$ is not in the interior of $\mathbb{R}^2$ and thus, gradient-based optimization with regard to these parameters will diverge. However, this problem is easily solved by imposing a weakly informative prior on the prior variance of $\boldsymbol{\Gamma}$ with little influence on the result (see Appendix D for more details).

The marginal likelihood approximation, $Q\left(\boldsymbol{Y}\middle|\boldsymbol{\Omega}\right)$, depends on the hyperparameters $\boldsymbol{\Omega}$ directly as well as through the site parameters, but the latter dependency can be ignored in gradients computations when the EP fixed point conditions hold (Seeger, 2005). The hyperparameter optimization procedure proceeds in an iterative two-stage fashion, where we first run EP until convergence and then we take a gradient step with regard to the hyperparameters and then repeat.

6.2 Approximate Marginalization using Numerical Integration

As a better approximation of eq. (65), we propose to approximate the marginalization integral using numerical integration with a finite sum using a central composite design (CCD) grid (Rue et al., 2009). This method has previously been successfully applied for marginalizing over hyperparameters in Gaussian process based models and the accuracy is reported to be between empirical Bayes and full marginalization using a dense grid (Vanhatalo et al., 2010). We approximate the marginal posterior distribution as follows

$$p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}|\boldsymbol{Y}) \approx \int Q(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}|\boldsymbol{Y}, \boldsymbol{\Omega}) Q\left(\boldsymbol{\Omega}\middle|\boldsymbol{Y}\right) \mathrm{d}\boldsymbol{\Omega} \tag{68}$$

$$\approx \sum_{m=1}^{M} Q(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}|\boldsymbol{Y}, \boldsymbol{\Omega}_m) Q\left(\boldsymbol{\Omega}_m\middle|\boldsymbol{Y}\right) w_m, \tag{69}$$

for a set of points $\{\boldsymbol{\Omega}_m\}_{m=1}^M$, a set of integration weights $\{w_m\}_{m=1}^M$. Thus, the resulting approximate marginal posterior distribution becomes a Gaussian mixture model with mixing weights $\pi_m = Q\left(\boldsymbol{\Omega}_m\middle|\boldsymbol{Y}\right) w_m$ and components $Q(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Gamma}|\boldsymbol{Y}, \boldsymbol{\Omega}_m)$.

To keep the computational burden to a minimum, we use a so-called Central Composite Design (CCD) to choose the points and weights. Most of the hyperparameters are variance or scale parameters and hence constrained to be positive. Therefore, we first transform these parameters into an unconstrained space using a log transformation, $\lambda_i = \ln \Omega_i$. Next, we locate the mode in the transformed parameter space, $\hat{\boldsymbol{\lambda}}_{\mathrm{MAP}}$, by optimizing $Q\left(\boldsymbol{Y}\middle|\boldsymbol{\lambda}\right)$ with regard to $\boldsymbol{\lambda}$ using gradient-based optimization methods and numerically estimate the inverse Hessian, $\hat{\boldsymbol{S}} = \boldsymbol{H}^{-1}$, at the mode $\hat{\boldsymbol{\lambda}}_{\mathrm{MAP}}$.

The CCD integration points are then obtained as $\boldsymbol{\lambda}_m = \hat{\boldsymbol{S}}^{\frac{1}{2}} \boldsymbol{p}_m + \hat{\boldsymbol{\lambda}_{\mathrm{MAP}}}$, where $\{\boldsymbol{p}_m\}_{m=1}^M$ is a CCD design grid (Rue et al., 2009) in L-dimensions. The points on the CCD grid consist of a fractional factorial design as well as $2K$ star points and a center point. All points, except for the center point, lives on the surface of a L-dimensional ball with radius $\sqrt{L}$. This specific design choice requires a much smaller number of points compared to a dense grid. For example, for $L = 2, 3, 4, 5$ parameters, the number of CCD points are $M = 9, 15, 25, 43$, respectively. The integration weights $\{w_m\}_{m=1}^M$ are chosen such that the integral match for the first three moments of a L-dimensional standardized Gaussian random variable, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_L)$, and $\mathbb{E}\left[1\right] = 1$, $\mathbb{E}\left[\boldsymbol{z}\right] = 0$, $\mathbb{E}\left[\boldsymbol{z}^T\boldsymbol{z}\right] = L$.

6.3 Bayesian Optimization

There can be some challenges with gradient-based optimization of the marginal likelihood approximation. Firstly, the optimization problem is in general non-convex and thus, the results can suffer from poor local optima. Secondly, for some problem instances the marginal likelihood approximate can exhibit discontinuities (as discussed in experiment 7.1).

To counter these issues, we consider Bayesian optimization (Shahriari et al., 2016; Snoek et al., 2012) as a third strategy to model selection as it does not depend on gradient information. As indicated by the name, Bayesian optimization is a probabilistic approach to

optimization, where the objective function is modelled as a random function. Thus, the approach allows us to model the potential discontinuities. Specifically, we use a Gaussian process to model log posterior density as follows

$$\ln Q(\boldsymbol{\Omega}|Y) \sim \mathcal{GP}\left(\mu\left(\boldsymbol{\Omega}\right), k\left(\boldsymbol{\Omega}, \boldsymbol{\Omega}'\right)\right), \tag{70}$$

where $\mu : \mathbb{R}^K \to \mathbb{R}$ is a mean function and $k : \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}$ is the kernel function. Rather than following the direction of the gradient, Bayesian optimization works by exploring values of $\boldsymbol{\Omega}$, that are likely to improve the value of the objective function as measured by a so-called acquisition function. For more details on Bayesian optimization, we refer to (Shahriari et al., 2016; Snoek et al., 2012; Brochu et al., 2010) .

7. Numerical Experiments

In this section, we conduct a series of experiments designed to investigate the properties of the proposed model and the associated EP inference scheme.

We describe seven experiments with a Gaussian observation model and one experiment with a probit observation model. In the first five experiments, we focus on problem instances with a single measurement vector. Experiment 1 investigates the effect of the prior by analyzing a synthetic data set with a range of different values for the hyperparameters. In the second experiment, we compare the three different approximation schemes (low rank, common precision, group) to standard EP. Specifically, we analyze a synthetic data set with all four methods and compare the results. Experiment 3 is designed to investigate how the EP algorithms perform as a function of the undersampling ratio N/D giving rise to the so-called *phase transition curves* (Donoho and Tanner, 2010). In experiment 4, we apply the proposed model to a compressed sensing problem and in experiment 5, we apply our model to a binary classification task, where the goal is to discriminate between utterances of two different vowels using log-periodograms as features.

In Experiment 6-8, we turn our attention to problems with multiple measurement vectors. In the sixth experiment, we qualitatively study the properties of the proposed methods in the multiple measurement vector setting. We demonstrate the benefits of modeling both the spatial and temporal structure of the support and discuss the marginal likelihood approximation for hyperparameter tuning. Experiment 7 studies the performance of the EP algorithms as a function of the undersampling ratio when multiple measurement vectors are available and compare the results to competing methods. Finally, in Experiment 8 we apply the proposed method to an EEG source localization problem (Baillet et al., 2001).

For the subset of experiments, where the ground truth solutions are available, we use the *Normalized Mean Square Error (NMSE)* and the *F-measure* (Rijsbergen, 1979) to quantify the performance of the algorithms. In particular, we compute the NMSE between the posterior mean $\hat{\boldsymbol{X}} = \mathbb{E}_{Q(\boldsymbol{X}|\boldsymbol{Y})}\left[\boldsymbol{X}\right]$ and the true solution $\boldsymbol{X}_0$ to quantify the algorithms abilities to reconstruct the true signal $\boldsymbol{X}_0$

$$\mathrm{NMSE} = \frac{\|\hat{\boldsymbol{X}} - \boldsymbol{X}_0\|_{\mathrm{F}}^2}{\|\boldsymbol{X}_0\|_{\mathrm{F}}^2}, \tag{71}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm. We use the F-measure to quantify the algorithms' abilities to recover the true support set

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}, \tag{72}$$

where *precision* (positive predictive value) is the fraction of non-zero weights found by the algorithm that are also non-zero in the true model, while recall *sensitivity* is the fraction of non-zeros in the true model that have been identified by the algorithm. Here a given weight $x_{i,t}$ is identified as being non-zero if the posterior mean of $z_{i,t}$ is above 0.5.

The code is available at `https://github.com/MichaelRiis/SSAS`.

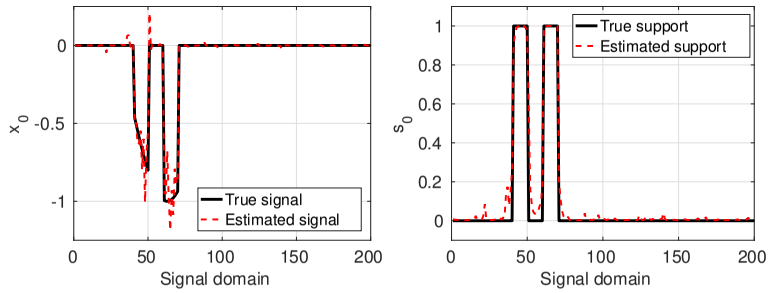
7.1 Experiment 1: The Effect of the Prior

In this experiment, we investigate the effect of the structured spike-and-slab prior on the reconstructed support set. For simplicity we only consider spatial structure, $T = 1$, and we further assume that the spatial coordinates are on a regular 1D grid. We construct a sparse 1D test signal $\boldsymbol{x}_0 \in \mathbb{R}^{200}$, where the active coefficients are sampled from a cosine function, see Figure 1(a)–(b). Based on this signal we generate a synthetic data set using the linear model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{e}$, where $A_{ij} \sim \mathcal{N}(0,1)$, $\boldsymbol{e} \sim \mathcal{N}(0, 5\boldsymbol{I})$ is isotropic Gaussian noise (SNR $\approx$ 5dB) and the number of samples is $N = 0.5D$. The prior on $\boldsymbol{\gamma}$ is of the form $p(\boldsymbol{\gamma}) = \mathcal{N}(\boldsymbol{\gamma}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\boldsymbol{\mu} = \nu \cdot \mathbf{1} \quad \text{and} \quad \boldsymbol{\Sigma}_{ij} = \kappa_1^2 \exp\left(-\frac{D_{ij}^2}{2\kappa_2^2}\right).$$

We sample the length-scale κ_2 equidistantly 100 times in $\left[10^{-3}, 50\right]$ and run the algorithm on the synthetic data set for each value of κ_2. For this experiment we use the standard EP scheme with no further approximations. The noise variance is fixed to the true value and the remaining hyperparameters are fixed $\nu = 0$, $\tau = 1, \kappa_1^2 = 5$. The posterior results are shown in the panels in leftmost column in Figure 2. The topmost panel shows the marginal likelihood approximation as a function of the spatial length scale κ_2. The panel in the middle shows the posterior mean $\mathbb{E}_{Q(z_i|\boldsymbol{y})}[\gamma_i]$, as a function of the scale κ_2. That is, each column in the image corresponds to the posterior mean of $\boldsymbol{\gamma}$ for a specific value of κ_2. The panel in the bottom shows a similar plot for the posterior support probabilities $\mathbb{E}_{Q(z_i|\boldsymbol{y})}[z_i]$.

When κ_2 is close to zero the posterior mean vectors for both $\boldsymbol{\gamma}$ and $\boldsymbol{z}$ are very irregular and resemble the solution of an independent spike-and-slab prior. As the length-scale increases the posterior mean vector $\boldsymbol{\gamma}$ becomes more and more smooth and eventually give rise to well-defined clusters in the support. The algorithm recovers the correct support for $\kappa_2 \in [3, 15]$. However, at $\kappa_2 \approx 15$ a discontinuity is seen. Since the prior distribution on $\boldsymbol{z}$ does not exhibit any phase transitions with regard to κ_2, this is likely to be an effect of a unimodal approximation to a highly multimodal distribution. The discontinuity is also present in the marginal likelihood approximation as seen in the top panel and therefore one should be cautious when optimizing the marginal likelihood using line search based methods. We repeated this experiment for multiple realizations of the noise and the discontinuity was



(a) Signal (b) Support

Figure 1: (a) Synthetic test signal $\boldsymbol{x}_0$ superimposed with the posterior mean of the test signal. The active coefficients are sampled from a cosine function (b) The support of the test signal superimposed with the posterior support probabilities.

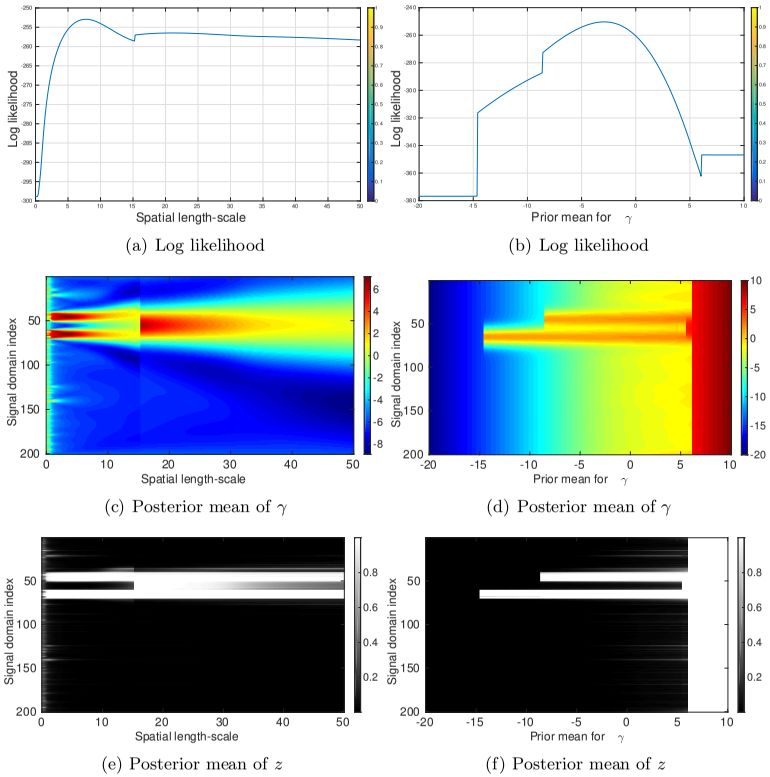
only present occasionally.

The rightmost column in Figure 2 shows equivalent figures for a sweep over ν, which is the prior mean of γ_i, where it is seen that the algorithm recovers the correct support for $\nu \in [-15, 0]$. It is seen that when ν is below some threshold ν_{lower}, the posterior mean of z_i is close to zero for all $i \in [D]$. The total number of active variables increases with ν, until ν surpasses an upper threshold ν_{upper}, where all variables are included in the support set. It is also seen that variables are included cluster-wise rather than individually, which gives rise to discontinuities in the marginal likelihood in the topmost panel.

Figure 1 shows the estimated signal and the estimated support probabilities for the optimal hyperparameter values in the top row of Figure 2, that is the prior mean $\nu = -2.93$ and lengthscale $\kappa_2 = 7.72$, where it is seen that both the estimated coefficients $\hat{\boldsymbol{x}}$ and the estimated support $\hat{\boldsymbol{s}}$ are high-quality approximations of the true quantities. We will make these relationships more quantitative in experiment 3.

7.2 Experiment 2: Comparison of Approximation Schemes

In this experiment, we investigate the properties of the proposed algorithm and the three approximation schemes: standard EP (EP), the low rank approximation (LR-EP), the common precision approximation (CP-EP) and the group approximation (G-EP). Using a similar setup as in Experiment 1, we generated a sample of $\boldsymbol{\gamma}_0$, $\boldsymbol{z}_0$ and $\boldsymbol{x}_0$ from the prior distribution specified in eq. (5)-(7) with $\rho_0 = 0, \tau_0 = 1$ and a squared exponential kernel with variance $\kappa_1^2 = 100$ and lengthscale $\kappa_2 = 75$. The generated sample is shown in the leftmost panels in Figure 3. We generated observations from a linear measurement model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{e}$, where $A_{ij} \sim \mathcal{N}(0,1)$ and the noise variance σ^2 is chosen such that the signal-to-noise is 20dB. Next, we computed the posterior distributions of $\boldsymbol{x}_0$, $\boldsymbol{z}_0$ and $\boldsymbol{\gamma}_0$ from



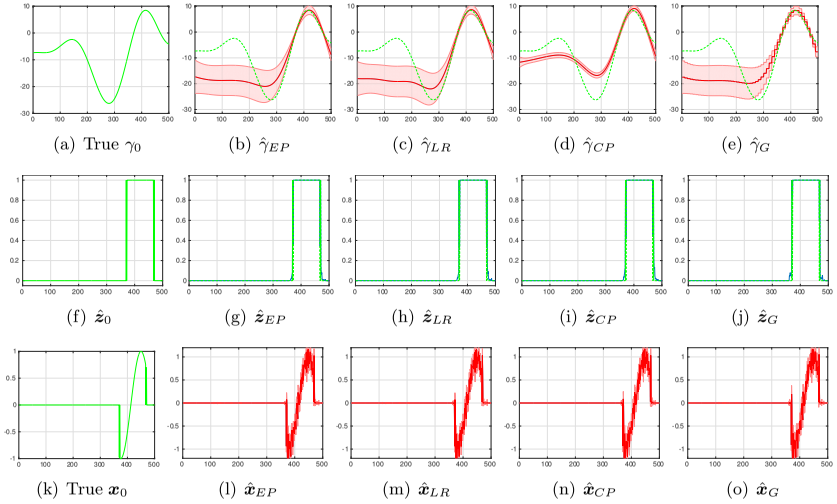
(a) Log likelihood (b) Log likelihood

(c) Posterior mean of γ (d) Posterior mean of γ

(e) Posterior mean of z (f) Posterior mean of z

Figure 2: The effect of the spatial prior distribution. The left-most column shows the approximate marginal log likelihood, posterior mean of $\boldsymbol{\gamma}$ and posterior mean of z as a function of the prior length-scale of $\boldsymbol{\gamma}$. The right-most column shows similar plots as a function of the prior mean ν_0 of $\boldsymbol{\gamma}$.

the observed measurements $\boldsymbol{y}$ using standard EP and the three approximation schemes. For the low rank approximation we included 7 eigenvectors corresponding to 99% percent of the variance and for G-EP we used a group size of 10 variables. Columns 2-5 in Figure 3 show the posterior mean values for $\boldsymbol{x}, \boldsymbol{z}$, and $\boldsymbol{\gamma}$ for EP, LR-EP, CP-EP and G-EP, respectively. Consider the posterior mean and standard deviation for $\boldsymbol{\gamma}$ for standard EP (topmost row, second column). In the region where $\boldsymbol{\gamma}_0$ is positive the posterior mean accurately recovers $\boldsymbol{\gamma}_0$ with high precision, whereas both the accuracy and the precision is lower in regions where $\boldsymbol{\gamma}_0$ is negative. The reason for the additional uncertainty is that negative values of γ_i are in



(a) True $\boldsymbol{\gamma}_0$ (b) $\hat{\boldsymbol{\gamma}}_{EP}$ (c) $\hat{\boldsymbol{\gamma}}_{LR}$ (d) $\hat{\boldsymbol{\gamma}}_{CP}$ (e) $\hat{\boldsymbol{\gamma}}_{G}$

(f) $\hat{\boldsymbol{z}}_0$ (g) $\hat{\boldsymbol{z}}_{EP}$ (h) $\hat{\boldsymbol{z}}_{LR}$ (i) $\hat{\boldsymbol{z}}_{CP}$ (j) $\hat{\boldsymbol{z}}_{G}$

(k) True $\boldsymbol{x}_0$ (l) $\hat{\boldsymbol{x}}_{EP}$ (m) $\hat{\boldsymbol{x}}_{LR}$ (n) $\hat{\boldsymbol{x}}_{CP}$ (o) $\hat{\boldsymbol{x}}_{G}$

Figure 3: Comparison of approximation schemes. The panels in the first column shows a realization $\boldsymbol{x}, \boldsymbol{z}$, and $\boldsymbol{\gamma}$ from the prior distribution in eq. (5)-(7). The columns 2-5 show the posterior mean quantities for EP, the low rank approximation (LR-EP), the common precision approximation (CP-EP) and the group approximation (G-EP), respectively. The pink shaded areas depict $\pm$ standard deviation.

general associated with a small value of $|x_i|$, but $|x_i|$ can be small for two reason. Recall that each x_i can be considered as a product $x_i = z_i \cdot c_i$, where $z_i \in \{0, 1\}$ and $c_i \in \mathbb{R}$. If $z_i = 0$, then clearly $x_i = 0$, but we can still have that $x_i \approx 0$ even if $z_i = 1$ and $c_i \approx 0$ and thus the increased uncertainty.

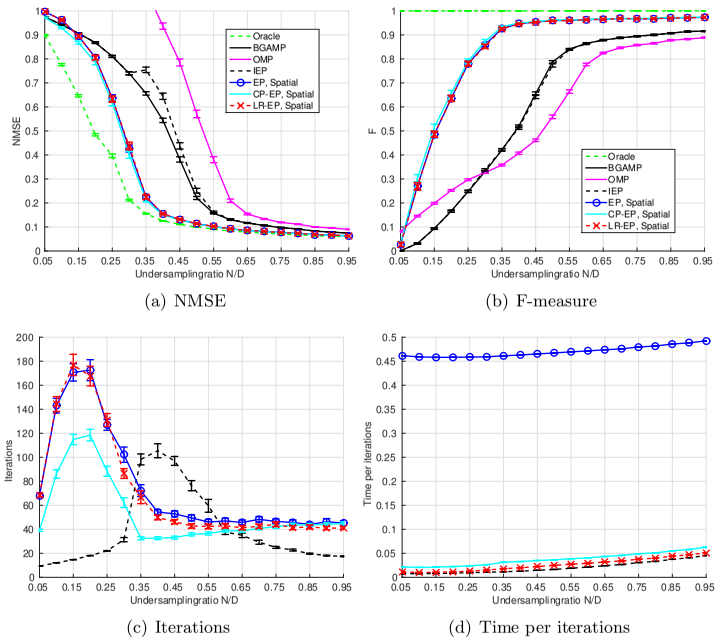
We can now compare the posterior distribution of $\boldsymbol{\gamma}_i$ for standard EP with the three approximations. Based on visual inspection one cannot tell the difference between the standard EP and EP with the low rank approximation, but the results for CP-EP and G-EP are quite different. For CP-EP it is seen that the posterior mean in the positive region is accurate, but the CP-EP approximation underestimates the uncertainty in general. The grouping effect for G-EP is clearly seen in the topmost panel in the last column, but despite the staircase pattern the posterior mean and variance are accurately recovered. The second and the third row in Figure 3 show the reconstructions of $\boldsymbol{x}$ and $\boldsymbol{z}$. We see that all of the four approaches accurately reconstruct the true quantities despite the approximation of the posterior distribution of $\boldsymbol{\gamma}$.

7.3 Experiment 3: Phase Transitions for a Single Measurement Vector

The purpose of this experiment is two-fold. The experiment serves to validate the inference algorithm, but it also serves to quantify the relationship between the recovery performance of the algorithm as a function of the undersampling ratio. It is well-known that the quality of the inferred solutions strongly depend on both the undersampling ratio $\delta = N/D$ and the number of non-zeros $K = ||\boldsymbol{x}||_0$ and that linear inverse problems exhibit a phase transition from almost perfect recovery to no recovery of solution $\boldsymbol{x}$ in the (δ, K)-space (Donoho and Tanner, 2010; Donoho et al., 2011). We hypothesize that the phase transition curves for signals with structured support can be improved, so that we can recover structured sparse signals using fewer measurements for a given number of non-zero coefficients K by taking advantage of the structure. We investigate this hypothesis by measuring the recovery performance of the EP algorithms as a function of the undersampling ratio N/D and compare with state-of-the-art probabilistic methods that ignore the structure of the support.

Using a squared exponential kernel for $\boldsymbol{\gamma}$ with variance $\kappa_1^2 = 50$ and lengthscale $\kappa_2 = 10$, we generated 100 realizations of $\boldsymbol{x}_0$ from the prior for $D = 500$. We fixed the expected sparsity to $K = \frac{1}{4}D = 125$ by choosing the prior mean of $\boldsymbol{\gamma}$ to $\nu = \phi^{-1}\left(\frac{1}{4}\right)(1 + \kappa_1^2)$. As the recovery performance is very sensitive to the number of non-zero coefficients, we conditioned each sample of $\boldsymbol{x}$ on $||\boldsymbol{x}||_0 = K$ by discarding samples where $||\boldsymbol{x}||_0 \neq K$ to reduce the variance of the resulting curves for NMSE and F-measure. For each of the samples, we generated measurements $\boldsymbol{y} \in \mathbb{R}^N$ through the linear observation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{e}$ for a range of values for N. The forward model $\boldsymbol{A}$ is a Gaussian i.i.d. ensemble, where the column have been scaled to unit ℓ_2-norm. The noise $e \sim \mathcal{N}\left(0, \sigma^2\right)$ is zero-mean Gaussian noise, where the noise variance σ^2 is chosen such that the signal-to-noise (SNR) ratio is fixed to 20dB. We choose values of N such that $\frac{N}{D} \in [0.05, 0.10, ..., 0.95]$.

We compare our methods with Bernoulli-Gaussian Approximate Message Passing (BG-AMP) (Vila and Schniter, 2013), Orthogonal Matching Pursuit (OMP) (Needell and Tropp,



(a) NMSE (b) F-measure

(c) Iterations (d) Time per iterations

Figure 4: Performance of the methods as a function of undersampling ratio N/D for $T = 1$. We compare full EP (EP, Spatial), EP with diagonal prior covariance (IEP), the common precision approximation (CP-EP) and the low rank approximation (LR-EP). The results are averaged over 100 realization.

2010) and an "oracle" estimator, which computes a ridge regression estimate based on knowledge of the true support. In this work, we use the BG-AMP method as baseline. It uses a (generalized) approximate message passing algorithm (Sundeep, 2010) for inference in a probabilistic model with i.i.d. spike-and-slab priors and a Gaussian likelihood. The AMP algorithm is closely related to EP algorithm (Meng et al., 2015), and the phase transition curve for BG-AMP is state of the art to the best of our knowledge. The OMP algorithm is a non-probabilistic greedy method, that iteratively select the column of $\boldsymbol{A}$ that correlate best with the current residuals until a pre-specified number of columns have been selected. The regularization parameters for the ridge regression is fixed to $\lambda = 10^{-3}$ for all runs. Finally, for comparison we also apply the proposed EP algorithm with a diagonal prior covariance matrix, which correspond to the conventional independent spike-and-slab prior (IEP). We provide BG-AMP and OMP with prior knowledge of the true number of non-zero variables in $\boldsymbol{x}_0$ and the noise variance used to generate the observations. The results are shown in Figure 4.

The two black curves in Figure 4 show the results for BG-AMP (black, solid) and EP with diagonal prior covariance (black, dashed). Both of these methods are based on conventional independent spike-and-slab priors. It is seen that the methods with prior correlation, that is EP (blue), CP-EP (cyan), and LR-EP (red, dashed), are uniformly better than the methods with independent priors both in terms of NMSE and F-measure. In fact, these methods achieve as good performance as the support-aware oracle estimator around $N/D = 0.6$ in terms of NMSE. Furthermore, it is also seen that the two approximations CP-EP and LR-EP are indistinguishable from the full EP algorithm in terms of accuracy. Panel (c) and (d) show the number iterations and the run time per iteration for the EP-based methods. Here it is seen that IEP has the lowest computational complexity per iteration, but the CP-EP and LR-EP are almost as fast.

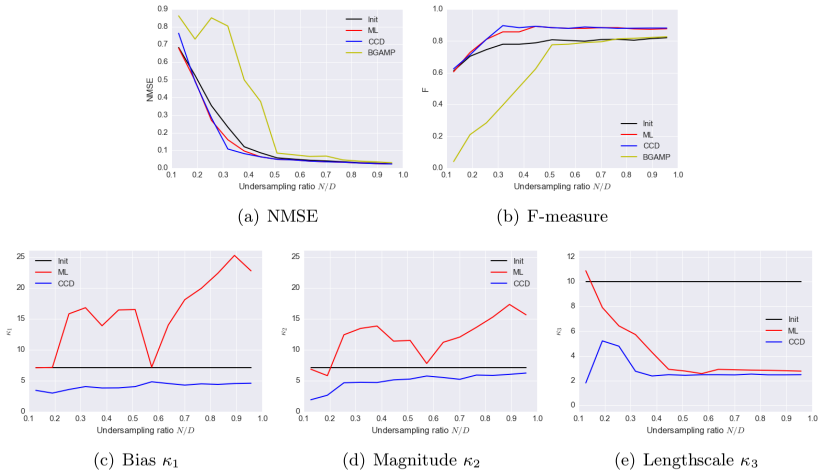
7.4 Experiment 4: Compressed Sensing of Optical Characters

In this experiment, we apply the structured spike-and-slab model with Gaussian likelihood to an application of compressed sensing (Donoho, 2006) of numerical characters from the MNIST data set (LeCun et al., 1998; Hernández-Lobato et al., 2013). The images of the numerical digits are 28 pixels $\times$ 28 pixels and they are represented as vectors $\boldsymbol{x}_0 \in \mathbb{R}^{784}$. The objective is to reconstruct $\boldsymbol{x}_0$ from a small set of linear and noisy measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_0 + \boldsymbol{\epsilon}$. The sensing matrix $\boldsymbol{A}$ is sampled independently from a standardized Gaussian distribution, that is $A_{ni} \sim \mathcal{N}(0, 1)$ and the noise variance is scaled such that the SNR is fixed 10dB.

We use a squared exponential kernel with a single lengthscale defined on the 2D image grid to encourage the neighbourhood structure expected in the images. We impose a Gaussian prior distribution on ν_0 with zero mean and variance κ_1^2, that is $\nu_0 \sim \mathcal{N}(0, \kappa_1^2)$ and integrate over ν_0 analytically to get the kernel function

$$k(i, j) = \kappa_1^2 + \kappa_2^2 \exp\left(-\frac{\|\boldsymbol{d}_i - \boldsymbol{d}_j\|_2^2}{2\kappa_3^2}\right), \tag{73}$$

where $\boldsymbol{d}_i$ is the image grid coordinates of γ_i. We assume that the noise variance is known and we fix the prior mean and variance of the 'slab' component to a standardized Gaussian with $\rho_0 = 0$ and $\tau_0 = 1$. Thus, the hyperparameters to be learned are $\boldsymbol{\Omega} = \{\kappa_1, \kappa_2, \kappa_3\}$. For the CCD procedure, we have to choose prior distributions for the hyperparameters. For the lengthscale parameter, we can use the fact that the 'pen' is roughly a few pixels wide on average and choose a log-normal prior with mean 4 and standard deviation 2, that is $\kappa_3 \sim \mathcal{LN}\left(4, 2^2\right)$. The 10'th and 90'th percentiles for this distribution are approximately 2 and 7, respectively. For the remaining two hyperparameters, we will use the same prior distribution, that is $\kappa_1, \kappa_2 \sim \mathcal{LN}\left(4, 2^2\right)$, but for a different reason than for the lengthscale parameter κ_3. The mode of the distribution $\mathcal{LN}\left(4, 2^2\right)$ is approximate 2.9 and then the 10'th and 90'th percentiles of the distribution of $\phi(\gamma)$ for $\gamma \sim \mathcal{N}(0, 2.9^2)$ are approximately 0.0001 and 0.9999, respectively. Furthermore, the choice of lognormal priors generally works well for the CCD scheme, which can yield poor performance if the distributions have too heavy tails.



(a) NMSE (b) F-measure

(c) Bias κ_1 (d) Magnitude κ_2 (e) Lengthscale κ_3

Figure 5: Performance of compressed sensing of numerical digits as a function of undersampling ratio. Panels (a) and (b) show the NMSE and F-measure, while panels (c)–(e) show the estimated values of the hyperparameters as a function of the undersampling ratio. For the CCD method, the panels (c)–(e) show the CCD-weighted average of the hyperparameters.

We use the low-rank approximation for all computations in this experiment. Figure 5(a)–(b) show the NMSE reconstruction error and F measure as a function of the undersampling ratio $\frac{N}{D}$. In this experiment, we also compare with the BGAMP method, which is informed about the noise level. We also use a standardized Gaussian as slab distribution for BGAMP. The black curves in panels (a) and (b) show the performance for the model when the hyperparameter are fixed to the initial values. It is seen that for small undersampling rates $N/D < 0.5$, we obtain slightly better results in terms of NMSE when adapting the hyperparameters, but we get a uniform improvement in terms of the F measure. Figure 5(c)–(e) show the estimated values for the hyperparameters as a function of the undersampling ratio. It is seen that the ML solution tends to overestimate the lengthscale for small sample sizes. In this case, only weak information are propagated from the observations to the prior of $\boldsymbol{\gamma}$ and thus the model becomes over-regularized. It is also seen that the bias and magnitude parameters are correlated as expected from the relationship in eq. 8, (see Appendix D for more details).

Figure 6 shows the posterior mean of the support for each method for $N/D \approx 0.3$, where it is seen that we obtain a qualitative and quantitative improvement of the support estimate by taking a priori knowledge into account and integrating over the uncertainty. Figure 7 shows the posterior distribution for $\boldsymbol{\gamma}$, $\boldsymbol{z}$ and $\boldsymbol{x}$ for the line indicated by the green dashed

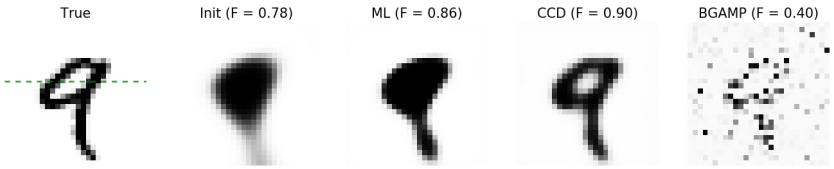
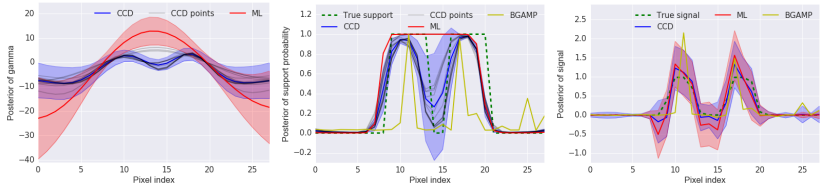
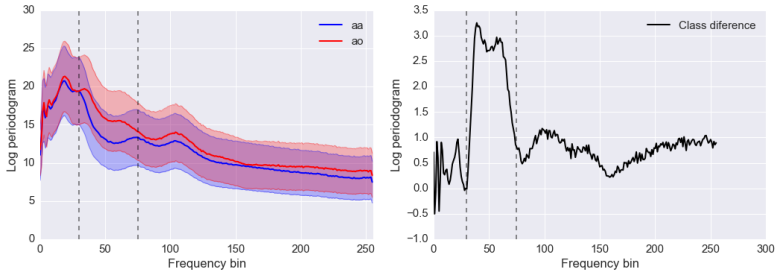


Figure 6: Posterior mean of $\boldsymbol{z}$ for compressed sensing of numerical digits, where $N/D \approx 0.3$. The panels in the top row show the posterior mean of the support, while the panels in the bottom row show the posterior mean of signal. Figure 7 shows the posterior distributions of the row indicated the dashed line.



(a) Posterior $\boldsymbol{\gamma}$ (b) Posterior support (c) Posterior signal

Figure 7: Comparison of the posterior distribution of $\boldsymbol{\gamma}$, $\boldsymbol{z}$ and $\boldsymbol{x}$ for the row shown by the green dashed line in Figure 6 for $N/D \approx 0.3$.



(a) The two classes (b) Difference in mean

Figure 8: a) The frequency-wise mean and standard deviation of the log-periodogram of the two spoken vowels "aa" and "ao". b) The difference of the two mean signals.

Method	**Training error**	**Test error**	**Train LPPD**	**Test LPPD**
NBSBC	9.7 (±0.3)	19.5 (±0.1)	−42.7 (±0.8)	−698.6 (±3.3)
LR-EP (ML)	13.3 (±0.3)	19.4 (±0.1)	−50.6 (±0.8)	−673.8 (±2.2)
LR-EP (CCD)	13.4 (±0.3)	**19.2** (±0.1)	−50.7 (±0.8)	**−665.5** (±1.5)

Table 2: Results for phoneme classification experiment

line in Figure 6(a). Recall, that the posterior distributions obtained using CCD are finite mixture models. The thin gray lines in left and center columns show the posterior mean of the individual mixture components, while the solid colored lines and the shaded areas show the mean and variance of the mixture distributions, respectively. From the center panel, it is seen all methods fail to capture the true support perfectly, but the mean of the support of the CCD solution are significantly improved compared to the ML solution and more interestingly, the CCD solution also has high variance in the region, where it is wrong. These uncertainties are not properly reflected in the NMSE and F metrics, but the log posterior density of the true support of the ML solution is −181.654, while the same quantity for the CCD method and BG-AMP evaluate to −74.181 and −339.065, respectively.

7.5 Experiment 5: Phoneme Recognition

In this experiment, we consider the task of phoneme recognition (Hastie et al., 2001; Hernandez-Lobato et al., 2011). In particular, we consider the problem of discriminating between the spoken vowels "aa" and "ao" using their log-periodograms as features. The data set consists of 695 and 1022 utterances of the vowels "aa" and "ao", respectively, along with their corresponding labels. The response variable in this experiment is binary and therefore we use the probit model rather than the Gaussian observation model.

Each log-periodogram has been sampled at 256 uniformly spaced frequencies. The left-most

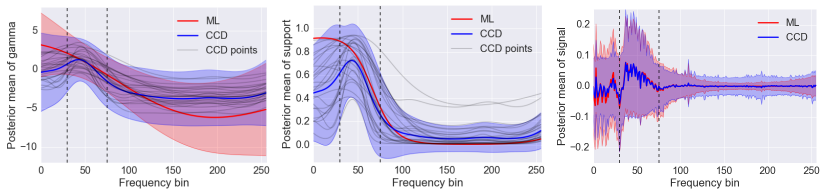
panel in Figure 2 shows the frequency-wise mean and standard deviation of the two classes and the right-most panel shows the difference of the two mean signals. We choose a squared exponential kernel for $\boldsymbol{\gamma}$ since it is assumed that frequency bands rather than single frequencies are relevant for discriminating between the two classes. The total number of observations is 1717 and we use $N = 150$ examples for training and the remaining 1567 examples for testing. We repeat the experiment 100 times using different partitions into training and test sets. The training and test sets are generated such that the prior odds of the two classes are the same for both training and test. The number of input features is $D = 257$ (256 frequencies + bias).

We use the low rank approximation for this experiment, and we choose the number of eigenvectors such that 99% of the total variance in $\boldsymbol{\Sigma}_0$ is explained. We use the maximum likelihood method and the CCD marginalization method for the hyperparameter inference. We choose the prior mean of $\boldsymbol{x}$ as $\rho_0 = 0$ to reflect our ignorance on the sign of the active weights and we impose a half Student's t distribution on the prior standard deviation of the $\boldsymbol{x}$, that is $\sqrt{\tau_0} \sim t^+(\mathrm{df} = 4)$, which is considered to be weakly informative.

As in the compressed sensing experiment, we impose a zero-mean normal distribution on the prior mean of $\boldsymbol{\gamma}$ and integrate it out analytically to obtain a kernel of the form given in eq. (73). Compared to the compressed sensing example, our a priori knowledge of the structure of the support are more diffuse, but we expect that the lengthscale is significantly smaller than the number of frequency bins. Therefore, we choose a log-normal prior with mean 40 and standard deviation 30, that is $\kappa_3 \sim \mathcal{LN}\left(40, 30^2\right)$. The 5'th and 95'th percentiles for this distribution is approximately 10 and 100, respectively. For the remaining two hyperparameters, we use the same two prior distributions as in the previous experiment, that is $\kappa_1, \kappa_2 \sim \mathcal{LN}\left(4, 2^2\right)$. To predict the label of a new observation, we compute the predictive distribution by integrating the probit likelihood with respect to the approximate posterior distribution of the weights.

We compare our method against the network-based sparse Bayesian classification (NBSBC) method, which also uses EP to approximate the posterior distribution of linear model with coupled spike-and-slab priors. Instead of using a Gaussian process to encode the structure of the support, the NBSBC model encodes the structure in a network using a Markov random field prior. This method has been shown to outperform competing method on this specific problems (Hernandez-Lobato et al., 2011).

Table 2 summarizes the performance of the methods based on the average number of misclassifications and average log posterior predictive density (LPPD). It is seen that the LR-EP (ML) method achieves similar performance in terms test error as the NBSBC method, but it performs marginally better in terms of test LPPD. On the other hand, it is seen that the LR-EP (CCD) method outperforms both other methods. The panels in Figure 9 shows the posterior distributions of $\boldsymbol{\gamma}$, $\boldsymbol{z}$ and $\boldsymbol{x}$. The posterior distribution for the CCD approximation is a finite mixture model and each of the thin black lines shows the posterior mean for each individual mixture component. However, these are omitted for the posterior of $\boldsymbol{x}$ to improve the visual clarity of the figure. Based on Figure 2(b), we expect the weights for the frequencies between bin 35 and bin 70 to most discriminative of the two classes and



(a) Posterior $\boldsymbol{\gamma}$ (b) Posterior support (c) Posterior signal

Figure 9: Comparison of the posterior distribution of $\boldsymbol{\gamma}$, $\boldsymbol{z}$ and $\boldsymbol{x}$ for ML and CCD hyperparameter inference. The posterior distribution for the CCD approximation is a mixture model and the thin black solid lines show the posterior mean of each individual component. These are omitted for the posterior of the signal to improve visual clarity.

it is seen that both the ML method and the CCD method have high posterior probabilities for the support in the region.

7.6 Experiment 6: Spatio-temporal Example

In the previous experiments the focus was on problems with only one measurement vector, whereas in this and the following experiments we consider problems with multiple measurement vectors. Specifically, in this experiment we qualitatively study the properties of the proposed algorithm in the spatio-temporal setting using simulated data. We have synthesized a signal, where the support set satisfies the following three properties: 1) non-stationarity, 2) spatiotemporal correlation, and 3) the number of active coefficients change over time. The support of the signal is shown in panel (a) in Figure 10. Based on the support set, we sample the active coefficients from a zero-mean isotropic Gaussian distribution. We then observe the desired signal $\boldsymbol{X}$ through linear measurements $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E}$, where both the forward model $\boldsymbol{A}$ and the noise $\boldsymbol{E}$ is sampled from a zero-mean isotropic Gaussian distribution. The noise variance is scaled such that the SNR is 5dB. We apply our proposed method to estimate $\boldsymbol{X}$ given the forward model $\boldsymbol{A}$ and the observations $\boldsymbol{Y}$.

Panel (b) in Figure 10 shows the reconstructed support $\boldsymbol{Z}$ using the proposed EP algorithm with a diagonal prior covariance matrix on $\boldsymbol{\Gamma}$, which implies no prior correlation in the support. The panels (c)–(f) shows the reconstructed support for full EP, low rank EP, common precision EP and group EP, which all assumes that the prior covariance matrix for $\boldsymbol{\Gamma}$ is a Kronecker product of two squared exponential components. For the group approximation the group size is chosen to 5 and 10 in the spatial and temporal dimension, respectively, and for the low-rank approximation the rank is chosen such that the minimum number of eigenvectors explain 99% of the variance in the prior. All hyperparameters are chosen by maximizing the approximate marginal likelihood. By inspecting the panels (a)–(f) it is seen that the reconstructed support is qualitatively improved by modeling the additional

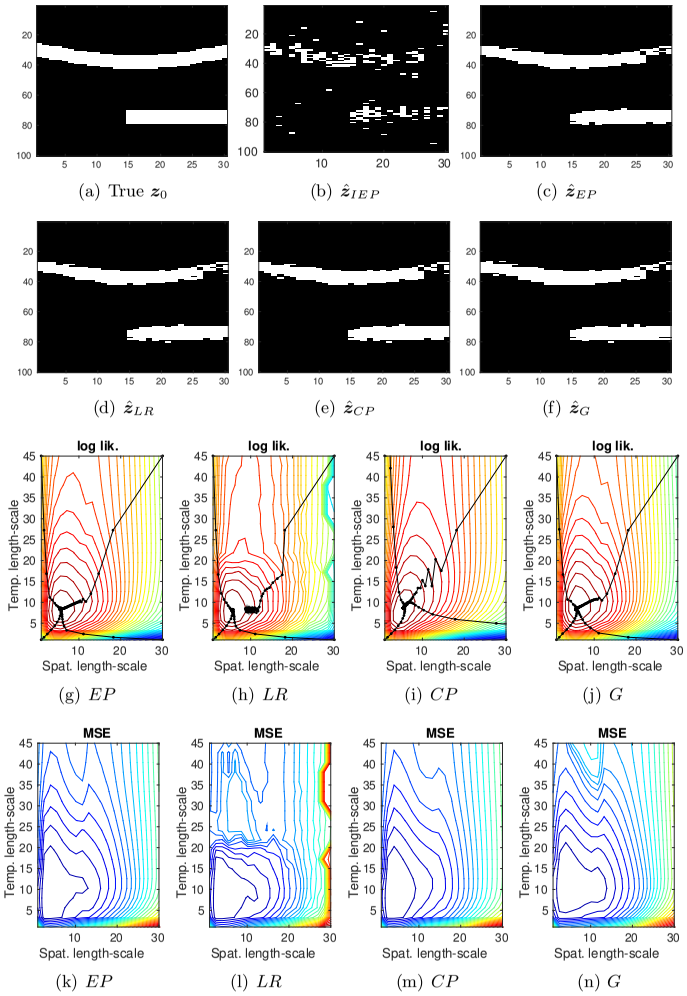
structure. Furthermore, the reconstructions using the approximation schemes do not differ significantly from the result using full EP.

Panels (g)–(j) shows the marginal likelihood approximation as a function of the spatial and temporal length scale of the prior covariance matrix for the proposed methods, while the panels (k)–(g) show the corresponding NMSE between the reconstructed coefficients $\hat{\boldsymbol{X}}$ and the true coefficient $\boldsymbol{X}$. The black curves superimposed on the marginal likelihood plots show the trajectories of the optimization path for the length-scales of the prior distribution starting from four different initial values. It is seen that the marginal likelihood approximation is unimodal and correlates strongly with the NMSE surface, which suggests that it is reasonable to tune the length-scales of the prior covariance using the marginal likelihood approximation. However, we emphasize that this is not always the case and for some problems this indeed leads to suboptimal results.

7.7 Experiment 7: Phase Transitions for Multiple Measurement Vectors

The multiple measurement vector problem also exhibits a phase transition analogously to the single measurement vector problem described in Experiment 3 (Cotter et al., 2005; Ziniel and Schniter, 2013a; Andersen et al., 2015). In this experiment, we investigate how the location of the phase transition of the EP algorithms improves when the sparsity pattern of the underlying signal is smooth both in space and time and multiple measurement vectors are available. Using a similar setup as in Experiment 3, we generate 100 realizations of $\boldsymbol{X}$ from the prior specified in eq. (10)–(12) such that the total number of active components is fixed to $K = \frac{1}{4}DT = 2500$. The covariance structure is of the form $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_{\text{temporal}} \otimes \boldsymbol{\Sigma}_{\text{spatial}}$, where both the temporal and spatial components are chosen to be squared exponential kernels. Figure 11 shows an example of a sample realization of $\boldsymbol{\Gamma}$, $\boldsymbol{Z}$ and $\boldsymbol{X}$ from the prior distribution. For each of the realizations of $\boldsymbol{X}$, we generate a set of linear observations $Y = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{E}$, where the forward model $\boldsymbol{A}$ is Gaussian i.i.d. and $E_{nt} \sim \mathcal{N}\left(0, \sigma^2\right)$ is zero-mean Gaussian scaled such that the SNR is fixed to 20dB. For reference we compare our methods against BG-AMP (Vila and Schniter, 2013) and DCS-AMP (Ziniel and Schniter, 2013a). The DCS-AMP method is a temporal extension to the BG-AMP method (see Experiment 3 for a brief description), and it uses approximate message passing inference based on spatially i.i.d. spike-and-slab priors, but assumes that the binary support variables evolve in time according to a first order Markov process. Both BG-AMP and DCS-AMP methods are informed about the true number of active coefficients and the true noise level. The results are shown in Figure 12.

The method LR-EP (blue) assumes that the sparsity pattern is spatially correlated, but independent in time. The method LR-K-EP (red, dashed) applies the low rank approximation to EP and assumes that the sparsity pattern is spatio-temporally correlated and that the prior covariance for $\boldsymbol{\Gamma}$ is described by a Kronecker product (hence the prefix "-K"). Similarly, the methods CP-K-EP (cyan) and G-K-EP (magenta) have the same assumptions about the sparsity pattern, but use the common precision approximation and the group approximation, respectively. For G-K-EP we use groups of 5 in both the spatial dimension and temporal dimension. In this experiment, we do not run full EP with the spatiotemporal prior because it would be prohibitively slow.



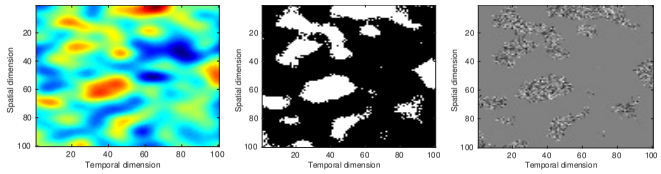
(a) True $\boldsymbol{z}_0$ (b) $\hat{\boldsymbol{z}}_{IEP}$ (c) $\hat{\boldsymbol{z}}_{EP}$

(d) $\hat{\boldsymbol{z}}_{LR}$ (e) $\hat{\boldsymbol{z}}_{CP}$ (f) $\hat{\boldsymbol{z}}_{G}$

(g) EP (h) LR (i) CP (j) G

(k) EP (l) LR (m) CP (n) G

Figure 10: Results for a simulated spatio-temporal example with $D = 100$, $T = 30$, $N = 33$ and SNR $= 5dB$. Panels (a)–(f) compare the true sparsity pattern and the reconstructed sparsity patterns and panels (g)–(n) show the approximate marginal likelihood and the MSE error metric as a function of the spatial and temporal length-scale. For the low rank approximation (LR-EP), the number of eigenvectors is chosen to explain 99% of the variance and the group size for group approximation (G-EP) is chosen to 5 and 10 in the spatial and temporal dimensions, respectively.



(a) Sample of $\mathbf{\Gamma}$ (b) Sample of $\boldsymbol{Z}$ (c) Sample of $\boldsymbol{X}$

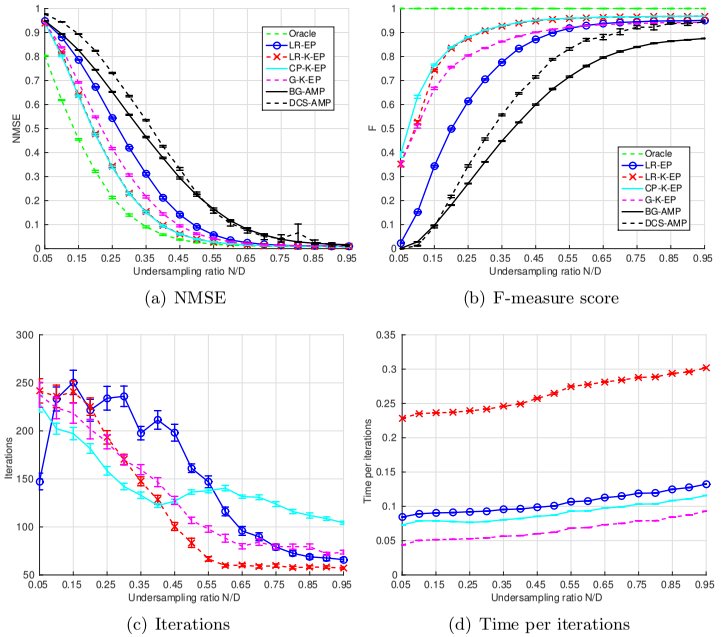
Figure 11: Example of a realization of the synthetic signals in Experiment 6.

On panel (a) in Figure 12 it is seen that as the number of measurements increase, all methods eventually reach the NMSE level of the support-aware oracle estimator, but the general picture is that the more structure a method takes into account (i.i.d. sparsity vs. spatial sparsity vs. spatio-temporal sparsity), the better it performs in terms of NMSE. In particular, at $N/D \approx 0.3$ BG-AMP achieves NMSE ≈ 0.63 and LR-EP achieves NMSE ≈ 0.44 while LR-K-EP and CP-K-EP achieve NMSE ≈ 0.24. Panel (b) shows a similar picture for F-measure. Furthermore, it is seen that the performance of LR-K-EP and CP-K-EP are similar and slightly better than the performance of G-K-EP both in terms of NMSE and F-measure. However, the G-K-EP approximation has the lowest computational complexity per iteration as seen in panel (d). In terms of run time the EP-methods are slower compared to the AMP-based methods, which have linear time complexity in all dimensions. However, the EP methods are not limited to Gaussian i.i.d. ensembles as the AMP-based methods are.

7.8 Experiment 8: EEG Source Localization

In the final experiment, we apply the proposed method to an EEG source localization problem (Baillet et al., 2001), where the objective is to infer the locations of the active sources on the cortical surface of the human brain based on electroencephalogram (EEG) measurements. The brain is modelled using a discrete set of current dipoles distributed on the cortical surface and Maxwell's equations then describe how the magnitudes of the dipole sources relate to the EEG signals measured at the scalp. We apply the proposed method to an EEG data set, where the subjects are presented with pictures of faces and scrambled faces. The data set is publicly available and the experimental paradigm is described in (Henson et al., 2003). The data set has $N = 128$ electrodes and contains a total of 304 epochs evenly distributed between the two conditions: face or scrambled face. Each epoch has a duration of $800ms$ corresponding to $T = 161$ samples in time and the stimuli are presented at $t = 0s$. We generated a forward model[1] with 5124 dipole sources, that is $\boldsymbol{A} \in \mathbb{R}^{128\times 5124}$. To encourage spatio-temporal coherence of the sources, we choose the covariance matrix for $\mathbf{\Gamma}$ to be of the form $\mathbf{\Sigma}_0 = \kappa^2 \cdot \mathbf{\Sigma}_{\text{temporal}} \otimes \mathbf{\Sigma}_{\text{spatial}}$, where both the temporal component and spatial component are squared exponential kernels with individual length-scales. For simplicity, we use the Euclidean distance to compute the pairwise distances among the

1. We used the SPM8 software (Ashburner et al., 2010).



(a) NMSE (b) F-measure score

(c) Iterations (d) Time per iterations

Figure 12: Performance of the methods as a function of undersampling ratio N/D for $T = 100$. We compare low-rank EP with spatial structure only (LR-EP, Spatial), low rank EP spatio-temporal kronecker structure (LR-L-EP), the common precision EP with spatio-temporal kronecker structure (CP-K-EP), group EP with spatio-temporal kronecker structure (G-K-EP), the low rank approximation (LR-EP) with BGAMP and DCSAMP. The results are averaged over 100 realization.

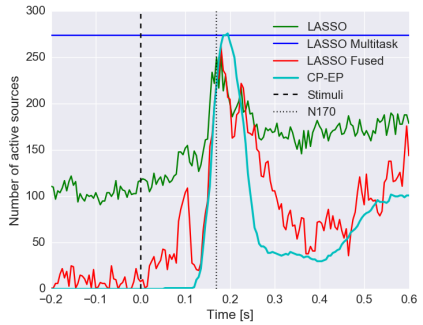


Figure 13: The number of active dipoles sources as a function of the time for the common precision approximation (CP-EP) and the LASSO, the Multi-task LASSO and the fused LASSO for a face perception experiment. The stimuli were presented at time $t = 0$.

dipole sources as opposed to the more advances approach, where the distances are computed within the manifold defined by the cortex.

The resulting inverse problem has $N = 128$ measurements, $T = 161$ measurement vectors and $D = 5124$ unknowns per time point and a total of $DT = 5124 \cdot 151 = 824964$ unknowns. The forward model has a condition number of $\text{cond}(\boldsymbol{A}) = 3.1099 \cdot 10^{15}$. Thus, the problem instance is both heavily ill-posed and ill-conditioned. Because of the dimensions of this problem we use the common precision approximation for this data set. In fact, a low rank approximation of the prior covariance matrix $\boldsymbol{\Sigma}_0$ will require 3961 eigenvectors to explain 90% of the variance and the matrix low rank eigenvector matrix $\boldsymbol{U} \in \mathbb{R}^{824964 \times 3691}$ would then require more than $20GB$ of memory to store in 64 bit double precision.

Tuning the hyperparameters using the approximate marginal likelihood estimate leads to poor solutions for this data set. In particular, the length-scales were significantly overestimated, which is consistent with what we observed in the compressed sensing experiment for very small samples sizes (see Figure 5(e)). However, manually specifying the hyperparameters using prior knowledge (spatial lengthscale $10mm$, temporal lengthscale $50ms$ and magnitude $\kappa^2 = 10$, prior mean 0), yields a posterior approximation with several interesting aspects. Ideally, we would compare the findings with the same posterior quantities for the BG-AMP and DCS-AMP methods as discussed earlier, but the highly correlated columns of the forward model make the AMP-approximations break down as they assume that the entries of the forward model are sampled from an Gaussian i.i.d. distribution. Instead, we compare with the LASSO[2] (Tibshirani, 1994), the multi-task LASSO [2](Obozinski et al., 2006) and

2. We used the implementation in scikit-learn toolbox (Pedregosa et al., 2011).

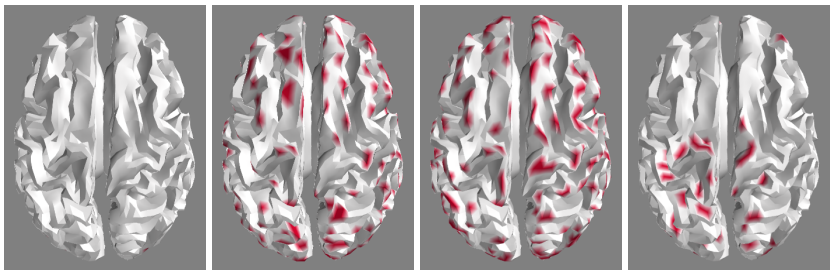
the fused LASSO[3] (Tibshirani et al., 2005). The LASSO, the multi-task LASSO and the fused LASSO all minimize a quadratic reconstruction error subject to an ℓ_1 constraint, but the multi-task LASSO also assumes that the sparsity pattern is constant in time (joint sparsity) and the fused LASSO has an additional constraint on the temporal first-order difference of the solution $\sum_{i,t} |x_{i,t} - x_{i,t-1}|$.

The CP-EP method has been informed with appropriate values of the kernel hyperparameters. This is possible because the structure of the sparsity pattern is encoded using generic covariance functions that are easily interpretable. However, this is not an option for the LASSSO methods as the regularization parameters of these methods are more difficult to interpret in terms of the geometry of the problem. To compensate, we used the estimated solution obtained using the CP-EP method to inform the LASSO methods as follows. For the regular LASSO and the multi-task LASSO, we chose the value of the regularization parameter such that the number of active sources matches the CP-EP solution at the time point with the largest number of active sources (see Figure 13). For the fused LASSO, we also matched the average autocorrelation across all sources.

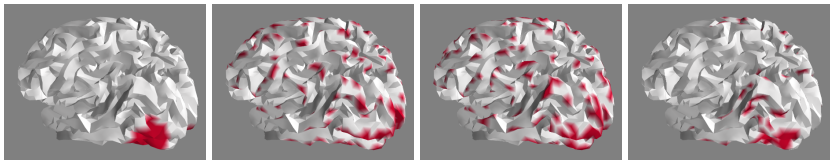
Figure 13 shows the number of active dipole sources as a function of time for each method. The reconstructed support for both CP-EP and the fused LASSO are well-localized in time, whereas the distribution of active sources for LASSO are very diffuse in time. For the CP-EP method, it is seen that the number of active sources is zero until roughly time $t \approx 150ms$, where the number of active sources increase and peaks at $t \approx 180ms$, which is consistent with the known time delay of approximately 170ms for the face perception, that is the so-called N170 ERP component (Itier and Taylor, 2004). Figure 14 shows a visualization of the estimated sets of active sources for time $t = 180ms$ from a top view, a side view and a bottom view, respectively. Interestingly, CP-EP detects four spatially coherent areas: left and right occipital and fusiform face areas that are associated with the face perception (Henson et al., 2009). The LASSO, the Multi-task LASSO and the fused LASSO also detect several active dipoles in the left and right occipital areas, but they also detect active sources distributed over the entire cortex as seen in the top row.

Thus, from this experiment we conclude that this problem is too ill-posed for learning the hyperparameters of the model, but we can still extract meaningful information from the data using the model if we have access to additional a priori information. We note that learning hyperparameters is often difficult in neuroimaging due to high-dimensional signals and poor signal to noise conditions (Varoquaux et al., 2017).

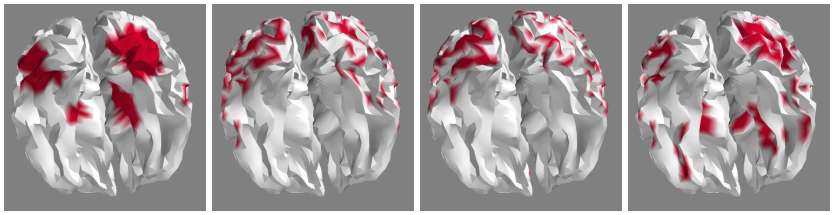
3. We used the implementation in SPAMS toolbox (Jenatton et al., 2010; Mairal et al., 2010).



(a) CP-EP (b) LASSO (c) LASSO Multitask (d) LASSO Fused



(e) CP-EP (f) LASSO (g) LASSO Multitask (h) LASSO Fused



(i) CP-EP (j) LASSO (k) LASSO Multitask (l) LASSO Fused

Figure 14: Estimate support sets for each method at time $t \approx 180ms$ for the face perception experiment. The top, middle and bottom rows show the brain from the top, side and bottom respectively.

8. Summary and Outlook

In this work, we have addressed the problem of solving multiple underdetermined linear inverse problems subject to a sparsity constraint. We have proposed a new generalization of the spike-and-slab prior distribution to encode a priori correlation of the support of the solution in both space and time by imposing a transformed Gaussian process on the spike-and-slab probabilities. An expectation propagation (EP) algorithm for posterior inference under the proposed model has been derived. Computations involved in EP updates scale like $\mathcal{O}\left(D^3T^3\right)$ where D is the number of features and T is number of inverse problems, hence for large scale problems, the standard EP algorithm can be prohibitively slow. We therefore introduced three different approximation schemes for the covariance structure to reduce the computational complexity. First, assuming that the prior has a Kronecker decomposition brings complexity $\mathcal{O}\left(D^3 + T^3\right)$, based on this decomposition, a further K-rank approximation brings a reduction of complexity to $\mathcal{O}\left(K^2DT\right)$, we also proposed a common precision approximation of complexity $\mathcal{O}\left(D^2T + T^2D\right)$, and finally a scheme based on spatio-temporal grouping of variables effectively reducing D and T by the grouping factor. We also discussed several ways to handle unknown hyperparameters, including maximum likelihood estimates, maximum a posterior estimates and efficient numerical integration using central composite design (CCD) approach.

We investigated the role of the spatio-temporal prior and the approximation schemes in a series of experiments. First we studied a simple 1D problem with spatial, translational invariant smoothness of the support (single measurement case, $T = 1$). For a signal with two small connected components in the support, we illustrate the solutions for variable smoothness of the prior. For a wide range of prior parameters the correct form of the support is recovered, while the two support regions were found to merge in a single region as the smoothness length scale approaches the distance between the two regions. In the second experiment we investigated the role of the three covariance function approximations, also in a single measurement setup ($T = 1$). We found that all approaches accurately reconstruct the true simulated support and inverse problem solutions despite the approximation of the Gaussian process posterior. It is well-known that the quality of the inferred solutions strongly depends on both the undersampling ratio and the sparsity level of the true solution. We investigated how the location of the phase transition is improved by invoking the smoothness prior. We found that the methods based on assumed prior correlation, were uniformly better than the methods with independent priors both in terms of the quality of normalized mean square error and in terms of their accuracy of support recovery (F-measure). The covariance approximation schemes are almost as fast as the scheme without smoothness, while yielding greatly improved performance.

In the experiments 4 and 5, we investigated two applications: compressed sensing of numerical characters and phoneme recognition, respectively. In the former, we demonstrated how the quality of the reconstructed digits was improved using the structured prior. We also found that for the severely undersampled problems, maximum likelihood learning tends to overestimate the lengthscale of the kernel, which in turn lead to poor estimates of the support of and the weight. However, we also demonstrated how this could be alleviated by

imposing a proper prior distribution to the lengthscale parameter and integrating over the uncertainty using CCD. In the latter experiment, we demonstrated how the probit likelihood can be used to extend to model binary sparse classification problems and we found that our algorithm compare well with published benchmarks.

In a sixth experiment we studied the properties of the proposed algorithms in the spatio-temporal setting using simulated data. Signals were synthesized so that the support set showed non-stationarity, spatio-temporal correlation and so that the cardinality of the support set changed over time. We estimated prior hyperparameters by optimizing the approximate marginal likelihood and found they converged to optimal settings in all cases. We found that there was a good correspondence between the approximate marginal likelihood and the solution's quantitative performance measure (NMSE).

Also for the multiple measurement vector problem it is known that there is a phase transition-like dependence of the solution quality on undersampling ratio. In the sixth experiment we investigated how the location of the phase transition of the EP algorithms improved when the sparsity pattern of the underlying signal is smooth in both space and time for the multiple measurements case. We compare our various approximate solvers with the state-of-the-art tools based on approximate message passing: BG-AMP, DCS-AMP, both of which were informed about the true number of active coefficients and the true noise level. The full EP was too demanding to run for this problem. Significant improvement were found for the methods that exploited sparsity structure. Comparing performance with AMP methods, the EP methods performed best both in terms of identifying the support (F measure) and in terms of NMSE. Run times for the EP-methods were longer compared to the AMP-based methods, which have linear complexity in all dimensions. We noted importantly that the EP methods also can be used for more general forward model ensembles (A), while the AMP-based methods assume a Gaussian i.i.d. ensemble.

In the final experiment, we applied the proposed methods to the hard problem of EEG source localization; data for this experiment was derived from a publicly available brain imaging data set designed to detect brain areas involved in face perception (Henson et al., 2003). This was a larger scale application with $N = 128$ measurements and a total number of 824964 unknowns, hence, only the common precision approximation was feasible. Furthermore, the forward model was very ill-conditioned in contrast to the well-conditioned i.i.d. ensembles considered in the simulations. For this data set, the hyperparameters of the kernel, for example, spatial and temporal lengthscale, could not be estimated from the data and thus, additional prior knowledge was required to perform inference. In spite of these challenges highly interesting results were obtained: All four main foci of activation as earlier detected by fMRI, but not in these EEG data by other inference schemes, were here found to have well-defined and spatially extended support by the new approximate inference scheme. In contrast to fMRI EEG allowed us to monitor the dynamics in these areas in high temporal resolution.

This work has led to several interesting lines of research. First of all, from the the compressed sensing experiment as well as the source localization experiment, we concluded that the

lengthscale parameter of the kernel cannot be learned from the data if the problem is too ill-posed. Thus, in future work we will extend the model to handle EEG data for multiple subjects simultaneously in a hierarchical manner, which allows us to use much more data to estimate the hyperparameters. Future studies also include an analysis of the phase transitions of the approximate log marginal likelihood in the hyperparameter space of the spatiotemporal prior as discussed in Experiment 1. Furthermore, we also plan to apply the proposed algorithms to brain decoding problems, for example, in classification of fMRI task pattern data sets. Finally, we also plan to investigate the use of spatio-temporal sparsity priors for factor models like PCA and ICA.

Appendix A. Moments Computations for $f_{2,t,j}$

In this section, we consider the update for the terms $\tilde{f}_{2,t,j}\left(x_{t,j}, z_{t,j}\right)$. First we compute the so-called cavity distribution $Q^{\backslash 2,t,j}\left(x_{t,j}, z_{t,j}\right)$ by removing the contribution of $f_{2,t,j}(x_{t,j}, t, j)$ from the marginals of the joint approximation $Q\left(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\gamma}\right)$

$$\begin{aligned} Q^{\backslash 2,t,j}(x_{t,j}, z_{t,j}) &= \frac{Q\left(x_{t,j}, z_{t,j}\right)}{\tilde{f}_{2,t,j}\left(x_{t,j}, z_{t,j}\right)} = \frac{\mathcal{N}\left(x_{t,j} \middle| \hat{m}_{t,j}, \hat{V}_{t,j}\right) \mathrm{Ber}\left(z_{t,j} \middle| \phi\left(\hat{\gamma}_{t,j}\right)\right)}{\mathcal{N}\left(x_{t,j} \middle| \hat{m}_{2,t,j}, \hat{V}_{2,t,j}\right) \mathrm{Ber}\left(z_{t,j} \middle| \phi\left(\hat{\gamma}_{2,t,j}\right)\right)} \\ &= K^{\backslash 2,t,j} \cdot \mathcal{N}\left(x_{t,j} \middle| \hat{m}^{\backslash 2,t,j}, \hat{V}^{\backslash 2,t,j}\right) \mathrm{Ber}\left(z_{t,j} \middle| \phi\left(\hat{\gamma}^{\backslash 2,t,j}\right)\right), \end{aligned} \tag{74}$$

where

$$\hat{v}^{\backslash 2,t,j} = \left[\hat{V}_{jj}^{-1} - \hat{v}_{2,t,j}^{-1}\right]^{-1}, \tag{75}$$

$$\hat{m}^{\backslash 2,t,j} = \hat{v}^{\backslash 2,t,j}\left[\hat{V}_{t,jj}^{-1}\hat{m}_{t,j} - \hat{v}_{2,t,j}^{-1}\hat{m}_{2,t,j}\right], \tag{76}$$

$$\hat{\gamma}^{\backslash 2,t,j} = \hat{\gamma}_{3,t,j}. \tag{77}$$

Note that the cavity parameter for γ for $f_{2,t,j}$ is simply equal to $\hat{\gamma}_{3,t,j}$ (and vice versa) since $\hat{\gamma}_{2,t,j}$ and $\hat{\gamma}_{3,t,j}$ are the only two terms contributing to $\gamma_{t,j}$.

Next, we minimize the KL-divergence between $f_{2,t,j}Q^{\backslash 2,t,j}$ and q or equivalently matching the moments between the two distributions. Following the latter approach we first compute the (unnormalized) moment w.r.t. $z_{t,j}$

$$\begin{aligned} Z_1 &= \sum_{z_{t,j}} \int z_{t,j} f_{2,t,j}\left(x_{t,j}, z_{t,j}\right) Q^{\backslash 2,t,j}\left(x_{t,j}, z_{t,j}\right) \mathrm{d}x_{t,j} \\ &= \phi\left(\hat{\gamma}^{\backslash 2,t,j}\right) \mathcal{N}\left(0 \middle| \hat{m}^{\backslash 2,i} - \rho_0, \hat{V}^{\backslash 2,t,j} + \tau_0\right). \end{aligned} \tag{78}$$

Next, the zeroth moment w.r.t $x_{t,i}$ or the normalization constant of $f_{2,t,j}Q^{\setminus 2,t,j}$

$$\begin{aligned} X_0 &= \sum_{z_{t,j}} \int f_{2,t,j}\left(x_{t,j}, z_{t,j}\right) Q^{\setminus 2,t,j}\left(x_{t,j}, z_{t,j}\right) \mathrm{d}x_{t,j} \qquad (79)\\ &= \sum_{z_{t,j}} \int \left[\left(1 - z_{t,j}\right)\delta(x_{t,j}) + z_{t,j}\mathcal{N}\left(x_{t,j}\middle|\rho_0, \tau_0\right)\right] \\ &\quad \mathcal{N}\left(x_{t,j}\middle|\hat{m}^{\setminus 2,t,j}, \hat{V}^{\setminus 2,t,j}\right) \mathrm{Ber}\left(z_{t,j}\middle|\phi\left(\hat{\gamma}^{\setminus 2,t,j}\right)\right) \mathrm{d}x_{t,j} \\ &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 2,t,j}\right)\right)\mathcal{N}\left(0\middle|\hat{m}^{\setminus 2,i}, \hat{V}^{\setminus 2,i}\right) + \phi\left(\hat{\gamma}^{\setminus 2,t,j}\right)\mathcal{N}\left(0\middle|\hat{m}^{\setminus 2,t,j} - \rho_0, \hat{V}^{\setminus 2,t,j} + \tau_0\right) \\ &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 2,t,j}\right)\right)\mathcal{N}\left(0\middle|\hat{m}^{\setminus 2,i}, \hat{V}^{\setminus 2,i}\right) + Z_1 \qquad (80) \end{aligned}$$

We now compute the (unnormalized) first moment w.r.t. $x_{t,j}$

$$\begin{aligned} X_1 &= \sum_{z_{t,j}} \int x_{t,j} f_{2,t,j}\left(x_{t,j}, z_{t,j}\right) Q^{\setminus 2,i}\left(x_{t,j}, z_{t,j}\right) \mathrm{d}x_{t,j} \\ &= \phi\left(\hat{\gamma}^{\setminus 2,t,j}\right)\mathcal{N}\left(0\middle|\hat{m}^{\setminus 2,t,j} - \rho_0, \hat{V}^{\setminus 2,t,j} + \tau_0\right) \frac{\frac{\hat{m}^{\setminus 2,t,j}}{\hat{V}^{\setminus 2,t,j}} + \frac{\rho_0}{\tau_0}}{\frac{1}{\tau_0} + \frac{1}{\hat{V}^{\setminus 2,t,j}}} \\ &= Z_1 \frac{\hat{m}^{\setminus 2,t,j}\tau_0 + \rho_0 \hat{V}^{\setminus 2,t,j}}{\tau_0 + \hat{V}^{\setminus 2,t,j}} \qquad (81) \end{aligned}$$

and the second (unnormalized) moment w.r.t. $x_{t,j}$

$$\begin{aligned} X_2 &= \sum_{z_{t,j}} \int x_i^2 f_{2,i}\left(x_i, z_i\right) Q^{\setminus 2,i}\left(x_i, z_i\right) \mathrm{d}x_i \\ &= \phi\left(\hat{\gamma}^{\setminus 2,i}\right)\mathcal{N}\left(0\middle|\hat{m}^{\setminus 2,i} - \rho_0, \hat{V}^{\setminus 2,i} + \tau_0\right)\left[\left(\frac{\frac{\hat{m}^{\setminus 2,i}}{\hat{V}^{\setminus 2,i}} + \frac{\rho_0}{\tau_0}}{\frac{1}{\tau_0} + \frac{1}{\hat{V}^{\setminus 2,i}}}\right)^2 + \frac{1}{\frac{1}{\tau_0} + \frac{1}{\hat{V}^{\setminus 2,i}}}\right] \\ &= Z_1\left[\left(\frac{\hat{m}^{\setminus 2,t,j}\tau_0 + \rho_0\hat{V}^{\setminus 2,t,j}}{\tau_0 + \hat{V}^{\setminus 2,t,j}}\right)^2 + \frac{\tau_0\hat{V}^{\setminus 2,i}}{\hat{V}^{\setminus 2,i} + \tau_0}\right] \qquad (82) \end{aligned}$$

The central moments for Q^* in eq. (21) are given by

$$E\left[x_{t,j}\right] = \frac{X_1}{X_0}, \qquad V\left[x_{t,j}\right] = \frac{X_2}{X_0} - \frac{X_1^2}{X_0^2}, \qquad E\left[z_{t,j}\right] = \frac{Z}{X_0}. \qquad (83)$$

Appendix B. Moment Computations for $\tilde{f}_{3,t,j}$

The moments matching for $\tilde{f}_{3,t,j}$ is derived in a similar manner as for $\tilde{f}_{2,t,j}$ (see appendix A for details). First we compute the cavity distribution $Q^{\setminus 3,t,j}\left(z_{t,j}, \gamma_{t,j}\right)$ by removing the

contribution of $f_{3_t,j}(z_{t,j}, \gamma_{t,j})$ from the marginals of the joint approximation Q

$$\begin{aligned} Q^{\setminus 3,t,j}(z_{t,j}, \gamma_{t,j}) &= \frac{Q\left(z_{t,j}, \gamma_{t,j}\right)}{\tilde{f}_{3,t,j}\left(z_{t,j}, \gamma_{t,j}\right)} = \frac{\mathrm{Ber}\left(z_{t,j}\middle|\phi\left(\hat{\gamma}_{t,j}\right)\right)\mathcal{N}\left(\gamma_{t,j}, \hat{\mu}_{t,j}, \hat{\Sigma}_{t,jj}\right)}{\mathrm{Ber}\left(z_{t,j}\middle|\phi\left(\hat{\gamma}_{3,t,j}\right)\right)\mathcal{N}\left(\gamma_{t,j}, \hat{\mu}_{3,t,j}, \hat{\Sigma}_{3,t,j}\right)} \\ &= K^{\setminus 3,t,j} \cdot \mathrm{Ber}\left(z_{t,j}\middle|\phi\left(\hat{\gamma}^{\setminus 3,t,j}\right)\right)\mathcal{N}\left(\gamma_{t,j}\middle|\hat{\mu}^{\setminus 3,t,j}, \hat{\Sigma}^{\setminus 3,t,j}\right), \qquad (84) \end{aligned}$$

where

$$\hat{\Sigma}^{\setminus 3,t,j} = \left(\hat{\Sigma}_{t,jj}^{-1} - \Sigma_{3,t,j}^{-1}\right)^{-1}, \qquad (85)$$

$$\hat{\mu}^{\setminus 3,t,j} = \hat{\Sigma}^{\setminus 3,t,j}\left(\hat{\Sigma}_{t,jj}^{-1}\hat{\mu}_{t,j} - \hat{\Sigma}_{3,t,j}^{-1}\hat{\mu}_{3,t,j}\right), \qquad (86)$$

$$\hat{\gamma}^{\setminus 3,t,j} = \hat{\gamma}_{2,t,j}. \qquad (87)$$

Once again we minimize the KL-divergence between $f_{3,t,j}Q^{\setminus 3,t,j}$ and Q or equivalently matching the moments between the two distributions. We now compute the moments w.r.t. $\gamma_{j,t}$ and $z_{j,t}$ of the (unnormalized) tilted distribution

$$G_m = \sum_{z_{j,t}} \int \gamma_{j,t}^m \cdot f_{3,t,j}\left(z_{j,t}, \gamma_{j,t}\right) Q^{\setminus 3,t,j}\left(z_{j,t}, \gamma_{j,t}\right) \mathrm{d}\gamma_{j,t} \quad \text{for} \quad m = 0, 1, 2, \qquad (88)$$

$$Z_1 = \sum_{z_{j,t}} \int z_{j,t} \cdot f_{3,t,j}\left(z_{j,t}, \gamma_{j,t}\right) Q^{\setminus 3,t,j}\left(z_{j,t}, \gamma_{j,t}\right) \mathrm{d}\gamma_{j,t} \qquad (89)$$

We first compute the normalization constant of $f_{3,t,j}Q^{\setminus 3,t,j}$

$$\begin{aligned} G_0 &= \sum_{z_{t,j}} \int f_{3,t,j}\left(z_{t,j}, \gamma_{t,j}\right) Q^{\setminus 3,t,j}\left(z_{t,j}, \gamma_i\right) \mathrm{d}\gamma_{t,j} \\ &= \sum_{z_{t,j}} \int \mathrm{Ber}\left(z_{t,j}\middle|\phi\left(\gamma_{t,j}\right)\right)\mathrm{Ber}\left(z_{t,j}\middle|\phi\left(\hat{\gamma}^{\setminus 3,t,j}\right)\right)\mathcal{N}\left(\gamma_{t,j}\middle|\hat{\mu}^{\setminus 3,t,j}, \hat{\Sigma}^{\setminus 3,t,j}\right)\mathrm{d}\gamma_{t,j} \\ &= \sum_{z_i} \int \left[\left(1 - z_i\right)\left(1 - \phi\left(\gamma_i\right)\right)\left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) + z_i\phi\left(\gamma_i\right)\phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right]\mathcal{N}\left(\gamma_i\middle|\hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right)\mathrm{d}\gamma_i \\ &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right)\int\left(1 - \phi\left(\gamma_i\right)\right)\mathcal{N}\left(\gamma_i\middle|\hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right)\mathrm{d}\gamma_i \\ &\quad + \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\int\phi\left(\gamma_i\right)\mathcal{N}\left(\gamma_i\middle|\hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right)\mathrm{d}\gamma_i \end{aligned}$$

Integrals of the form $\int \phi\left(\gamma_i\right)\mathcal{N}\left(\gamma_i\middle|\hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right)\mathrm{d}\gamma_i$ can be solved analytically (Rasmussen and Williams, 2006),

$$\int \phi\left(\gamma_i\right)\mathcal{N}\left(\gamma_i\middle|\hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right)\mathrm{d}\gamma_i = \phi\left(c_{3,i}\right), \quad c_{3,i} \triangleq \frac{\hat{\mu}^{\setminus 3,i}}{\sqrt{1 + \hat{\Sigma}^{\setminus 3,i}}}. \qquad (90)$$

Inserting this result back into the expression for G_0 yields

$$G_0 = \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right)\left(1 - \phi\left(c_{3,i}\right)\right) + \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\phi\left(c_{3,i}\right). \qquad (91)$$

We can now compute the moments of the unnormalized distribution

$$\begin{aligned} Z_1 &= \sum_{z_i} \int z_i f_{3,i}\left(z_i, \gamma_i\right) Q^{\setminus 3,i}\left(z_i, \gamma_i\right) \mathrm{d}\gamma_i \\ &= \phi\left(\hat{\gamma}^{\setminus 3,i}\right) \phi\left(c_{3,i}\right), \end{aligned} \tag{92}$$

Then the first moment w.r.t. to $z_{i,t}$ is obtained as $E\left[z_{i,t}\right] = Z_1/G_0$.

For the moments w.r.t. γ_i, we get

$$\begin{aligned} G_1 &= \sum_{z_i} \int \gamma_i f_{3,i}\left(z_i, \gamma_i\right) Q^{\setminus 3,i}\left(z_i, \gamma_i\right) \mathrm{d}\gamma_i \\ &= \sum_{z_i} \int \gamma_i \left[\left(1 - z_i\right)\left(1 - \phi\left(\gamma_i\right)\right)\left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) + z_i \phi\left(\gamma_i\right) \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right] \\ &\quad \mathcal{N}\left(\gamma_i \middle| \hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right) \mathrm{d}\gamma_i \\ &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) \int \gamma_i \left(1 - \phi\left(\gamma_i\right)\right) \mathcal{N}\left(\gamma_i \middle| \hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right) \mathrm{d}\gamma_i \\ &\quad + \phi\left(\hat{\gamma}^{\setminus 3,i}\right) \int \gamma_i \phi\left(\gamma_i\right) \mathcal{N}\left(\gamma_i \middle| \hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right) \mathrm{d}\gamma_i \\ &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) \left[\hat{\mu}^{\setminus 3,i} - \int \gamma_i \phi\left(\gamma_i\right) \mathcal{N}\left(\gamma_i \middle| \hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right) \mathrm{d}\gamma_i\right] \\ &\quad + \phi\left(\hat{\gamma}^{\setminus 3,i}\right) \int \gamma_i \phi\left(\gamma_i\right) \mathcal{N}\left(\gamma_i \middle| \hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right) \mathrm{d}\gamma_i \end{aligned} \tag{93}$$

Again we turn to (Rasmussen and Williams, 2006) for the analytical solution of the above integrals

$$\begin{aligned} \int \gamma_i \phi\left(\gamma_i\right) \mathcal{N}\left(\gamma_i \middle| \hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right) \mathrm{d}\gamma_i &= \phi\left(c_{3,i}\right) \hat{\mu}^{\setminus 3,i} + \phi\left(c_{3,i}\right) \frac{\hat{\Sigma}^{\setminus 3,i} \mathcal{N}\left(c_{3,i} \middle| 0, 1\right)}{\phi\left(c_{3,i}\right) \sqrt{1 + \hat{\Sigma}^{\setminus 3,i}}} \\ &= \phi\left(c_{3,i}\right) \hat{\mu}^{\setminus 3,i} + \phi\left(c_{3,i}\right) d_{3,i}, \end{aligned} \tag{94}$$

where we have defined

$$d_{3,i} \triangleq \frac{\hat{\Sigma}^{\setminus 3,i} \mathcal{N}\left(c_{3,i} \middle| 0, 1\right)}{\phi\left(c_{3,i}\right) \sqrt{1 + \hat{\Sigma}^{\setminus 3,i}}}. \tag{95}$$

Plugging eq. (94) back into eq. (93) and simplifying yields

$$\begin{aligned} G_1 &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) \left[\left(1 - \phi\left(c_{3,i}\right)\right) \hat{\mu}^{\setminus 3,i} - \phi\left(c_{3,i}\right) d_{3,i}\right] + \phi\left(\hat{\gamma}^{\setminus 3,i}\right) \phi\left(c_{3,i}\right) \left[\hat{\mu}^{\setminus 3,i} + d_{3,i}\right] \\ &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) \left(1 - \phi\left(c_{3,i}\right)\right) \hat{\mu}^{\setminus 3,i} - \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) \phi\left(c_{3,i}\right) d_{3,i} + Z_1 \left[\hat{\mu}^{\setminus 3,i} + d_{3,i}\right] \\ &= \left(G_0 - Z_1\right) \hat{\mu}^{\setminus 3,i} - \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) \phi\left(c_{3,i}\right) d_{3,i} + Z_1 \left[\hat{\mu}^{\setminus 3,i} + d_{3,i}\right] \\ &= G_0 \hat{\mu}^{\setminus 3,i} + \left(2 Z_1 - \phi\left(c_{3,i}\right)\right) d_{3,i} \end{aligned} \tag{96}$$

Thus, the first moment w.r.t. $\gamma_{i,t}$ is given by $\mathbb{E}\left[\gamma_{i,t}\right] = G1/G0$.

Similarly, we compute the second moment w.r.t. γ_i

$$\begin{aligned} G_2 &= \sum_{z_i} \int \gamma_i^2 f_{3,i}\left(z_i, \gamma_i\right) Q^{\setminus 3,i}\left(z_i, \gamma_i\right) \mathrm{d}\gamma_i \\ &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) \int \gamma_i^2 \left(1 - \phi\left(\gamma_i\right)\right) \mathcal{N}\left(\gamma_i \middle| \hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right) \mathrm{d}\gamma_i \\ &\quad + \phi\left(\hat{\gamma}^{\setminus 3,i}\right) \int \gamma_i^2 \phi\left(\gamma_i\right) \mathcal{N}\left(\gamma_i \middle| \hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right) \mathrm{d}\gamma_i \end{aligned} \tag{97}$$

The solution to the above integrals are given by (Rasmussen and Williams, 2006)

$$\begin{aligned} &\int \gamma_i^2 \phi\left(\gamma_i\right) \mathcal{N}\left(\gamma_i \middle| \hat{\mu}^{\setminus 3,i}, \hat{\Sigma}^{\setminus 3,i}\right) \mathrm{d}\gamma_i \\ &= \phi\left(c_{3,i}\right) \left[2 \hat{\mu}^{\setminus 3,i} \left(\hat{\mu}^{\setminus 3,i} + d_{3,i}\right) + \left(\hat{\Sigma}^{\setminus 3,i} - \left(\hat{\mu}^{\setminus 3,i}\right)^2\right) - b_{3,i}\right] \end{aligned} \tag{98}$$

where

$$b_{3,i} \triangleq \frac{\left(\hat{\Sigma}^{\setminus 3,i}\right)^2 c_{3,i} \mathcal{N}\left(c_{3,i} \middle| 0, 1\right)}{\phi\left(c_{3,i}\right)\left(1 + \hat{\Sigma}^{\setminus 3,i}\right)} \tag{99}$$

Furthermore, we define

$$w_{3,i} \triangleq 2 \hat{\mu}^{\setminus 3,i} \left(\hat{\mu}^{\setminus 3,i} + d_{3,i}\right) + \left(\hat{\Sigma}^{\setminus 3,i} - \left(\hat{\mu}^{\setminus 3,i}\right)^2\right) - b_{3,i} \tag{100}$$

Substituting the above result back into eq. (97) and rearranging yields

$$\begin{aligned} G_2 &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) \left[\left(\hat{\mu}^{\setminus 3,i}\right)^2 + \hat{\Sigma}^{\setminus 3,i} - \phi\left(c_{3,i}\right) w_{3,i}\right] + \phi\left(\hat{\gamma}^{\setminus 3,i}\right) \phi\left(c_{3,i}\right) w_{3,i} \\ &= \left(1 - \phi\left(\hat{\gamma}^{\setminus 3,i}\right)\right) \left[\left(\hat{\mu}^{\setminus 3,i}\right)^2 + \hat{\Sigma}^{\setminus 3,i} - \phi\left(c_{3,i}\right) w_{3,i}\right] + Z_1 w_{3,i} \end{aligned} \tag{101}$$

Thus, the second moment is given by $\mathbb{E}\left[\gamma_{i,t}^2\right] = G2/G0$. Finally, the central moments of Q^* then becomes

$$\mathbb{E}\left[\gamma_{j,t}\right] = \frac{G_1}{G_0}, \qquad \mathbb{V}\left[\gamma_{j,t}\right] = \frac{G_2}{G_0} - \frac{G_1^2}{G_0^2}, \qquad \mathbb{E}\left[z_{j,t}\right] = \frac{Z_1}{G_0}. \tag{102}$$

These moments completely determine the distribution $Q^{3,\text{new}}$ and thus, we compute the updates for $f_{3,i}$ as follows

$$\hat{\Sigma}_{3,i}^{\text{new}} = \left[\mathbb{V}\left[\gamma_i\right]^{-1} - \left(\hat{\Sigma}^{\setminus 3,i}\right)^{-1}\right]^{-1}, \tag{103}$$

$$\hat{\mu}_{3,i}^{\text{new}} = \hat{\Sigma}_{3,i}^{\text{new}} \left[\mathbb{V}\left[\gamma_i\right]^{-1} \mathbb{E}\left[\gamma_i\right] - \left(\hat{\Sigma}^{\setminus 3,i}\right)^{-1} \hat{\mu}^{\setminus 3,i}\right], \tag{104}$$

$$\hat{\gamma}_{3,i}^{\text{new}} = d\left(\phi\left(\mathbb{E}\left[z_i\right]\right), \hat{\gamma}^{\setminus 3,i}\right), \tag{105}$$

Appendix C. Moments Computations for Probit Likelihood

The purpose of this section is to describe the details of the EP approximation of the structured spike-and-slab prior with a probit likelihood. Using the notation described in Section 4, the probit likelihood term is given by

$$f_{1,t}\left(\boldsymbol{x}_t\right) = p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \prod_{n=1}^{N} \phi\left(y_{n,t}\boldsymbol{A}_{n,\cdot}\boldsymbol{x}_t\right). \tag{106}$$

First we compute the cavity distribution $Q^{\backslash 1,t,n}\left(\boldsymbol{x}\right)$ by removing the contribution of $\tilde{f}_{1t,n}(\boldsymbol{x})$ from the marginals of the joint approximation Q

$$Q^{\backslash 1,t,n}(\boldsymbol{x}_t) = \frac{\mathcal{N}\left(\boldsymbol{x}_t|\boldsymbol{m}_t, \boldsymbol{V}_t\right)}{\tilde{f}_{1,t,n}(\boldsymbol{x}_t)} = K^{\backslash 1,t,n}\mathcal{N}\left(\boldsymbol{x}_t|\boldsymbol{m}^{\backslash 1,t,n}, \boldsymbol{V}^{\backslash 1,t,n}\right), \tag{107}$$

where

$$\boldsymbol{V}^{\backslash 1,t,n} = \left(\hat{\boldsymbol{V}}_t^{-1} - \hat{\boldsymbol{V}}_{1,t,n}^{-1}\right)^{-1}, \tag{108}$$

$$\boldsymbol{m}^{\backslash 1,t,n} = \boldsymbol{V}^{\backslash 1,t,n}\left(\hat{\boldsymbol{V}}_t^{-1}\hat{\boldsymbol{m}}_t - \hat{\boldsymbol{V}}_{1,t}^{-1}\hat{\boldsymbol{m}}_{1,t,n}\right). \tag{109}$$

for diagonal matrices $\boldsymbol{V}_t$ and $\boldsymbol{V}^{\backslash 1,t,n}$. The tilted distribution then becomes

$$\hat{q}_{1,t,n}\left(\boldsymbol{x}_t\right) = \frac{1}{z_{1,t,n}}\phi\left(y_{n,t}\boldsymbol{A}_{n,\cdot}\boldsymbol{x}_t\right)\mathcal{N}\left(\boldsymbol{x}_t|\boldsymbol{m}^{\backslash 1,t,n}, \boldsymbol{V}^{\backslash 1,t,n}\right),$$

First we compute the normalization constant, which is given by

$$z_{1,t,n} = \int \phi\left(y_{n,t}\boldsymbol{A}_{n,\cdot}\boldsymbol{x}_t\right)\mathcal{N}\left(\boldsymbol{x}_t|\boldsymbol{m}^{\backslash 1,t,n}, \boldsymbol{V}^{\backslash 1,t,n}\right)\mathrm{d}\boldsymbol{x}_t \tag{110}$$

$$= \int \phi\left(u\right)\mathcal{N}\left(u|a_{1,t,n}, b_{1,t,n}\right)\mathrm{d}u \tag{111}$$

$$= \phi\left(c_{1,t,n}\right), \tag{112}$$

where $a_{1,t,n} = y_{n,t}\boldsymbol{A}_{n,\cdot}\boldsymbol{m}^{\backslash 1,t,n}$, $b_{1,t,n} = \boldsymbol{A}_{n,\cdot}\boldsymbol{V}^{\backslash 1,t,n}\boldsymbol{A}_{n,\cdot}^T$ and $c_{1,t,n} = \frac{a_{1,t,n}}{\sqrt{1+b_{1,t,n}}}$. Since $y_{n,t} \in \{-1, 1\}$, $y_{n,t}$ does not appear in the expression for $b_{1,t,n}$ due to square form. Define the row-vector $\tilde{\boldsymbol{A}}_n = y_{n,t}\boldsymbol{A}_{n,\cdot} \in \mathbb{R}^{1\times D}$, then the first moment w.r.t. $x_{t,j}$ is given by

$$\mathbb{E}\left[x_{t,j}\right] = \frac{1}{z_{1,t,n}}\int x_{t,j}\phi\left(y_{n,t}\boldsymbol{A}_{n,\cdot}\boldsymbol{x}_t\right)\mathcal{N}\left(\boldsymbol{x}_t|\boldsymbol{m}^{\backslash 1,t,n}, \boldsymbol{V}^{\backslash 1,t,n}\right)\mathrm{d}\boldsymbol{x}_t \tag{113}$$

$$= \frac{1}{z_{1,t,n}}\int x_{t,j}\phi\left(\tilde{\boldsymbol{A}}_{n,\cdot}\boldsymbol{x}_t\right)\mathcal{N}\left(\boldsymbol{x}_t|\boldsymbol{m}^{\backslash 1,t,n}, \boldsymbol{V}^{\backslash 1,t,n}\right)\mathrm{d}\boldsymbol{x}_t \tag{114}$$

$$= \frac{1}{z_{1,t,n}}\int x_{t,j}\int \phi\left(\tilde{\boldsymbol{A}}_{n,-j}\boldsymbol{x}_{-j} + \tilde{a}_{n,j}x_{t,j}\right)\mathcal{N}\left(\boldsymbol{x}_{t,-j}|\boldsymbol{m}_{-j}^{\backslash 1,t,n}, \boldsymbol{V}_{-j}^{\backslash 1,t,n}\right)\mathrm{d}\boldsymbol{x}_{t,-j}$$
$$\mathcal{N}\left(x_{t,j}|\boldsymbol{m}_j^{\backslash 1,t,n}, \boldsymbol{V}_{jj}^{\backslash 1,t,n}\right)\mathrm{d}x_{t,j} \tag{115}$$

Performing a change of variable, $z = \tilde{A}_{n,-j}\boldsymbol{x}_{t,-j}$, reduces the inner integral to a one-dimensional integral and thus, the resulting two nested one-dimensional integrals can be solved using standard results for Gaussian integrals Rasmussen and Williams (2006). The resulting moment becomes:

$$\mathbb{E}\left[x_{t,j}\right] = m_j^{\backslash 1,t,n} + \alpha_{1,t,n}\tilde{a}_{n,j}V_{jj}^{\backslash 1,t,n}, \tag{116}$$

where we have defined $\alpha_{1,t,n} = \frac{\mathcal{N}(z)}{\sqrt{1+b_{1,t,n}}\phi(z)}$. Therefore,

$$\mathbb{E}\left[\boldsymbol{x}_t\right] = \boldsymbol{m}^{\backslash 1,t,n} + \alpha_{1,t,n}\cdot\left(\tilde{\boldsymbol{A}}_{n,\cdot}\circ\mathrm{diag}\left(\boldsymbol{V}^{\backslash 1,t,n}\right)\right). \tag{117}$$

Carrying out similar calculations for $\boldsymbol{x}\boldsymbol{x}^T$ yields

$$\mathbb{V}\left[\boldsymbol{x}_t\right] = \mathrm{diag}\left(\boldsymbol{V}^{\backslash 1,t,n}\right)$$
$$- \alpha_{1,t,n}\cdot\frac{\left(\tilde{\boldsymbol{A}}_{n,\cdot}\mathbb{E}\left[\boldsymbol{x}_t\right] + \alpha_{1,t,n}\right)}{1+b_{1,t,n}}\left(\tilde{\boldsymbol{A}}_{n,\cdot}\circ\mathrm{diag}\left(\boldsymbol{V}^{\backslash 1,t,n}\right)\right)\circ\left(\tilde{\boldsymbol{A}}_{n,\cdot}\circ\mathrm{diag}\left(\boldsymbol{V}^{\backslash 1,t,n}\right)\right). \tag{118}$$

Using these moments, we compute the updates for $\tilde{f}_{1,t,n}$ as follows

$$\hat{\boldsymbol{V}}_{1,t,n}^{\mathrm{new}} = \left[\boldsymbol{V}\left[\boldsymbol{x}_t\right]^{-1} - \left(\boldsymbol{V}^{\backslash 1,t,n}\right)^{-1}\right]^{-1}, \tag{119}$$

$$\hat{\boldsymbol{m}}_{1,t,n}^{\mathrm{new}} = \hat{\boldsymbol{V}}_{1,t,n}^{\mathrm{new}}\left[\boldsymbol{V}\left[\boldsymbol{x}_t\right]^{-1}\mathbb{E}\left[\boldsymbol{x}_t\right] - \left(\boldsymbol{V}^{\backslash 1,t,n}\right)^{-1}\boldsymbol{m}^{\backslash 1,t,n}\right]. \tag{120}$$

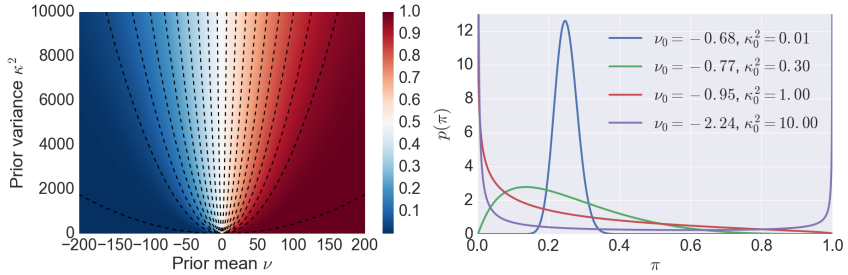
Appendix D. On the Prior Mean and Variance of Γ

The purpose of this appendix is to elaborate on the interplay between the prior mean and the prior variance of $\boldsymbol{\Gamma}$. For this analysis we will assume that the $\boldsymbol{\Gamma}$ has constant mean $\boldsymbol{\mu}_0 = \nu_0\mathbf{1}$ for $\nu_0 \in \mathbb{R}$, and covariance $\boldsymbol{\Sigma}_0 = \kappa_0^2\boldsymbol{R}_0$, where $\mathbf{1} \in \mathbb{R}^D$ is a column vector of ones and $\boldsymbol{R}_0 \in \mathbb{R}^{D\times D}$ is a correlation matrix. Recall from eq. (8) that the marginal prior probability of $z_i = 1$ is given by

$$\hat{p} = p(z_i = 1) = \int p(z_i = 1|\gamma_i)p(\gamma_i)\mathrm{d}\gamma_i = \int \phi(\gamma_i)\mathcal{N}\left(\gamma_i|\mu_i, \Sigma_{0,ii}\right)\mathrm{d}\gamma_i = \phi\left(\frac{\nu_0}{\sqrt{1+\kappa_0^2}}\right). \tag{121}$$

It is seen from the above expression that the marginal expected sparsity level is controlled by ν_0 and κ_0^2. Figure 15(a) shows the surface of $p(z_i = 1)$ as a function of ν_0 and κ_0^2, where the black dashed isocontours confirm that the same level of marginal expected sparsity can be obtained for any combination of (ν_0, κ_0^2) that satisfies the relationship in eq. (121) for some $\hat{p} \in (0, 1)$. Also, note that the prior probability $\hat{p}$ is by definition equal to the expectation of $\phi(\gamma_i)$, that is $\hat{p} = \mathbb{E}_{p(\gamma_i)}\left[\phi(\gamma_i)\right]$. However, as ϕ is a monotonic function, we can derive the full distribution of $\pi = \phi(\gamma)$ through a change of variable as follows

$$p(\pi) = p_\gamma(\phi^{-1}(\pi))\left|\frac{\mathrm{d}\phi^{-1}(\pi)}{\mathrm{d}\pi}\right| = \mathcal{N}\left(\phi^{-1}(\pi)|\nu_0, \kappa_0^2\right)\left|\frac{\mathrm{d}\phi^{-1}(\pi)}{\mathrm{d}\pi}\right| = \frac{\mathcal{N}\left(\phi^{-1}(\pi)|\nu_0, \kappa_0^2\right)}{\mathcal{N}\left(\phi^{-1}(\pi)|0, 1\right)}. \tag{122}$$

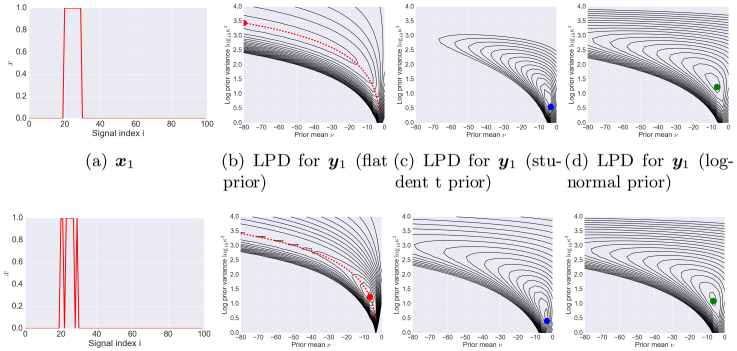


(a) Expected sparsity vs. prior mean and variance

(b) Distribution of $\phi(\gamma_i)$ for $p(z_i = 1) = \frac{1}{4}$

Figure 15: Properties of the prior distribution. (a) Marginal prior probability $p(z_i = 1)$ as a function of (ν_0, κ_0^2). The black dashed lines are isocontours. (b) Distribution of $\pi = \phi(\gamma_i)$ for 4 different pairs of (ν_0, κ_0^2), but for fixed value of $p(z_i = 1)$.

Figure 15(b) shows a plot of the density of π for 4 pairs of (ν_0, κ_0^2) that all satisfy $\hat{p} = \mathbb{E}[\pi] = \frac{1}{4}$. Thus, increasing κ_0^2 while keeping $\mathbb{E}[\pi]$ fixed pushes the mass of $p(\pi)$ to the boundary values. Informally, the distribution of $p(\pi)$ will approach a mixture of two Dirac distributions at 0 and 1 with weights $1 - \mathbb{E}[\pi]$ and $\mathbb{E}[\pi]$, respectively, for very large values of κ_0^2 relative to $\nu_0 \neq 0$. In Section 6, we discussed maximum likelihood among other methods for learning the hyperparameters of the structured spike-and-slab model. However, maximum likelihood learning of ν and κ can in some instances give rise to the similar problems as encountered in maximum likelihood learning of logistic regression models on data sets, that are completely separated in one or more dimensions (Gelman et al., 2008). The following small example illustrates the problem. Consider an instance of $\boldsymbol{y}_1 = \boldsymbol{A}\boldsymbol{x}_1 + \boldsymbol{\epsilon}$, where $\boldsymbol{x}_1$ is the signal shown in Figure 16(a) and where the signal to noise ratio is such that the true support of the signal can be recovered exactly. The dimensions of the forward model is $\boldsymbol{A} \in \mathbb{R}^{50\times 100}$. Let $\mathbf{R}$ be the squared exponential kernel with lengthscale fixed to 8. Figure 16(c) shows the surface of the marginal likelihood approximation as a function of ν_0 and κ_0^2 while the remaining hyperparameters are kept fixed. The red dot indicates the maximum likelihood solution constrained to the domain shown in the figure. The red dashed line shows a plot of the implicit function $\hat{p}_{ML} = p(z_i = 1) = \phi\left(\frac{\nu_0}{\sqrt{1+\kappa_0^2}}\right)$ that intersects the maximum likelihood solution. It is clear that the likelihood surface has a ridge along the curve satisfying $\hat{p}_{ML} = \phi\left(\frac{\nu_0}{\sqrt{1+\kappa_0^2}}\right)$ and that the likelihood is increasing along that ridge as the magnitude of ν_0 and κ_0^2 increase. Thus, the maximum likelihood solutions pushes to magnitude of ν_0 and κ_0^2 to larger and larger values while keeping the sparsity level $\hat{p}_{ML}$ fixed and therefore, gradient-based optimization of the maximum likelihood w.r.t. (ν_0, κ_0^2) will never converge. However, this problem only occurs when the support is separated as in Figure 16(a). Figure 16(f) shows the marginal likelihood approximation surface for $\boldsymbol{y}_2 = \boldsymbol{A}\boldsymbol{x}_2 + \boldsymbol{\epsilon}$, where $\boldsymbol{x}_2$ in Figure 16(b). It is now seen that the maximum likelihood solution is well-defined within the



(a) $\boldsymbol{x}_1$

(b) LPD for $\boldsymbol{y}_1$ (flat prior)

(c) LPD for $\boldsymbol{y}_1$ (student t prior)

(d) LPD for $\boldsymbol{y}_1$ (log-normal prior)

(e) $\boldsymbol{x}_2$

(f) LPD for $\boldsymbol{y}_2$ (flat prior)

(g) LPD for $\boldsymbol{y}_2$ (student t prior)

(h) LPD for $\boldsymbol{y}_2$ (log-normal prior)

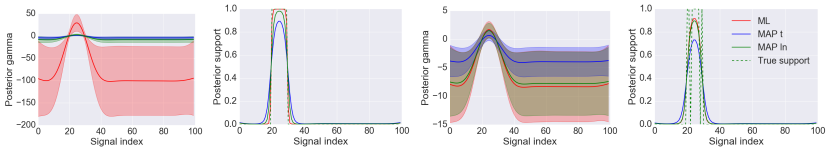
Figure 16: (a) Signal, where the support is contiguous. (b), (c), (d): Log posterior density for $\boldsymbol{y}_1 = \boldsymbol{A}\boldsymbol{x}_1 + \boldsymbol{\epsilon}$ with a flat prior, half student t prior (df = 4) and a log normal prior (mean 6, std. dev 3) for κ_0, respectively. (e) Signal, where the support is not contiguous.(f), (g), (h): Log posterior density (LPD) for $\boldsymbol{y}_2 = \boldsymbol{A}\boldsymbol{x}_2 + \boldsymbol{\epsilon}$ with a flat prior, half student t prior (df = 4) and a log normal prior (mean 6, std. dev 3) for κ_0, respectively. The red dashed line shows a plot of the implicit function $\hat{p}_{ML} = p(z_i = 1) = \phi\left(\nu_0 (1 + \kappa_0^2)^{-\frac{1}{2}}\right)$ that intersects the maximum likelihood solution.

interior of $\mathbb{R}^2$. The problem is easily fixed by imposing a weakly informative prior on κ_0 to ensure that the solution is always well-defined. To illustrate this, we re-run this experiment shown in Figure 16(c) with two different priors on κ_0. Figures 16(d)-(e) show the results for a standardized half student t prior with 4 degrees of freedom and a log-normal prior with mean 6 and standard deviation 3, respectively. Figures 16(g)-(h) show the same plots for the signal $\boldsymbol{x}_2$. Figure 17 shows the resulting posterior distribution for both signals with and without priors distributions.

References

M. R. Andersen, O. Winther, and L. K. Hansen. Bayesian inference for structured spike and slab priors. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1745–1753. Curran Associates, Inc., 2014.

Michael R. Andersen, Ole Winther, and Lars Kai Hansen. Spatio-temporal spike and slab priors for MMV problems. *arXiv preprint arXiv:1508.04556*, 2015.



(a) Posterior of $\mathbf{\Gamma}$ for $\boldsymbol{y}_1$ (b) Posterior mean support for $\boldsymbol{y}_1$ (c) Posterior of $\mathbf{\Gamma}$ for $\boldsymbol{y}_2$ (d) Posterior mean support for $\boldsymbol{y}_2$

Figure 17: Posterior distributions for $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ for hyperparameter values indicated by the colors dots in Figure 16.

J. Ashburner, G. Barnes, C. Chen, J. Daunizeau, G. Flandin, K. Friston, D. Gitelman, S. Kiebel, J. Kilner, V. Litvak, Rosalyn Moran, W. Penny, K. Stephan, D. Gitelman, R. Henson, C. Hutton, V. Glauche, J. Mattout, and C. Phillips. *SPM8 manual*, July 2010. URL `http://www.fil.ion.ucl.ac.uk/spm/doc/manual.pdf`.

S. Baillet, J. C. Mosher, J. C. Mosher, R. M. Leahy, and R. M. Leahy. Electromagnetic brain mapping. *IEEE Signal Processing Magazine, IEEE Signal Process Mag*, 18(6):14–30, 2001. ISSN 10535888. doi: 10.1109/79.962275.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. ISBN 0387310738, 9780387310732.

E. Brochu, V. M. Cora, and N de Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *CoRR*, abs/1012.2599, 2010. URL `http://arxiv.org/abs/1012.2599`.

E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.

P. Carbonetto and M. Stephens. Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Anal.*, 7(1):73–108, March 2012.

C. M. Carvalho, N. G. G. Polson, and J. G. Scott. Handling sparsity via the horseshoe. In David A. Van Dyk and Max Welling, editors, *AISTATS*, volume 5 of *JMLR Proceedings*, pages 73–80. JMLR.org, 2009.

V. Cevher, M. F. Duarte, C. Hegde, and R. Baraniuk. Sparse signal recovery using Markov random fields. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 257–264. Curran Associates, Inc., 2009.

S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *Signal Processing, IEEE Transactions on*, 53(7):2477–2488, 2005.

D. L. Donoho. Compressed sensing. *IEEE Trans. Inf. Theor.*, 52(4):1289–1306, April 2006. ISSN 0018-9448. doi: 10.1109/TIT.2006.871582. URL `http://dx.doi.org/10.1109/TIT.2006.871582`.

D. L. Donoho and J. Tanner. Precise undersampling theorems. *Proceedings of the IEEE*, 98 (6):913–924, 2010.

D. L. Donoho, A. Maleki, and A. Montanari. The noise-sensitivity phase transition in compressed sensing. *IEEE Transactions on Information Theory*, 57(10):6920–6941, 2011.

B. E. Engelhardt and R. P. Adams. Bayesian structured sparsity from Gaussian fields. 8 July 2014.

A. Gelman, A. Jakulin, M. G. Pittau, and Y. Su. A weakly informative default prior distribution for logistic and other regression models. *Ann. Appl. Stat.*, 2(4):1360–1383, December 2008.

M. V. Gerven, B. Cseke, R Oostenveld, and T. Heskes. Bayesian source localization with the multivariate Laplace prior. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1901–1909. Curran Associates, Inc., 2009.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

R. N. Henson, Y. Goshen-Gottstein, T. Ganel, L. J. Otten, A. Quayle, and M. D. Rugg. Electrophysiological And Haemodynamic Correlates Of Face Perception, Recognition And Priming. *Cerebral cortex (New York, N.Y. : 1991)*, 13(7):793–805, July 2003. ISSN 1047-3211. doi: 10.1093/cercor/13.7.793. URL `http://dx.doi.org/10.1093/cercor/13.7.793`.

R. N. A. Henson, E. Mouchlianitis, and K. J. Friston. MEG and EEG data fusion: Simultaneous localisation of face-evoked responses. *NeuroImage*, 47(2):581–589, 2009.

D. Hernández-Lobato and J. M. Hernández-Lobato. Learning feature selection dependencies in multi-task learning. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 746–754. Curran Associates, Inc., 2013.

D. Hernandez-Lobato, J. M. Hernandez-Lobato, and A. Suarez. Expectation propagation for microarray data classification. *Pattern recognition letters*, 31(12):1618–1626, 2010. ISSN 01678655, 18727344. doi: 10.1016/j.patrec.2010.05.007.

D. Hernandez-Lobato, J. M. Hernandez-Lobato, and A. Suarez. Network-based sparse Bayesian classification. *Pattern recognition*, 44(4):886–900, 2011. ISSN 00313203, 18735142. doi: 10.1016/j.patcog.2010.10.016.

D. Hernández-Lobato, J. M. Hernández-Lobato, and P. Dupont. Generalized spike-and-slab priors for Bayesian group feature selection using expectation propagation. *Journal of Machine Learning Research*, 14:1891–1945, 2013.

J. M. Hernández-Lobato, D. Hernández-Lobato, and A. Suárez. Expectation propagation in linear regression models with spike-and-slab priors. *Machine Learning*, 99(3): 437–487, 2015. doi: 10.1007/s10994-014-5475-7. URL `http://dx.doi.org/10.1007/s10994-014-5475-7`.

J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 417–424, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553429. URL `http://doi.acm.org/10.1145/1553374.1553429`.

R. J. Itier and M. J. Taylor. N170 or N1? spatiotemporal differences between object and face processing using ERPs. *Cereb. Cortex*, 14(2):132–142, February 2004.

L. Jacob, G. Obozinski, and J. Vert. Group Lasso with overlap and graph Lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 433–440, New York, NY, USA, 2009a. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553431. URL `http://doi.acm.org/10.1145/1553374.1553431`.

L. Jacob, G. Obozinski, and J. Vert. Group Lasso with overlap and graph Lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 433–440, New York, NY, USA, 2009b. ACM. ISBN 978-1-60558-516-1. doi: 10.1145/1553374.1553431. URL `http://doi.acm.org/10.1145/1553374.1553431`.

R. Jenatton, J. Mairal, F. R. Bach, and G. R. Obozinski. Proximal methods for sparse hierarchical dictionary learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 487–494, 2010.

P. Jylänki, J. Vanhatalo, and A. Vehtari. Robust Gaussian process regression with a Student-t likelihood. *Journal of Machine Learning Research*, 12:3227–3257, 2011.

P. Jylänki, A. Nummenmaa, and A. Vehtari. Expectation propagation for neural networks with sparsity-promoting priors. *Journal of Machine Learning Research*, 15:1849–1901, 2014. URL `http://jmlr.org/papers/v15/jylanki14a.html`.

Y. LeCun, C. Cortes, and C. J. C. Burges. The MNIST database of handwritten digits, 1998.

Y. Li, J. M. Hernández-Lobato, and R. E. Turner. Stochastic expectation propagation. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2323–2331. Curran Associates, Inc., 2015. URL `http://papers.nips.cc/paper/5760-stochastic-expectation-propagation.pdf`.

J. Mairal, R. Jenatton, F. R. Bach, and G. R. Obozinski. Network flow algorithms for structured sparsity. In J D Lafferty, C K I Williams, J Shawe-Taylor, R S Zemel, and A Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1558–1566. Curran Associates, Inc., 2010.

X Meng, S Wu, L Kuang, and J Lu. An expectation propagation perspective on approximate message passing. *IEEE Signal Process. Lett.*, 22(8):1194–1197, August 2015.

T. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 362–369, San Francisco, CA, 2001. Morgan Kaufmann.

T. Minka. Divergence measures and message passing. Technical report, 2005.

T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear-regression. *Journal of the american statistical association*, 83(404):1023–1032, 1988. ISSN 01621459, 1537274x.

F. S. Nathoo, A. Babul, A. Moiseev, N. Virji-Babul, and M. F. Beg. A variational Bayes spatiotemporal model for electromagnetic brain mapping. *Biometrics*, 70(1):132–143, 2014. ISSN 1541-0420. doi: 10.1111/biom.12126. URL `http://dx.doi.org/10.1111/biom.12126`.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag, 1996.

D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Commun. ACM*, 53(12):93–100, December 2010. ISSN 0001-0782. doi: 10.1145/1859204.1859229. URL `http://doi.acm.org/10.1145/1859204.1859229`.

G. Obozinski, B. Taskar, and M. Jordan. Multi-task feature selection. *Statistics Department, UC Berkeley,*, 2006.

M. Opper and O. Winther. Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

T. Park and G. Casella. The Bayesian Lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

T. Peltola, P. Jylänki, and A. Vehtari. Expectation propagation for likelihoods depending on an inner product of two multivariate random variables. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 769–777, 2014.

W. D. Penny, N. J. Trujillo-Barreto, and K. J. Friston. Bayesian fMRI time series analysis with spatial priors. *NeuroImage*, 24(2):350–362, January 2005. ISSN 10538119. doi: 10.1016/j.neuroimage.2004.08.034. URL `http://dx.doi.org/10.1016/j.neuroimage.2004.08.034`.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN 0262256835, 1423769902, 9780262256834, 9781423769903.

J. Riihimäki, A. Vehtari, et al. Laplace approximation for logistic Gaussian process density estimation and regression. *Bayesian Analysis*, 9(2):425–448, 2014.

C. J. V. Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979. ISBN 0408709294.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, 2009. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2008.00700.x. URL `http://dx.doi.org/10.1111/j.1467-9868.2008.00700.x`.

M. Seeger. Expectation propagation for exponential families. Technical report, 2005.

B Shahriari, K Swersky, Z Wang, R P Adams, and N de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proc. IEEE*, 104(1):148–175, January 2016.

J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In F Pereira, C J C Burges, L Bottou, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2951–2959. Curran Associates, Inc., 2012.

R. Sundeep. Generalized approximate message passing for estimation with random linear mixing. *CoRR*, abs/1010.5141, 2010. URL `http://arxiv.org/abs/1010.5141`.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Series B Stat. Methodol.*, 67(1):91–108, 1 February 2005.

M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, September 2001. ISSN 1532-4435. doi: 10.1162/15324430152748236. URL `http://dx.doi.org/10.1162/15324430152748236`.

M. K. Titsias and M. Lazaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011, Nips 2011, Adv. Neural Inf. Process. Syst.: Annu. Conf. Neural Inf. Process. Syst., Nips*, 2011.

J. Vanhatalo, V. Pietiläinen, and A. Vehtari. Approximate inference for disease mapping with sparse Gaussian processes. *Stat. Med.*, 29(15):1580–1607, 10 July 2010.

G. Varoquaux, P. R. Raamana, D. A. Engemann, Y. Hoyos-Idrobo, A.and Schwartz, and B. Thirion. Assessing and tuning brain decoders: Cross-validation, caveats, and guidelines. *Neuroimage*, 145(Pt B):166–179, 15 January 2017.

J. P. P Vila and P. Schniter. Expectation-maximization Gaussian-mixture approximate message passing. *Signal Processing, IEEE Transactions on*, 61(19):4658–4672, 2013.

A. Wu, M. Park, O. O. Koyejo, and J. W. Pillow. Sparse Bayesian structure learning with dependent relevance determination priors. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1628–1636. Curran Associates, Inc., 2014a.

A. Wu, M. Park, O. O. Koyejo, and J. W. Pillow. Sparse Bayesian structure learning with dependent relevance determination priors. In *Advances in Neural Information Processing Systems*, pages 1628–1636, 2014b.

L. Yu, H. Sun, J. P. Barbot, and G. Zheng. Bayesian compressive sensing for cluster structured sparse signals. *Signal Processing*, 92(1):259 – 269, 2012. ISSN 0165-1684. doi: http://dx.doi.org/10.1016/j.sigpro.2011.07.015.

Z. Zhang and B. Rao. Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning. *IEEE Journal of Selected Topics in Signal Processing*, 5 (5):912–926, 2011.

J. Ziniel and P. Schniter. Dynamic compressive sensing of time-varying signals via approximate message passing. *IEEE Transactions on signal processing*, 2013a.

J. Ziniel and P. Schniter. Efficient high-dimensional inference in the multiple measurement vector problem. *IEEE Transactions on Signal Processing*, 61(2):340–354, 2013b.

Journal of Machine Learning Research 18 (2017) 1-30 Submitted 4/15; Revised 7/16; Published 11/17

Adaptive Randomized Dimension Reduction on Massive Data

Gregory Darnell GDARNELL@PRINCETON.EDU
Lewis-Sigler Institute
Princeton University
Princeton, NJ 08544, USA

Stoyan Georgiev SGEORG80@GMAIL.COM
Google
Palo Alto, CA 94043, USA

Sayan Mukherjee SAYAN@STAT.DUKE.EDU
Departments of Statistical Science
Mathematics, and Computer Science
Duke University
Durham, NC 27708, USA

Barbara E Engelhardt BEE@PRINCETON.EDU
Department of Computer Science
Center for Statistics and Machine Learning
Princeton University
Princeton, NJ 08540, USA

Editor: Robert McCulloch

Abstract

The scalability of statistical estimators is of increasing importance in modern applications. One approach to implementing scalable algorithms is to compress data into a low dimensional latent space using dimension reduction methods. In this paper, we develop an approach for dimension reduction that exploits the assumption of low rank structure in high dimensional data to gain both computational and statistical advantages. We adapt recent randomized low-rank approximation algorithms to provide an efficient solution to principal component analysis (PCA), and we use this efficient solver to improve estimation in large-scale linear mixed models (LMM) for association mapping in statistical genomics. A key observation in this paper is that randomization serves a dual role, improving both computational and statistical performance by implicitly regularizing the covariance matrix estimate of the random effect in an LMM. These statistical and computational advantages are highlighted in our experiments on simulated data and large-scale genomic studies.

Keywords: dimension reduction, generalized eigendecompositon, low-rank, genomics, linear mixed models, supervised, random projections, randomized algorithms, Krylov subspace methods

1. Introduction

In the current era of information, large amounts of complex high dimensional data are routinely generated across science and engineering disciplines. One perspective is that the signal in high dimensional data is often concentrated in low dimensional structure, and estimating and exploring this latent structure is of fundamental importance in a variety of applications. As the size of the data sets increases, the problem of statistical inference and computational feasibility become inextricably linked. Dimension reduction is a natural approach to summarizing massive data and has historically played a central role in data analysis, visualization, and predictive modeling. Dimension reduction has had a significant impact on both statistical inference (Adcock, 1878; Edegworth, 1884; Fisher, 1922; Hotelling, 1933; Young, 1941), and on numerical analysis research and applications (Golub, 1969; Golub and Van Loan, 1996; Gu and Eisenstat, 1996; Golub et al., 2000); for a recent review see Mahoney (2011). Historically, statisticians have focused on the study of theoretical properties of estimators often in the context of asymptotically large number of samples. Numerical analysts and computational mathematicians, on the other hand, have been instrumental in the development of useful and tractable algorithms with provable stability and convergence guarantees. Naturally, many of these algorithms have been successfully applied to compute estimators grounded on solid statistical foundations. A classic example of this interplay is principal component analysis (PCA) (Hotelling, 1933). In PCA, an objective function is defined based on statistical considerations about the sample variance, which can then be efficiently computed using a variety of singular value decomposition (SVD) algorithms developed by the numerical analysis community.

In this paper, we consider the problem of dimension reduction, focusing on the integration of i) statistical considerations of *estimation accuracy* and out-of-sample prediction error of matrices with latent low-rank, and ii) computational considerations of run time and *numerical accuracy*. The methodology that we develop builds on a classical approach to modeling large data, which first compresses the data, minimizing the loss of relevant information, and then applies statistical estimators appropriate for small-scale problems. In particular, we focus on dimension reduction via generalized eigendecomposition as the means for data compression, and on out-of-sample residual error as the measure of information loss. The scope of this work includes applications to a large number of dimension reduction methods, which can be implemented as solutions to truncated generalized eigendecomposition problems (Hotelling, 1933; Fisher, 1936; Li, 1991; Wu et al., 2010). In this paper our first focus is on the increasing need to compute an SVD of massive data using randomized algorithms developed in the numerical analysis community (Drineas et al., 2006; Sarlos, 2006; Liberty et al., 2007; Boutsidis et al., 2009; Rokhlin et al., 2009; Halko et al., 2011) to simultaneously reduce the dimension and regularize, or control the impact of independent random noise.

The second focus in this paper is to provide efficient solvers for the linear mixed models that arise in statistical and quantitative genomics. In high-throughput genomics experiments, a vast amount of sequencing data is collected—on the order of tens of millions of genetic variants. The goal of genome-wide association studies (GWAS) is to test for a statistical association at each genetic variant (polymorphic position) to a response of interest (e.g., gene expression levels or disease status) in a sample cohort. However, as the dimension of these genomic data and sample sizes continue to increase, there is an urgent need to improve the statistical and computational performance of standard tests.

It is typical to collect several thousand individuals for one study. These individuals may come from several genetically heterogeneous populations. It has been recognized since 2001 (Pritchard and Donnelly, 2001) that the ancestry makeup of the individuals in the study has great potential to influence study results—in particular, spurious associations arise when genetic variants with differential frequencies may appear to be associated with the biased response variable via latent population structure.

The earliest methods (e.g., genomic control) accounted for population structure by using covariate estimates to correct for these confounding signals. More recently, linear mixed models have been used successfully to correct spurious results in the presence of population structure. LMMs

©2017 Gregory Darnell, Stoyan Georgiev, Sayan Mukherjee, Barbara E Engelhardt.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/15-143.html.

have been shown to improve power in association studies while reducing false positives (Yang et al., 2014). However, mixed models incur a high computational cost when performing association studies because of the computational burden of computing and inverting the covariance matrix for the random effect controlling for population structure. Significant work has gone into mitigating such costs using spectral decompositions for efficient covariance estimation (Kang et al., 2008, 2010; Yang et al., 2011; Zhou and Stephens, 2012; Listgarten et al., 2012).

In this work we show, using simulations of genomic data with latent population structure and real data from large-scale genomic studies, that our approach, adaptive randomized SVD (ARSVD), is effective in terms of both computational efficiency and numerical accuracy. Under certain settings, we find that the LMM using ARSVD outperforms current state-of-the-art approaches by implicitly performing regularization of the covariance matrix.

There are three key contributions of this paper:

(i) We develop an adaptive algorithm for randomized singular value decomposition (SVD) in which both the number of relevant singular vectors and the number of iterations of the algorithm are inferred from the data based on informative statistical criteria.

(ii) We use our adaptive randomized SVD (ARSVD) algorithm to construct truncated generalized eigendecomposition estimators for PCA and linear mixed models (LMMs) (Listgarten et al., 2012; Zhou and Stephens, 2012).

(iii) We demonstrate on simulated and real data examples that the randomized estimators provide a computationally efficient solution, and, furthermore, often improve statistical accuracy of the predictions. We show that, in an over-parametrized setting, this improvement in accuracy is due to implicit regularization imposed by the randomized approximation.

In Section 2, we describe the adaptive randomized SVD procedure we use for the various dimension reduction methods. In Section 2.5, we provide randomized estimators for linear mixed models used in statistical genetics. In Section 3, we give an explanation for why the randomized estimator for linear (mixed) models imposes regularization. In Section 4, we validate the proposed methodology on simulated and real data and compare our approach with state-of-the-art approaches. In particular, we show results from our approach for estimating low dimensional geographic structure in genomic data and for genetic association mapping applications.

2. Randomized Algorithms for Dimension Reduction

In this section, we develop algorithmic extensions for PCA. We state an algorithm that provides a numerically efficient and statistically robust estimate of the highest variance directions in the data using a randomized algorithm for singular value decomposition (Randomized SVD) (Rokhlin et al., 2009; Halko et al., 2011). In this problem, the objective is linear unsupervised dimension reduction with the low-dimensional subspace estimated via an eigendecomposition. Randomized SVD will serve as the core computational engine for the other estimators we develop in this paper.

2.1 Notation

Given positive integers p and d with $p \gg d$, $\mathbb{R}^{p\times d}$ denotes the class of all matrices of dimension $p \times d$ with real entries. We denote symmetric positive semi-definite matrices as $\mathbb{S}_+^p$. For $B \in \mathbb{R}^{p\times d}$,

span(B) denotes the subspace of $\mathbb{R}^p$ spanned by the columns of B. A *basis matrix* for a subspace $\mathcal{S}$ is any full column rank matrix $B \in \mathbb{R}^{p\times d}$ such that $\mathcal{S} = \text{span}(B)$, where $d = \dim(\mathcal{S})$. We denote the data matrix $X = (x_1, \ldots, x_n)^T \in \mathbb{R}^{n\times p}$ with observations drawn from p-dimensional marginal distribution, $x_i \sim \mathcal{P}_X$. When we consider supervised problems such as regression we denote the response vector as either a quantitative response $Y \in \mathbb{R}^m$ or a categorical response $Y \in \{1, \ldots, C\}$, here C is the number of categories. For the joint setting of response and predictor variables we assume a joint distribution, $(X, Y) \sim \mathcal{P}_{X\times Y}$. We denote the orthonormal left eigenvector basis of the data matrix X as eigen-basis(X).

2.2 Computational Considerations

The main computational tool we use is a randomized algorithm for approximate eigendecompositon, which factorizes a $n \times p$ matrix of rank r in time $\mathcal{O}(npr)$ using randomized methods that take advantage of the intrinsic low-rank of the input matrix, rather than the $\mathcal{O}(np \times \min(n, p))$ time required by deterministic approaches. This is relevant to statistical applications to high dimensional data but reflects a highly constrained process (e.g., from genomic or financial applications), which suggests that the data have low intrinsic dimensionality, i.e., $r \ll n < p$. Further improvements have been made to randomized algorithms for approximate eigendecomposition by noting that a structured random projection (such as the subsampled random Fourier transform) can achieve computational complexity of $O(np\log(r)+(m+p)r^2)$ assuming the input matrix fits in main memory (Halko et al., 2011). Since we use power iterations to decay the eigenspectrum and achieve a numerically accurate result independent of the particular spectral gaps, most of the computational gains from subsampling methods would be lost when applied in our framework. Furthermore, since matrix-matrix multiplies are highly optimized on many computational architectures, parallel implementations can reduce our asymptotic complexity to yield excellent run times in practice (Halko et al., 2011).

An appealing characteristic of our randomized algorithm is the explicit control of the trade-off between estimation accuracy relative to the exact estimates and computational efficiency. Rapid convergence to the exact estimates has been shown both empirically as well as in theory (Rokhlin et al., 2009). From the perspective of theoretical computer science and numerical analysis, the objective of randomized SVD algorithms is, given a matrix X, to efficiently compute an approximate eigendecomposition that is close to the exact eigendecomposition; we call this view the *approximation perspective*.

2.3 Statistical Considerations

A *statistical perspective* will deviate from the approximation perspective in two ways: the data matrix X is not fixed, but a noisy random sample drawn from a population, and the inferential objective is to obtain estimates of population quantities from the sample X, not estimates of the eigendecomposition of X itself. Taking a statistical perspective will drive two central concepts in this paper. The first concept is that there is utility in considering randomized algorithms as statistical models. The second concept is that many formulations of ARSVD implicitly impose regularization constraints.

The acceptable error for the approximation perspective and the statistical perspective differ; typically larger error is tolerated in the statistical perspective. For many statistical estimators, the error between the estimator and the population quantity scales as $\varepsilon = O(\frac{1}{\sqrt{n}})$ where n is the sample

size. This is much coarser than the approximation error sought in numerical analysis, where the desired error between the exact and approximate algorithms scales as ε^2, the squared error in the statistical estimate. This observation highlights that, in the statistical setting, one can use fewer computations than are typically considered in the numerical analysis setting because the accuracy of the finer approximation will be lost to the error due to sampling.

This observation about error will impact the parameters of the randomized estimators that we propose in in this paper. An important parameter in our ARSVD algorithms is the number of power iterations t that the randomized algorithm executes (Section 2.4). Increasing the number of power iterations results in a closer approximation to the exact solution (Rokhlin et al., 2009), but also increases the runtime of the algorithm. The observation that we can afford coarser error rates between the exact and approximate solutions suggests that very few power iterations may be required. We provide empirical evidence (Section 4) that fewer power iterations of the approximate algorithm provide results that are both faster and also more accurate with respect to out-of-sample predictions. This observation suggests that the approximation induced by the randomized algorithm is a form of regularization.

2.4 Adaptive Randomized Low-Rank Approximation

In this section, we provide a brief description of a randomized estimator for the best low-rank matrix approximation, introduced by Rokhlin et al. (2009); Halko et al. (2011), which combines random projections with numerically stable matrix factorization. We consider this numerical framework as implementing a computationally efficient shrinkage estimator of the subspace capturing the largest variance directions in the data. The procedure is well suited for matrices that are low rank or matrices where the signal is low rank. Detailed discussion of the estimation accuracy of Randomized SVD in the absence of noise is provided in Rokhlin et al. (2009).

The idea of random projection was first developed as a proof technique to study the distortion induced by the low dimensional embedding of high-dimensional vectors (Johnson and Lindenstrauss, 1984), with much literature simplifying and sharpening the results (Frankl and Maehara, 1987; Indyk and Motwani, 1998; Achlioptas, 2001; Dasgupta and Gupta, 2003). More recently, the theoretical computer science and the numerical analysis communities discovered that random projections can be used for efficient approximation algorithms for a variety of applications (Drineas et al., 2006; Sarlos, 2006; Liberty et al., 2007; Boutsidis et al., 2009; Rokhlin et al., 2009; Halko et al., 2011). We focus on one such approach proposed by Rokhlin et al. (2009); Halko et al. (2011), which targets the accurate low-rank approximation of a given large data matrix $X \in \mathbb{R}^{n\times p}$. In particular, we extend the randomization methodology to the noisy setting, in which the estimation error is due to both the approximation of the low-rank structure in X and also added noise. A simple working model capturing this scenario is

$$X = X_{d^*} + E, \quad X_{d^*} \in \mathbb{R}^{n\times p}, \quad \text{rank}(X_{d^*}) = d^*,$$

where X_{d^*} captures the low dimensional signal and E is independent additive noise.

2.4.1 ALGORITHM FOR ARSVD

Given an upper bound on the target rank $d_{\max}$ and the number of necessary power iterations $t_{\max}$ ($t_{\max} \in \{5, \dots 10\}$ is sufficient in most cases), the algorithm proceeds in two stages: (1) estimate a

basis for the range of X_{d^*}, (2) project the data onto this basis and apply SVD:

Algorithm: *Adaptive Randomized SVD*(X, $t_{\max}$, $d_{\max}$, Δ)

(1) Find orthonormal basis for the range of X;

  (i) Set the number working directions: $\ell = d_{\max} + \Delta$;

  (ii) Generate random matrix: $\Omega \in \mathbb{R}^{n\times\ell}$ with $\Omega_{ij} \overset{iid}{\sim} \text{N}(0, 1)$;

  (iii) Construct blocks: $F^{(t)} = XX^TF^{(t-1)}$ with $F^{(0)} = \Omega$ for $t \in \{1, \dots, t_{\max}\}$;

  (iv) Select the optimal block $t^* \in \{1, \dots, t_{\max}\}$ and rank estimate $d^* \in \{1, \dots, d_{\max}\}$, using the stability criterion and Bi-Cross-Validation stated in Section 2.4.3;

  (v) Compute a basis for the selected block: $F^{(t^*)} = QR \in \mathbb{R}^{n\times\ell}, Q^TQ = I$;

(2) Project data onto the range basis and compute the SVD;

  (i) Project onto the basis: $B = X^TQ \in \mathbb{R}^{p\times\ell}$;

  (ii) Factorize: $B \overset{\text{svd}}{=} U\Sigma W^T$, where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_\ell)$;

  (iii) Compute the rank d^* approximation: $\widehat{X}_{d^*} = U_{d^*}\Sigma_{d^*}V_{d^*}^T$
  $U_{d^*} = (U_1|\dots|U_{d^*}) \in \mathbb{R}^{n\times d^*}$
  $\Sigma_{d^*} = \text{diag}(\sigma_1, \dots, \sigma_{d^*}) \in \mathbb{R}^{d^*\times d^*}$
  $V_{d^*} = Q \times (W_1|\dots|W_{d^*}) \in \mathbb{R}^{p\times d^*}$;

In stage (1), we set the number of working directions $\ell = d_{\max} + \Delta$ to be the sum of the upper bound on the rank of the data $d_{\max}$, and a small oversampling parameter Δ, which ensures a more stable approximation of the top $d_{\max}$ sample variance directions; the estimator tends to be robust to changes in Δ, so we use $\Delta = 10$ as a suggested default. In step (1.iii), the random projection matrix Ω is applied to powers of XX^T to randomly sample linear combinations of eigenvectors of the data weighted by powers of the eigenvalues:

$$\underbrace{F^{(t)}}_{n\times\ell} = (XX^T)^t\Omega = US^{2t}U^T\Omega = US^{2t}\Omega^*, \quad \text{where } X \overset{\text{svd}}{=} USV^T.$$

The power iterations shrink small eigenvalues and increase large eigenvalues while leaving the eigenvectors unchanged. Observe that each column of $F^{(t)}$ can be thought of as drawn from a multivariate normal, $F_j^{(t)} \sim \mathcal{N}(0, US^{4t}U^T)$. The covariance structure of this matrix is biased towards higher directions of variation as t increases. The fact that the power iterations shrink noise directions shows that power iterations impose a form of regularization. The multivariate normal structure of this shrinkage is related to local shrinkage priors developed in Polson and Scott (2010). In step (iv), we select an optimal block $F^{(t^*)}$ for $t^* \in \{1, \dots, t_{\max}\}$ and estimate an orthonormal basis for the column space. For numerical stability, each block in the intermediate power iterations should be orthgonalized (Halko et al., 2011; Gu, 2015). In previous work (Rokhlin et al., 2009), the authors assumed fixed target rank d^* and approximated X rather than X_{d^*}. They showed that the optimal strategy is to set $t^* = t_{\max}$, which typically achieves excellent d^*-rank approximation accuracy for X, even for relatively small values of $t_{\max}$.

In stage (2), we rotate the orthogonal basis Q computed in stage (1) to the canonical eigenvector basis and scale according to the corresponding eigenvalues. In step (2.i) the data is projected onto the low dimensional orthogonal basis Q. Step (2.ii) computes the exact SVD in the projected space.

In this work, we focus on the noisy case, where $E \neq 0$, and propose to adaptively set both d^* and t^*, aiming to optimize the generalization or out-of-sample performance of the randomized estimator. The estimation strategy for d^* and t^* is described in detail in Section 2.4.3.

2.4.2 COMPUTATIONAL COMPLEXITY

The computational complexity of the randomization step is $\mathcal{O}(np \times d_{\max} \times t_{\max})$ and the factorizations in the lower dimensional space have complexity $\mathcal{O}(np \times d_{\max} + n \times d_{\max}^2)$. With $d_{\max}$ small relative to n and p, the runtime in both steps is dominated by the multiplication by the data matrix; in the case of sparse data, fast multiplication can further reduce the run time. We use a normalized version of the above algorithm that has the same run time complexity but is numerically more stable (Martinsson et al., 2010).

2.4.3 ADAPTIVE METHOD TO ESTIMATE d^* AND t^*

We propose to use ideas of stability under random projections in combination with cross-validation to estimate the intrinsic dimensionality of the reduced subspace d^* and the optimal value of the eigenvalue shrinkage parameter t^*.

2.4.4 ESTIMATION OF t^* USING BI-CROSS-VALIDATION

We propose a procedure for selecting an optimal value for $t \in \{1, \ldots, t_{\max}\}$ by using the Bi-Cross-Validation procedure of Owen and Perry (2009), which was used to estimate the rank or cutoff for SVD. For our procedure, we consider a Bi-Cross-Validation formulation that uses the generalized Gabriel holdout pattern (Gabriel, 2002) to partition the data matrix by partitioning the rows and columns into $r = 2$ and $c = 2$ groups respectively that are non-overlapping as suggested in Owen and Perry (2009). We then compute the following Bi-Cross-Validation error by holding out each of the four blocks and estimating a block using the other three blocks

$$\begin{aligned}\text{BiCV}(t) =& \frac{1}{4}\Big[\|A - BD_t^\dagger C|\|_F^2 + \|B - AC_t^\dagger D\|_F^2 + \|B - AC_t^\dagger D\|_F^2 + \\ & \|C - DB_t^\dagger A\|_F^2 + \|D - CA_t^\dagger B\|_F^2\Big], \quad \text{here } X = \begin{pmatrix} A & B \\ C & D \end{pmatrix}.\end{aligned} \tag{1}$$

In the above equation, $\|\cdot\|_F^2$ is the Frobenius norm, $U_t^\dagger$ is the Moore-Penrose pseudoinverse of U where the SVD of U is computed using Adaptive Randomized SVD$(t, d(t), \delta = 10)$, and $d(t)$ is set using the stability criterion developed in the next section. We optimize over the range $\{1, \ldots, t_{\max}\}$ to estimate t^*

$$\hat{t} \;=\; \underset{t \in \{1,\ldots,t_{\max}\}}{\arg\min}\; \text{BiCV}(t).$$

In our simulation results, the value of t^* is estimated small enough to not incur much computational overhead, yet still yields accurate results (Results Section). The original formulation of Bi-Cross-Validation defined the hold-out error to be the Frobenius norm between the predicted and

true submatrix (Owen and Perry, 2009). While the Frobenius and spectral norms both have upper bounds with respect to approximation accuracy of truncated spectral decompositions, more recent results suggest that the spectral norm may generalize better if the goal is to produce an accurate dimension reduction of massive data such as principal components analysis (Mahoney, 2011; Szlam et al., 2014). We recognize the limitation of the Frobenius norm in this context, and acknowledge that it may be wise for Bi-Cross-Validation error to use the spectral norm.

2.4.5 ESTIMATION OF d^* USING STABILITY CRITERION

Given the number of power iterations t, we describe a procedure to estimate the rank parameter $d^*(t)$ using a stability criterion based on random projections of the data. We start with rough upper-bound estimate $d_{\max}$ for the dimension parameter d^*. We then apply a small number ($B = 5$) of independent Gaussian random projections $\Omega^{(b)} \in \mathbb{R}^{n \times d_{\max}}$, $\Omega_{ij}^{(b)} \overset{iid}{\sim} \text{N}(0, 1)$, for $b \in \{1, \ldots, B\}$. Given the projections, we compute an estimate of the eigenvector basis of the column space onto the projected data. We then denoise the estimate by raising all the eigenvalues to the power t:

$$U_b^{(t)} \equiv (U_{b1}^{(t)}|\ldots|U_{bd}^{(t)}) = \text{SVD}[(XX^T)^t\Omega^{(b)}] \;\text{ for }\; b \in \{1, \ldots, B\}.$$

The k-th principal basis left singular vector estimate ($k \in \{1, \ldots, d\}$) is assigned a *stability score*:

$$\text{stab}(t, k, B) = \frac{1}{N}\sum_{j_1=1}^{B-1}\sum_{j_2=j_1+1}^{B}\left|\text{cor}\left(U_{j_1k}^{(t)}, U_{j_2k}^{(t)}\right)\right|, \quad \text{where } N = \frac{B(B-1)}{2}.$$

Here $U_{rk}^{(t)}$ is the estimate of the kth principal eigenvector of X^TX based on the r-th random projection and cor $\left(U_{j_1k}^{(t)}, U_{j_2k}^{(t)}\right)$ denotes the Spearman rank-sum correlation between $U_{j_1k}^{(t)}$ and $U_{j_2k}^{(t)}$. Eigenvector directions that are not dominated by independent noise are expected to have higher stability scores. When the data has approximately low-rank, we expect a sharp transition in the eigenvector stability between the directions corresponding to signal and to noise. In order to estimate this change point, we apply a non-parametric location shift test (Wilcoxon rank-sum) to each of the $d_{\max} - 2$ stability score partitions of eigenvectors with larger versus smaller eigenvalues. The subset of principal eigenvectors that can be stably estimated from the data for the given value of t is determined by the change point with smallest p-value among all $d_{\max} - 2$ non-parametric tests.

$$\hat{d}_t = \underset{k \in \{2,\ldots,d_{\max}-1\}}{\arg\min}\; \text{p-value}(k, t),$$

where p-value(k,t) is the p-value from the Wilcoxon rank-sum test applied to the $\{\text{stab}(t, i, B)\}_{i=1}^{k-1}$ and $\{\text{stab}(t, i, B)\}_{i=k}^{d_{\max}}$.

2.5 Fast Linear Mixed Models

Multivariate linear mixed models (LMMs) are a workhorse in statistical and quantitative genetics because they allow for the regression of explanatory variables on outcome variables while capturing potentially confounding relatedness between samples (Henderson, 1984; Price et al., 2011; Krote et al., 2012). In the context of the genetic association mapping of complex traits, LMMs are used to control for observed (known covariates) and unobserved (random effects) statistical confounding, particularly the presence of population structure amongst samples.

The linear mixed models in this paper take the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u} + e,$$

where $\mathbf{y}$ is an $n \times 1$ response vector of observed phenotypes, $\mathbf{X}$ is an $n \times q$ matrix of fixed effects that includes the genotypes (SNPs) and other confounding variables, $\boldsymbol{\beta}$ is a $q \times 1$ vector representing coefficients of the fixed effects, $\mathbf{u} \sim N(0, \sigma_g^2 K)$ is the random vector of additive genetic effects with incidence matrix $\mathbf{Z}$, and the vector $e \sim \mathcal{N}(0, \sigma_e^2 I_n)$ is the residual error. The matrix K is the kinship or genetic relatedness matrix and may be computed from genotype data. Parameter σ_g^2 is the proportion of variance in the phenotypes explained by genetic factors. The overall phenotypic variance-covariance matrix, integrating out the random effects, is $\mathbf{V} = \sigma_g^2 \mathbf{Z} K \mathbf{Z}^T + \sigma_e^2 I_n$.

The main goal in genetic association studies is to test every genetic locus for significant association to the phenotype based on the effect size of the coefficient $\boldsymbol{\beta}_j$. For each genetic locus, we apply the following hypothesis test:

$$\text{null } H_0 : \boldsymbol{\beta}_j = 0 \quad \text{alternative } H_1 : \boldsymbol{\beta}_j \neq 0, \qquad \text{for } j = 1, \ldots, p.$$

The standard procedure for inferring SNPs associated with a phenotype while correcting for population structure using an LMM proceeds in the following steps:

(1) *Construct the genetic relatedness matrix (GRM)*: The GRM captures the genetic relationship between individuals in the study to model population structure, family structure, and cryptic relatedness. There are cases where the GRM may be obtained directly if the pedigree of the individuals in the sample is known (see Thompson, 1976). In the more common setting, we are not given pedigree information, but instead we have the p dimensional genotype vector G for each of the n individuals. Given the genotype matrix, we can compute the GRM matrix as $\text{GRM} = GG^T$. In the machine learning literature, the GRM would be defined as the Gram matrix for a linear kernel. The biological interpretation of a GRM computed from genotype data differs slightly from the one specified by a pedigree.

(2) *Estimate variance components*: We first use the restricted maximum likelihood estimation (REML) method to estimate σ_g^2, the proportion of phenotypic variance attributable to additive genetic effects. Given the estimate $\hat{\sigma}_g^2$, we estimate σ_ϵ^2, the proportion of phenotypic variance attributable to environmental factors. There are several efficient algorithms for computing the REML (Johnson and Thompson, 1995; Gilmour et al., 1995; Lin et al., 2013; Matilainen et al., 2013). If one computes the random effects first and then the fixed effects, unbiased estimates of the random effects can be found; this is typically the order of computations in genomics.

(3) *Compute an association statistic at each genotype location*: There are a variety of procedures to test for the significance of a genotype j using coefficients $\widehat{\boldsymbol{\beta}}_j$. One approach is to use an F-statistic to test for whether $(G\boldsymbol{\beta})_j = 0$ for each $j = 1, \ldots, p$ (Kang et al., 2008; Kennedy et al., 1992; Henderson, 1984). Another approach is to use a likelihood ratio test considering the variance components. Denote $\ell_1(\hat{\sigma}_1$ as the likelihood under the alternate hypothesis with $\hat{\sigma}_1 = \hat{\sigma}_g^2$ as the estimate of the additive genetic variance component under the alternate model. Then denote $\ell_0(\hat{\sigma}_0)$ as the likelihood under the null ($\boldsymbol{\beta} = 0$) with $\hat{\sigma}_0 = \hat{\sigma}_g^2$ the estimate of the additive genetic variance component under the null model. The log ratio test statistic $2 \log \frac{\ell_1(\hat{\sigma}_1)}{\ell_0(\hat{\sigma}_0)}$ follows a χ^2 distribution and may be used as a test statistic (Kang et al., 2008, 2010; Zhou and Stephens, 2012).

Reducing the computational complexity of the LMM has been an active area of research driven by the increasing size of association studies. A variety of methods have been proposed to increase computational speed (see Kang et al., 2008, 2010; Lippert et al., 2011; Zhou and Stephens, 2012; Lippert et al., 2013). The software we implemented for the results in this paper as well as our methodology is based on EMMAX (Kang et al., 2010). EMMAX improved on EMMA (Kang et al., 2008), which dramatically reduced the computational cost of a standard LMM by exploiting properties of a spectral decomposition of the genotype matrix. EMMAX improves on EMMA by approximating the variance component for each SNP based on an estimate that is computed only once rather than for each SNP. Although we will speed up the model used in EMMAX, our approach can be applied to other fast LMM solvers.

Our contribution to accelerating parameter estimation is using ARSVD to reduce the computational complexity of estimating the random effect $\mathbf{u}$ associated with the design matrix $\mathbf{Z}$. An SVD of the matrix $\mathbf{Z}$ has complexity $\mathcal{O}(n^3)$. If we instead apply ARSVD to $\mathbf{Z}$, we reduce the computational complexity to $\mathcal{O}(np \times d_{\max} + n \times d_{\max}^2)$. In addition, using ARSVD to decompose the design matrix serves to denoise the GRM by retaining the low-rank structure present in the GRM. This avoids the need to manually or heuristically subset the data to achieve a low-rank representation. We will observe in both simulated and real data examples that this application of the ARSVD leads to both a substantial acceleration of the computational speed and an implicit regularization of the design matrix, which reduces type I errors substantially.

3. Regularization of ARSVD

The idea of adding randomness or noise to algorithms for the purpose of regularization has been repeatedly rediscovered (Bishop, 1995; Simard et al., 1993; Mahoney, 2011; Srivastava et al., 2014). Adding independent and identical noise to input variables was observed and rigorously shown to be identical to Tikhonov regularization (Bishop, 1995). Furthermore, Tikhonov regularization is closely related to early stopping, as both regularization methods act as low pass filters (Yao et al., 2007). On the other hand, early stopping is not subject to saturation (Vito et al., 2005; Smale and Zhou, 2007; Yao et al., 2007). In this section we explain why principal components regression (PCR) using ARSVD is a form of regularization. We are confident more refined analyses as well as sharper statements and bounds can be made; our results are more motivational than a detailed analysis.

We will use the spectral filtering framework. In particular, we will use kernel least squares ridge regression (KRR) to illustrate spectral filtering. We then show that PCR using ARSVD is also a spectral filter, and regularization is imposed by weighting and truncating eigenvalues of a positive semidefinite matrix that the algorithm constructs.

3.1 Kernel Ridge Regression

We consider the regression setting, where the number of variables is much larger than the number of observations, $p \gg n$. Given an $n \times p$ design matrix X, the ordinary least squares solution (OLS) is computed based on the normal equations:

$$\hat{\beta} = (X^T X)^{-1} X^T Y,$$

where Y is an $n \times 1$ vector of responses and $\hat{\beta}$ is the OLS estimate for the regression coefficients. When $p \gg n$, the OLS estimator does not work because $X^T X$ is not invertible. In this high-dimensional $p \gg n$ setting, ridge regression (Hoerl and Kennard, 1970) addresses many of the shortcomings of OLS. The estimation problem in ridge regression is formulated as

$$\hat{\alpha} = (XX^T + n\lambda I)^{-1} Y,$$

where λ is a regularization parameter and the induced regression function is

$$\hat{y} = \sum_{j=1}^{p} \hat{\alpha}_j x_j^T x = \hat{\beta}^T x, \quad \hat{\beta} = \sum_{j=1}^{p} \hat{\alpha}_j x_j.$$

When $\lambda = 0$, we recover the OLS estimator, and, when $\lambda = \infty$, one obtains the zero solution $\hat{\alpha} = 0$; λ trades off between fitting the observations and shrinking the solution towards zero.

A standard nonlinear extension to ridge regression is kernel ridge regression (Poggio and Girosi, 1990; Williams and Seeger, 2001) where the regression function takes the form

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i),$$

and $k(u, v)$ is a positive (semi) definite function called a *kernel*. One example of a kernel function is the Gaussian kernel, $k(u, v) = \exp(-h^2 \|u - v\|^2)$. The parameters α to be estimated in kernel ridge regression (KRR) are given by the formula

$$\hat{\alpha} = (K + n\lambda I)^{-1} Y,$$

where the kernel matrix K is defined as $K_{ij} = k(x_i, x_j)$.

3.2 KRR as a Spectral Filter

For the purposes of this paper we will consider spectral filtering as a procedure to filter or smooth a signal (vector) by filtering the eigenvalues of a positive (semi) definite matrix. A signal processing perspective of KRR as a filtering operation is as follows: given the response signal Y and matrix K, the filtering procedure is a map $F(K) : \mathbb{R}^n \to \mathbb{R}^n$, where

$$\hat{Y} = FY, \quad F = K(K + n\lambda I)^{-1}. \tag{2}$$

The basic idea behind spectral filtering is that the filter F operates on the spectrum of the positive (semi) definite matrix, K. In the KRR setting, a natural basis for the matrix F is the eigenvectors of K, and we define the orthonormal matrix $V = [v_1 \cdots v_m]$ with $(v_j)_{j=1}^m$ the m eigenvectors of K with nonzero eigenvalues. The filter F can be written as

$$F = (v_1\ v_2\ \cdots\ v_m) \begin{pmatrix} f(\sigma_1) & & & \\ & f(\sigma_2) & & 0 \\ & & \ddots & \\ & 0 & & \ddots \\ & & & f(\sigma_m) \end{pmatrix} \begin{pmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_m^T \end{pmatrix}, \tag{3}$$

where the spectrum of K is filtered by the function $f(\sigma_i) = \frac{\sigma_i}{\sigma_i + \lambda}$. The filter given in equation (3) can be thought of as a low pass filter that is smoothing the signal Y by shrinking higher frequency eigenvectors—those eigenvectors corresponding to small eigenvalues. In the case of the linear kernel, which is the focus of our paper, K is the Gram matrix, $K_{ij} = x_i^T x_j$.

3.3 Randomized Principal Component Regression as a Spectral Filter

The standard formulation of principal components regression (PCR) is specified by the model

$$y_i = \beta^T z_i + \varepsilon_i, \quad \varepsilon_i \overset{iid}{\sim} \mathrm{N}(0, \sigma^2), \tag{4}$$

where $z_i = x_i^T V$ with x_i an observation in $\mathbb{R}^p$ and $V = [v_1 \cdots v_m]$ are the m eigenvectors corresponding to the top eigenvalues of the empirical covariance matrix. The idea of PCR is that projection onto the top principal component reduces variance without much loss in bias. The nonzero eigenvalues of the Gram matrix and the empirical covariance matrix are identical, so one can compute V from the Gram matrix $K_{ij} = x_i^T x_j$.

Again, we consider the case where $p \gg n$ and assume that the ARSVD procedure sets the maximum rank $d_{\max} \ll n \ll p$. In the following, we will formulate principal components regression as a spectral filter analogous to a filtering formulation of KRR in (2). We will make some approximations in this analysis as the form of the filter in the case of ARSVD is not straightforward due to randomization. The randomization and power iterations of ARSVD impact the spectral filter F in two ways: The filter no longer operates on the eigenvalues of the Gram matrix, and the eigenvectors of the spectral filter are not given by the eigenvalues of the Gram matrix.

Our analysis will consist of two observations. The first is that the eigenvectors of the exact gram matrix K and the eigenvectors of the Gram matrix induced by the ARSVD procedure are close approximations. This observation will allow us to use the eigenvectors of the exact Gram matrix in our analysis of the spectral filter. The second observation is based on a series of papers (Gerfo et al., 2008; Rudi et al., 2013, 2015) that illustrated a common regularization framework for some families of truncation-based algorithms including truncated SVD and PCR, early stopping of iterative procedures, and regularization algorithms such as ridge regression. The filter function for PCR is

$$f(\sigma) = \begin{cases} 1 & \sigma \geq \tau \\ 0 & \text{otherwise,} \end{cases}$$

here τ is the eigenvalue cutoff.

In the case of ARSVD, we generate a random matrix Ω with $\Omega_{ij} \overset{iid}{\sim} \mathrm{N}(0, 1)$. The following power iterations are then taken of a random projection onto the Gram matrix

$$G^{(t)} = (XX^T)^t \Omega = U\Sigma^{2t} U^T \Omega = U\Sigma^{2t} \Omega^*, \quad \text{with } X = U\Sigma V^T,$$

where U, Σ, and V correspond to a standard SVD of X. A basis is computed from matrix $G^{(t)}$ via QR decomposition,

$$G^{(t)} = QR \in \mathbb{R}^{n \times \ell}, \quad Q^T Q = I.$$

The data are then projected onto this basis $B = X^T Q \in \mathbb{R}^{p \times \ell}$ and a standard SVD is run on the much lower rank matrix B. If the eigenvectors of XX^T and BB^T are equal, then the following

spectral filter can be specified for the PCR with ARSVD based on the eigenvalues of the Gram matrix of the data

$$\hat{Y} = FY, \quad F = V\,\Lambda_{f(\sigma)}\,V^T, \quad \Lambda_{f(\sigma)} = \text{diag}(f(\sigma_1), ..., f(\sigma_n)), \quad f(\sigma_i) = \frac{\sigma_i^{2t}}{\sigma_i^{2t} + \tau}, \quad (5)$$

where τ is a threshold parameter. The derivation of this spectral filter is based on results in Gerfo et al. (2008) and Rudi et al. (2013). In the limit of infinite power iterations, the spectral filter is simply a hard thresholding algorithm

$$\lim_{t\to\infty} f(\sigma_i) = \begin{cases} 1 & \sigma_i > 1 \\ 0 & \sigma_i > 1. \end{cases}$$

This asymptotic analysis suggests that scaling each eigenvalue $\sigma_i := \sigma_i/\tau$ can be used to threshold at the level $1/\tau$.

We now show that the eigenvectors for XX^T and BB^T are equivalent, at least for eigenvectors corresponding to larger eigenvalues. This allows us to interpret the filter in terms of the eigenvalues of the Gram matrix XX^T. If the matrix Ω is orthogonal, then the eigenvectors corresponding to the top ℓ eigenvalues of XX^T and BB^T would be equivalent, modulo a constant scale term, which we can set without loss of generality. We argue that the matrix Ω is ε-quasiorthogonal. A set of unit norm vectors $\mu_1, ..., \mu_M \in \mathcal{R}^n$ is ε-quasiorthogonal (Kainen and Kůrková, 1993; Hecht-Nielsen and Kůrková, 1992) if their inner products are small $|\mu_i \cdot \mu_j| \leq \varepsilon$. In Indyk and Motwani (1998, Appendix A), it was shown that, for a random matrix with elements drawn exactly as Ω, the columns are ε-quasiorthogonal.

Beyond principal component regression, the top eigenvalues are important to estimate for many machine learning methods, including graph Laplacian objectives. Graph Laplacians form a key component of practically important algorithms including computing the heat kernel of a graph and PageRank (Mahoney and Orecchia, 2010). It has been shown that approximation algorithms such as ARSVD solve an exact optimization problem with an explicit regularization term (Mahoney and Orecchia, 2010; Perry and Mahoney, 2011).

3.4 Randomization and Leverage Scores

The idea of subsampling observations to generate the Gram matrix is at the heart of Nyström methods (Williams and Seeger, 2001; Drineas and Mahoney, 2005). For runtime considerations, a Gram matrix is constructed as the approximation $\tilde{G} = C^T W^{-1} C$, where C is a $p \times c$ matrix where c is a uniform subsample of the n observations and W is an $n \times n$ incidence matrix of which rows and columns are included in the subsample. There is a great deal of work in the machine learning literature arguing why it is that the Nyström method results in faster algorithms, and recent work illustrating why this numerical approximation can be formulated as a regularization method (Rudi et al., 2013, 2015). Another subsampling perspective is based on leverage scores. Given a design matrix X and the corresponding left singular vectors U the leverage score of a sample is $\ell_j = ||U_j||^2$ (Gittens and Mahoney, 2013) and can be thought of as a measure of relevance of the jth sample to the linear regression function. An alternative to uniformly sampling observations is to sample them according to the leverage score, $p_j \propto \ell_j = ||U_j||^2$, so points with higher leverage scores will more likely be sampled. Indeed, it has been shown that leverage scores drive the accuracy of the the Nyström method, and the uniform sampling approach is optimal when the leverage

scores are almost equal; in the case when the leverage scores are variable, importance sampling according to the leverage score has been used (Drineas et al., 2012). It has been shown previously that random projections project into a space where leverage scores are nearly uniform (Drineas et al., 2012; Mahoney, 2011).

4. Results on Real and Simulated Data

We use real and simulated data to highlight the following three major contributions of this paper

1. In the presence of informative *low-rank* structure in the data, randomized algorithms tend to be much faster than exact methods with minimal loss in approximation accuracy.

2. The *rank* and *subspace* containing information in the data can be reliably estimated and used to provide efficient solutions for dimension reduction.

3. The randomized algorithms implicitly impose regularization, which can be adaptively controlled in a computationally efficient manner to produce improved out-of-sample performance.

4.1 Simulated Data

4.1.1 UNSUPERVISED DIMENSION REDUCTION

We begin with unsupervised dimension reduction of data with low-rank structure contaminated with Gaussian noise, and we focus on evaluating the application of *Adaptive Randomized SVD* for PCA (see Section 2.4). In particular, we demonstrate that the proposed method estimates the sample singular values with exponentially decreasing relative error in t. Then we show that achieving similar low-rank approximation accuracy to a state-of-the-art Lanczos method requires the same run time complexity, which scales linearly in both dimensions of the input matrix. This makes our proposed method applicable to large data matrices. Lastly, we demonstrate the ability to adaptively estimate the underlying rank of the data, given a coarse upper bound. In all our simulations, we set the oversampling parameter in ARSVD, $\Delta = 10$.

We note that the oversampling parameter is crucial in scientific computing applications such as ours. In particular, setting this parameter will be application-specific and depend heavily on the structure of the input data, such as sparsity and distributional properties. For worst-case matrices, our algorithm has potential to produce sub-optimal results (Mahoney, 2011). In general, high quality results—both empirical and theoretical—are achieved by setting the oversampling parameter between five and ten (Halko et al., 2011).

4.1.2 SIMULATION MODEL

We first state the simulation model used for most of the results in this subsection. The data matrix $X \in \mathbb{R}^{n\times p}$, is generated as follows: $X = USV^T + E$, where $U^TU = V^TV = I_{d^*}$. The d^* columns of U and V are drawn uniformly at random from the corresponding unit sphere and the singular values S = diag$(s_1, \ldots, s_{d^*})$ are randomly generated starting from a baseline value, which is a fraction of the maximum noise singular value, with exponential increments separating consecutive

entries:

$$\begin{aligned} s_j &= s_{j-1} + \nu_j, \text{ for } j \in \{2, \ldots, d^*\} \\ \nu_j &\overset{\text{iid}}{\sim} \text{Exp}(\lambda), \quad \nu_0 = s_1^{(E)}. \end{aligned}$$

The noise is iid Gaussian: $E_{ij} \overset{\text{iid}}{\sim} N\left(0, \frac{1}{n}\right)$. The gaps between singular values, ν_j, follow an exponential distribution with rate parameter λ to control the signal-to-noise ratio (Table 1). The sample variance has the SVD decomposition $E \overset{\text{svd}}{=} U_E S_E V_E^T$, where $S_E = \text{diag}\left(s_1^{(E)}, \ldots, s_{\min(n,p)}^{(E)}\right)$ are the singular values in decreasing order. While there exist other working models for the noise structure, here we chose to investigate the current model and that of latent population structure because of its relevance to the genetic data that we wish to model (Section 4.1.4). Our simulations and genetic data experiments show that the assumptions we make on this particular noise model generalize well to genetics data.

4.1.3 RESULTS

The first objective is to show that we can accurately estimate singular values with very few power iterations. Our focus is on understanding the effect of the regularization parameter t controlling the singular value shrinkage. Larger values correspond to a stronger weighting on directions with large eigenvalues. In our first simulation we assume the rank d^* is fixed to 50 and the input matrix is $2,000 \times 5,000$. Studying the estimates of the percent relative error of the *singular values* averaged over ten simulated data sets. The relative error given a singular value estimate $\hat{\sigma}$ and singular value σ is $\left(\frac{\sigma - \hat{\sigma}}{\sigma}\right)$. We observed exponential convergence to the sample estimates with increasing t (Table 1). This suggests that we can capture the variation in the data with a few data matrix multiplications. We measure the error in our estimates using a signal-to-noise (S/N) metric

$$\frac{\|S\|_F^2}{\|E\|_F^2}, \quad S \text{ is the signal matrix and } E \text{ is the error or residual matrix.}$$

The signal matrix has a maximum of d^* non-zero singular values, and thus the calculation of the Frobenius norm only includes the top d^* singular values in computing both $\|S\|_F^2$ and $\|E\|_F^2$.

λ	S/N	$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$
2	2.39	2.34 ± 1.23	1.18 ± 0.63	0.72 ± 0.43	0.48 ± 0.32	0.35 ± 0.25
4	0.61	3.32 ± 1.32	1.67 ± 0.47	1.00 ± 0.28	0.68 ± 0.20	0.50 ± 0.16
6	0.15	5.04 ± 1.53	2.97 ± 0.66	1.86 ± 0.41	1.30 ± 0.30	0.97 ± 0.24
8	0.13	6.26 ± 1.87	3.48 ± 0.42	2.14 ± 0.28	1.47 ± 0.21	1.08 ± 0.18

Table 1: **Singular values from ARSVD.** We report the relative error for singular value estimates with ± 1 standard deviation. A linear increase in the regularization parameter t results in a exponential decrease in the error. S/N is the signal to noise ratio and a function of λ.

It is also of interest to characterize the decay in accuracy in estimating singular values using the randomized method as the magnitude of the true singular values decreases. When we consider

the rank-ordered singular values for a fixed matrix as well as the distribution of the singular values computed via various runs of the ARSVD, an interesting observation is that the estimates are biased for small singular values (Figure 1). Data were generated from $n = 1,000$ and $p = 1,000$ with true rank $d^* = 50$. For the smallest singular values, ARSVD tends to underestimate the singular values, and the standard error is larger. This bias can be considered a form of regularization that shrinks directions corresponding to small singular values. We will discuss this property further in Section 4.1.4.

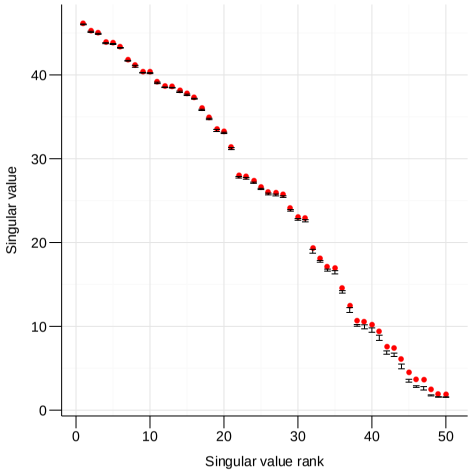


Figure 1: **Singular value accuracy of ARSVD.** Simulation results comparing estimation accuracy of singular values for ARSVD versus SVD on pseudo-random matrices of dimension $n = 1,000$ and $p = 1,000$. The exact singular values are in red and confidence intervals for the singular values computed using ARSVD are in blue.

We can compare the runtime of randomized SVD (RSVD) to two standard spectral decompositions methods. We denote the singular value decomposition of a data matrix X as SVD. We denote as eig the procedure first computing $\widehat{\Sigma} = XX^T$ and then computing the spectral decomposition of $\widehat{\Sigma}$. For this simulation, we generate pseudo-random matrices that are $n \times p$ such that $n = \frac{p}{10}$. For the RSVD procedure we will set the rank parameter to 100. We ran the SVD procedure on matrices with $p = [2000, 40000]$, we did not exceed 40,000 due to computational constraints. We ran the eigenvalue procedure on matrices with $p = [2000, 80000]$. We ran our RSVD procedure on matrices with $p = [2000, 100000]$. Examining the runtime of the three methods in terms of CPU-seconds compared to the size of the matrix, which we index as p, we see on a log scale that RSVD dramatically outperforms the other two methods (Figure 2).

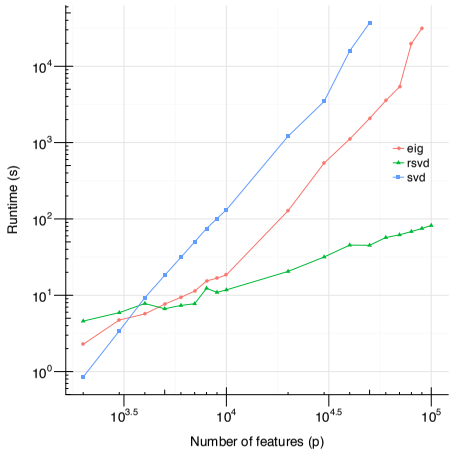


Figure 2: **Comparison of three spectral methods.** We compare RSVD, SVD, and eig. The x-axis indexes matrix size as p and the y-axis is the runtime in seconds. Both axes are on a log scale.

A natural question is whether the randomization offers any advantage over well developed efficient methods, such as Lanczos-Krylov Subspace estimation, which also operate on the data matrix only through matrix multiplies (Saad, 1992; Lehoucq et al., 1998; Stewart, 2001; Baglama and Reichel, 2006). These subspace methods are also iterative in nature, with the runtime complexity typically scaling as $O(qnp)$, where q is small. We compare the runtime ratio of our RSVD with a state-of-the-art low-rank approximation algorithm, the blocked Lanczos method implemented in the CRAN package `irlba` (Baglama and Reichel, 2006). Data were generated from $n = 1000$ and $p = 1000$ with true rank $d^* = 50$, and we varied t in ARSVD. We ran both ARSVD and blocked Lanczos until the Frobenius norm reconstruction error to the original matrix was equal to one degree of precision. We report the ratio of the runtime of ARSVD over blocked Lanczos computed on ten simulated data sets (Table 2). The relative runtime remains approximately constant with simultaneous increase in both data dimensions, which suggests similar order of complexity for both methods when the latent rank of the data (d^*) is supplied as a static parameter to each method. The relative runtime decreases exponentially when using ARSVD to dynamically estimate the latent rank versus using Bi-Cross-Validation to dynamically estimate the rank supplied to block Lanczos.

We now examine our ability to accurately estimate the rank of the matrix using the adaptive stability-based approach outlined in Section 2.4.3. We generated 50 random data sets with $n = 1000, p = 1000, d^* \overset{\text{iid}}{\sim} \text{Uniform}[10, 50]$. We set the initial rank upper bound estimate to be $2 \times d^*$ and used the stability based method (Section 2.4.3) to estimate both optimal t^* and the corresponding d^*. We compared the *true rank* and the corresponding estimates of the regularization parameter t^* for

n + p	6,000	7,500	9,000	10,500	12,000
relative time (static rank)	2.5 ± 0.05	1.84 ± 0.03	1.82 ± 0.03	1.83 ± 0.02	1.84 ± 0.04
n + p	2,000	3,500	5,000	6,500	8,000
relative time (dynamic rank)	3.49 ± 1.10	0.58 ± 0.26	0.46 ± 0.21	0.04 ± 0.01	0.25 ± 0.11

Table 2: **Runtime ratio of ARSVD versus block Lanczos.** We report the sample mean and standard error of ARSVD over block Lanczos based on ten random replicate data sets across dimensionality $n+p$. n is incremented by 500 and p is incremented by $1,000$. In the static rank experiments, the latent rank d^* is supplied as a static parameter to both methods. In the dynamic rank experiments, ARSVD estimates the latent rank using the algorithm previously described, and we use Bi-Cross-Validation to dynamically estimate the latent rank supplied to block Lanczos. ARSVD is run for as many iterations as needed until the sample error is equal to within one degree of precision, thus we do not dynamically estimate the number of power iterations, t^*, in these experiments.

two different signal-to-noise scenarios (Figure 3). In both scenarios, the rank estimates agreed with the true rank values. If the signal-to-noise ratio is low, then our procedure slightly underestimates the rank. We suspect this underestimate is due to the fact that the few smallest variance signal directions tend to be difficult to distinguish from the random noise and hence are less stable under random projections. Our approach tends to select small values for t^*, especially when there is a clear separation between the signal and the noise.

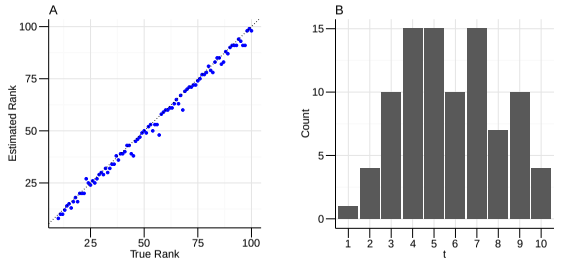


Figure 3: **Rank estimation of ARSVD.** Rank estimation results after performing ARSVD on 50 pseudo-random matrices of dimension $n = 10000$ and $p = 10000$, with true rank $d^* \overset{\text{iid}}{\sim} \text{Uniform}[10, 50]$. Matrices are generated as described in Section 4.1.1. Panel A: True rank (d^*) on the x-axis, and estimated rank ($\hat{d}^*$) on the y-axis. Panel B: estimations of t^* with max t set to ten.

4.1.4 LATENT POPULATION STRUCTURE

We examine how accurately ARSVD can be used to correct latent or cryptic population structure. Specifically, we compared the performance of an LMM using ARSVD versus a standard LMM. We simulated genotype and phenotype data where the genotypes have latent population structure that, once corrected for, there remains no association between genotype and phenotype. In other words, the phenotype is conditionally independent of the genotype given the latent (population) structure. This relation is sometimes called the confounding effect of cryptic structure in genomic data, and motivates the need for LMMs in genome-wide studies. Given that simulations are entirely under the null hypothesis of no association between genotype and phenotype, p-values should follow a uniform distribution if the random effect controls population structure appropriately. We consider any result exceeding an α-threshold a false positive, occurring at rate α.

We use the model stated in Mimno et al. (2014) to simulate admixed genotypes with K ancestral populations. The genotype of an individual is generated by the following hierarchical model

$$\begin{aligned}
\theta_i &\sim \text{Dir}_K(\alpha), \\
\phi_k &\sim \text{Beta}(1,1),\ k = 1, ..., K, \\
(z_{1,ij}, z_{2,ij}) &\sim \Big(\text{Mult}(\theta_i), \text{Mult}(\theta_i)\Big) \text{ for } j = 1, ..., p \\
(x_{1,ij}, x_{2,ij}) &\sim \Big(\text{Bin}(\phi_{z_{1,ij}}), \text{Bin}(\phi_{z_{2,ij}})\Big) \text{ for } j = 1, ..., p.
\end{aligned}$$

The first step samples the the admixture proportions for individual i. The second step samples the allele frequency distribution for populations $k = 1, \ldots, K$. The third step samples the population of origin for both allele copies over all loci, $j = 1, \ldots, p$. The final step samples both copies of the alleles at each locus j. We generated the phenotype using the following relation: $y_i \sim \text{Be}(0.5\theta_k + 0.1(1 - \theta_k))$.

We looked at four simulation settings $n = p = 1000$, $n = p = 5000$, $n = 1000, p = 5000$, and $n = 5000, p = 1000$. Using the p-values for a LMM using ARSVD with the rank parameter d^* of the ARSVD specified, we see that the standard LMM is recovered in the limit of $d^* = p$ (Figure 4). Of these settings, the case where $n = 1000, p = 5000$ is the most similar to the standard genomics case where the number of SNPs p is much larger than the number of observations n. A summary of these simulation results is that the ARSVD method is much faster than the standard LMM and performs similarly with respect to correcting for population structure and controlling false positives. We observed that, for the simulation with $n = 1000, p = 5000$, using ARSVD leads to a substantial reduction of computational complexity with similar performance. In general, we expect the tradeoff between computational efficiency and accuracy to depend on the data. In the case of structured genomic data, we report in these simulations and in Section 4.1.4 that massive computational savings are accompanied by numerical accuracy.

4.2 Association Mapping in Large Genomic Data

We applied our LMM with ARSVD to a large genomic data set to illustrate that we can achieve considerable computational efficiency without loss in accuracy. In particular, we applied our method to the Wellcome Trust Case Control Consortium (WTCCC) data (Consortium, 2007). The data we consider consist of a case-control study of $4,684$ individuals. The cases are individuals with Crohn's disease, and the number of features are 478,765 genetic variants across the 22 autosomal chromosomes in the genome. We compared our ARSVD method with state-of-the-art LMMs designed for

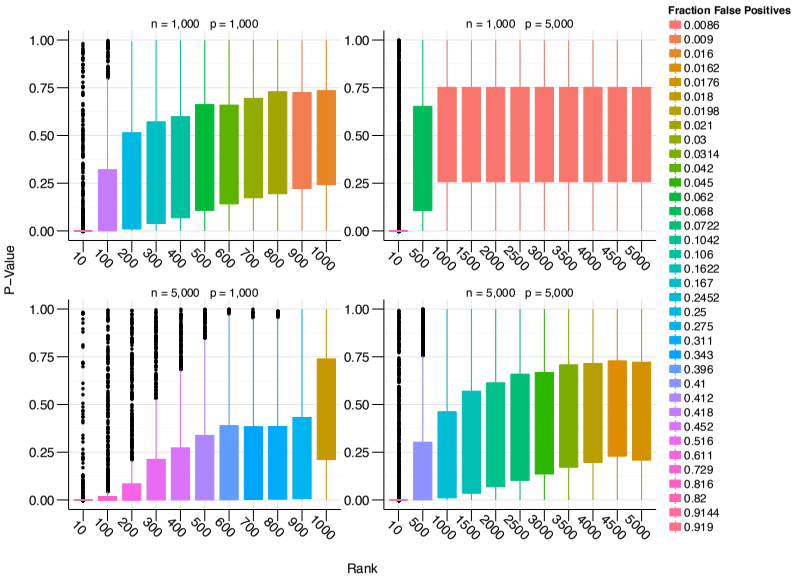


Figure 4: **Controlling for structure.** RSVD with pylmm is applied to four simulation settings with varying n and p. The x-axis is the setting of the ARSVD parameter d^*. The y-axis corresponds to the p-values from the LMM for each of the p features. The color of each box plot represents the fraction of false positive rate of the LMM using ARSVD.

association studies. Our LMM with ARSVD procedure uses EMMAX (pylmm) to solve the LMM. The two methods we compared to are EMMAX with the addition of ARSVD and GEMMA (Zhou et al., 2013) (GEMMA is executed with the `-lmm` option to most closely approximate the analysis that pylmm performs).

ARSVD on the whole genome took 82.2 seconds, while a traditional eigendecomposition of the covariance matrix in pylmm took 88 mins 23.9 seconds. In order to most accurately control for the test statistic computed in a LMM and to achieve maximal statistical power, it is suggested that a covariance matrix is constructed once per (22) chromosomes, performed by holding out the test chromosome and concatenating the remaining chromosomes (Yang et al., 2014). Our method performs the 22 decompositions in a total of 5 mins 4.8 secs, while the traditional decomposition method takes 4 hrs 24 mins.

In the remainder of this subsection, we denote the culling of EMMAX with ARSVD as pylmm. It has been observed (Zhou et al., 2013) that the effect sizes of the coefficients are similar whether one applies linear regression or logistic regression to most case-control genomic studies. Estimates

of β for pylmm and GEMMA are strongly correlated. The distributions of p-values computed by pylmm show enrichment in low p-values. This suggests that the regularization in pylmm may capture additional associations.

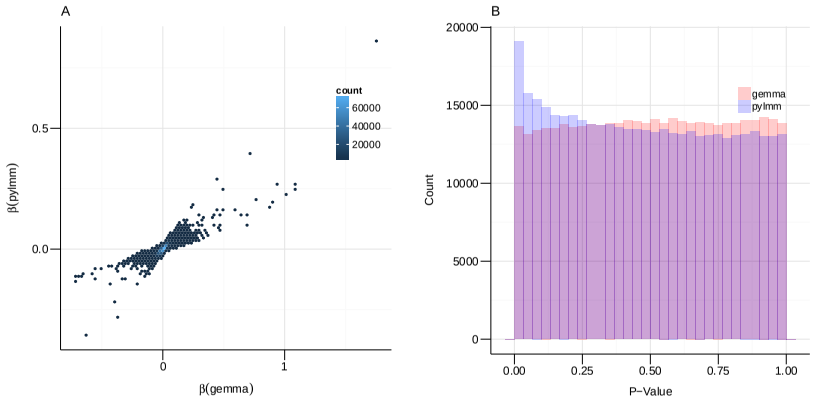


Figure 5: **Comparison of pylmm and GEMMA.** A) A scatter plot of the β-values for GEMMA on the x-axis versus the β-values for pylmm on the y-axis. B) Histogram of p-values for both methods, GEMMA in red and pylmm in blue.

We compared the most significant associations that we identified to results from other analyses of Crohn's disease. One source of associations is from a large-scale meta-analysis of Crohn's disease consisting of 6,333 affected individuals (cases) and 15,056 controls, and the top association signals were followed up on in 15,694 cases, 14,026 controls, and 414 parent-offspring trios (Franke et al., 2010). We denote this list as MA. In Listgarten et al. (2012), the list of associated genetic variants collected includes the MA list as well as the WTCCC list. The major histocompatibility complex (MHC) is a region that has been previously associated to Crohn's disease and autoimmune disease in general. We denote the list of variants in this region as MHC. We report the overlap between the results obtained from our method to the MA list, the union of the WTCCC and MA lists, and the union of the WTCCC, MA, and MHC lists. (Table 3). We select our top hits using two cutoffs: the top 0.5% with respect to negative log p-value, and the associations that pass a local false discovery rate (LFDR) of 5% (Strimmer, 2008).

We identified several potential genetic variants associated with Crohn's that were previously unidentified. In particular, there are a few genetic variants within the 3.6MB region that defines the MHC on chromosome 6 in the human genome (sequencing consortium, 1999) at an LFDR threshold of 10%. Genetic variant rs9269186 ($p \leq 7.03 \times 10^{-7}$) is below the LFDR threshold of 1% and lies within 5 kilobases (KB) of the start of the *HLA-DRB5* gene, putatively acting to regulate transcription of this protein-coding gene that produces a membrane-bound class II molecule. The

HLA-DRB5 protein is an antigen that has an important role in the human immune system, and thus may play a role in an autoimmune disorder such as Crohn's disease.

source	significant variants	top 0.5%	LFDR 5%
MA+WTCC+MHC	151	61	35
MA + WTCCC	93	48	30
MA	81	47	29

Table 3: **Overlap of genetic associations with prior studies.** Columns represent the number of associations (significant variants) in the MA, WTCCC studies and the number of variants in the MHC region; the overlap of our list with the associations in these lists with a cutoff of the top 0.5% of our associations; and overlap at an LFDR threshold of 5%.

5. Discussion

Massively high-dimensional data sets are ubiquitous in modern data analysis settings. In this paper, we provide a scale method for accurate computation using spectral decompositions for data analysis. The main computational tool we use is based on recent randomized algorithms developed by the numerical analysis community. To address the issue of noise, we provide an adaptive procedure to estimate both the rank d^* of the lower dimensional projection and the number of Krylov iterations t^* for the randomized approximate SVD. Using this adaptive estimator of low-rank structure, we implement efficient algorithms for PCA and linear mixed models. An interesting observation both from an empirical and theoretical perspective is that our randomized algorithm implicitly imposes regularization.

In simulated experiments we show high accuracy in recovering the true (latent) rank of matrices with low-rank substructure, without the need for many Krylov iterations. Additionally, we show in simulations that our method performs implicit regularization and improves quantitative properties of results under various types of data structure. Furthermore, our results on large genome-wide association studies show that our approach to using ARSVD in linear mixed models fills a critical need for methods with computational efficiency that do not sacrifice the desirable statistical properties of traditional LMMs.

Some important open questions still remain:

(1) There is need for a more refined theoretical framework to quantify what generalization guarantees the randomization algorithm can provide on out-of-sample data, and the dependence of this bound on the noise and the structure in the data on one hand and on the parameter settings on the other.

(2) A probabilistic interpretation of the algorithm could contribute additional insights (Mahoney, 2011) into the practical utility of the proposed approach under different assumptions. In particular, it would be interesting to relate our work to a Bayesian model with posterior modes that correspond to the subspaces estimated by the randomized approach.

(3) The implicit regularization on latent factors imposed by ARSVD should be further explored with respect to the structure of the noise, and its impact on estimates of random effects in LMMs (Runcie and Mukherjee, 2013).

Acknowledgments

SM would like to acknowledge Lek-Heng Lim, Michael Mahoney, Qiang Wu, and Ankan Saha. SM is pleased to acknowledge support from grants NIH (Systems Biology): 5P50-GM081883, AFOSR: FA9550-10-1-0436, NSF CCF-1049290, and NSF-DMS-1209155. GD and BEE would like to acknowledge Joel Tropp. BEE is pleased to acknowledge support from grants NIH R00 HG006265, NIH R01 MH101822, NIH U01 HG007900, and a Sloan Faculty Fellowship. SG would like to acknowledge Uwe Ohler, Jonathan Pritchard and Ankan Saha. The software for ARSVD is available at `https://github.com/gdarnell/arsvd`.

Appendix A. Generalized Eigendecomposition and Dimension Reduction

The appendix states a variety of dimension reduction methods supervised, unsupervised, and non-linear that can use the ARSVD engine to scale to massive data. The key requirement is a formulation of the truncated generalized eigendecomposition problem that can be implemented by the *Adaptive Randomized SVD* from Section 2.4. The dimension reduction methods we will focus on are sliced inverse regression (SIR) and localized sliced inverse regression (LSIR).

A.1 Problem Formulation

Assume we are given $\Sigma \in \mathbb{S}^p_{++}, \Gamma \in \mathbb{S}^p_+$ that characterize pairwise relationships in the data and let $r \ll \min(n,p)$ be the "intrinsic dimensionality" of the information contained in the data. In the case of supervised dimension reduction methods this corresponds to the dimensionality of the linear subspace to which the joint distribution of (X,Y) assigns non-zero probability mass. Our objective is to find a basis for that subspace. For SIR and LSIR this corresponds to the span of the generalized eigenvectors $\{g_1,\dots,g_r\}$ with largest eigenvalues $\{\lambda_{\max} = \lambda_1 \geq \dots \geq \lambda_r\}$:

$$\Gamma g = \lambda \Sigma g. \tag{6}$$

An important structural constraint we impose on Γ, which is applicable to a variety of high-dimensional data settings, is that it has low-rank: $r \leq d^* \equiv \text{rank}(\Gamma) \ll p$. It is this constraint that we will take advantage of in the randomized methods. In the case of $\Sigma = \text{I}$ (unsupervised case), $r = d^*$.

A.2 Sufficient Dimension Reduction

Dimension reduction is often a first step in the statistical analysis of high-dimensional data and could be followed by data visualization or predictive modeling. If the ultimate goal is the latter, then the statistical quantity of interest is a low dimensional summary $Z \equiv R(X)$ which captures all the predictive information in X relevant to Y:

$$Y = f(X) + \varepsilon = h(Z) + \varepsilon, \quad X \in \mathbb{R}^p, Z \in \mathbb{R}^r,\ r \ll p.$$

Sufficient dimension reduction (SDR) is one popular approach for estimating Z (Li, 1991; Cook and Weisberg, 1991; Li, 1992; Li et al., 2005; Nilsson et al., 2007; Sugiyama, 2007; Cook, 2007; Wu et al., 2010). In this appendix we focus on linear SDRs: $G = (g_1,\dots,g_r) \in \mathbb{R}^{p\times r} \Rightarrow R(X) = G^T X$, which provide a prediction-optimal reduction of X.

$$(Y \mid X) \overset{d}{=} (Y \mid G^T X), \quad \overset{d}{=} \text{ is equivalence in distribution.}$$

We will consider two specific dimension reduction methods: Sliced Inverse Regression (SIR) (Li, 1991) and Localized Sliced Inverse Regression (LSIR) (Wu et al., 2010). SIR is effective when the predictive structure in the data is global, i.e., there is single predictive subspace over the support of the marginal distribution of X. In the case of local or manifold predictive structure in the data, LSIR can be used to compute a projection matrix G that contains this non-linear (manifold) structure.

A.3 Efficient Solutions and Approximate SVD

SIR and LSIR reduce to solving a truncated generalized eigendecomposition problem as formulated in (6). Since we consider estimating the dimension reduction based on sample data we focus on the sample estimators $\widehat{\Sigma} = \frac{1}{n}X^T X$ and $\widehat{\Gamma}_{XY} = X^T K_{XY} X$, where K_{XY} is symmetric and encodes the method-specific grouping of the samples based on the response Y. In the classic statistical setting, when $n > p$, both $\widehat{\Sigma}$ and $\widehat{\Gamma}_{XY}$ are positive definite almost surely. Then, a typical solution proceeds by first sphering the data: $Z = \widehat{\Sigma}^{-\frac{1}{2}} X$, e.g., using a Cholesky or SVD representation $\widehat{\Sigma} = \widehat{\Sigma}^{\frac{1}{2}}(\widehat{\Sigma}^{\frac{1}{2}})^T$. This is followed by eigendecomposition of $\widehat{\Gamma}_{ZY}$ Li (1991); Wu et al. (2010) and back-transformation of the top eigenvectors directions to the canonical basis. The computational time is $O(np^2)$. When $n < p$, $\hat{\Sigma}$ and $\hat{\Gamma}$ are rank-deficient and a unique solution to the problem (6) does not exist. One widely-used approach, which allows us to make progress in this problematic setting, is to restrict our attention to the directions in the data with positive variance. Then we can proceed as before, using an orthogonal projection onto the span of the data. The total computation time in this case is $O(n^2 p)$. In many modern data analysis applications both n and p are very large, and hence algorithmic complexity of $O[\max(n,p) \times \min(n,p)^2]$ could be prohibitive, rendering the above approaches unusable. We propose an approximate solution that explicitly recovers the low-rank structure in Γ using *Adaptive Randomized SVD* from Section 2.4. In particular, assume $\text{rank}\,(\Gamma) = d^* \geq r$ (where r is the dimensionality of the optimal dimension reduction subspace). Then $\Gamma \overset{\text{svd}}{=} US^2U^T$, where $U \in \mathbb{R}^{p\times d^*}$. The generalized eigendecomposition problem (6) solution becomes restricted to the subspace spanned by the columns of Γ:

$$S^{-1}U^T \Sigma U S^{-1} e \;=\; \frac{1}{\lambda} e, \qquad e \equiv SU^T g. \tag{7}$$

The dimension reduction subspace is contained in the span(G), where

$$G = (US^{-1}e_1,\dots,US^{-1}e_r).$$

References

D. Achlioptas. Database-friendly random projections. In *Proceedings of the Twentieth Symposium on Principles of Database Systems*, PODS '01, pages 274–281, New York, NY, USA, 2001. ACM. ISBN 1-58113-361-8.

R.J. Adcock. A problem in least squares. *The Analyst*, 5:53–54, 1878.

J. Baglama and L. Reichel. Restarted block Lanczos bidiagonalization methods. *Numerical Algorithms*, 43(3):251–272, 2006.

C.M. Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Comput.*, 7 (1):108–116, January 1995. ISSN 0899-7667. doi: 10.1162/neco.1995.7.1.108. URL `http://dx.doi.org/10.1162/neco.1995.7.1.108`.

C. Boutsidis, M.W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 968–977. Society for Industrial and Applied Mathematics, 2009.

The Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447(7145):661–678, 2007.

R.D. Cook. Fisher Lecture: Dimension Reduction in Regression. *Statistical Science*, 22(1):1–26, 2007.

R.D. Cook and S. Weisberg. Discussion of Li (1991). *J. Amer. Statist. Assoc.*, 86:328–332, 1991.

S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

P. Drineas and M.W. Mahoney. On the Nystrom Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *J. Mach. Learn. Res.*, 6:2153–2175, December 2005.

P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo Algorithms for Matrices II: Computing a Low-Rank Approximation to a Matrix. *SIAM J. Comput.*, 36:158–183, July 2006.

P. Drineas, M. Magdon-Ismail, M.W. Mahoney, and D.P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.

F.Y. Edegworth. On the reduction of observations. *Philosophical Magazine*, pages 135–141, 1884.

R.A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Statistical Society A*, 222:309–368, 1922.

R.A. Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7 (2):179–188, 1936.

A. Franke, D.P.B. McGovern, J.C Barrett, K. Wang, G.L. Radford-Smith, T. Ahmad, C.W. Lees, T. Balschun, J. Lee, R. Roberts, et al. Genome-wide meta-analysis increases to 71 the number of confirmed crohn's disease susceptibility loci. *Nature genetics*, 42(12):1118–1125, 2010.

P. Frankl and H. Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *J. Comb. Theory Ser. A*, 44:355–362, June 1987.

K.R. Gabriel. Le biplot - outil d'exploration de données multidimensionnelles. *Journal de la société française de statistique*, 143(3-4):5–55, 2002.

L.L. Gerfo, L. Rosasco, F. Odone, E. De Vito, and A. Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008.

A.R. Gilmour, R. Thompson, and B.R Cullis. Average information reml: an efficient algorithm for variance parameter estimation in linear mixed models. *Biometrics*, pages 1440–1450, 1995.

A. Gittens and M.W. Mahoney. Revisiting the Nystrom method for improved large-scale machine learning. *arXiv preprint arXiv:1303.1849*, 2013.

G.H. Golub. Matrix decompositions and statistical calculations. In *Statistical Computation*, pages 365–397. New York: Academic Press, 1969.

G.H. Golub and C.F. Van Loan. *Matrix computations*. John Hopkins University Press, Baltimore, MD, 3rd edition, 1996. ISBN 0-8018-5413-X.

G.H. Golub, K. Sløna, and P. Van Dooren. Computing the SVD of a General Matrix Product/Quotient. *SIAM J. Matrix Anal. Appl*, 22:1–19, 2000.

M. Gu. Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing*, 37(3):A1139–A1173, 2015.

M. Gu and S.C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Sci. Comput.*, 17(4):848–869, July 1996.

N. Halko, P-G. Martinsson, and J.A. Tropp. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review*, 53(2):217–288, 2011.

R. Hecht-Nielsen and Věrá Kůrková. Quasiorthogonal dimension of euclidean spaces. *Technical Report Series*, INC-9205, 1992.

C.R. Henderson. *Applications of Linear Models in Animal Breeding*. University of Guelph, 1984.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

H. Hotelling. Analysis of a complex of statistical variables in principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, STOC '98, pages 604–613, New York, NY, USA, 1998. ACM. ISBN 0-89791-962-9.

D.L. Johnson and R. Thompson. Restricted maximum likelihood estimation of variance components for univariate animal models using sparse matrix techniques and average information. *Journal of dairy science*, 78(2):449–456, 1995.

W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society, 1984.

P.C. Kainen and V. Kůrková. Quasiorthogonal dimension of euclidean spaces. *Applied mathematics letters*, 6(3):7–10, 1993.

H.M. Kang, N.A. Zaitlen, C.M. Wade, A. Kirby, D. Heckerman, MJ. Daly, and E. Eskin. Efficient control of population structure in model organism association mapping. *Genetics*, 178(3):1709–1723, 2008.

H.M. Kang, J.H. Sul, S.K. Service, N.A. Zaitlen, S. Kong, N.B. Freimer, C. Sabatti, and E. Eskin. Variance component model to account for sample structure in genome-wide association studies. *Nature genetics*, 42(4):348–354, 2010.

B.W. Kennedy, M. Quinton, and J.A. Van Arendonk. Estimation of effects of single genes on quantitative traits. *Journal of Animal Science*, 70(7):2000–2012, 1992.

A. Krote, B.J. Vilhjálmsson, V. Segura, A. Platt, Q. Long, and M. Nordburg. A mixed-model approach for genome-wide association studies of correlated traits in structured populations. *Nature genetics*, 44(9):106610711, 2012.

R.R.B. Lehoucq, D.D.C. Sorensen, and C-C. Yang. *Arpack User's Guide: Solution of Large-Scale Eigenvalue Problems With Implicitly Restorted Arnoldi Methods*, volume 6. SIAM, 1998.

B. Li, H. Zha, and F. Chiaromonte. Contour Regression: A General Approach to Dimension Reduction. *The Annals of Statistics*, 33(4):1580–1616, 2005.

K.C. Li. Sliced inverse regression for dimension reduction (with discussion). *J. Amer. Statist. Assoc.*, 86:316–342, 1991.

K.C. Li. On principal Hessian directions for data visulization and dimension reduction: another application of Stein's lemma. *J. Amer. Statist. Assoc.*, 87:1025–1039, 1992.

E. Liberty, F. Woolfe, P-G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104(51): 20167–20172, 2007.

B. Lin, Z. Pang, and J. Jiang. Fixed and random effects selection by reml and pathwise coordinate optimization. *Journal of Computational and Graphical Statistics*, 22(2):341–355, 2013.

C. Lippert, J. Listgarten, Y. Liu, C.M. Kadie, R.I. Davidson, and D. Heckerman. Fast linear mixed models for genome-wide association studies. *Nature Methods*, 8(10):833–835, 2011.

C. Lippert, G. Quon, E.Y. Kang, C.M. Kadie, J. Listgarten, and D. Heckerman. The benefits of selecting phenotype-specific variants for applications of mixed models in genomics. *Scientific reports*, 3, 2013.

J. Listgarten, C. Lippert, C.M. Kadie, R.I. Davidson, E. Eskin, and D. Heckerman. Improved linear mixed models for genome-wide association studies. *Nature methods*, 9(6):525–526, 2012.

M.W. Mahoney. Randomized Algorithms for Matrices and Data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

M.W. Mahoney and L. Orecchia. Implementing regularization implicitly via approximate eigenvector computation. *arXiv preprint arXiv:1010.0703*, 2010.

P.-G. Martinsson, A. Szlam, and M. Tygert. Normalized power iterations for the computation of SVD. *NIPS workshop on low-rank methods for large-scale machine learning*, 2010.

K. Matilainen, E.A. Mäntysaari, M.H. Lidauer, I. Strandén, and R. Thompson. Employing a monte carlo algorithm in newton-type methods for restricted maximum likelihood estimation of genetic parameters. *PloS one*, 8(12):e80821, 2013.

D. Mimno, D.M. Blei, and B.E. Engelhardt. Posterior predictive checks to quantify lack-of-fit in admixture models of latent population structure. *arXiv preprint arXiv:1407.0050*, 2014.

J. Nilsson, F. Sha, and M.I. Jordan. Regression on Manifolds Using Kernel Dimension Reduction. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

A.B. Owen and P.O. Perry. Bi-cross-validation of the SVD and the nonnegative matrix factorization. *The Annals of Applied Statistics*, 3:564–594, 2009. doi: 10.1214/08-AOAS227.

P.O. Perry and M.W. Mahoney. Regularized laplacian estimation and fast eigenvector approximation. In *Advances in Neural Information Processing Systems*, pages 2420–2428, 2011.

T. Poggio and F. Girosi. Regularization Algorithms for Learning that are Equivalent to Multilayer Networks. *Science*, 247:978–982, February 1990. doi: 10.1126/science.247.4945.978.

N.G. Polson and J.G. Scott. Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction. In *Bayesian Statistics 9*. Oxford University Press, 2010.

A.L. Price, A. Helgason, G. Thorleifsson, S.A. McCarroll, A. Kong, and K. Stefansson. Single-tissue and cross-tissue heritability of gene expression via identity-by-descent in related or unrelated individuals. *PLoS Genet*, 7, 02 2011.

J.K. Pritchard and P. Donnelly. Case–control studies of association in structured or admixed populations. *Theoretical population biology*, 60(3):227–237, 2001.

V. Rokhlin, A. Szlam, and M. Tygert. A Randomized Algorithm for Principal Component Analysis. *SIAM J. Matrix Anal. Appl.*, 31(3):1100–1124, August 2009.

A. Rudi, G.D. Cañas, and L. Rosasco. On the sample complexity of subspace learning. In *Advances in Neural Information Processing Systems*, pages 2067–2075, 2013.

A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1648–1656, 2015.

D.E. Runcie and S. Mukherjee. Dissecting high-dimensional phenotypes with bayesian sparse factor analysis of genetic covariance matrices. *Genetics*, 194(3):753–767, 2013.

Y. Saad. *Numerical methods for large eigenvalue problems*, volume 158. SIAM, 1992.

T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *Foundations of Computer Science, 2006. FOCS '06. 47th Annual IEEE Symposium on*, pages 143 –152, October 2006.

The MHC sequencing consortium. Complete sequence and gene map of a human major histocompatibility complex. *Nature*, 401(6756):921–923, 1999.

P. Simard, Y. LeCun, and J.S. Denker. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, pages 50–58, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc. ISBN 1-55860-274-7. URL http://dl.acm.org/citation.cfm?id=645753.668226.

S. Smale and D. Zhou. Learning theory estimates via integral operators and their approximations. *Constructive approximation*, 26(2):153–172, 2007.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2627435.2670313.

G.W. Stewart. *Matrix Algorithms: Volume 2, Eigensystems*, volume 2. SIAM, 2001.

K. Strimmer. fdrtool: a versatile r package for estimating local and tail area-based false discovery rates. *Bioinformatics*, 24(12):1461–1462, 2008.

M. Sugiyama. Dimension reduction of multimodal labeled data by local Fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.

A. Szlam, Y. Kluger, and M. Tygert. An implementation of a randomized algorithm for principal component analysis. *arXiv preprint arXiv:1412.3510*, 2014.

E.A. Thompson. Population correlation and population kinship. *Theoretical population biology*, 10 (2):205–226, 1976.

E.D. Vito, L. Rosasco, A. Caponnetto, U.D. Giovannini, and F. Odone. Learning from examples as an inverse problem. *Journal of Machine Learning Research*, 6(May):883–904, 2005.

C.K.I. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001. URL http://papers.nips.cc/paper/1866-using-the-nystrom-method-to-speed-up-kernel-machines.pdf.

Q. Wu, F. Liang, and S. Mukherjee. Localized Sliced Inverse Regression. *Journal of Computational and Graphical Statistics*, 19(4):843–860, 2010.

J. Yang, S.H. Lee, M.E. Goddard, and P.M Visscher. GCTA: a tool for genome-wide complex trait analysis. *The American Journal of Human Genetics*, 88(1):76–82, 2011.

J. Yang, N.A. Zaitlen, M.E. Goddard, P.M. Visscher, and A.L. Price. Advantages and pitfalls in the application of mixed-model association methods. *Nature genetics*, 46(2):100–106, 2014.

Y. Yao, L. Rosasco, and A. Caponnetto. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

G. Young. Maximum likelihood estimation and factor analysis. *Psychometrika*, 6:49–53, 1941.

X. Zhou and M. Stephens. Genome-wide efficient mixed-model analysis for association studies. *Nature genetics*, 44(7):821–824, 2012.

X. Zhou, P. Carbonetto, and M. Stephens. Polygenic modeling with Bayesian sparse linear mixed models. *PLoS genetics*, 9(2):e1003264, 2013.

Journal of Machine Learning Research 18 (2017) 1-35 Submitted 11/16; Revised 10/17; Published 11/17

A Nonconvex Approach for Phase Retrieval: Reshaped Wirtinger Flow and Incremental Algorithms

Huishuai Zhang HZHAN23@SYR.EDU
Department of EECS
Syracuse University
Syracuse, NY 13244 USA

Yi Zhou ZHOU.1172@OSU.EDU
Yingbin Liang LIANG.889@OSU.EDU
Yuejie Chi CHI.97@OSU.EDU
Department of ECE
The Ohio State University
Columbus, OH 43210 USA

Editor: Benjamin Recht

Abstract

We study the problem of solving a quadratic system of equations, i.e., recovering a vector signal $\boldsymbol{x} \in \mathbb{R}^n$ from its magnitude measurements $y_i = |\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle|, i = 1, ..., m$. We develop a gradient descent algorithm (referred to as RWF for reshaped Wirtinger flow) by minimizing the quadratic loss of the magnitude measurements. Comparing with Wirtinger flow (WF) (Candès et al., 2015), the loss function of RWF is nonconvex and nonsmooth, but better resembles the least-squares loss when the phase information is also available. We show that for random Gaussian measurements, RWF enjoys linear convergence to the true signal as long as the number of measurements is $\mathcal{O}(n)$. This improves the sample complexity of WF ($\mathcal{O}(n \log n)$), and achieves the same sample complexity as truncated Wirtinger flow (TWF) (Chen and Candès, 2015), but without any sophisticated truncation in the gradient loop. Furthermore, RWF costs less computationally than WF, and runs faster numerically than both WF and TWF. We further develop an incremental (stochastic) version of RWF (IRWF) and connect it with the randomized Kaczmarz method for phase retrieval. We demonstrate that IRWF outperforms existing incremental as well as batch algorithms with experiments.

Keywords: gradient descent, phase retrieval, nonconvex optimization, regularity condition, stochastic algorithms

1. Introduction

Many problems in machine learning and signal processing can be reduced to solving a quadratic system of equations. For instance, in phase retrieval applications, i.e., X-ray crystallography and coherent diffraction imaging (Drenth, 2007; Miao et al., 1999, 2008), the structure of an object is to be recovered from the measured far field diffracted intensity when the object is illuminated by a source light. Mathematically, such a problem amounts to recovering the signal from only the magnitudes of its linear measurements. Specifically, the problem is formulated as follows.

©2017 Huishuai Zhang, Yi Zhou, Yingbin Liang and Yuejie Chi.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-572.html.

Problem 1 *Recover $\boldsymbol{x} \in \mathbb{R}^n/\mathbb{C}^n$ from the measurements given as*

$$y_i = |\langle \boldsymbol{a}_i, \mathbf{x}\rangle|, \quad \text{for } i = 1, \cdots, m, \tag{1}$$

where $\boldsymbol{a}_i \in \mathbb{R}^n/\mathbb{C}^n$ are known design vectors.

Various algorithms have been proposed to solve this problem since the 1970s. The error-reduction methods (Gerchberg, 1972; Fienup, 1982) work well empirically but lack theoretical guarantees. More recently, convex relaxation of the problem has been formulated, for example, via PhaseLift (Candès et al., 2013; Candès and Li, 2014; Chen et al., 2015) and PhaseCut (Waldspurger et al., 2015), and the correspondingly developed algorithms typically come with performance guarantees. The readers can refer to the review paper (Shechtman et al., 2015) to learn more about applications and algorithms of the phase retrieval problem.

Nonetheless, these convex approaches often suffer from high computational complexity particularly when the signal dimension is large. It is therefore desirable to develop more efficient nonconvex approaches that can provably recover the true signal. Netrapalli et al. (2013) proposed the *AltMinPhase* algorithm, which alternates between updates of the phase and the signal, and showed that it converges linearly and recovers the true signal with $\mathcal{O}(n \log^3 n)$ Gaussian measurements, when $\boldsymbol{a}_i$'s are composed of *independent and identically distributed* (i.i.d.) standard Gaussian entries. More recently, Candès et al. (2015) introduced the *Wirtinger flow* (WF) algorithm, which guarantees signal recovery via a simple gradient descent algorithm with only $\mathcal{O}(n \log n)$ Gaussian measurements and attains ϵ-accuracy within $\mathcal{O}(mn^2 \log 1/\epsilon)$ flops[1]. More specifically, WF obtains a good initialization by the spectral method, and then minimizes the following nonconvex loss function based on the quadratic loss of the squared magnitude measurements

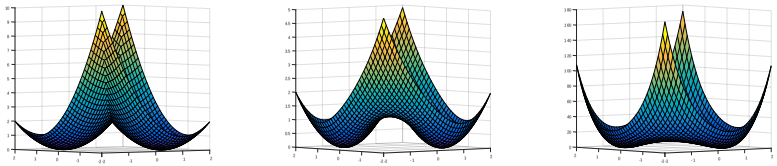
$$\ell_{WF}(\boldsymbol{z}) := \frac{1}{4m}\sum_{i=1}^{m}(|\boldsymbol{a}_i^T \boldsymbol{z}|^2 - |y_i|^2)^2, \tag{2}$$

via the gradient descent scheme.

WF is further improved by *truncated Wirtinger flow* (TWF) (Chen and Candès, 2015), which adopts a Poisson loss function of $|\boldsymbol{a}_i^T \boldsymbol{z}|^2$, and keeps only well-behaved measurements based on carefully designed truncation thresholds for calculating the initialization and the gradient updates. Such truncation assists to achieve a linear convergence rate using a fixed step size; as a consequence, TWF reduces both the sample complexity to $\mathcal{O}(n)$ and the computational complexity to $\mathcal{O}(mn \log 1/\epsilon)$.

Furthermore, incremental/stochastic methods have been proposed to solve Problem 1. Specifically, the randomized Kaczmarz method has been adopted to solve the phase retrieval problem (Wei, 2015; Li et al., 2015), which was shown to have excellent empirical performance, but no nonasymptotic global convergence guarantee was established. Incremental truncated Wirtinger flow (ITWF) (Kolte and Özgür, 2016) is another stochastic algorithm developed based on TWF.

1. WF has computational cost $\mathcal{O}(mn^2 \log(1/\epsilon))$ (Candès et al., 2015, Theorem 3.1) because it needs $\log(1/\epsilon) \cdot \frac{1}{-\log(1-c/n)} \approx \frac{n}{c}\log(1/\epsilon)$ iterations to achieve ϵ-accuracy due to the convergence rate $\left(1 - \frac{c}{n}\right)^k$ and each iteration needs $\mathcal{O}(mn)$ flops. Contrastingly, RWF has computational cost $\mathcal{O}(mn \log 1/\epsilon)$ because it requires only $c \log(1/\epsilon)$ iterations to achieve ϵ-accuracy due to the convergence rate $(1-c)^k$.



(a) Expected loss of LS (b) Expected loss of RWF (c) Expected loss of WF

Figure 1: Surface of the expected loss function of (a) least-squares (mirrored symmetrically), (b) RWF, and (c) WF when $\boldsymbol{x} = [1, -1]^T$.

1.1 Our Contribution

This paper adopts the following loss function

$$\ell(\boldsymbol{z}) := \frac{1}{2m}\sum_{i=1}^{m}\left(|\boldsymbol{a}_i^T\boldsymbol{z}| - y_i\right)^2, \tag{3}$$

which is the quadratic loss of the magnitude measurements. Compared to the loss function (2) of WF that adopts the quadratic loss of $|\boldsymbol{a}_i^T\boldsymbol{z}|^2$, the above loss function adopts the quadratic loss of $|\boldsymbol{a}_i^T\boldsymbol{z}|$ and hence has lower-order variables. While both loss functions are nonconvex in $\boldsymbol{z}$, the WF loss function (2) is a fourth-order smooth function of $\boldsymbol{a}_i^T\boldsymbol{z}$ and our loss function (3) in contrast is nonsmooth.

To minimize such a nonconvex and nonsmooth loss function (3), we develop a gradient descent algorithm, which sets the "gradient" to zero corresponding to nonsmooth samples. We refer to such an algorithm together with an initialization using a new spectral method (different from that employed in TWF or WF) as *reshaped Wirtinger flow* (RWF). We show that the new loss function has great advantage in both statistical and computational efficiency, despite nonsmoothness. In fact, the curvature of such a loss function behaves similarly to that of a least-squares problem with phase information in the neighborhood of global optimizers, and hence yields faster convergence. To provide further insights, consider the standard problem of solving $\boldsymbol{x}$ from *linear measurements* $\langle \boldsymbol{a}_i, \mathbf{x}\rangle$, $i = 1, \cdots, m$, where $\boldsymbol{a}_i$'s are composed of i.i.d. standard Gaussian entries. In this case, it is natural to use the least-squares loss function

$$\ell_{LS}(\boldsymbol{z}) := \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{a}_i^T\boldsymbol{z} - \boldsymbol{a}_i^T\boldsymbol{x}\right)^2. \tag{4}$$

Examining the expected (with respect to $\boldsymbol{a}_i$'s) loss surface of $\min\{\ell_{LS}(\boldsymbol{z}), \ell_{LS}(-\boldsymbol{z})\}$ (to mimic sign ambiguity), $\ell(\boldsymbol{z})$, and $\ell_{WF}(\boldsymbol{z})$ in Figure 1, whose expressions can be found in Appendix A, it can be seen that the loss of RWF, rather than the loss of WF, has a similar curvature to the quadratic least-squares loss around the global optimizers, which justifies its better performance than WF.

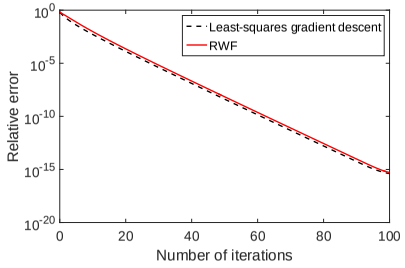


Figure 2: Comparison of convergence behavior between RWF and the least-squares gradient descent with the same initialization, the same parameters $n = 1000$, $m = 6n$, and the same step size $\mu = 0.8$.

The nonsmoothness of the loss function (3) does not negatively impact the performance of RWF because only with negligible probability the algorithm encounters nonsmooth points for some samples, which furthermore are set not to contribute to the gradient direction by RWF. The gradient of the RWF loss (3) is given as

$$\nabla\ell(\boldsymbol{z}) := \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{a}_i^T\boldsymbol{z} - y_i\cdot\mathrm{sgn}(\boldsymbol{a}_i^T\boldsymbol{z})\right)\boldsymbol{a}_i, \tag{5}$$

where $\mathrm{sgn}(0) = 0$ by convention. Comparing this with the gradient of the least-squares loss

$$\nabla\ell_{LS}(\boldsymbol{z}) = \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{a}_i^T\boldsymbol{z} - \boldsymbol{a}_i^T\boldsymbol{x}\right)\boldsymbol{a}_i, \tag{6}$$

one can see that RWF uses estimated phase information $\mathrm{sgn}(\boldsymbol{a}_i^T\boldsymbol{z})$ to generate the gradient updates, and the convergence behavior of RWF is much similar to that of least-squares with phase information if initialized properly. Indeed, Figure 2 illustrates that RWF takes almost the *same* number of iterations for recovering a signal (with only the magnitude information) as the least-squares gradient descent method for recovering a signal (with both the magnitude and the sign information).

We further develop incremental/stochastic versions of RWF using mini-batches of measurements, called incremental RWF (IRWF), and show that IRWF also enjoys the advantageous local curvature of RWF, and achieves excellent statistical and computational performance. Along the way, we establish the performance guarantee of the Kaczmarz-PR update rule by interpreting it as a variant of IRWF. We conduct extensive numerical experiments to demonstrate that IRWF performs better than other competitive incremental algorithms (ITWF and Kaczmarz-PR) as well as batch algorithms (RWF, TWF, WF and AltMinPhase).

We summarize our main results as follows.

- Statistically, we show that RWF recovers the true signal with $\mathcal{O}(n)$ Gaussian measurements, which is order-wise optimal. Thus, RWF improves the sample complexity

$\mathcal{O}(n \log n)$ of WF, and achieves the same sample complexity as TWF but without truncation in the gradient descent step.

- Computationally, RWF converges linearly to the true signal with a constant step size, requiring $\mathcal{O}(mn \log(1/\epsilon))$ flops to reach ϵ-accuracy. Again, without truncation in the gradient descent step, RWF improves the computational cost $\mathcal{O}(mn^2 \log(1/\epsilon))$ of WF and achieves the same computational cost as TWF.

- We also show that RWF is robust to bounded additive noise. The estimation error is shown to diminish at a linear rate with respect to the noise level up to a scaling difference. Experiments on Poisson noise further corroborate the stability guarantee.

- The incremental versions of RWF (IRWF) is shown to shrink the estimation error in expectation for one update, which implies the linear convergence of the algorithm if the iteration path lies in the neighborhood of the true signal. Furthermore, we show that Kaczmarz-PR can be viewed as IRWF under a specific choice of step size, and establish a similar convergence result for Kaczmarz-PR under the same sample complexity.

- Numerically, RWF and its incremental versions require fewer parameters, e.g., truncation thresholds, than TWF in practice. RWF is generally two times faster than TWF and four to six times faster than WF in terms of both the number of iterations and time cost. IRWF also outperforms existing incremental as well as batch algorithms.

Compared to WF and TWF, the nonsmoothness in the loss function requires new bounding techniques in order to establish the theoretical guarantees. On the other hand, our technical proof is much simpler, because the lower-order loss function allows to bypass higher-order moments of variables as well as truncation of samples in the gradient descent steps.

1.2 Related Work

The quadratic loss function of magnitudes (3) was also used in the early literature of phase retrieval (Fienup, 1982) with Fourier magnitude measurements. However, no global convergence guarantee was available in (Fienup, 1982). Along the line of developing nonconvex algorithms with global performance guarantees for the phase retrieval problem, Netrapalli et al. (2013) and Waldspurger (2016) developed alternating minimization algorithms, Candès et al. (2015); Chen and Candès (2015); Zhang et al. (2016); Cai et al. (2016) developed first-order algorithms, and a recent study Sun et al. (2016) characterized the geometric structure of the nonconvex objective and designed a second-order trust-region algorithm. This paper is closely related to (Candès et al., 2015; Chen and Candès, 2015; Zhang et al., 2016), but develops a new gradient descent algorithm based on a lower-order nonsmooth (as well as nonconvex) loss function that yields advantageous statistical and computational efficiency.

Stochastic algorithms were also developed for the phase retrieval problem. Kolte and Özgür (2016) studied the incremental truncated Wirtinger flow (ITWF) and showed that ITWF needs much fewer passes of data than TWF to reach the same accuracy. Wei (2015) adapted the Kaczmarz method to solve the phase retrieval problem and demonstrated its

fast empirical convergence. We show that IRWF is closely related to Kaczmarz-PR, and empirically runs faster than ITWF thanks to the advantageous curvature of the loss function.

After our work was posted on arXiv, an independent work (Wang et al., 2016) was subsequently posted, which also adopts the same loss function but develops a slightly different algorithm called TAF (i.e., truncated amplitude flow). One major difference is that RWF does not require truncation in the gradient loops while TAF still employs truncation. Hence, RWF has fewer parameters to tune, and is easier to implement than TAF in practice. Furthermore, RWF demonstrates the performance advantage of adopting a lower-order loss function even without truncation, which cannot be observed from TAF. The two algorithms also employ different initialization strategies. Moreover, we analyze stochastic algorithms based on the new loss function while Wang et al. (2016) do not.

More generally, various high-dimensional signal estimation problems have been studied by minimizing nonconvex loss functions. For example, a partial list of these studies include matrix completion (Keshavan et al., 2010; Jain et al., 2013; Sun and Luo, 2016; Hardt, 2014; Sa et al., 2015; Zheng and Lafferty, 2016; Jin et al., 2016; Ge et al., 2016), low-rank matrix recovery (Bhojanapalli et al., 2016; Chen and Wainwright, 2015; Tu et al., 2015; Zheng and Lafferty, 2015; Park et al., 2016; Wei et al., 2016; Li et al., 2017), robust PCA (Netrapalli et al., 2014), robust tensor decomposition (Anandkumar et al., 2016), dictionary learning (Arora et al., 2015; Sun et al., 2015), community detection (Bandeira et al., 2016), phase synchronization (Boumal, 2016), blind deconvolution (Lee et al., 2016b; Li et al., 2016), etc.

For minimizing a general nonconvex nonsmooth objective, various algorithms have been proposed, such as gradient sampling (Burke et al., 2005; Kiwiel, 2007) and majorization-minimization (Ochs et al., 2015). These algorithms are often shown to converge to critical points which may be local minimizers or saddle points, without an explicit characterization of convergence rates. In contrast, our algorithm is specifically designed for the phase retrieval problem, and can be shown to converge linearly to the global optimum under an appropriate initialization.

The advantage of using nonsmooth loss functions in our study is analogous in spirit to that of the rectifier activation function (of the form $\max\{0, \cdot\}$) in neural networks. It has been shown that rectified linear unit ($ReLU$) enjoys superb advantage in reducing the training time (Krizhevsky et al., 2012) and promoting sparsity (Glorot et al., 2011) over its counterparts of sigmoid and hyperbolic tangent functions, in spite of non-linearity and non-differentiability at zero. Our results in fact also demonstrate that a nonsmooth but simpler loss function yields improved performance.

1.3 Paper Organization and Notations

The rest of this paper is organized as follows. Section 2 describes the RWF algorithm in detail and establishes its performance guarantee. Section 3 introduces the IRWF algorithm, establishes its performance guarantee and compares it with existing stochastic algorithms. Section 4 compares RWF and IRWF with other competitive algorithms numerically. Finally, Section 5 concludes the paper with comments on future directions.

Throughout the paper, boldface lowercase letters such as $\boldsymbol{a}_i, \boldsymbol{x}, \boldsymbol{z}$ denote vectors, and boldface capital letters such as $\boldsymbol{A}, \boldsymbol{Y}$ denote matrices. For two matrices, $\boldsymbol{A} \prec \boldsymbol{B}$ means that $\boldsymbol{B} - \boldsymbol{A}$ is positive definite. For a complex matrix and a vector, $\boldsymbol{A}^*$ and $\boldsymbol{z}^*$ denote conjugate

transposes of $\boldsymbol{A}$ and $\boldsymbol{z}$, respectively. For a real matrix and a vector, $\boldsymbol{A}^T$ and $\boldsymbol{z}^T$ denote transposes of $\boldsymbol{A}$ and $\boldsymbol{z}$, respectively. We let $\odot$ denotes element-wise product. The indicator function $\mathbf{1}_A = 1$ if the event A occurs, and $\mathbf{1}_A = 0$ otherwise. We let $\|\boldsymbol{x}\|_1$ and $\|\boldsymbol{x}\|$ denote the l_1 norm and l_2 norm of a vector $\boldsymbol{x}$, respectively. Moreover, let $\|\boldsymbol{A}\|_F$ and $\|\boldsymbol{A}\|$ denote the Frobenius norm and the spectral norm of a matrix $\boldsymbol{A}$, respectively. We note that the constants c, C, c_0, c_1, c_2 may be different in different equations, for the sake of notational simplicity.

2. Reshaped Wirtinger Flow

Consider the Problem 1 for the complex case. It can be observed that if $\boldsymbol{z}$ is a solution, i.e., satisfying Equation (1), then $\boldsymbol{z}e^{-j\phi}$ is also the solution of the problem where ϕ is an arbitrary phase constant. Therefore, the recovery is up to a phase difference. Thus, we define the Euclidean distance between two complex vectors up to a global phase difference (Candès et al., 2015) as,

$$\operatorname{dist}(\boldsymbol{z}, \boldsymbol{x}) := \min_{\phi \in [0, 2\pi)} \|\boldsymbol{z}e^{-j\phi} - \boldsymbol{x}\|, \tag{7}$$

which is simply $\min \|\boldsymbol{z} \pm \boldsymbol{x}\|$ for the real case.

In this paper, we focus on the real-valued case in our analysis, but the algorithm designed below is applicable to the complex-valued case and the case with other measurement vectors such as *coded diffraction pattern* (CDP) as we demonstrate via numerical experiments in Section 4.

We design RWF (see Algorithm 1) for solving the phase retrieval problem, which contains two stages: spectral initialization and gradient loop. The suggested values for the parameters[2] are given by $\alpha_l = 1, \alpha_u = 5$ and $\mu = 0.8$. The rescaling coefficient in λ_0 and the conjugate transpose $\boldsymbol{a}_i^*$ allow the algorithm readily applicable to the complex and CDP cases. We next describe the two stages of RWF in details in Sections 2.1 and 2.2, respectively, and establish the convergence guarantee in Section 2.3. Finally, we provide the stability guarantee of RWF in Section 2.4.

2.1 Initialization via the Spectral Method

When solving nonconvex problems by iterative algorithms, the starting point is critical. The spectral method is a popular choice in the literature (Keshavan et al., 2010; Netrapalli et al., 2013; Candès et al., 2015; Chen and Candès, 2015), which often provides a good initialization. Different from the spectral initialization used in AltMinPhase (Netrapalli et al., 2013), WF (Candès et al., 2015) and TWF (Chen and Candès, 2015), which are based on the squared magnitudes as the weight of each rank-one matrix $\boldsymbol{a}_i\boldsymbol{a}_i^*$, we propose an alternative initialization in Algorithm 1 that uses the magnitudes instead, and truncates samples that are either too large or too small. We show that such an initialization achieves a smaller sample complexity than WF and the same sample complexity as TWF order-wise, and furthermore, performs better than both WF and TWF numerically.

Our initialization consists of estimation of both the norm and the direction of $\boldsymbol{x}$. The norm estimation of $\boldsymbol{x}$ is given by λ_0 in Algorithm 1. Intuitively, with real-valued Gaussian

2. For the complex Gaussian case, we suggest $\mu = 1.2$.

Algorithm 1 Reshaped Wirtinger Flow

Input: $\boldsymbol{y} = \{y_i\}_{i=1}^m$, $\{\boldsymbol{a}_i\}_{i=1}^m$;

Parameters: Lower and upper thresholds α_l, α_u for truncation in initialization, step size μ;

Initialization: Let $\boldsymbol{z}^{(0)} = \lambda_0 \tilde{\boldsymbol{z}}$, where $\lambda_0 = \frac{mn}{\sum_{i=1}^m \|\boldsymbol{a}_i\|_1} \cdot \left(\frac{1}{m}\sum_{i=1}^m y_i\right)$ and $\tilde{\boldsymbol{z}}$ is the leading eigenvector of

$$\boldsymbol{Y} := \frac{1}{m}\sum_{i=1}^m y_i \boldsymbol{a}_i \boldsymbol{a}_i^* \mathbf{1}_{\{\alpha_l \lambda_0 < y_i < \alpha_u \lambda_0\}}. \tag{8}$$

Gradient loop: for $t = 0 : T-1$ do

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \frac{\mu}{m}\sum_{i=1}^m \left(\boldsymbol{a}_i^*\boldsymbol{z}^{(t)} - y_i \cdot \frac{\boldsymbol{a}_i^*\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_i^*\boldsymbol{z}^{(t)}|}\right)\boldsymbol{a}_i. \tag{9}$$

Output $\boldsymbol{z}^{(T)}$.

measurement vectors (i.e., $\boldsymbol{a}_i \sim \mathcal{N}(0, \boldsymbol{I}_{n\times n})$), the scaling coefficient $\frac{mn}{\sum_{i=1}^m \|\boldsymbol{a}_i\|_1} \approx \sqrt{\frac{\pi}{2}}$. Moreover, for $i = 1, \ldots, m$, $y_i = |\boldsymbol{a}_i^T\boldsymbol{x}|$ are independent sub-Gaussian random variables with mean $\sqrt{\frac{2}{\pi}}\|\boldsymbol{x}\|$, and thus $\frac{1}{m}\sum_{i=1}^m y_i \approx \sqrt{\frac{2}{\pi}}\|\boldsymbol{x}\|$. Combining these two facts yields the desired argument.

The direction of $\boldsymbol{x}$ is approximated by the leading eigenvector of $\boldsymbol{Y}$, because $\boldsymbol{Y}$ approaches $\mathrm{E}[\boldsymbol{Y}]$ by concentration of measure arguments, and the leading eigenvector of $\mathrm{E}[\boldsymbol{Y}]$ takes the form $c\boldsymbol{x}$ for some scalar $c \in \mathbb{R}$. We note that (8) involves truncation of samples from both sides, in contrast to truncation only by an upper threshold in TWF (Chen and Candès, 2015). The truncation parameters α_l and α_u are related to the eigenvalue gap between the top two eigenvalues of $\boldsymbol{Y}$, which determines the estimation accuracy of the eigenvector after running k iterations of the power method[3]. The larger the eigenvalue gap, the better the accuracy of the power method given a number of iterations. We note that $\alpha_l = 1, \alpha_u = \infty$ yield the largest eigenvalue gap from the developments in Appendix B. We explicitly set a bounded α_u for the development of the proof and for numerical stability.

We next provide the formal statement that with high probability, the proposed initialization lands in a small neighborhood around the true signal.

Proposition 1 *Fix $\delta > 0, \alpha_l = 1$ and $\alpha_u \gg 1$. The initialization step in Algorithm 1 yields $\boldsymbol{z}^{(0)}$ satisfying $dist(\boldsymbol{z}^{(0)}, \boldsymbol{x}) \le \delta\|\boldsymbol{x}\|$ with the probability at least $1 - \exp(-cm\epsilon^2)$, if $m > C(\delta, \epsilon)n$, where c is some positive constant and C is a positive constant affected by δ and ϵ.*

Proof. See Appendix B. ■

Finally, we numerically compare different initialization methods in Figure 3. It is demonstrated that RWF achieves better initialization accuracy in terms of the relative error $\operatorname{dist}(\boldsymbol{z}^{(0)}, \boldsymbol{x})/\|\boldsymbol{x}\|$ than WF, TWF, as well as the initialization method proposed in TAF (Wang et al., 2016).

3. Numerical linear algebra shows that after k iterations of the power method, the estimation accuracy of the eigenvector is given by $\mathcal{O}((\lambda_2/\lambda_1)^k)$, where λ_1 and λ_2 are the top two eigenvalues.

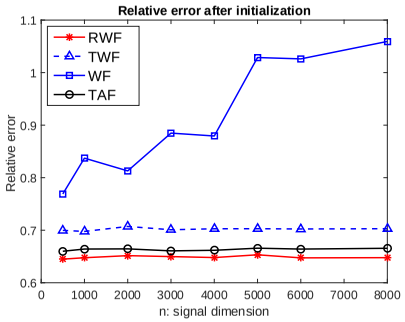


Figure 3: Comparison of different initialization methods with $m = 6n$ and under 50 iterations of power method.

2.2 Gradient Loop

The gradient loop of Algorithm 1 is based on the loss function (3), and the update rule (9) makes itself suitable for the complex case. For the real case the update direction is given as follows:

$$\nabla \ell(\boldsymbol{z}) := \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{a}_i^T\boldsymbol{z} - y_i\cdot \mathrm{sgn}(\boldsymbol{a}_i^T\boldsymbol{z})\right)\boldsymbol{a}_i = \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{a}_i^T\boldsymbol{z} - y_i\cdot \frac{\boldsymbol{a}_i^T\boldsymbol{z}}{|\boldsymbol{a}_i^T\boldsymbol{z}|}\right)\boldsymbol{a}_i, \tag{10}$$

where $\mathrm{sgn}(\cdot)$ is the sign function for nonzero arguments. We further set $\mathrm{sgn}(0) = 0$ and $\frac{0}{|0|} = 0$. In fact, $\nabla\ell(\boldsymbol{z})$ equals the gradient of the loss function (3) over the samples that satisfy $\boldsymbol{a}_i^T\boldsymbol{z} \neq 0$. For samples corresponding to the nonsmooth point, i.e., $\boldsymbol{a}_i^T\boldsymbol{z} = 0$, we adopt the Fréchet superdifferential (Kruger, 2003) for nonconvex functions to set the gradient component to be zero (as zero is an element in the Fréchet superdifferential). With abuse of terminology, we still refer to $\nabla\ell(\boldsymbol{z})$ in Equation (10) as the "gradient" for simplicity, which rather represents the update direction in the gradient loop of Algorithm 1.

2.3 Linear Convergence of RWF

We characterize the convergence guarantee of RWF in the following theorem.

Theorem 2 *Consider the problem of solving any given $\boldsymbol{x} \in \mathbb{R}^n$ from a system of equations (1) with Gaussian measurement vectors. There exist some universal constants $\mu_0 > 0$ (μ_0 can be set as 0.8 in practice), $0 < \rho, \nu < 1$ and $c_0, c_1, c_2 > 0$ such that if $m \geq c_0 n$ and $\mu < \mu_0$, then with probability at least $1 - c_1\exp(-c_2 m)$, Algorithm 1 yields*

$$dist(\boldsymbol{z}^{(t)}, \boldsymbol{x}) \leq \nu(1-\rho)^t\|\boldsymbol{x}\|, \quad \forall t \in \mathbb{N}. \tag{11}$$

Proof We outline the proof here with the details delegated to Appendix C. Compared to WF and TWF, our proof is much simpler due to the lower-order loss function that RWF relies on. We first introduce a global phase notation for the real case as follows:

$$\Phi_{\boldsymbol{x}}(\boldsymbol{z}) := \begin{cases} 0, & \text{if } \|\boldsymbol{z}-\boldsymbol{x}\| \leq \|\boldsymbol{z}+\boldsymbol{x}\|, \\ \pi, & \text{otherwise.} \end{cases} \tag{12}$$

For the sake of simplicity, we let $\boldsymbol{z}$ be $e^{-j\Phi_{\boldsymbol{x}}(\boldsymbol{z})}\boldsymbol{z}$, which indicates that $\boldsymbol{z}$ is always in the neighborhood of $\boldsymbol{x}$.

Here, the central idea is to show that within the neighborhood of the global minimizer, RWF satisfies the *Regularity Condition* $\mathsf{RC}(\mu, \lambda, c)$ (Chen and Candès, 2015), i.e.,

$$\langle \nabla\ell(\boldsymbol{z}), \boldsymbol{z}-\boldsymbol{x}\rangle \geq \frac{\mu}{2}\|\nabla\ell(\boldsymbol{z})\|^2 + \frac{\lambda}{2}\|\boldsymbol{z}-\boldsymbol{x}\|^2 \tag{13}$$

for all $\boldsymbol{z}$ obeying $\|\boldsymbol{z}-\boldsymbol{x}\| \leq c\|\boldsymbol{x}\|$, where $0 < c < 1$ is some small constant. Then, as shown in (Candès et al., 2015; Chen and Candès, 2015), once the initialization lands into this neighborhood, linear convergence can be guaranteed with a proper choice of the constant step size, i.e.,

$$\mathrm{dist}^2\left(\boldsymbol{z} - \mu\nabla\ell(\boldsymbol{z}), \boldsymbol{x}\right) \leq (1-\mu\lambda)\mathrm{dist}^2(\boldsymbol{z}, \boldsymbol{x}), \tag{14}$$

for any $\boldsymbol{z}$ satisfying $\|\boldsymbol{z}-\boldsymbol{x}\| \leq c\|\boldsymbol{x}\|$.

Lemmas 6 and 7 in Appendix C yield that for all $\boldsymbol{z}$ satisfying $\|\boldsymbol{z}-\boldsymbol{x}\| \leq c\|\boldsymbol{x}\|$,

$$\langle \nabla\ell(\boldsymbol{z}), \boldsymbol{z}-\boldsymbol{x}\rangle \geq (1-0.26-3\epsilon)\|\boldsymbol{z}-\boldsymbol{x}\|^2 = (0.74-3\epsilon)\|\boldsymbol{z}-\boldsymbol{x}\|^2$$

with high probability. Moreover, Lemma 8 in Appendix C further yields that

$$\|\nabla\ell(\boldsymbol{z})\| \leq (1+\delta)\cdot 2\|\boldsymbol{z}-\boldsymbol{x}\| \tag{15}$$

with high probability. Therefore, the above two bounds imply that the Regularity Condition (13) holds for μ and λ satisfying

$$0.74 - 3\epsilon \geq \frac{\mu}{2}\cdot 4(1+\delta)^2 + \frac{\lambda}{2} \tag{16}$$

for sufficiently small ϵ and δ. ■

We note that Equation (16) implies an upper bound for the step size $\mu \leq \frac{0.74}{2} = 0.37$, by taking ϵ and δ to be sufficiently small. However, in practice, the step size μ can be set much larger than such a bound, say 0.8, while still keeping the algorithm convergent. This is because the coefficients in the proof are set for the convenience of the proof rather than being tightly chosen.

Theorem 2 indicates that RWF recovers the true signal with $\mathcal{O}(n)$ samples, which is order-wise optimal. Such an algorithm improves the sample complexity $\mathcal{O}(n\log n)$ of WF. Furthermore, RWF does not require truncation of samples in the gradient step to achieve the same sample complexity as TWF. This is mainly because RWF benefits from the lower-order loss function given in Equation (3), the curvature of which behaves similarly to the least-squares loss function locally as we explain in the introduction.

Theorem 2 also suggests that RWF converges linearly with a constant step size. To reach ϵ-accuracy, it requires a computational cost of $\mathcal{O}(mn\log(1/\epsilon))$ flops, which is better than WF ($\mathcal{O}(mn^2\log(1/\epsilon))$ and on par with TWF. Numerically, as we demonstrate in Section 4, RWF is two times faster than TWF and four to six times faster than WF in terms of both the iteration counts and the time cost in various examples.

2.4 Stability to Bounded Noise

We have established that RWF guarantees exact recovery at a linear convergence rate for noise-free measurements. We now study RWF in the presence of noise. Suppose the measurements are corrupted by bounded noise, and are given by

$$y_i = |\boldsymbol{a}_i^T\boldsymbol{x}| + w_i, \quad 1 \le i \le m, \tag{17}$$

where $\boldsymbol{w} = \{w_i\}_{i=1}^m$ denote the additive noise. Then the following theorem shows that RWF is robust under bounded noise.

Theorem 3 *Consider the model* (17). *Suppose that the measurement vectors are independently Gaussian, i.e., $\boldsymbol{a}_i \sim \mathcal{N}(0, \boldsymbol{I})$ for $1 \le i \le m$, and the noise is bounded, i.e., $\|\boldsymbol{w}\|/\sqrt{m} \le c\|\boldsymbol{x}\|$ where c is a positive constant. Then there exist some universal constants $\mu_0 > 0$ (μ_0 can be set as 0.8 in practice), $0 < \rho < 1$ and $c_0, c_1, c_2 > 0$ such that if $m \ge c_0 n$ and $\mu < \mu_0$, then with probability at least $1 - c_1\exp(-c_2 m)$, Algorithm 1 yields*

$$dist(\boldsymbol{z}^{(t)}, \boldsymbol{x}) \lesssim \frac{\|\boldsymbol{w}\|}{\sqrt{m}} + (1-\rho)^t\|\boldsymbol{x}\|, \quad \forall t \in \mathbb{N}. \tag{18}$$

Proof See Appendix D. ■

Theorem 3 shows that under the same sample complexity, RWF converges at a linear rate to a neighborhood around the true signal, whose radius is on the level of the noise. The numerical result under the Poisson noise model in Section 4 further corroborates the stability of RWF.

3. Incremental Reshaped Wirtinger Flow

In large-sample and online scenarios, stochastic algorithms are preferred due to their potential advantage of faster convergence and lower memory requirement. Thus, in this section, we develop stochastic versions of RWF, referred to as incremental reshaped Wirtinger flow (IRWF). We show that IRWF guarantees exact recovery at a linear convergence rate under the same sample complexity. We further draw the connection between IRWF and the randomized Kaczmarz method recently developed for phase retrieval (Wei, 2015; Li et al., 2015; Chi and Lu, 2016), and establish its global convergence as a side product.

3.1 (Mini-batch) IRWF: Algorithm and Convergence

In oder to fully exploit the processing throughput of CPU/GPU, we develop a mini-batch IRWF, described in Algorithm 2. The mini-batch IRWF applies the same initialization step as in RWF, and uses a mini-batch of measurements for each gradient update.

Algorithm 2 Mini-batch Incremental Reshaped Wirtinger Flow (mini-batch IRWF)

Input: $\boldsymbol{y} = \{y_i\}_{i=1}^m$, $\{\boldsymbol{a}_i\}_{i=1}^m$, mini-batch size k;
Initialization: Same as in RWF (Algorithm 1);
Gradient loop: for $t = 0 : T-1$ do
Choose Γ_t uniformly at random from the subsets of $\{1, 2, \ldots, m\}$ with the cardinality k, and let

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \mu \cdot \boldsymbol{A}_{\Gamma_t}^* \left(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)} - \boldsymbol{y}_{\Gamma_t} \odot \mathrm{Ph}(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)})\right), \tag{19}$$

where $\boldsymbol{A}_{\Gamma_t}$ is a matrix stacking $\boldsymbol{a}_i^*$ for $i \in \Gamma_t$ as its rows, $\boldsymbol{y}_{\Gamma_t}$ is a vector stacking y_i for $i \in \Gamma_t$ as its elements, and $\mathrm{Ph}(\boldsymbol{z})$ denotes the phase vector of $\boldsymbol{z}$.
Output $\boldsymbol{z}^{(T)}$.

If the gradient update uses only a single sample, i.e., $k = 1$, we refer to Algorithm 2 as IRWF, where the step (19) becomes

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \mu\left(\boldsymbol{a}_{i_t}^*\boldsymbol{z}^{(t)} - y_{i_t} \cdot \frac{\boldsymbol{a}_{i_t}^*\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_{i_t}^*\boldsymbol{z}^{(t)}|}\right)\boldsymbol{a}_{i_t}. \tag{20}$$

We characterize the convergence of mini-batch IRWF in the following theorem.

Theorem 4 *Consider the problem of solving any given $\boldsymbol{x} \in \mathbb{R}^n$ from a system of equations* (1) *with independent Gaussian measurement vectors $\boldsymbol{a}_i \sim \mathcal{N}(0, \boldsymbol{I})$. There exist some universal constants $0 < \rho, \rho_0 < 1$ and $c_0, c_1, c_2 > 0$ such that if $m \ge c_0 n$ and $\mu = \rho_0/n$ for the update rule* (19), *then with probability at least $1 - c_1\exp(-c_2 m)$, we have that*

$$\mathrm{E}_{\Gamma_t}\left[dist^2(\boldsymbol{z}^{(t+1)}, \boldsymbol{x})\right] \le \left(1 - \frac{k\rho}{n}\right) \cdot dist^2(\boldsymbol{z}^{(t)}, \boldsymbol{x}) \tag{21}$$

holds for all $\boldsymbol{z}^{(t)}$ satisfying $\frac{dist(\boldsymbol{z}^{(t)}, \boldsymbol{x})}{\|\boldsymbol{z}\|} \le \frac{1}{10}$.

Proof See Appendix E.1. ■

We suggest that $\rho_0 = 1$ and hence the step size $\mu = \frac{1}{n}$ in practice. Theorem 4 characterizes that the error decays exponentially fast in expectation if the estimate lands into the neighborhood of the global minimizer. For a generic optimization objective, it is not anticipated that incremental/stochastic first-order methods achieve linear convergence due to the variance of stochastic gradients. However, for our specific problem, the variance of stochastic gradients reduces as the estimate approaches the true signal, and hence a fixed step size can be employed and exponential decay can be established. A result similar in spirit was also established for the stochastic algorithm based on TWF (referred to as ITWF) (Kolte and Özgür, 2016). We provide further comparisons between IRWF and ITWF in Section 3.3. On the other hand, it was shown in (Moulines and Bach, 2011; Needell et al., 2016) that stochastic gradient methods yield linear convergence to the minimizer $\boldsymbol{x}_\star$ if the objective $F(\boldsymbol{x}) = \sum_i f_i(\boldsymbol{x})$ is a smooth and strongly convex function and $\boldsymbol{x}_\star$ minimizes

all components $f_i(\boldsymbol{x})$. The summands of our objective (3) also share the same minimizer (although it is neither convex nor smooth), which also helps to explain the convergence property of the IRWF.

3.2 Connection to the Kaczmarz Method for Phase Retrieval

The *Kaczmarz method* was originally developed for solving systems of linear equations (Kaczmarz, 1937). In literature (Wei, 2015; Li et al., 2015), it was adapted to solve the phase retrieval problem, which we refer to as *Kaczmarz-PR*. It has been demonstrated in Wei (2015) that Kaczmarz-PR exhibits better empirical performance than error reduction (ER) (Gerchberg, 1972; Fienup, 1982) and WF (Candès et al., 2015). However, global convergence of Kaczmarz-PR has not been well established yet, although the randomized Kaczmarz method for the least-squares problem is known to converge at a linear rate (Strohmer and Vershynin, 2009; Zouzias and Freris, 2013). For instance, Wei (2015) obtained a bound on the estimation error which can be as large as the signal energy no matter how many iterations are taken. Li et al. (2015) established the asymptotic convergence in the regime when both m and n go to infinity but their ratio is fixed.

In this section, we draw connection between IRWF and Kaczmarz-PR, which enables us to establish the theoretical guarantee of Kaczmarz-PR by adapting that of IRWF. This is analogous to the connection made in Needell et al. (2016) between the Kaczmarz method and the stochastic gradient method for solving the least-squares problem. Here, the connection is made possible due to the lower-order loss function of RWF, which was not evident in previous studies of WF and TWF.

To be more specific, the Kaczmarz-PR (Wei, 2015, Algorithm 3) employs the following update rule

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \frac{1}{\|\boldsymbol{a}_{i_t}\|^2}\left(\boldsymbol{a}_{i_t}^*\boldsymbol{z}^{(t)} - y_{i_t}\cdot\frac{\boldsymbol{a}_{i_t}^*\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_{i_t}^*\boldsymbol{z}^{(t)}|}\right)\boldsymbol{a}_{i_t}, \tag{22}$$

where i_t is selected either in a deterministic manner or randomly. We focus on the randomized case where i_t is selected uniformly at random from $\{1,\dots,m\}$.

Comparing Equation (22) and Equation (20), the update rule of Kaczmarz-PR becomes equivalent to IRWF, if we replace the step size μ by $\frac{1}{\|\boldsymbol{a}_{i_t}\|^2}$. Moreover, these two update rules are close if μ is set as suggested, i.e., $\mu = \frac{1}{n}$, because for Gaussian measurements, $\|\boldsymbol{a}_{i_t}\|^2$ concentrates around n by the law of large numbers. As we demonstrate in the numerical experiments (see Table 1), Kaczmarz-PR and IRWF have similar performance as anticipated. Thus, following the convergence guarantee for IRWF in Theorem 4, we establish the convergence guarantee for the randomized Kaczmarz-PR as follows.

Theorem 5 *Assume the measurement vectors are independent and each $\boldsymbol{a}_i \sim \mathcal{N}(0,\boldsymbol{I})$. There exist some universal constants $0<\rho<1$ and $c_0,c_1,c_2>0$ such that if $m \ge c_0 n$, then with probability at least $1-c_1 m\exp(-c_2 n)$, the randomized Kaczmarz-PR update rule* (22) *yields*

$$\mathrm{E}_{i_t}\left[dist^2(\boldsymbol{z}^{(t+1)},\boldsymbol{x})\right] \le \left(1-\frac{\rho}{n}\right)\cdot dist^2(\boldsymbol{z}^{(t)},\boldsymbol{x}) \tag{23}$$

for all $\boldsymbol{z}^{(t)}$ satisfying $\frac{dist(\boldsymbol{z}^{(t)},\boldsymbol{x})}{\|\boldsymbol{z}\|} \le \frac{1}{10}$.

Proof See Appendix E.2. ■

Theorem 5 implies that as long as the iteration sequence $\boldsymbol{z}^{(t)}$ lies in the neighborhood of the true signal, the error decays exponentially fast in expectation. However, Theorem 5 does not guarantee the full trajectory of the estimates to be in the neighborhood of the true signal. After submission of our work, more recent studies (Jeong and Gunturk, 2017; Tan and Vershynin, 2017) established the convergence of full trajectory by employing martingale theory.

Furthermore, Wei (2015) also provided a *block* Kaczmarz-PR (similar to the mini-batch version), whose update rule is given by

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \boldsymbol{A}_{\Gamma_t}^{\dagger}\left(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)} - \boldsymbol{y}_{\Gamma_t}\odot \mathrm{Ph}(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)})\right), \tag{24}$$

where Γ_t is a selected block at iterate t containing row indices, and $\dagger$ represents *Moore-Penrose pseudoinverse*, which is computed as follows:

$$\boldsymbol{A}^{\dagger} = \begin{cases} (\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*, & \text{if } \boldsymbol{A} \text{ has linearly independent columns;} \\ \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{A}^*)^{-1}, & \text{if } \boldsymbol{A} \text{ has linearly independent rows.} \end{cases} \tag{25}$$

Comparing Equation (24) and the mini-batch IRWF update in Equation (19), these two update rules are similar to each other if $\boldsymbol{A}_{\Gamma_t}\boldsymbol{A}_{\Gamma_t}^*$ approaches $\frac{n}{\rho_0}\boldsymbol{I}_{|\Gamma_t|}$. For the case with Gaussian measurements, $\boldsymbol{A}_{\Gamma_t}$ has linearly independent rows with high probability if $|\Gamma_t| \le n$ and hence $\boldsymbol{A}_{\Gamma_t}\boldsymbol{A}_{\Gamma_t}^*$ is not far from $n\boldsymbol{I}_{|\Gamma_t|}$. Our numerical experiments (see Table 1) further suggest similar convergence rates for these two algorithms with the same block/mini-batch size.

Next, we argue that for the CDP setting, block Kaczmarz-PR is the same as the mini-batch IRWF with $\mu = 1$. The CDP measurements are collected in the following form

$$\boldsymbol{y}^{(l)} = |\boldsymbol{F}\boldsymbol{D}^{(l)}\boldsymbol{x}|, \quad 1 \le l \le L, \tag{26}$$

where $\boldsymbol{F}$ represents the discrete Fourier transform (DFT) matrix, $\boldsymbol{D}^{(l)}$ denotes a diagonal matrix (mask), and L denotes the number of masks. We choose the block size $|\Gamma_t|$ to be the dimension n of the signal for the convenience of Fourier transform. Then $\boldsymbol{A}_{\Gamma_t}$ becomes the Fourier transform composed with $\boldsymbol{D}^{(l)}$ (mask effect) and $\boldsymbol{A}_{\Gamma_t}^*$ becomes $\boldsymbol{D}^{(l)*}$ multiplied by the inverse Fourier transform. Therefore, $(\boldsymbol{A}_{\Gamma_t}\boldsymbol{A}_{\Gamma_t}^*) = \boldsymbol{I}$ if the diagonal elements of $\boldsymbol{D}^{(l)}$ have unit magnitude. Taking the step size $\mu = 1$, the two algorithms are identical.

On the other hand, since the block Kaczmarz-PR needs to calculate the matrix inverse or to solve an inverse problem, the block size cannot be too large. However, mini-batch IRWF works well for a wide range of the mini-batch sizes, which can even grow with the signal dimension n as long as a batch of data is loadable into the memory.

3.3 Comparison with Incremental Truncated Wirtinger Flow (ITWF)

Recently, (Kolte and Özgür, 2016) designed and analyzed an incremental algorithm based on TWF, which is referred to as ITWF. More specifically, ITWF employs the same initialization procedure as TWF and randomly chooses one sample with the index i_t selected uniformly

at random from $\{1, 2, \ldots, m\}$ for the gradient update as follows:

$$\boldsymbol{z}^{(t+1)} = \boldsymbol{z}^{(t)} - \mu \cdot \frac{|\boldsymbol{a}_{i_t}^T \boldsymbol{z}|^2 - y_{i_t}^2}{\boldsymbol{a}_{i_t}^T \boldsymbol{z}} \boldsymbol{a}_{i_t} \mathbf{1}_{\mathcal{E}_{1,t}^{i_t} \cap \mathcal{E}_3^{i_t}}, \tag{27}$$

where $\mathbf{1}_{\mathcal{E}_{1,t}^{i_t} \cap \mathcal{E}_3^{i_t}}$ represents the truncation rule determined by the events $\mathcal{E}_{1,t}^{i_t}$ and $\mathcal{E}_3^{i_t}$. As a comparison, the update rule of IRWF is much simpler due to the use of the lower-order loss function and does not require any truncation in the gradient loop. Kolte and Özgür (2016) proved that the update (27) shrinks the estimate error as long as m/n is large enough and the estimate $\boldsymbol{z}$. Compared to ITWF, the IRWF update (19) also shrinks the estimate error, but runs faster than ITWF numerically as demonstrated in Section 4.

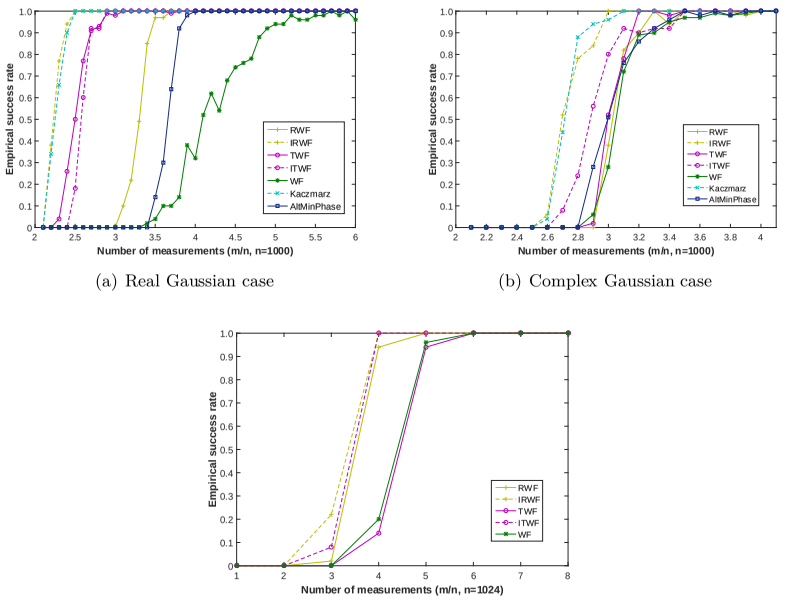
4. Numerical Experiments

In this section, we demonstrate the numerical efficiency of RWF and (mini-batch) IRWF by comparing their performance with other competitive algorithms. Our experiments are conducted not only for the real Gaussian case but also for the complex Gaussian and the CDP cases. All the experiments are implemented in Matlab 2015b and conducted on a computer equipped with Intel Core i7 3.4GHz CPU and 12GB RAM.

We first compare the sample complexity of RWF and IRWF with those of TWF, WF, Kaczmarz-PR and AltMinPhase via the empirical successful recovery rate versus the number of measurements. For RWF, we follow Algorithm 1 with the suggested parameters. For IRWF, we adopt a block size 64 for efficiency and set the step size $\mu = 1/n$. For WF, TWF, we use the code provided in the original papers with the suggested parameters. For ITWF, we also adopt a block size 64 and set the step size $\mu = 0.6/n$ (optimal step size). We conduct the experiments for the real Gaussian, complex Gaussian and CDP cases respectively. For the real and complex cases, we set the signal dimension n to be 1000, and set the ratio m/n to take values from 2 to 6 with a step 0.1. For each m, we run 100 trials and count the number of successful trials. For each trial, we run a maximal number of iterations/passes $T = 10000$ for all algorithms, and a trial is declared to be successful whenever the iterate satisfies $\text{dist}(\boldsymbol{z}^{(T)}, \boldsymbol{x})/\|\boldsymbol{x}\| \leq 10^{-5}$. For the real Gaussian case, we generate the signal $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{n\times n})$, and generate the measurement vectors $\boldsymbol{a}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{n\times n})$ i.i.d. for $i = 1, \ldots, m$. For the complex Gaussian case, we generate the signal $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{I}_{n\times n}) + j\mathcal{N}(0, \boldsymbol{I}_{n\times n})$ and the measurement vectors $\boldsymbol{a}_i \sim \frac{1}{2}\mathcal{N}(0, \boldsymbol{I}_{n\times n}) + j\frac{1}{2}\mathcal{N}(0, \boldsymbol{I}_{n\times n})$ i.i.d. for $i = 1, \ldots, m$. For the CDP case (26), we set $n = 1024$ for the convenience of FFT and $m/n = L = 1, 2, \ldots, 8$. All other settings are the same as those for the real case.

We note that for the CDP case, the Kaczmarz-PR algorithm is identical to the IRWF with step size $\mu = 1/n$ due to the argument in Section 3.2. Moreover under the CDP case, the AltMinPhase algorithm is identical to the RWF with step size $\mu = 1$ because the inverse of the Fourier measurement matrix is nothing but its conjugate transpose. In the following experiments for the CDP case, we choose the step size $\mu = 1/n$ for the IRWF and $\mu = 1$ for the RWF, under which the Kaczmarz-PR algorithm coincides with the IRWF and the AltMinPhase algorithm coincides with the RWF.

Figure 4 plots the fraction of successful trials out of 100 trials for all algorithms, with respect to m/n. It can be seen that IRWF and Kaczmarz-PR exhibit a similar sample



(a) Real Gaussian case

(b) Complex Gaussian case

(c) CDP case

Figure 4: Comparison of sample complexity among RWF, IRWF, TWF, ITWF, WF, Karzmarz-PR and AltMinPhase.

complexity, which is the best for all three cases, and is close to the theoretical limit (Bandeira et al., 2014). It can also be seen that the two incremental methods (IRWF and ITWF) outperform the batch methods (RWF, TWF, AltMinPhase and WF). This can be due to the inherent noise in incremental methods, which helps to escape bad local minima. This can be extremely helpful in the regime with a small number of samples, where local minima do exist near the global minima. Comparing among the batch methods (RWF, TWF AltMinPhase and WF), it can be seen that although RWF outperforms only WF and AltMinPhase (not TWF) for the real Gaussian case, it has a comparable performance for the complex case and outperforms TWF and WF in the CDP case. An intuitive explanation for the real case is that a substantial number of samples with small $|\boldsymbol{a}_i^T \boldsymbol{z}|$ can deviate the gradient direction so that truncation indeed helps to stabilize the algorithm if the number of measurements is not large.

We next compare the convergence rate of RWF, IRWF with those of TWF, ITWF, WF, Kaczmarcz and AltMinPhase. We run all of the algorithms with the suggested parameters in the original code. We generate the signal and measurements in the same way as those

Table 1: Comparison of iteration count and time cost among algorithms ($n = 5000, m = 8n$).

		Real Gaussian		Complex Gaussian	
		#passes	time(s)	# passes	time(s)
Batch methods	RWF	72	**12.66**	176	**122.4**
	TWF	186	32.36	487	395.1
	WF	319	54.83	932	887.8
	AltMinPhase	**6**	79.58	**159**	9637
Incremental methods	IRWF	9	44.77	21	233.2
	mini-batch IRWF (64)	9	**8.076**	**21**	**48.58**
	mini-batch ITWF (64)	16	37.38	29	149.5
	Kaczmarz-PR	9	50.68	21	248.4
	block Kaczmarz-PR (64)	**8**	28.50	22	89.31

in the first experiment with $n = 5000, m = 8n$. All algorithms are seeded with the RWF initialization. In Table 1, we list the number of passes and the time cost for all the algorithms to achieve a relative error of 10^{-14} averaged over 10 trials. For the incremental methods, one update passes k samples and one pass amounts to m/k updates. Clearly, IRWF with mini-batch size 64 runs the fastest for both the real and complex cases. Moreover, among the batch (deterministic) algorithms, RWF takes much fewer passes as well as runs much faster than TWF and WF. Although RWF takes more iterations than AltMinPhase, it runs much faster than AltMinPhase due to the fact that each iteration of AltMinPhase needs to solve a least-squares problem that takes much longer than a simple gradient update in RWF.

We also compare the performance of the above algorithms on the recovery of a real image from the Fourier intensity measurements (the two dimensional CDP case). The image (see Figure 5) is the Milky Way Galaxy with resolution 1920×1080. Table 2 lists the number of passes and the time cost for the above six algorithms to achieve the relative error of 10^{-15} for one R/G/B channel. All algorithms are seeded with the RWF initialization. To explore the advantage of FFT, we run the incremental/stochastic methods with the mini-batch size equal to the number of pixels for one R/G/B channel. We note that with such a mini-batch size, IRWF is equivalent to block Kaczmarz-PR from the discussion in Section 3.2. It can be seen that in general, the incremental/stochastic methods (IRWF and ITWF) run faster than the batch methods (RWF, TWF, WF). Moreover, among the batch methods, RWF outperforms the other three algorithms in both the number of passes and the computational time. In particular, RWF runs two times faster than TWF and six times faster than WF in terms of both the number of iterations and the computational time.

We next demonstrate the robustness of RWF to noise and compare it with TWF. We consider the phase retrieval problem in imaging applications, where Poisson noise is often used to model the sensor and electronic noise (Fogel et al., 2016). Specifically, the noisy measurements of intensity can be expressed as $y_i = \sqrt{\alpha \cdot \text{Poisson}\left(|\boldsymbol{a}_i^T \boldsymbol{x}|^2/\alpha\right)}$, for



Figure 5: Milky way Galaxy.

Table 2: Comparison of iterations and time cost among algorithms on recovery of Galaxy image (shown in Figure 5), where $L = m/n$ denotes the number of CDP masks.

	Algorithms	RWF	IRWF	TWF	ITWF	WF
$L = 6$	#passes	140	**24**	410	41	fail
	time cost(s)	110	**21.2**	406	43	fail
$L = 12$	#passes	70	**8**	190	12	315
	time cost(s)	107	**13.7**	363.6	25.9	426

$i = 1, 2, ...m$ where α denotes the level of the input noise, and Poisson(λ) denotes a random sample generated by the Poisson distribution with mean λ. It can be observed from Figure 6 that RWF performs better than TWF in terms of the recovery accuracy under two different noise levels.

5. Conclusion

In this paper, we proposed RWF and its incremental version IRWF to recover a signal for the phase retrieval problem, based on a *nonconvex and nonsmooth* quadratic loss function of magnitude measurements. This loss function sacrifices the smoothness but enjoys advantages in statistical and computational efficiency. It has potential to be extended in various scenarios. One interesting direction is to extend such an algorithm to exploit signal structures (e.g., non-negativity, sparsity, etc) to assist the recovery. The lower-order loss function may offer great simplicity to prove the performance guarantee in such cases.

Another interesting direction is to study the convergence of algorithms from random initialization. In the regime with a large sample size ($m \gg n$), the empirical loss surface

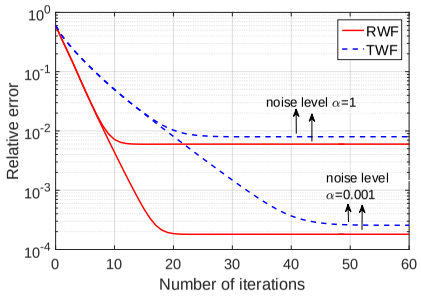


Figure 6: Comparison of relative error under Poisson noise between RWF and TWF.

approaches the asymptotic loss Figure 1(b) and hence has no spurious local minima. Due to the result (Lee et al., 2016a), it is conceivable that the gradient descent algorithm converges from any random starting point. Similar phenomena have been observed in (Sun et al., 2016; Ge et al., 2016). However, under a moderate number of measurements ($m < 10n$), we observe that genuine local minima do exist and often locate not far from the global minima. In such a regime, the batch gradient method often fails with random initialization. As anticipated, stochastic algorithms are efficient in escaping bad local minima or saddle points in nonconvex optimization because of the inherent noise (Ge et al., 2015; Sa et al., 2015). We observe numerically that IRWF and block IRWF from a random starting point still converge to a global minimum even with a very small sample size which is close to the theoretical limit. It is of interest to analyze theoretically why stochastic methods escape these local minima (not just saddle points) efficiently.

Acknowledgments

The work of H. Zhang, Y. Zhou and Y. Liang is supported in part by the grants NSF CCF-1704169, NSF ECCS-1609916 and AFOSR FA9550-16-1-0077. The work of Y. Chi is supported in part by the grants AFOSR under FA9550-15-1-0205, ONR N00014-15-1-2387, NSF CCF-1704245, and ECCS-1650449.

Appendices

Appendix A. Expected Loss Surfaces

It is easy to check that the expectation of the least-squares loss is given by $\mathrm{E}[\ell_{LS}(\boldsymbol{z})] = \|\boldsymbol{z}-\boldsymbol{x}\|^2$. The expectation of the WF loss function (2) is given by (Sun et al., 2016) as

$$\mathrm{E}[\ell_{WF}(\boldsymbol{z})] = \frac{3}{4}\|\boldsymbol{x}\|^4 + \frac{3}{4}\|\boldsymbol{z}\|^4 - \frac{1}{2}\|\boldsymbol{x}\|^2\|\boldsymbol{z}\|^2 - |\boldsymbol{z}^T\boldsymbol{x}|^2. \tag{28}$$

We next show that the expectation of the RWF loss function (3) has the following form:

$$\mathrm{E}[\ell(\boldsymbol{z})] = \frac{1}{2}\|\boldsymbol{x}\|^2 + \frac{1}{2}\|\boldsymbol{z}\|^2 - \|\boldsymbol{x}\|\|\boldsymbol{z}\| \cdot \mathrm{E}\left[\frac{|\boldsymbol{a}_i^T\boldsymbol{z}|}{\|\boldsymbol{z}\|} \cdot \frac{|\boldsymbol{a}_i^T\boldsymbol{x}|}{\|\boldsymbol{x}\|}\right], \tag{29}$$

where

$$\mathrm{E}\left[\frac{|\boldsymbol{a}_i^T\boldsymbol{z}|}{\|\boldsymbol{z}\|} \cdot \frac{|\boldsymbol{a}_i^T\boldsymbol{x}|}{\|\boldsymbol{x}\|}\right] = \begin{cases} \frac{(1-\rho^2)^{3/2}}{\pi}\int_0^\infty t(e^{\rho t}+e^{-\rho t})K_0(t)dt, & \text{if } |\rho|<1; \\ 1, & \text{if } |\rho|=1; \end{cases} \tag{30}$$

where $\rho = \frac{\boldsymbol{z}^T\boldsymbol{x}}{\|\boldsymbol{x}\|\|\boldsymbol{z}\|}$ and $K_0(\cdot)$ is the modified Bessel function of the second kind.

In order to derive (30), we first define

$$u := \frac{\boldsymbol{a}_i^T\boldsymbol{z}}{\|\boldsymbol{z}\|} \text{ and } v := \frac{\boldsymbol{a}_i^T\boldsymbol{x}}{\|\boldsymbol{x}\|},$$

and it suffices to drive $\mathrm{E}[|uv|]$. Note that $(u,v) \sim \mathcal{N}(0,\Sigma)$, where

$$\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}, \quad \text{and} \quad \rho = \frac{\boldsymbol{z}^T\boldsymbol{x}}{\|\boldsymbol{x}\|\|\boldsymbol{z}\|}.$$

Following (Donahue, 1964), the density function of uv is given by

$$\phi_{uv}(x) = \frac{1}{\pi\sqrt{1-\rho^2}}\exp\left(\frac{\rho x}{1-\rho^2}\right)K_0\left(\frac{|x|}{1-\rho^2}\right), \quad x \neq 0.$$

Thus, the density function of $|uv|$ is given by

$$\psi_{|uv|}(x) = \frac{1}{\pi\sqrt{1-\rho^2}}\left[\exp\left(\frac{\rho x}{1-\rho^2}\right)+\exp\left(-\frac{\rho x}{1-\rho^2}\right)\right]K_0\left(\frac{|x|}{1-\rho^2}\right), \quad x>0, \tag{31}$$

for $|\rho|<1$. Therefore, if $|\rho|<1$, we have

$$\begin{aligned}\mathrm{E}[|uv|] &= \int_0^\infty x\cdot\psi_\rho(x)dx \\ &= \int_0^\infty x\cdot\frac{1}{\pi\sqrt{1-\rho^2}}\left[\exp\left(\frac{\rho x}{1-\rho^2}\right)+\exp\left(-\frac{\rho x}{1-\rho^2}\right)\right]K_0\left(\frac{|x|}{1-\rho^2}\right)dx \\ &= \frac{(1-\rho^2)^{3/2}}{\pi}\int_0^\infty t(e^{\rho t}+e^{-\rho t})K_0(t)dt,\end{aligned}$$

where the last step follows from change of variables.

If $|\rho| = 1$, then $|uv|$ becomes a χ_1^2 random variable, with the density given by

$$\psi_{|uv|}(x) = \frac{1}{\sqrt{2\pi}} x^{-1/2} \exp(-x/2), \quad x > 0,$$

and hence $\mathrm{E}[|uv|] = 1$.

Appendix B. Proof of Proposition 1

The idea of using truncation to bound some non-sub-Gaussian sequences has appeared in the literature (Candès et al., 2013, Lemma 2.3) and (Chen and Candès, 2015). Compared to the proof of the initialization for TWF (Chen and Candès, 2015), new technical developments are needed to address the magnitude measurements and truncation from both sides.

We first estimate the norm of $\boldsymbol{x}$ as

$$\lambda_0 = \frac{mn}{\sum_{i=1}^m \|\boldsymbol{a}_i\|_1} \cdot \left(\frac{1}{m} \sum_{i=1}^m y_i \right). \tag{32}$$

Since $\boldsymbol{a}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{\mathbf{n}\times\mathbf{n}})$, by Hoeffding-type inequality, it can be shown that

$$\left| \frac{\sum_{i=1}^m \|\boldsymbol{a}_i\|_1}{mn} - \sqrt{\frac{2}{\pi}} \right| < \frac{\epsilon}{3} \tag{33}$$

holds with probability at least $1 - 2\exp(-c_1 mn\epsilon^2)$ for some constant $c_1 > 0$.

Moreover, for a fixed $\boldsymbol{x}$, y_i's are independent sub-Gaussian random variables. Thus, by Hoeffding-type inequality, it can be shown that

$$\left| \sqrt{\frac{\pi}{2}} \left(\frac{1}{m} \sum_{i=1}^m y_i \right) - \|\boldsymbol{x}\| \right| < \frac{\epsilon}{3} \|\boldsymbol{x}\| \tag{34}$$

holds with probability at least $1 - 2\exp(-c_1 m\epsilon^2)$ for some constant $c_1 > 0$.

On the event $E_1 = \{\text{both (33) and (34) hold}\}$, it can be argued that

$$|\lambda_0 - \|\boldsymbol{x}\|| < \epsilon \|\boldsymbol{x}\|. \tag{35}$$

Without loss of generality, we let $\|\boldsymbol{x}\| = 1$. Then on the event E_1, the truncation function satisfies the following bounds

$$\mathbf{1}_{\{\alpha_l(1+\epsilon) < |\boldsymbol{a}_i^T \boldsymbol{x}| < \alpha_u(1-\epsilon)\}} \le \mathbf{1}_{\{\alpha_l \lambda_0 < y_i < \alpha_u \lambda_0\}} \le \mathbf{1}_{\{\alpha_l(1-\epsilon) < |\boldsymbol{a}_i^T \boldsymbol{x}| < \alpha_u(1+\epsilon)\}}.$$

Thus, define

$$\begin{aligned}
\boldsymbol{Y}_1 :=& \frac{1}{m} \sum_{i=1}^m \boldsymbol{a}_i \boldsymbol{a}_i^T |\boldsymbol{a}_i^T \boldsymbol{x}| \mathbf{1}_{\{\alpha_l(1+\epsilon) < |\boldsymbol{a}_i^T \boldsymbol{x}| < \alpha_u(1-\epsilon)\}} \\
\boldsymbol{Y}_2 :=& \frac{1}{m} \sum_{i=1}^m \boldsymbol{a}_i \boldsymbol{a}_i^T |\boldsymbol{a}_i^T \boldsymbol{x}| \mathbf{1}_{\{\alpha_l(1-\epsilon) < |\boldsymbol{a}_i^T \boldsymbol{x}| < \alpha_u(1+\epsilon)\}},
\end{aligned}$$

and we have $\boldsymbol{Y}_1 \prec \boldsymbol{Y} \prec \boldsymbol{Y}_2$. We further compute the expectations of $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ and obtain

$$\mathrm{E}[\boldsymbol{Y}_1] = (\beta_1 \boldsymbol{x}\boldsymbol{x}^T + \beta_2 \boldsymbol{I}), \quad \mathrm{E}[\boldsymbol{Y}_2] = (\beta_3 \boldsymbol{x}\boldsymbol{x}^T + \beta_4 \boldsymbol{I}), \tag{36}$$

where

$$\begin{aligned}
\beta_1 &:= \mathrm{E}[|\xi|^3 \mathbf{1}_{\{\alpha_l(1+\epsilon) < |\xi| < \alpha_u(1-\epsilon)\}}] - \mathrm{E}[|\xi| \mathbf{1}_{\{\alpha_l(1+\epsilon) < |\xi| < \alpha_u(1-\epsilon)\}}], \\
\beta_2 &:= \mathrm{E}[|\xi| \mathbf{1}_{\{\alpha_l(1+\epsilon) < |\xi| < \alpha_u(1-\epsilon)\}}] \\
\beta_3 &:= \mathrm{E}[|\xi|^3 \mathbf{1}_{\{\alpha_l(1-\epsilon) < |\xi| < \alpha_u(1+\epsilon)\}}] - \mathrm{E}[|\xi| \mathbf{1}_{\{\alpha_l(1-\epsilon) < |\xi| < \alpha_u(1+\epsilon)\}}], \\
\beta_4 &:= \mathrm{E}[|\xi| \mathbf{1}_{\{\alpha_l(1-\epsilon) < |\xi| < \alpha_u(1+\epsilon)\}}]
\end{aligned}$$

where $\xi \sim \mathcal{N}(0,1)$. For given α_l and α_u, a small value of ϵ yields arbitrarily close β_1 and β_3, as well as arbitrarily close β_2 and β_4. For example, taking $\alpha_l = 1, \alpha_u = 5$ and $\epsilon = 0.01$, we have $\beta_1 = 0.9678, \beta_2 = 0.4791, \beta_3 = 0.9688, \beta_4 = 0.4888$.

Now applying the standard results on random matrices with non-isotropic sub-Gaussian rows (Vershynin, 2012, equation (5.26)) and noticing that $\boldsymbol{a}_i \boldsymbol{a}_i^T |\boldsymbol{a}_i^T \boldsymbol{x}| \mathbf{1}_{\{\alpha_l(1+\epsilon) < |\boldsymbol{a}_i^T \boldsymbol{x}| < \alpha_u(1-\epsilon)\}}$ can be rewritten as $\boldsymbol{b}_i \boldsymbol{b}_i^T$ for a sub-Gaussian vector $\boldsymbol{b}_i := \boldsymbol{a}_i \sqrt{|\boldsymbol{a}_i^T \boldsymbol{x}|} \mathbf{1}_{\{\alpha_l(1+\epsilon) < |\boldsymbol{a}_i^T \boldsymbol{x}| < \alpha_u(1-\epsilon)\}}$, one can derive

$$\|\boldsymbol{Y}_1 - \mathrm{E}[\boldsymbol{Y}_1]\| \le \delta, \quad \|\boldsymbol{Y}_2 - \mathrm{E}[\boldsymbol{Y}_2]\| \le \delta \tag{37}$$

with probability at least $1 - 4\exp(-c_1(\delta)m)$ for some positive c_1 which is only affected by δ, provided that m/n exceeds a certain constant. Furthermore, when ϵ is sufficiently small, one further has $\|\mathrm{E}[\boldsymbol{Y}_1] - \mathrm{E}[\boldsymbol{Y}_2]\| \le \delta$. Combining the above facts together, one can show that

$$\|\boldsymbol{Y} - (\beta_1 \boldsymbol{x}\boldsymbol{x}^T + \beta_2 \boldsymbol{I})\| \le 3\delta. \tag{38}$$

Let $\tilde{\boldsymbol{z}}^{(0)}$ be the normalized leading eigenvector of $\boldsymbol{Y}$. Following the arguments in (Candès et al., 2015, Section 7.8) and taking δ and ϵ to be sufficiently small, one has

$$\mathrm{dist}(\tilde{\boldsymbol{z}}^{(0)}, \boldsymbol{x}) \le \tilde{\delta}, \tag{39}$$

for a given $\tilde{\delta} > 0$, as long as m/n exceeds a certain constant.

Appendix C. Proof of Theorem 2

The general structure of the proof follows that for WF in (Candès et al., 2015) and TWF in (Chen and Candès, 2015). However, the proof requires the development of new bounds due to the nonsmoothness of the loss function and the magnitude measurements. On the other hand, the proof is much simpler due to the lower-order loss function adopted in RWF.

The idea of the proof is to show that within the neighborhood of the global optimizers, RWF satisfies the *Regularity Condition* $\mathsf{RC}(\mu, \lambda, c)$, i.e.,

$$\langle \nabla \ell(\boldsymbol{z}), \boldsymbol{h} \rangle \ge \frac{\mu}{2} \|\nabla \ell(\boldsymbol{z})\|^2 + \frac{\lambda}{2} \|\boldsymbol{h}\|^2 \tag{40}$$

for all $\boldsymbol{z}$ and $\boldsymbol{h} = \boldsymbol{z} - \boldsymbol{x}$ obeying $\|\boldsymbol{h}\| \le c\|\boldsymbol{x}\|$, where $0 < c < 1$ is some constant. Then, as shown in (Chen and Candès, 2015), once the initialization lands into the neighborhood that satisfies $\|\boldsymbol{z} - \boldsymbol{x}\| \le c\|\boldsymbol{x}\|$, linear convergence can be guaranteed, i.e.,

$$\mathrm{dist}^2\left(\boldsymbol{z} - \mu\nabla\ell(\boldsymbol{z}), \boldsymbol{x}\right) \le (1 - \mu\lambda)\mathrm{dist}^2(\boldsymbol{z}, \boldsymbol{x}), \tag{41}$$

for any $\boldsymbol{z}$ with $\|\boldsymbol{z} - \boldsymbol{x}\| \le c\|\boldsymbol{x}\|$.

To show the regularity condition, we first define a set $\mathcal{S} := \{i : 1 \le i \le m, (\boldsymbol{a}_i^T\boldsymbol{z})(\boldsymbol{a}_i^T\boldsymbol{x}) < 0\}$ (which depends on $\boldsymbol{x}$ and $\boldsymbol{z}$), and then derive the following bound:

$$\begin{aligned}
\langle\nabla\ell(\boldsymbol{z}), \boldsymbol{h}\rangle &= \frac{1}{m}\sum_{i=1}^{m}\left(\boldsymbol{a}_i^T\boldsymbol{z} - |\boldsymbol{a}_i^T\boldsymbol{x}|\mathrm{sgn}(\boldsymbol{a}_i^T\boldsymbol{z})\right)(\boldsymbol{a}_i^T\boldsymbol{h}) \\
&= \frac{1}{m}\left[\sum_{i=1}^{m}(\boldsymbol{a}_i^T\boldsymbol{h})^2 + 2\sum_{i\in\mathcal{S}}(\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{h})\right] \\
&\ge \frac{1}{m}\left[\sum_{i=1}^{m}(\boldsymbol{a}_i^T\boldsymbol{h})^2 - 2\left|\sum_{i\in\mathcal{S}}(\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{h})\right|\right] \\
&\ge \frac{1}{m}\left[\sum_{i=1}^{m}(\boldsymbol{a}_i^T\boldsymbol{h})^2 - \sum_{i\in\mathcal{S}}2\left|(\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{h})\right|\right].
\end{aligned} \tag{42}$$

The first term in (42) can be bounded using Lemma 3.1 in (Candès et al., 2013), which we state below.

Lemma 6 *For any $0 < \epsilon < 1$, there exist constants $c_0, c_1 > 0$ such that if $m > c_0 n\epsilon^{-2}$, then with probability at least $1 - 2\exp(-c_1\epsilon^2 m)$,*

$$(1 - \epsilon)\|\boldsymbol{h}\|^2 \le \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{a}_i^T\boldsymbol{h})^2 \le (1 + \epsilon)\|\boldsymbol{h}\|^2 \tag{43}$$

holds for all non-zero vectors $\boldsymbol{h} \in \mathbb{R}^n$.

For the second term in (42), we derive

$$\begin{aligned}
\sum_{i\in\mathcal{S}}2\left|\boldsymbol{a}_i^T\boldsymbol{x}\right|\left|\boldsymbol{a}_i^T\boldsymbol{h}\right| &\le \sum_{i\in\mathcal{S}}\left[(\boldsymbol{a}_i^T\boldsymbol{x})^2 + (\boldsymbol{a}_i^T\boldsymbol{h})^2\right] \\
&= \sum_{i=1}^{m}[(\boldsymbol{a}_i^T\boldsymbol{x})^2 + (\boldsymbol{a}_i^T\boldsymbol{h})^2]\cdot\mathbf{1}_{\{(\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{z})<0\}} \\
&= \sum_{i=1}^{m}[(\boldsymbol{a}_i^T\boldsymbol{x})^2 + (\boldsymbol{a}_i^T\boldsymbol{h})^2]\cdot\mathbf{1}_{\{(\boldsymbol{a}_i^T\boldsymbol{x})^2 + (\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{h})<0\}} \\
&\le \sum_{i=1}^{m}[(\boldsymbol{a}_i^T\boldsymbol{x})^2 + (\boldsymbol{a}_i^T\boldsymbol{h})^2]\cdot\mathbf{1}_{\{|\boldsymbol{a}_i^T\boldsymbol{x}|<|\boldsymbol{a}_i^T\boldsymbol{h}|\}} \\
&\le 2\sum_{i=1}^{m}(\boldsymbol{a}_i^T\boldsymbol{h})^2\cdot\mathbf{1}_{\{|\boldsymbol{a}_i^T\boldsymbol{x}|<|\boldsymbol{a}_i^T\boldsymbol{h}|\}}.
\end{aligned} \tag{44}$$

The right hand side of the above equation can be further upper bounded by the following lemma.

Lemma 7 *For any $\epsilon > 0$, there exist constants $c_0, c_1, C > 0$ such that if $m > c_0 n\epsilon^{-2}\log\epsilon^{-1}$, then with probability at least $1 - C\exp(-c_1\epsilon^2 m)$,*

$$\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{a}_i^T\boldsymbol{h})^2\cdot\mathbf{1}_{\{|\boldsymbol{a}_i^T\boldsymbol{x}|<|\boldsymbol{a}_i^T\boldsymbol{h}|\}} \le (0.13 + \epsilon)\|\boldsymbol{h}\|^2 \tag{45}$$

holds for all non-zero vectors $\boldsymbol{h} \in \mathbb{R}^n$ satisfying $\|\boldsymbol{h}\| \le \frac{1}{10}\|\boldsymbol{x}\|$.

Proof See Section C.1. ■

Therefore, combining Lemmas 6 and 7 with (42) yields

$$\langle\nabla\ell(\boldsymbol{z}), \boldsymbol{h}\rangle \ge (1 - 0.26 - 3\epsilon)\|\boldsymbol{h}\|^2 = (0.74 - 3\epsilon)\|\boldsymbol{h}\|^2. \tag{46}$$

We further provide an upper bound on $\|\nabla\ell(\boldsymbol{z})\|$ in the following lemma.

Lemma 8 *Fix $\delta > 0$. There exist constatnts $c_0, c, C > 0$ such that if $m \ge c_0 n$ then with probability at least $1 - C\exp(-cm)$,*

$$\|\nabla\ell(\boldsymbol{z})\| \le (1 + \delta)\cdot 2\|\boldsymbol{h}\| \tag{47}$$

holds for all non-zero vectors $\boldsymbol{h}, \boldsymbol{z} \in \mathbb{R}^n$ satisfying $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{h}$ and $\|\boldsymbol{h}\| \le \frac{1}{10}\|\boldsymbol{x}\|$.

Proof See Section C.2. ■

Thus, applying Lemma 8 and (46) to (40), we conclude that *Regularity Condition* holds for μ and λ satisfying

$$0.74 - 3\epsilon \ge \frac{\mu}{2}\cdot 4(1 + \delta)^2 + \frac{\lambda}{2}, \tag{48}$$

which concludes the proof.

C.1 Proof of Lemma 7

We first prove bounds for any fixed $\boldsymbol{h} \le \frac{1}{10}\|\boldsymbol{x}\|$, and then develop a uniform bound later on. We introduce a series of auxiliary random Lipschitz functions to approximate the indicator functions. For $i = 1, \ldots, m$, define

$$\chi_i(t) := \begin{cases} t, & \text{if } t > (\boldsymbol{a}_i^T\boldsymbol{x})^2; \\ \frac{1}{\delta}(t - (\boldsymbol{a}_i^T\boldsymbol{x})^2) + (\boldsymbol{a}_i^T\boldsymbol{x})^2, & \text{if } (1-\delta)(\boldsymbol{a}_i^T\boldsymbol{x})^2 \le t \le (\boldsymbol{a}_i^T\boldsymbol{x})^2; \\ 0, & \text{else;} \end{cases} \tag{49}$$

and then $\chi_i(t)$'s are random Lipschitz functions with Lipschitz constant $\frac{1}{\delta}$. We further have

$$|\boldsymbol{a}_i^T\boldsymbol{h}|^2\mathbf{1}_{\{|\boldsymbol{a}_i^T\boldsymbol{x}|<|\boldsymbol{a}_i^T\boldsymbol{h}|\}} \le \chi_i(|\boldsymbol{a}_i^T\boldsymbol{h}|^2) \le |\boldsymbol{a}_i^T\boldsymbol{h}|^2\mathbf{1}_{\{(1-\delta)|\boldsymbol{a}_i^T\boldsymbol{x}|^2<|\boldsymbol{a}_i^T\boldsymbol{h}|^2\}}. \tag{50}$$

For convenience, we denote $\gamma_i := \frac{|\boldsymbol{a}_i^T\boldsymbol{h}|^2}{\|\boldsymbol{h}\|^2}\mathbf{1}_{\{(1-\delta)|\boldsymbol{a}_i^T\boldsymbol{x}|^2<|\boldsymbol{a}_i^T\boldsymbol{h}|^2\}}$ and $\theta := \|\boldsymbol{h}\|/\|\boldsymbol{x}\|$. We next estimate the expectation of γ_i, by the following conditional expectation,

$$\mathrm{E}[\gamma_i] = \int_\Omega \gamma_i d\mathbb{P} = \iint_{-\infty}^{\infty} \mathrm{E}\left[\gamma_i | \boldsymbol{a}_i^T\boldsymbol{x} = \tau_1\|\boldsymbol{x}\|, \boldsymbol{a}_i^T\boldsymbol{h} = \tau_2\|\boldsymbol{h}\|\right] \cdot f(\tau_1,\tau_2)d\tau_1 d\tau_2, \tag{51}$$

where $f(\tau_1,\tau_2)$ is the density of two joint Gaussian random variables with the correlation $\rho = \frac{\boldsymbol{h}^T\boldsymbol{x}}{\|\boldsymbol{h}\|\|\boldsymbol{x}\|} \neq \pm 1$. We then continue to derive

$$\begin{aligned}
\mathrm{E}[\gamma_i] &= \iint_{-\infty}^{\infty} \tau_2^2 \cdot \mathbf{1}_{\{\sqrt{1-\delta}|\tau_1|<|\tau_2|\theta\}} \cdot f(\tau_1,\tau_2)d\tau_1 d\tau_2 \\
&= \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\infty}^{\infty} \tau_2^2 \exp\left(-\frac{\tau_2^2}{2}\right) \cdot \int_{\frac{-|\tau_2|\theta}{\sqrt{1-\delta}}}^{\frac{|\tau_2|\theta}{\sqrt{1-\delta}}} \exp\left(-\frac{(\tau_1-\rho\tau_2)^2}{2(1-\rho^2)}\right) d\tau_1 d\tau_2 \qquad (52) \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \tau_2^2 \exp\left(-\frac{\tau_2^2}{2}\right) \cdot \int_{\frac{-\frac{|\tau_2|\theta}{\sqrt{1-\delta}}-\rho\tau_2}{\sqrt{1-\rho^2}}}^{\frac{\frac{|\tau_2|\theta}{\sqrt{1-\delta}}-\rho\tau_2}{\sqrt{1-\rho^2}}} \exp\left(-\frac{\tau^2}{2}\right) d\tau d\tau_2 \qquad \text{by change of variables} \\
&= \frac{1}{2\pi} \int_{-\infty}^{\infty} \tau_2^2 \exp\left(-\frac{\tau_2^2}{2}\right) \cdot \sqrt{\frac{\pi}{2}} \left(\mathrm{erf}\left(\frac{\frac{|\tau_2|\theta}{\sqrt{1-\delta}}-\rho\tau_2}{\sqrt{1-\rho^2}}\right) - \mathrm{erf}\left(\frac{-\frac{|\tau_2|\theta}{\sqrt{1-\delta}}-\rho\tau_2}{\sqrt{1-\rho^2}}\right)\right) d\tau_2 \\
&= \frac{1}{\sqrt{2\pi}} \int_0^{\infty} \tau_2^2 \exp\left(-\frac{\tau_2^2}{2}\right) \cdot \left(\mathrm{erf}\left(\frac{(\frac{\theta}{\sqrt{1-\delta}}-\rho)\tau_2}{\sqrt{1-\rho^2}}\right) + \mathrm{erf}\left(\frac{(\frac{\theta}{\sqrt{1-\delta}}+\rho)\tau_2}{\sqrt{1-\rho^2}}\right)\right) d\tau_2. \qquad (53)
\end{aligned}$$

For $|\rho|<1$, $\mathrm{E}[\gamma_i]$ is a continuous function of ρ. For $|\rho|=1$, $\mathrm{E}[\gamma_i]=0$. The last integral (53) can be calculated numerically. Figure 7 plots $\mathrm{E}[\gamma_i]$ for $\theta=0.1$ and $\delta=0.01$ over $\rho\in[-1,1]$. Furthermore, (52) indicates that $\mathrm{E}[\gamma_i]$ is monotonically increasing with both θ and δ. Thus, we obtain a universal bound

$$\mathrm{E}[\gamma_i] \leq 0.13 \quad \text{for } \theta<0.1 \text{ and } \delta = 0.01, \tag{54}$$

which implies $\mathrm{E}[\chi_i(|\boldsymbol{a}_i^T\boldsymbol{h}|^2)] \leq 0.13\|\boldsymbol{h}\|^2$ for $\theta<0.1$ and $\delta=0.01$.

Furthermore, $\chi_i(|\boldsymbol{a}_i^T\boldsymbol{h}|^2)$'s are sub-exponential with the sub-exponential norm $\mathcal{O}(\|\boldsymbol{h}\|^2)$. By the sub-exponential tail bound (Bernstein type) (Vershynin, 2012), we have

$$\mathbb{P}\left[\frac{1}{m}\sum_{i=1}^m \frac{\chi_i(|\boldsymbol{a}_i^T\boldsymbol{h}|^2)}{\|\boldsymbol{h}\|^2} > (0.13+\epsilon)\right] < \exp(-cm\epsilon^2), \tag{55}$$

for some universal constant c, as long as $\|\boldsymbol{h}\| \leq \frac{1}{10}\|\boldsymbol{x}\|$.

We have proved so far that the claim holds for a fixed $\boldsymbol{h}$. We next obtain a uniform bound over all $\boldsymbol{h}$ satisfying $\|\boldsymbol{h}\| \leq \frac{1}{10}\|\boldsymbol{x}\|$. We first show the claim holds for all $\boldsymbol{h}$ with $\|\boldsymbol{h}\| = \frac{1}{10}\|\boldsymbol{x}\|$ and then argue that the claim holds when $\|\boldsymbol{h}\| < \frac{1}{10}\|\boldsymbol{x}\|$ towards the end of the proof. Let $\epsilon' = \epsilon\frac{\|\boldsymbol{x}\|}{10}$ and we construct an ϵ'-net $\mathcal{N}_{\epsilon'}$ covering the sphere with the radius $\frac{1}{10}\|\boldsymbol{x}\|$ in $\mathbb{R}^n$ with the cardinality $|\mathcal{N}_{\epsilon'}| \leq (1+\frac{2}{\epsilon})^n$. Then for any $\|\boldsymbol{h}\| = \frac{1}{10}\|\boldsymbol{x}\|$, there exists an $\boldsymbol{h}_0 \in \mathcal{N}_{\epsilon'}$ such that $\|\boldsymbol{h}-\boldsymbol{h}_0\| \leq \epsilon\|\boldsymbol{h}\|$. Taking the union bound for all the points on the net, we claim that

$$\frac{1}{m}\sum_{i=1}^m \chi_i\left(|\boldsymbol{a}_i^T\boldsymbol{h}_0|^2\right) \leq (0.13+\epsilon)\|\boldsymbol{h}_0\|^2, \quad \forall \boldsymbol{h}_0 \in \mathcal{N}_{\epsilon'} \tag{56}$$

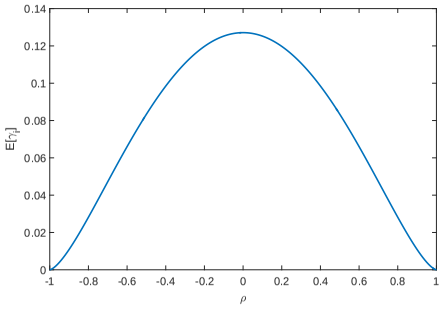


Figure 7: $\mathrm{E}[\gamma_i]$ with respect to ρ.

holds with probability at least $1-(1+2/\epsilon)^n\exp(-cm\epsilon^2)$.

Since $\chi_i(t)$'s are Lipschitz functions with the constant $1/\delta$, we have the following bound

$$\left|\chi_i(|\boldsymbol{a}_i^T\boldsymbol{h}|^2) - \chi_i(|\boldsymbol{a}_i^T\boldsymbol{h}_0|^2)\right| \leq \frac{1}{\delta}||\boldsymbol{a}_i^T\boldsymbol{h}|^2 - |\boldsymbol{a}_i^T\boldsymbol{h}_0|^2|. \tag{57}$$

Moreover, by (Chen and Candès, 2015, Lemma 1), we have

$$\frac{1}{m}\sum_{i=1}^m |\boldsymbol{a}_i^T\boldsymbol{M}\boldsymbol{a}_i| \leq c_2\|\boldsymbol{M}\|_F, \qquad \text{for all symmetric rank-2 matrices } \boldsymbol{M}\in\mathbb{R}^{n\times n}, \tag{58}$$

holds with probability at least $1-C\exp(-c_1m)$ as long as $m>c_0n$ for some constants $C, c_0, c_1, c_2>0$. Consequently, on the event that (58) holds, we have

$$\begin{aligned}
&\left|\frac{1}{m}\sum_{i=1}^m \chi_i\left(|\boldsymbol{a}_i^T\boldsymbol{h}|^2\right) - \frac{1}{m}\sum_{i=1}^m \chi_i\left(|\boldsymbol{a}_i^T\boldsymbol{h}_0|^2\right)\right| \\
&\leq \frac{1}{m}\sum_{i=1}^m \left|\chi_i\left(|\boldsymbol{a}_i^T\boldsymbol{h}|^2\right) - \chi_i\left(|\boldsymbol{a}_i^T\boldsymbol{h}_0|^2\right)\right| \\
&\leq \frac{1}{\delta}\cdot\frac{1}{m}\sum_{i=1}^m \left|\boldsymbol{a}_i^T(\boldsymbol{h}\boldsymbol{h}^T-\boldsymbol{h}_0\boldsymbol{h}_0^T)\boldsymbol{a}_i\right| \qquad \text{because of (57)} \\
&\leq \frac{1}{\delta}\cdot c_2\|\boldsymbol{h}\boldsymbol{h}^T-\boldsymbol{h}_0\boldsymbol{h}_0^T\|_F \qquad \text{because of (58)} \\
&\leq \frac{1}{\delta}\cdot 3c_2\|\boldsymbol{h}-\boldsymbol{h}_0\|\cdot\|\boldsymbol{h}\| \\
&\leq \frac{3c_2\epsilon}{\delta}\|\boldsymbol{h}\|^2,
\end{aligned}$$

where the second to last inequality is due to (Chen and Candès, 2015, Lemma 2).

On the intersection of the events over which (56) and (58) hold respectively, we have

$$\frac{1}{m}\sum_{i=1}^m \chi_i\left(|\boldsymbol{a}_i^T\boldsymbol{h}|^2\right) \leq (0.13+\epsilon+3c_2\epsilon/\delta)\|\boldsymbol{h}\|^2, \tag{59}$$

for all $\boldsymbol{h}$ with $\|\boldsymbol{h}\| = \frac{1}{10}\|\boldsymbol{x}\|$.

For the case when $\|\boldsymbol{h}'\| < \frac{1}{10}\|\boldsymbol{x}\|$, $\boldsymbol{h}' = \omega\boldsymbol{h}$ for some $\boldsymbol{h}$ satisfying $\|\boldsymbol{h}\| = \frac{1}{10}\|\boldsymbol{x}\|$ and $0 < \omega < 1$. By the definition of $\chi_i(\cdot)$, it can be verified that

$$\chi_i(|\boldsymbol{a}_i^T\boldsymbol{h}'|^2) = \chi_i(|\boldsymbol{a}_i^T(\omega\boldsymbol{h})|^2) \le \omega^2\chi_i(|\boldsymbol{a}_i^T\boldsymbol{h}|^2). \tag{60}$$

Applying (59), on the same event over which (56) and (58) hold, we have

$$\frac{1}{m}\sum_{i=1}^m \chi_i\left(|\boldsymbol{a}_i^T\boldsymbol{h}'|^2\right) \le (0.13 + \epsilon + 3c_2\epsilon/\delta)\,\|\boldsymbol{h}'\|^2, \tag{61}$$

for all $\|\boldsymbol{h}'\| < \frac{1}{10}\|\boldsymbol{x}\|$. Since ϵ can be arbitrarily small, the proof is completed.

C.2 Proof of Lemma 8

Denote $v_i := \boldsymbol{a}_i^T\boldsymbol{z} - |\boldsymbol{a}_i^T\boldsymbol{x}|\mathrm{sgn}(\boldsymbol{a}_i^T\boldsymbol{z})$ for $i = 1, \cdots, m$. Then

$$\nabla\ell(\boldsymbol{z}) = \frac{1}{m}\boldsymbol{A}^T\boldsymbol{v}, \tag{62}$$

where $\boldsymbol{A}$ is a matrix with each row being $\boldsymbol{a}_i^T$ and $\boldsymbol{v} = [v_1, \cdots, v_m]^T$. Thus,

$$\|\nabla\ell(\boldsymbol{z})\| = \left\|\frac{1}{m}\boldsymbol{A}^T\boldsymbol{v}\right\| \le \frac{1}{m}\|\boldsymbol{A}\|\cdot\|\boldsymbol{v}\| \le (1+\delta)\frac{\|\boldsymbol{v}\|}{\sqrt{m}} \tag{63}$$

as long as $m \ge c_1 n$ for some sufficiently large $c_1 > 0$, where the spectral norm bound $\|\boldsymbol{A}\| \le \sqrt{m}(1+\delta)$ follows from (Vershynin, 2012, Theorem 5.32).

We next bound $\|\boldsymbol{v}\|$. Let $\boldsymbol{v} = \boldsymbol{v}^{(1)} + \boldsymbol{v}^{(2)}$, where $v_i^{(1)} = \boldsymbol{a}_i^T\boldsymbol{h}$ and $v_i^{(2)} = 2\boldsymbol{a}_i^T\boldsymbol{x}\mathbf{1}_{\{(\boldsymbol{a}_i^T\boldsymbol{z})(\boldsymbol{a}_i^T\boldsymbol{x})<0\}}$. By the triangle inequality, we have $\|\boldsymbol{v}\| \le \|\boldsymbol{v}^{(1)}\| + \|\boldsymbol{v}^{(2)}\|$. Furthermore, given $m > c_0 n$, following from (Candès et al., 2013, Lemma 3.1), with probability at least $1 - \exp(-cm)$, we have

$$\frac{1}{m}\|\boldsymbol{v}^{(1)}\|^2 = \frac{1}{m}\sum_{i=1}^m(\boldsymbol{a}_i^T\boldsymbol{h})^2 \le (1+\delta)\|\boldsymbol{h}\|^2. \tag{64}$$

By Lemma 7, we have with probability at least $1 - C\exp(-c_1 m)$

$$\frac{1}{m}\|\boldsymbol{v}^{(2)}\|^2 = \frac{1}{m}\sum_{i=1}^m 4(\boldsymbol{a}_i^T\boldsymbol{x})^2\cdot\mathbf{1}_{\{(\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{z})<0\}} \le 4(0.13+\epsilon)\|\boldsymbol{h}\|^2. \tag{65}$$

Hence,

$$\frac{\|\boldsymbol{v}\|}{\sqrt{m}} \le \left(\sqrt{1+\delta} + 2\sqrt{0.13+\epsilon}\right)\|\boldsymbol{h}\|. \tag{66}$$

This concludes the proof.

Appendix D. Proof of Theorem 3

The initialization analysis is similar to Appendix B and is hence omitted. To analyze the gradient loop, we consider the following two regimes.

• **Regime 1**: $c_4\|\boldsymbol{z}\| \ge \|\boldsymbol{h}\| \ge c_3\frac{\|\boldsymbol{w}\|}{\sqrt{m}}$. In this regime, the error contraction by each gradient descent step is given by

$$\mathrm{dist}\left(\boldsymbol{z} - \mu\nabla\ell(\boldsymbol{z}), \boldsymbol{x}\right) \le (1-\rho)\mathrm{dist}(\boldsymbol{z}, \boldsymbol{x}). \tag{67}$$

It suffices to justify that $\nabla\ell(\boldsymbol{z})$ satisfies the RC. We first have

$$\begin{aligned}\nabla\ell(\boldsymbol{z}) &= \frac{1}{m}\sum_{i=1}^m\left(\boldsymbol{a}_i^T\boldsymbol{z} - y_i\cdot\frac{\boldsymbol{a}_i^T\boldsymbol{z}}{|\boldsymbol{a}_i^T\boldsymbol{z}|}\right)\boldsymbol{a}_i \\ &= \underbrace{\frac{1}{m}\sum_{i=1}^m\left(\boldsymbol{a}_i^T\boldsymbol{z} - |\boldsymbol{a}_i^T\boldsymbol{x}|\cdot\frac{\boldsymbol{a}_i^T\boldsymbol{z}}{|\boldsymbol{a}_i^T\boldsymbol{z}|}\right)\boldsymbol{a}_i}_{\nabla^{clean}\ell(\boldsymbol{z})} - \underbrace{\frac{1}{m}\sum_{i=1}^m\left(w_i\cdot\frac{\boldsymbol{a}_i^T\boldsymbol{z}}{|\boldsymbol{a}_i^T\boldsymbol{z}|}\right)\boldsymbol{a}_i}_{\nabla^{noise}\ell(\boldsymbol{z})}.\end{aligned} \tag{68}$$

All the proofs for Lemmas 6, 7 and 8 are still valid for $\nabla^{clean}\ell(\boldsymbol{z})$, and thus we have

$$\langle\nabla^{clean}\ell(\boldsymbol{z}), \boldsymbol{h}\rangle \ge 0.74\|\boldsymbol{h}\|^2, \tag{69}$$

$$\left\|\nabla^{clean}\ell(\boldsymbol{z})\right\| \le 2(1+\delta)\|\boldsymbol{h}\|. \tag{70}$$

Next, we analyze the contribution of the noise. Let $\tilde{w}_i = w_i\frac{\boldsymbol{a}_i^T\boldsymbol{z}}{|\boldsymbol{a}_i^T\boldsymbol{z}|}$, and then for sufficiently large m/n, we have

$$\|\nabla^{noise}\ell(\boldsymbol{z})\| = \left\|\frac{1}{m}\boldsymbol{A}^T\tilde{\boldsymbol{w}}\right\| \le \left\|\frac{1}{\sqrt{m}}\boldsymbol{A}^T\right\|\left\|\frac{\tilde{\boldsymbol{w}}}{\sqrt{m}}\right\| \le (1+\delta)\frac{\|\tilde{\boldsymbol{w}}\|}{\sqrt{m}} \le (1+\delta)\frac{\|\boldsymbol{w}\|}{\sqrt{m}}, \tag{71}$$

where the second inequality is due to the spectral norm bound $\|\boldsymbol{A}\| \le \sqrt{m}(1+\delta)$ (Vershynin, 2012, Theorem 5.32). Given the regime condition $\|\boldsymbol{h}\| \ge c_3\frac{\|\boldsymbol{w}\|}{\sqrt{m}}$, we further have

$$\|\nabla^{noise}\ell(\boldsymbol{z})\| \le \frac{(1+\delta)}{c_3}\|\boldsymbol{h}\|, \tag{72}$$

$$\left|\left\langle\nabla^{noise}\ell(\boldsymbol{z}), \boldsymbol{h}\right\rangle\right| \le \left\|\nabla^{noise}\ell(\boldsymbol{z})\right\|\cdot\|\boldsymbol{h}\| \le \frac{(1+\delta)}{c_3}\|\boldsymbol{h}\|^2. \tag{73}$$

Combining these together, one has

$$\langle\nabla\ell(\boldsymbol{z}), \boldsymbol{h}\rangle \ge \left\langle\nabla^{clean}\ell(\boldsymbol{z}), \boldsymbol{h}\right\rangle - \left|\left\langle\nabla^{noise}\ell(\boldsymbol{z}), \boldsymbol{h}\right\rangle\right| \ge \left(0.74 - \frac{(1+\delta)}{c_3}\right)\|\boldsymbol{h}\|^2, \tag{74}$$

and

$$\|\nabla\ell(\boldsymbol{z})\| \le \left\|\nabla^{clean}\ell(\boldsymbol{z})\right\| + \left\|\nabla^{noise}\ell(\boldsymbol{z})\right\| \le (1+\delta)\left(2 + \frac{1}{c_3}\right)\|\boldsymbol{h}\|. \tag{75}$$

The RC is guaranteed if μ, λ are chosen properly, c_3 is sufficiently large, and δ is sufficiently small.

• **Regime 2**: Once the iterate enters the regime with $\|\boldsymbol{h}\| \le \frac{c_3\|\boldsymbol{w}\|}{\sqrt{m}}$, the gradient descent update may not reduce the estimation error. However, in this regime, the size $\mu\nabla\ell(\boldsymbol{z})$ of each move is at most $\mathcal{O}(\|\boldsymbol{w}\|/\sqrt{m})$. Then the estimation error cannot increase by more than $\|\boldsymbol{w}\|/\sqrt{m}$ with a constant factor. Thus, one has

$$\operatorname{dist}\left(\boldsymbol{z}+\mu\nabla\ell(\boldsymbol{z}),\boldsymbol{x}\right) \le c_5\frac{\|\boldsymbol{w}\|}{\sqrt{m}} \tag{76}$$

for some constant c_5. As long as $\|\boldsymbol{w}\|/\sqrt{m}$ is sufficiently small, it is guaranteed that $c_5\frac{\|\boldsymbol{w}\|}{\sqrt{m}} \le c_4\|\boldsymbol{x}\|$. If the iterate jumps out of *Regime 2*, it falls back into *Regime 1*.

Appendix E. Convergence of Mini-batch IRWF and Kaczmarz-PR

E.1 Proof of Theorem 4

Without loss of generality, we assume $\boldsymbol{z}^{(t)}$ is in the neighborhood of $\boldsymbol{x}$ (otherwise it is in the neighborhood of $-\boldsymbol{x}$). Let $\boldsymbol{h}=\boldsymbol{z}^{(t)}-\boldsymbol{x}$. We follow the notations in Appendix C and let $\mathcal{S}=\{i:(\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{z}^{(t)})<0\}$. Then we have

$$\begin{aligned}
&\mathrm{E}_{\Gamma_t}\left[\operatorname{dist}^2\left(\boldsymbol{z}^{(t+1)},\boldsymbol{x}\right)\right]\\
&=\mathrm{E}_{\Gamma_t}\left[\left\|\boldsymbol{z}^{(t)}-\mu\boldsymbol{A}_{\Gamma_t}^T\left(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)}-\boldsymbol{y}_{\Gamma_t}\odot\operatorname{sgn}(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)})\right)-\boldsymbol{x}\right\|^2\right]\\
&=\|\boldsymbol{h}\|^2-2\mu\mathrm{E}_{\Gamma_t}\left[\left(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)}-\boldsymbol{y}_{\Gamma_t}\odot\operatorname{sgn}(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)})\right)^T\boldsymbol{A}_{\Gamma_t}\boldsymbol{h}\right]\\
&\quad+\mu^2\mathrm{E}_{\Gamma_t}\left[\left(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)}-\boldsymbol{y}_{\Gamma_t}\odot\operatorname{sgn}(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)})\right)^T\boldsymbol{A}_{\Gamma_t}\boldsymbol{A}_{\Gamma_t}^T\left(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)}-\boldsymbol{y}_{\Gamma_t}\odot\operatorname{sgn}(\boldsymbol{A}_{\Gamma_t}\boldsymbol{z}^{(t)})\right)\right]\\
&\overset{(a)}{=}\|\boldsymbol{h}\|^2-\frac{2\mu k}{m}\sum_{i=1}^m\boldsymbol{a}_i^T\boldsymbol{h}\left(\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}-y_i\cdot\frac{\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}|}\right)+\frac{\mu^2k}{m}\sum_{i=1}^m\|\boldsymbol{a}_i\|^2\left(\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}-y_i\cdot\frac{\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}|}\right)^2\\
&=\|\boldsymbol{h}\|^2-\frac{2\mu k}{m}\left(\sum_{i=1}^m(\boldsymbol{a}_i^T\boldsymbol{h})^2+\sum_{i\in\mathcal{S}}2(\boldsymbol{a}_i^T\boldsymbol{h})(\boldsymbol{a}_i^T\boldsymbol{x})\right)\\
&\quad+\frac{\mu^2k}{m}\left(\sum_{i=1}^m\|\boldsymbol{a}_i\|^2(\boldsymbol{a}_i^T\boldsymbol{h})^2+4\sum_{i\in\mathcal{S}}\|\boldsymbol{a}_i\|^2(\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{z}^{(t)})\right)\\
&\le\|\boldsymbol{h}\|^2-\frac{2\mu k}{m}\sum_{i=1}^m(\boldsymbol{a}_i^T\boldsymbol{h})^2+\frac{4\mu k}{m}\sum_{i\in\mathcal{S}}\left|(\boldsymbol{a}_i^T\boldsymbol{h})(\boldsymbol{a}_i^T\boldsymbol{x})\right|+\frac{\mu^2k}{m}\sum_{i=1}^m\|\boldsymbol{a}_i\|^2(\boldsymbol{a}_i^T\boldsymbol{h})^2,
\end{aligned}\tag{77}$$

where (a) is due to the fact that Γ_t is uniformly chosen from all subsets of $\{1,2,\ldots,m\}$ with the cardinality k.

By Lemma 6, we have that if $m\ge c_0\epsilon^{-2}n$ then with probability $1-2\exp(-c_1m\epsilon^2)$

$$(1-\epsilon)\|\boldsymbol{h}\|^2\le\frac{1}{m}\sum_{i=1}^m(\boldsymbol{a}_i^T\boldsymbol{h})^2\le(1+\epsilon)\|\boldsymbol{h}\|^2.$$

holds for all vectors $\boldsymbol{h}$. By Lemma 7, we have that with probability $1-C\exp(-c_1m\epsilon^2)$

$$\frac{1}{m}\sum_{i\in\mathcal{S}}\left|(\boldsymbol{a}_i^T\boldsymbol{h})(\boldsymbol{a}_i^T\boldsymbol{x})\right|\le(0.13+\epsilon)\|\boldsymbol{h}\|^2$$

holds for all $\boldsymbol{h}$ satisfying $\|\boldsymbol{h}\|/\|\boldsymbol{x}\|\le\frac{1}{10}$.

Define an event $E_1:=\{\max_{1\le i\le m}\|\boldsymbol{a}_i\|^2\le 6n\}$. It can be shown that E_1 holds with probability $1-m\exp(-1.5n)$. Then on the event E_1, Equation (77) is further upper bounded by

$$\begin{aligned}
\mathrm{E}_{\Gamma_t}\left[\operatorname{dist}^2\left(\boldsymbol{z}^{(t+1)},\boldsymbol{x}\right)\right]&\le\left(1-2\mu k(1-\epsilon)+4\mu k(0.13+\epsilon)+\mu^2k\cdot6n(1+\epsilon)\right)\|\boldsymbol{h}\|^2\\
&\le\left(1-2\mu k(0.74-3\epsilon-3n\mu(1+\epsilon))\right)\|\boldsymbol{h}\|^2.
\end{aligned}\tag{78}$$

By choosing the step size $\mu\le\frac{0.24}{n}$, the proposition is proved.

E.2 Proof of Theorem 5

Without loss of generality, we assume the $\boldsymbol{z}^{(t)}$ is in the neighborhood of $\boldsymbol{x}$ (otherwise it is in the neighborhood of $-\boldsymbol{x}$). Let $\boldsymbol{h}=\boldsymbol{z}^{(t)}-\boldsymbol{x}$. We follow the notations in Appendix C and let $\mathcal{S}=\{i:(\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{z}^{(t)})<0\}$. Then we have

$$\begin{aligned}
&\mathrm{E}_{i_t}\operatorname{dist}^2\left(\boldsymbol{z}^{(t+1)},\boldsymbol{x}\right)\\
&=\mathrm{E}_{i_t}\left\|\boldsymbol{z}^{(t)}-\frac{1}{\|\boldsymbol{a}_{i_t}\|^2}\left(\boldsymbol{a}_{i_t}^T\boldsymbol{z}^{(t)}-y_{i_t}\cdot\frac{\boldsymbol{a}_{i_t}^T\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_{i_t}^T\boldsymbol{z}^{(t)}|}\right)\boldsymbol{a}_{i_t}-\boldsymbol{x}\right\|^2\\
&=\|\boldsymbol{h}\|^2-2\mathrm{E}_{i_t}\frac{(\boldsymbol{a}_{i_t}^T\boldsymbol{h})}{\|\boldsymbol{a}_{i_t}\|^2}\left(\boldsymbol{a}_{i_t}^T\boldsymbol{z}^{(t)}-y_{i_t}\cdot\frac{\boldsymbol{a}_{i_t}^T\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_{i_t}^T\boldsymbol{z}^{(t)}|}\right)+\mathrm{E}_{i_t}\frac{1}{\|\boldsymbol{a}_{i_t}\|^2}\left(\boldsymbol{a}_{i_t}^T\boldsymbol{z}^{(t)}-y_{i_t}\cdot\frac{\boldsymbol{a}_{i_t}^T\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_{i_t}^T\boldsymbol{z}^{(t)}|}\right)^2\\
&\overset{(a)}{=}\|\boldsymbol{h}\|^2-\frac{2}{m}\sum_{i=1}^m\frac{(\boldsymbol{a}_i^T\boldsymbol{h})}{\|\boldsymbol{a}_i\|^2}\left(\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}-y_i\cdot\frac{\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}|}\right)+\frac{1}{m}\sum_{i=1}^m\frac{1}{\|\boldsymbol{a}_i\|^2}\left(\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}-y_i\cdot\frac{\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}}{|\boldsymbol{a}_i^T\boldsymbol{z}^{(t)}|}\right)^2\\
&=\|\boldsymbol{h}\|^2-\frac{2}{m}\left(\sum_{i=1}^m\frac{(\boldsymbol{a}_i^T\boldsymbol{h})^2}{\|\boldsymbol{a}_i\|^2}+\sum_{i\in\mathcal{S}}\frac{2(\boldsymbol{a}_i^T\boldsymbol{h})(\boldsymbol{a}_i^T\boldsymbol{x})}{\|\boldsymbol{a}_i\|^2}\right)+\frac{1}{m}\left(\sum_{i=1}^m\frac{(\boldsymbol{a}_i^T\boldsymbol{h})^2}{\|\boldsymbol{a}_i\|^2}+4\sum_{i\in\mathcal{S}}\frac{(\boldsymbol{a}_i^T\boldsymbol{x})(\boldsymbol{a}_i^T\boldsymbol{z}^{(t)})}{\|\boldsymbol{a}_i\|^2}\right)\\
&=\|\boldsymbol{h}\|^2-\frac{1}{m}\sum_{i=1}^m\frac{(\boldsymbol{a}_i^T\boldsymbol{h})^2}{\|\boldsymbol{a}_i\|^2}+\frac{4}{m}\sum_{i\in\mathcal{S}}\frac{(\boldsymbol{a}_i^T\boldsymbol{x})^2}{\|\boldsymbol{a}_i\|^2}
\end{aligned}\tag{79}$$

where (a) is due to the fact that i_t is sampled uniformly at random from $\{1,2,\cdots,m\}$. By the special case of Lemma 5.20 in (Vershynin, 2012), $\{\sqrt{n}\frac{\boldsymbol{a}_i}{\|\boldsymbol{a}_i\|}\}_{i=1}^m$ are independent isotropic random vectors in $\mathbb{R}^n$ and hence

$$\mathrm{E}\left[n\frac{(\boldsymbol{a}_i^T\boldsymbol{h})^2}{\|\boldsymbol{a}_i\|^2}\right]=\|\boldsymbol{h}\|^2.$$

Moreover, $\{\sqrt{n}\frac{\boldsymbol{a}_i}{\|\boldsymbol{a}_i\|}\}_{i=1}^m$ are sub-Gaussian and the sub-Gaussian norm is bounded by an absolute constant. Thus, we have that if $m\ge c_0\epsilon^{-2}n$, then with probability $1-2\exp(-c_1m\epsilon^2)$,

$$\frac{1}{m}\sum_{i=1}^m\frac{(\boldsymbol{a}_i^T\boldsymbol{h})^2}{\|\boldsymbol{a}_i\|^2}\ge\frac{(1-\epsilon)}{n}\|\boldsymbol{h}\|^2.$$

holds for all vectors $\boldsymbol{h}$. By Lemma 7, we have that with probability $1-C\exp(-c_1m\epsilon^2)$

$$\frac{1}{m}\sum_{i\in\mathcal{S}}\left|\boldsymbol{a}_i^T\boldsymbol{x}\right|^2\le\frac{1}{m}\sum_{i=1}^m\left|\boldsymbol{a}_i^T\boldsymbol{h}\right|^2\mathbf{1}_{\{|\boldsymbol{a}_i^T\boldsymbol{x}|<|\boldsymbol{a}_i^T\boldsymbol{h}|\}}\le(0.13+\epsilon)\|\boldsymbol{h}\|^2$$

holds for all $\boldsymbol{h}$ satisfying $\|\boldsymbol{h}\|/\|\boldsymbol{x}\| \le \frac{1}{10}$.

Define an event $E_2 := \{\min_{1\le i\le m} \|\boldsymbol{a}_i\|^2 \ge \frac{2}{3}n\}$. It can be shown that $\mathbb{P}\{E_2\} \ge 1 - m\exp(-n/12)$. Then on the event E_2, (79) is further upper bounded by

$$\mathrm{E}_{i_t}\left[\mathrm{dist}^2\left(\boldsymbol{z}^{(t+1)},\boldsymbol{x}\right)\right] \le \left(1 - \frac{1-\epsilon}{n} + \frac{6(0.13+\epsilon)}{n}\right)\|\boldsymbol{h}\|^2 \le \left(1 - \frac{0.22-7\epsilon}{n}\right)\|\boldsymbol{h}\|^2, \quad (80)$$

which concludes the proof.

References

Anima Anandkumar, Prateek Jain, Yang Shi, and Uma Naresh Niranjan. Tensor vs. matrix methods: Robust tensor decomposition under block sparse perturbations. In *Artificial Intelligence and Statistics*, pages 268–276, 2016.

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. In *COLT*, pages 113–149, 2015.

Afonso S Bandeira, Jameson Cahill, Dustin G Mixon, and Aaron A Nelson. Saving phase: Injectivity and stability for phase retrieval. *Applied and Computational Harmonic Analysis*, 37(1):106–125, 2014.

Afonso S Bandeira, Nicolas Boumal, and Vladislav Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In *29th Annual Conference on Learning Theory*, 2016.

Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 3873–3881, 2016.

Nicolas Boumal. Nonconvex phase synchronization. *SIAM Journal on Optimization*, 26(4): 2355–2377, 2016.

James V Burke, Adrian S Lewis, and Michael L Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3): 751–779, 2005.

Tony T Cai, Xiaodong Li, and Zongming Ma. Optimal rates of convergence for noisy sparse phase retrieval via thresholded Wirtinger flow. *The Annals of Statistics*, 44(5):2221–2251, 2016.

E. J. Candès and X. Li. Solving quadratic equations via phaselift when there are about as many equations as unknowns. *Foundations of Computational Mathematics*, 14(5): 1017–1026, 2014.

Emmanuel J Candès, Thomas Strohmer, and Vladislav Voroninski. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, 2013.

Emmanuel J Candès, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via Wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

Yudong Chen and Martin J Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *arXiv preprint arXiv:1509.03025*, 2015.

Yuxin Chen and Emmanuel Candès. Solving random quadratic systems of equations is nearly as easy as solving linear systems. In *Advances in Neural Information Processing Systems (NIPS)*. 2015.

Yuxin Chen, Yuejie Chi, and Andrea J Goldsmith. Exact and stable covariance estimation from quadratic sampling via convex programming. *IEEE Transactions on Information Theory*, 61(7):4034–4059, 2015.

Yuejie Chi and Yue M Lu. Kaczmarz method for solving quadratic equations. *IEEE Signal Processing Letters*, 23(9):1183–1187, 2016.

James D Donahue. Products and quotients of random variables and their applications. Technical report, DTIC Document, 1964.

Jan Drenth. *Principles of protein X-ray crystallography*. Springer Science & Business Media, 2007.

James R Fienup. Phase retrieval algorithms: a comparison. *Applied Optics*, 21(15):2758–2769, 1982.

Fajwel Fogel, Irène Waldspurger, and Alexandre dAspremont. Phase retrieval for imaging problems. *Mathematical programming computation*, 8(3):311–335, 2016.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points-online stochastic gradient for tensor decomposition. In *Conference on Learning Theory (COLT)*, pages 797–842, 2015.

Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

Ralph W Gerchberg. A practical algorithm for the determination of phase from image and diffraction plane pictures. *Optik*, 35:237, 1972.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Moritz Hardt. Understanding alternating minimization for matrix completion. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 651–660. IEEE, 2014.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, 2013.

Halyun Jeong and C. Sinan Gunturk. Convergence of the randomized Kaczmarz method for phase retrieval. *arXiv preprint arXiv:1706.10291*, 2017.

Chi Jin, Sham M Kakade, and Praneeth Netrapalli. Provable efficient online matrix completion via non-convex stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 4520–4528, 2016.

Stefan Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de lAcademie Polonaise des Sciences et des Lettres*, 35:355–357, 1937.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980 –2998, June 2010.

Krzysztof C Kiwiel. Convergence of the gradient sampling algorithm for nonsmooth nonconvex optimization. *SIAM Journal on Optimization*, 18(2):379–388, 2007.

Ritesh Kolte and Ayfer Özgür. Phase retrieval via incremental truncated Wirtinger flow. *arXiv preprint arXiv:1606.03196*, 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, 2012.

A Ya Kruger. On Fréchet subdifferentials. *Journal of Mathematical Sciences*, 116(3): 3325–3358, 2003.

Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent converges to minimizers. *arXiv preprint arXiv:1602.04915*, 2016a.

Kiryung Lee, Yanjun Li, Marius Junge, and Yoram Bresler. Blind recovery of sparse signals from subsampled convolution. *IEEE Transactions on Information Theory*, 2016b.

Gen Li, Yuantao Gu, and Yue M Lu. Phase retrieval using iterative projections: Dynamics in the large systems limit. In *The 53rd Annual Allerton Conference on Communication, Control, and Computing*, 2015.

Xiaodong Li, Shuyang Ling, Thomas Strohmer, and Ke Wei. Rapid, robust, and reliable blind deconvolution via nonconvex optimization. *arXiv preprint arXiv:1606.04933*, 2016.

Yuanxin Li, Yuejie Chi, Huishuai Zhang, and Yingbin Liang. Non-convex low-rank matrix recovery from corrupted random linear measurements. In *Sampling Theory and Applications (SampTA), International Conference on*, 2017.

Jianwei Miao, Pambos Charalambous, Janos Kirz, and David Sayre. Extending the methodology of X-ray crystallography to allow imaging of micrometre-sized non-crystalline specimens. *Nature*, 400(6742):342–344, 1999.

Jianwei Miao, Tetsuya Ishikawa, Qun Shen, and Thomas Earnest. Extending X-ray crystallography to allow the imaging of noncrystalline materials, cells, and single protein complexes. *Annu. Rev. Phys. Chem.*, 59:387–410, 2008.

Eric Moulines and Francis R Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Mathematical Programming*, 155(1-2): 549–573, 2016.

Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi. Phase retrieval using alternating minimization. *Advances in Neural Information Processing Systems (NIPS)*, 2013.

Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust PCA. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

Peter Ochs, Alexey Dosovitskiy, Thomas Brox, and Thomas Pock. On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIAM Journal on Imaging Sciences*, 8(1):331–372, 2015.

Dohyung Park, Anastasios Kyrillidis, Srinadh Bhojanapalli, Constantine Caramanis, and Sujay Sanghavi. Provable non-convex projected gradient descent for a class of constrained matrix optimization problems. *arXiv preprint arXiv:1606.01316*, 2016.

Christopher De Sa, Christopher Re, and Kunle Olukotun. Global convergence of stochastic gradient descent for some non-convex matrix problems. In *Proceedings of International Conference on Machine Learning (ICML)*, pages 2332–2341, 2015.

Yoav Shechtman, Yonina C Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev. Phase retrieval with application to optical imaging: a contemporary overview. *IEEE Signal Processing Magazine*, 32(3):87–109, 2015.

Thomas Strohmer and Roman Vershynin. A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009.

Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery using nonconvex optimization. In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

Ju Sun, Qing Qu, and John Wright. A geometric analysis of phase retrieval. *arXiv preprint arXiv:1602.06664*, 2016.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

Yan Shuo Tan and Roman Vershynin. Phase retrieval via randomized Kaczmarz: Theoretical guarantees. *arXiv preprint arXiv:1706.09993*, 2017.

Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Benjamin Recht. Low-rank solutions of linear matrix equations via Procrustes flow. *arXiv preprint arXiv:1507.03566*, 2015.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *Compressed Sensing, Theory and Applications*, pages 210 – 268, 2012.

Irène Waldspurger. Phase retrieval with random gaussian sensing vectors by alternating projections. *arXiv preprint arXiv:1609.03088*, 2016.

Irène Waldspurger, Alexandre d'Aspremont, and Stéphane Mallat. Phase recovery, maxcut and complex semidefinite programming. *Mathematical Programming*, 149(1-2):47–81, 2015.

Gang Wang, Georgios B Giannakis, and Yonina C Eldar. Solving systems of random quadratic equations via truncated amplitude flow. *arXiv preprint arXiv:1605.08285*, 2016.

Ke Wei. Solving systems of phaseless equations via Kaczmarz methods: a proof of concept study. *Inverse Problems*, 31(12):125008, 2015.

Ke Wei, Jian-Feng Cai, Tony F Chan, and Shingyu Leung. Guarantees of Riemannian optimization for low rank matrix recovery. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1198–1222, 2016.

Huishuai Zhang, Yuejie Chi, and Yingbin Liang. Median-truncated nonconvex approach for phase retrieval with outliers. In *Proceedings of International Conference on Machine Learning (ICML)*, 2016.

Qinqing Zheng and John Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.

Qinqing Zheng and John Lafferty. Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent. *arXiv preprint arXiv:1605.07051*, 2016.

Anastasios Zouzias and Nikolaos M Freris. Randomized extended Kaczmarz for solving least squares. *SIAM Journal on Matrix Analysis and Applications*, 34(2):773–793, 2013.

Journal of Machine Learning Research 18 (2017) 1-39 Submitted 8/16; Revised 9/17; Published 11/17

Consistency, Breakdown Robustness, and Algorithms for Robust Improper Maximum Likelihood Clustering

Pietro Coretto PCORETTO@UNISA.IT
Department of Economics and Statistics
University of Salerno
Fisciano (SA), Italy

Christian Hennig C.HENNIG@UCL.AC.UK
Department of Statistical Science
University College London
London, United Kingdom

Editor: Ingo Steinwart

Abstract

The robust improper maximum likelihood estimator (RIMLE) is a new method for robust multivariate clustering finding approximately Gaussian clusters. It maximizes a pseudo-likelihood defined by adding a component with improper constant density for accommodating outliers to a Gaussian mixture. A special case of the RIMLE is MLE for multivariate finite Gaussian mixture models. In this paper we treat existence, consistency, and breakdown theory for the RIMLE comprehensively. RIMLE's existence is proved under non-smooth covariance matrix constraints. It is shown that these can be implemented via a computationally feasible Expectation-Conditional Maximization algorithm.

Keywords: Robustness, Improper density, Mixture models, Model-based clustering, Maximum likelihood, ECM-algorithm

1. Introduction

Maximum likelihood estimation (MLE) in a Gaussian mixture model with mixture components interpreted as clusters is a popular approach to cluster analysis (see, e.g., Fraley and Raftery (2002)). In many datasets not all observations can be assigned appropriately to clusters that can be properly modelled by a Gaussian distribution, and it is also well known that the MLE can be strongly affected by outliers (Hennig (2004)). In this paper we investigate the recently introduced "robust improper maximum likelihood estimator" (RIMLE, see Coretto and Hennig, 2016), a method for robust clustering with clusters that can be approximated by multivariate Gaussian distributions. The basic idea of RIMLE is to fit an improper density to the data that is made up by a Gaussian mixture density and a "pseudo mixture component" defined by a small constant density, which is meant to capture outliers and observations in low density areas of the data that cannot properly be assigned to a Gaussian mixture component (called "noise" in the following). This is inspired by the addition of a uniform "noise component" to a Gaussian mixture suggested by Banfield and Raftery (1993). Hennig (2004) showed that using an improper density improves the breakdown robustness of this approach for one-dimensional datasets. As in many other statistical problems, violations of the model assumptions may cause problems in cluster analysis. Our general attitude to the use of statistical models in cluster analysis is that the models should not be understood as reflecting some underlying but in practice unobservable "truth", but rather as thought constructs implying a certain behaviour of methods derived from them (e.g., maximizing the likelihood), which may or may not be appropriate in a given application (more details on the general philosophy of clustering can be found in Hennig and Liao (2013); Hennig (2015b)). Using a model such as a mixture of multivariate Gaussian distributions, interpreting every mixture component as a "cluster", implies that we look for clusters that are approximately "Gaussian-shaped", but we do not want to rely on whether the data really were generated i.i.d. by a Gaussian mixture. We focus on situations in which the number of clusters G is fixed.

There is a number of proposals already in the literature for accounting for the presence of noise and outliers in model-based clustering problems. The contributions can be divided in two groups: methods based on mixture modelling, and methods based on fixed partition models. Within the first group Banfield and Raftery (1993) and Coretto and Hennig (2011) dealt with uniform distributions added as "noise components" to a finite Gaussian mixture. Peel and McLachlan (2000) proposed to model data based on Student t-distributions. Cuesta-Albertos et al. (1997) and García-Escudero and Gordaliza (1999) introduced and studied trimming in order to robustify the k-means partitioning method. Robust partitioning methods with homoscedastic clusters based on ML–type procedures where proposed in Gallegos (2002) and Gallegos and Ritter (2005). Heteroscedasticity in ML-type partitioning methods has been introduced with the TCLUST algorithm of García-Escudero et al. (2008) and the "k–parameters clustering" of Gallegos and Ritter (2013). More references and an in-depth overview are given in García-Escudero et al. (2015). Different from the methods based on fixed partition models, mixture models and RIMLE allow a smooth transition between different clusters and between clustered observations and noise, which improves parameter estimation in the presence of overlap between mixture components. The one-dimensional version of the RIMLE was introduced in Coretto and Hennig (2010) and was investigated based on Monte Carlo experiments. Extension of the methods to the multivariate setting is not straightforward. Existence and consistency of the MLE for the multivariate Gaussian mixtures is a long standing problem due to the ill-posedness of the likelihood function. Even for ML for plain multivariate Gaussian mixtures (i.e. the RIMLE with the improper constant density set to zero), consistency theory is limited to the situation in which the model is assumed to hold precisely, and restrictive conditions are required (e.g., Redner and Walker (1984)). Chen and Tan (2009) and Alexandrovich (2014) propose and study a penalized ML estimator. García-Escudero et al. (2014) studied a classification ML estimator for Gaussian mixture that is based on the TCLUST idea.

In this paper we study the theoretical properties of the RIMLE as well as its computation. A comprehensive treatment of existence, consistency and robustness is given. This treatment includes the case of ML for multivariate Gaussian mixture as special case. Particularly, the robustness properties of RIMLE are superior to those of the mixture-based methods proposed by Banfield and Raftery (1993) and Peel and McLachlan (2000), as demonstrated later in the paper. For fitting plain Gaussian mixtures, some issues that are treated here arise as well, particularly the need to constrain the covariance matrices in order to avoid degeneration of the likelihood. Some literature on this is cited in Section

©2017 Pietro Coretto and Christian Hennig.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-382.html.

3.2. The consistency results given here in Section 4 are of a nonparametric nature and show the consistency of the RIMLE for the RIMLE-functional defined for a general class of sampling distributions. Similar results have been shown for a partition likelihood model (Gallegos and Ritter (2013)) and for alternative, trimming-based approaches to robust clustering (e.g., García-Escudero et al. (2008); Gallegos and Ritter (2009)). Compared to these results, there is an additional difficulty for the RIMLE, namely that degeneration of the likelihood needs to be prevented also in the case that almost all observations are assigned to the noise component and the remaining observations are fitted arbitrarily well. This may look like a disadvantage, but in the literature cited above such problems are only avoided by fixing the trimming rate. An analysis like the one given here, and in Coretto and Hennig (2016), is required for understanding the case in which both the proportion of points considered as "noise" and the density level at which this happens are flexible. Coretto and Hennig (2016) introduce the OTRIMLE, a data-adaptive choice of the improper constant density, the method's tuning constant for achieving robustness. That paper also includes a comprehensive simulation study comparing the different approaches to robust clustering. In the study, every method turns out to be superior for one or more setups, but the OTRIMLE achieves the most satisfactory overall performance.

The paper is organized as follows. We first discuss in Section 2 an artificial dataset to illustrate the issues RIMLE is meant to deal with. The RIMLE is introduced and defined in Section 3. In Section 4 existence and consistency of the RIMLE are proved. Section 5 treats the computation of the RIMLE and the choice of input parameters for the algorithms. Section 6 studies the breakdown robustness of RIMLE. Numerical experiments are presented in Section 7. Section 8 concludes the paper.

2. Artificial data examples

Every clustering method is designed to recover certain types of clusters even when they are based on methods and algorithms that apply universally. For instance, the well known k-means method aims to discover spherical balanced clusters, although the algorithm will find a solution when this is not the case. In this section we introduce some issues in robust clustering by showing examples of data affected by noise that cause trouble to most clustering methods, including those that supposedly explicitly account for it. These examples will illustrate the kind of clustering problem that the method investigated in this paper aims to address. Two artificial data sets are generated in dimension $p = 20$ from two sampling designs, called AsyNoise and GEM respectively, also considered for the numerical experiments presented in Section 7. The two data sets are shown in Figure 1 and 3. A detailed description of the sampling designs is given in Section 7.

In AsyNoise (Figure 1) there are 500 observations in 5 moderately separated clusters from student-t distributions with varying degrees of freedom, 187 observations (37.4%) are background noise. We have symmetric and elliptical clusters that are not well separated along all directions, and they are of different size. Although there is a deviation from Gaussianity in the tails of the clusters' distribution, methods based on Gaussian shapes are candidates to reconstruct such groups. Plain Gaussian mixture clustering (without noise component) fixing the number of clusters at $G = 5$ using the popular R package mclust of Fraley et al. (2012) puts the clustered points into two big clusters and assigns the noise to the

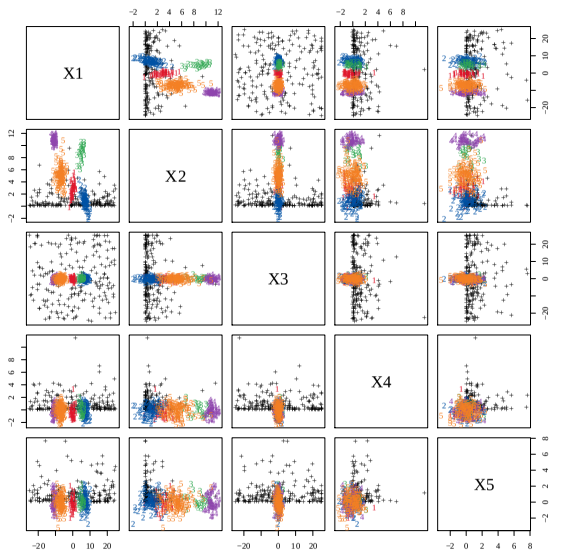


Figure 1: Scatter plots of $n = 500$ data points sampled from the AsyNoise design defined in Section 7. Marginals 1 to 5 are represented, further dimensions show a similar pattern. Colors denote the 5 clusters, while noise is represented by the "+" symbol.

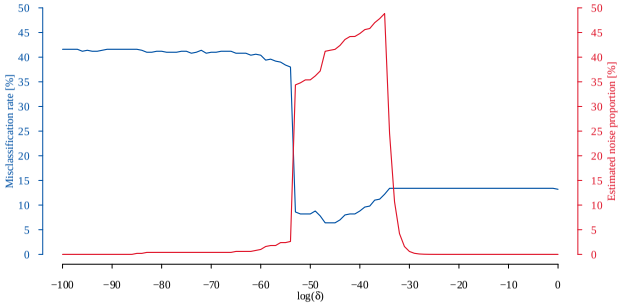


Figure 2: Misclassification rates (blue), and estimated noise proportions (red) by RIMLE over a grid of $\log(\delta)$ values for the data set in Figure 1.

remaining clusters achieving a misclassification rate of 61.4% (i.e., the best misclassification rate that can be achieved by permutation of the cluster labels so that no cluster is identified with the noise). Note that in this paper, mclust always automatically chooses an optimal covariance matrix parametrization by the BIC (see Fraley et al., 2012; Celeux and Govaert, 1995). One could wonder whether the data set may be an easy job for clustering methods that take into account outliers, but this is not the case as shown in Section 7. All robust methods require tuning that, directly or indirectly, controls the amount of noise present in the data set. Perhaps the only exception is the MLE for Gaussian mixtures with uniform noise of Banfield and Raftery (1993) implemented in the mclust package, but this can be led astray if the noise in fact behaves very differently from a uniform distribution. In real situations a priori information on the level of the noise is rarely available. The RIMLE method treated in this paper also requires tuning. The level of the noise is essentially controlled by the level of the improper noise density, called δ. For a given value of δ, the noise proportion then is estimated from the data.

For the data set in Figure 1 we computed the RIMLE for several values of $\log(\delta)$ (there are other constants required, chosen as $\gamma = 100$ and $\pi_{\max} = 0.5$, see Algorithm 2 and Sections 3). When choosing $\log(\delta)$ appropriately, namely $\log(\delta) \in [-53, -36]$, the RIMLE gets the structure of the data set right, and it stably produces a misclassification rate in the range [6.4%, 11%] with an estimated noise proportion in the range [34.38%, 45.8%]. This is clearly better than most other robust clustering methods we tried, see Section 7. Values of $\log(\delta)$ below -100 do not change the results. For large values of $\log(\delta)$ too much noise is found, hence the RIMLE's noise proportion constraint (see Section 3) becomes active and the resulting estimated noise proportion gets close to zero. The OTRIMLE criterion of Coretto and Hennig (2016) selects an optimal value $\log(\delta) = -40$, which is in the region where RIMLE shows its best performance. The RIMLE at $\log(\delta) = -40$ produces a misclassification rate of 8.8% with estimated noise proportion equal to 44.8%. Figure 2 shows how solutions change with changing values of $\log(\delta)$.

Another experimental situation considered in this paper is the GEM ("Gross Error Model") sampling design (Figure 3). In the GEM, 100 points are sampled from two normal populations with extremely different scatters, 2 points (2%) are outliers almost lying on a hyperplane. These outliers are not extremely separated from the regular points, and this can cause trouble to robust methods. Surprisingly, in a situation like this, some non-robust methods perform better than some robust alternatives. In fact, ML for plain Gaussian mixtures performed with mclust (without the noise component) assigns the two outliers to cluster 1 and achieves a misclassification rate of 2%, although the estimated mixture parameters are strongly biased. Robust methods may do worse if not well tuned (see Section 7). As for the previous data set the RIMLE has been computed for several values of $\log(\delta)$ maintaining all other parameters as before. The result can be seen in Figure 4. For any $-\infty < \log(\delta) \leq -46$ the RIMLE is 100% accurate and estimates a noise proportion of 2%. The OTRIMLE method for the data-driven choice of δ selects $\log(\delta) = -200$, which is in the region where the RIMLE achieves the best results.

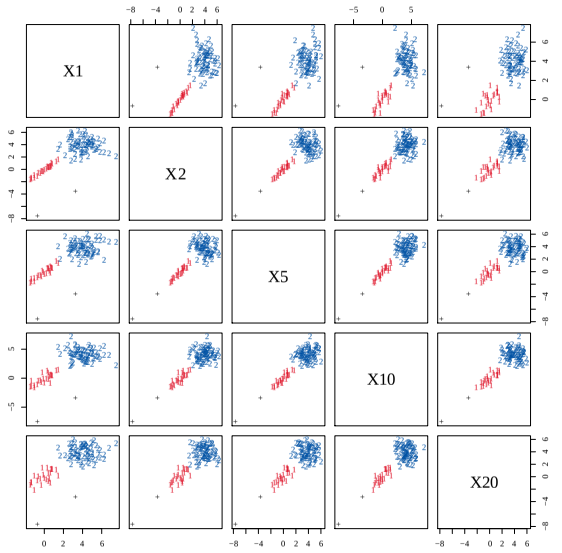


Figure 3: Scatter plots of $n = 100$ points sampled from the GEM sampling design defined in Section 7. Marginals 1,2,5,10, and 20 are represented, further dimensions show a similar pattern. Colors denote the 2 clusters, while noise is represented by the "+" symbol.

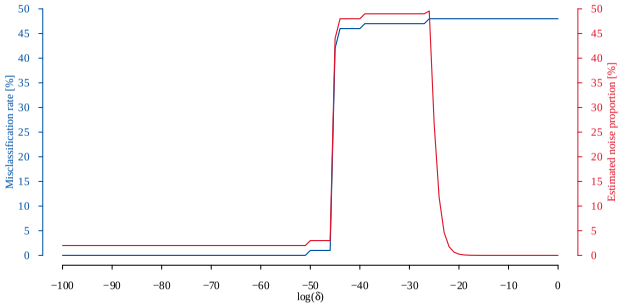


Figure 4: Misclassification rates (blue), and estimated noise proportions (red) by RIMLE over a grid of $\log(\delta)$ values for the data set in Figure 3.

3. Basic definitions

In this section we define the robust improper maximum likelihood estimator (RIMLE) along with its constrained parameter space.

3.1 RIMLE and clustering

The RIMLE is based on the "noise component" idea for robustification of the MLE based on the Gaussian mixture model. This models the noise by a uniform distribution, but in fact we are interested in more general patterns of noise or outliers. However, regions of high density are rather associated with clusters than with noise, so the noise regions should be those with the lowest density. This kind of distinction can be achieved by using the uniform density as in Banfield and Raftery (1993), but in the presence of gross outliers the dependence of the uniform distribution on the convex hull of the data causes a robustness problem (Hennig (2004)). The uniform distribution is not really used here as a model for the noise, but rather as a technical device to account for whatever goes on in low density regions. The RIMLE drives this idea further by using an improper uniform distribution the density value of which does not depend on how far awy extreme points in the data are from the main bulk. In the following, assume an observed sample $x_1, x_2, \ldots, x_n$, where x_i is the realization of a random variable $X_i \in \mathbb{R}^p$ with $p > 1$; $X_1, \ldots, X_n$ i.i.d. The goal is to cluster the sample points into G distinct groups. RIMLE then maximizes a pseudo-likelihood, which is based on the improper pseudo-density

$$\psi_\delta(x, \theta) = \pi_0 \delta + \sum_{j=1}^{G} \pi_j \phi(x; \mu_j, \Sigma_j), \tag{1}$$

where $\phi(\cdot, \mu, \Sigma)$ is the Gaussian density with mean μ and covariance matrix Σ, $\pi_0, \pi_j \in [0, 1]$ for $j = 1, 2, \ldots, G$, $\pi_0 + \sum_{i=1}^{G} \pi_j = 1$, while δ is the improper constant density. The parameter vector θ contains all Gaussian parameters plus all proportion parameters including π_0, ie. $\theta = (\mu_1, \ldots, \mu_G, \text{vect}(\Sigma_1), \ldots, \text{vect}(\Sigma_G), \pi_0, \ldots, \pi_G)$, where vect$(A)$ is the vectorized upper (or lower) triangle including the main diagonal of the symmetric square matrix A. δ and the number of Gaussian components G are considered fixed and known. Although this does not define a proper probability model, it yields a useful procedure for data modelled as a proportion of $(1 - \pi_0)$ of a mixture of Gaussian distributions, which have high enough density peaks to be interpreted as clusters plus a proportion π_0 times something unspecified with density smaller than or equal to δ (which may even contain further Gaussian components with so few points and/or so large within-component variation that they are not considered as "clusters"). The definition of the pseudo-model in (1) requires that the value of δ is fixed in advance. The choice of δ will be discussed in Section 5.2.

Given the sample improper pseudo-log-likelihood function

$$l_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log \psi_\delta(x_i, \theta), \tag{2}$$

the RIMLE is defined as

$$\theta_n(\delta) = \underset{\theta \in \Theta_n}{\arg\max}\, l_n(\theta), \tag{3}$$

where Θ_n is a constrained parameter space defined in Section 3.2. $\theta_n(\delta)$ is then used to cluster points using pseudo posterior probabilities for belonging to the Gaussian components or the improper uniform. These pseudo posterior probabilities are given by

$$\tau_j(x_i, \theta) := \begin{cases} \frac{\pi_0 \delta}{\psi_\delta(x_i, \theta)} & \text{if } j = 0 \\ \frac{\pi_j \phi(x_i, \mu_j, \Sigma_j)}{\psi_\delta(x_i, \theta)} & \text{if } j = 1, 2, \ldots, G; \end{cases} \qquad \text{for } i = 1, 2, \ldots, n.$$

Points are assigned to the component for which the pseudo posterior probability is maximized. The assignment rule is then given by

$$J(x_i, \theta) := \underset{j \in \{0,1,2,\ldots,G\}}{\arg\max}\, \tau_j(x_i, \theta). \tag{4}$$

The assignment based on maximum posterior probabilities is common to all model-based clustering methods. Here, an improper density is involved, and so these are "pseudo posterior probabilities".

We also define a population version of the RIMLE for later deriving consistency results for the sequence $\{\theta_n(\delta)\}_{n \in \mathbb{N}}$. Let $\text{E}_P f(x)$ be the expectation of $f(x)$ under P. The RIMLE population target function and the constrained parameter set can be obtained by replacing the empirical measure with P, and the population version of $l_n(\theta)$ is given by

$$L(\theta) = \text{E}_P \log(\psi_\delta(x, \theta)).$$

Define $L_G = \sup_{\theta \in \Theta_G(P)} L(\theta)$, where $\Theta_G(P)$ is a constrained parameter space defined in Section 3.2.

3.2 The constrained parameter space

Some notation: the kth element of μ_j is denoted by $\mu_{j,k}$ for $k = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, G$. Let $\lambda_{j,k}$ be the kth eigenvalue of Σ_j, define $\Lambda(\theta) = \{\lambda_{j,k} : \ j = 1, 2, \ldots, G; \ k = 1, 2, \ldots, p\}$, $\lambda_{\min}(\theta) = \min_{j,k}\{\Lambda(\theta)\}$, $\lambda_{\max}(\theta) = \max_{j,k}\{\Lambda(\theta)\}$.

Remark 1 *The p-dimensional Gaussian density can be written in terms of the eigen-decomposition of the covariance matrix:*

$$\phi(x; \mu, \Sigma) = (2\pi)^{-\frac{p}{2}} \left(\prod_{k=1}^{p} \lambda_k \right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2} \sum_{k=1}^{p} \lambda_k^{-1} (x - \mu)' v_k v_k' (x - \mu) \right),$$

where λ_k is the k-th eigenvalue of Σ, and v_k is its associated eigenvector, for $k = 1, 2, \ldots, p$. Let $\lambda_0 = \min\{\lambda_k; \ k = 1, 2, \ldots, p\}$. Then, $\lim_{\lambda_0 \searrow 0} \phi(\mu; \mu, \Sigma) = \infty$, with $\phi(\mu; \mu, \Sigma) = O(\lambda_0^{-p/2})$ as $\lambda_0 \searrow 0$. On the other hand $\lim_{\lambda_0 \searrow 0} \phi(x; \mu, \Sigma) = 0$ for all $x \neq \mu$, with $\phi(x; \mu, \Sigma) = o(\lambda_0^q)$ for any fixed q as $\lambda_0 \searrow 0$. This implies that

$$\lim_{\lambda_0 \searrow 0} \ \phi(\mu; \mu, \Sigma) \phi(x; \mu, \Sigma) \to 0 \quad \text{for any } x \neq \mu$$

Furthermore, each of the density components in $\psi_\delta(\cdot)$ can be bounded above in terms of $\lambda_{\max}(\theta)$ and $\lambda_{\min}(\theta)$:

$$\phi(x; \mu_j, \Sigma_j) \leq (2\pi \lambda_{\min}(\theta))^{-\frac{p}{2}} \exp\left\{ -\frac{1}{2} \lambda_{\max}(\theta)^{-1} \|x - \mu_j\|^2 \right\} \leq (2\pi \lambda_{\min}(\theta))^{-\frac{p}{2}}. \tag{5}$$

The optimization problem in (3) requires that Θ_n is suitably defined, otherwise $\theta_n(\delta)$ may not exist. Consider a sequence $(\theta_m)_{m\in\mathbb{N}}$, as discovered by Kiefer and Wolfowitz (1956), the likelihood of a Gaussian mixtures degenerates if $\lambda_{1,1,m} \searrow 0$ if $\mu_{1,m} = x_1$, and this holds for (2), too. We here use the eigenratio constraint

$$\lambda_{\max}(\theta)/\lambda_{\min}(\theta) \leq \gamma < +\infty \tag{6}$$

with a constant $\gamma \geq 1$, where $\gamma = 1$ constrains all component covariance matrices to be spherical and equal, as in k-means clustering. This type of constraint has been proposed by Jr (1981), while Hathaway (1985) showed consistency of the scale-ratio constrained MLE for one-dimensional Gaussian mixtures. Ingrassia (2004) and Ingrassia and Rocci (2007) introduced EM algorithms for implementing these constraints for multivariate datasets. TCLUST by García-Escudero et al. (2008) and García-Escudero et al. (2014) also makes use of eigenratio constraints. Moreover there are a number of alternative constraints, see Ingrassia and Rocci (2011); Gallegos and Ritter (2009). It may be seen as a disadvantage of (6) that the resulting estimator will not be affine equivariant (this would require allowing $\lambda_{\max}(\theta)/\lambda_{\min}(\theta) \to \infty$ within any component). Affine equivariance can be achieved by defining a sphered version of the RIMLE as

$$\theta_n^*(\delta) = \theta_n^*(\delta, x_1, \ldots, x_n) = \theta_n(\delta, x_1^*, x_2^*, \ldots, x_n^*)$$

with $x_i^* = A(x_i - m),\ i = 1, \ldots, n$, where $S(x_1, \ldots, x_n)^{-1} = A'A$; $S(x_1, \ldots, x_n)$ could be the sample covariance matrix or another scale matrix and m the mean vector or another location estimator. This yields affine equivariance because the sphered versions of $\{x_1, \ldots, x_n\}$ and $\{Bx_1 + b, \ldots, Bx_n + b\}$ with some invertible $p \times p$-matrix B and $b \in \mathbb{R}^p$ are the same. Affine equivariance is not necessarily desirable though, see Hennig (2015a), Sec. 31.3.4.

This defines the parameter space

$$\tilde{\Theta} := \left\{\theta :\ \pi_j \geq 0\ \forall j \geq 1,\ \pi_0 + \sum_{j=1}^{G} \pi_j = 1;\ \frac{\lambda_{\max}(\theta)}{\lambda_{\min}(\theta)} \leq \gamma\right\}. \tag{7}$$

Occasionally, later, the notation $\|\theta\|$ will refer to the Euclidean norm of a vector pieced together from all the parameters collected in θ, in which all covariance matrices are interpreted as subvectors of all the matrix entries.

Although (6) ensures the boundedness of the likelihood in standard mixture models and TCLUST, for RIMLE this is not enough. The Gaussian components could degenerate on a few points and all other points could be fitted by the improper uniform component. Therefore we impose an additional constraint:

$$\frac{1}{n}\sum_{i=1}^{n} \tau_0(x_i, \theta) \leq \pi_{\max}, \tag{8}$$

for fixed $0 < \pi_{\max} < 1$. The quantity $n^{-1}\sum_{i=1}^{n} \tau_0(x_i, \theta)$ can be interpreted as the estimated proportion of noise points. This constraint depends on the dataset. Unfortunately the similar looking constraint $\pi_0 \leq \pi_{\max}$ independent of the data will not do, because this could not stop more than a portion of π_0 points to be fitted by the improper uniform component.

There is therefore a constrained effective parameter space for RIMLE estimation depending on the dataset:

$$\Theta_n := \left\{\theta \in \tilde{\Theta} :\ \pi_j \geq 0\ \forall j \geq 1,\ \pi_0 + \sum_{j=1}^{G} \pi_j = 1;\ \frac{1}{n}\sum_{i=1}^{n} \tau_0(x_i, \theta) \leq \pi_{\max};\ \frac{\lambda_{\max}(\theta)}{\lambda_{\min}(\theta)} \leq \gamma\right\}. \tag{9}$$

Analogously, existence and consistency of the RIMLE functional can only be showed on a parameter subset of $\tilde{\Theta}$ that depends on the underlying distribution and enforces that enough probability mass is fitted by Gaussian components rather than the improper uniform:

$$\Theta_G(P) := \left\{\theta \in \tilde{\Theta} :\ \pi_j \geq 0\ \forall j \geq 1;\ \pi_0 + \sum_{j=1}^{G} \pi_j = 1;\ \mathrm{E}_P \frac{\pi_0 \delta}{\psi_\delta(x, \theta)} \leq \pi_{\max};\ \frac{\lambda_{\max}(\theta)}{\lambda_{\min}(\theta)} \leq \gamma\right\}. \tag{10}$$

4. RIMLE existence and consistency

We first show existence of the RIMLE for finite samples Let $\#(A)$ denote the cardinality of the set A. Let $\underline{x_n} = \{x_1, x_2, \ldots, x_n\}$. Lemma 2 concerns the important case of plain Gaussian mixtures ($\delta = 0$) and requires a weaker assumption A0(a) for existence than A0 required for the RIMLE with $\delta > 0$. Here are some assumptions:

A0(a) $\#(\underline{x_n}) > G$.

A0 $\#(\underline{x_n}) > G + \lceil n\pi_{\max}\rceil$.

Lemma 2 *Assume A0(a), $\delta = 0$. Let $(\theta_m)_{m\in\mathbb{N}}$ be a sequence such that $\lambda_{\max}(\theta_m)/\lambda_{\min}(\theta_m) \leq \gamma$. Assume also that for some $j = 1, 2, \ldots, G$ and $k = 1, 2, \ldots, p$, $\lambda_{k,j,m} \searrow 0$ as $m \to \infty$; then* $\sup l_n(\theta_m) \to -\infty$.

Proof $\lambda_{k,j,m} \searrow 0$ implies $\lambda_{\max}(\theta_m) \searrow 0$, $\lambda_{\min}(\theta_m) \searrow 0$ at the same speed because of (6). Assume w.l.o.g. (otherwise consider a suitable subsequence) that $(\theta_m)_{m\in\mathbb{N}}$ is such that $\mu_{j,m},\ j = 1, 2, \ldots, G$ either leave every compact set for m large enough or converge, and assume w.l.o.g., that if their limits are in $\{x_1, \ldots, x_n\}$, they are in $\underline{x_G} = \{x_1, \ldots, x_G\}$. A0(a) implies that $\exists\ x_i \notin \underline{x_G}$, and $\exists \nu > 0$ such that for all such $x_i,\ j = 1, 2, \ldots, G$ and large enough $m : \|x_i - \mu_{j,m}\| \geq \nu$. Because the likelihood

$$\mathcal{L}_n(\theta_m) = \prod_{x_i \in \underline{x_G}} \left\{\sum_{j=1}^{G} \pi_j \phi(x_i; \mu_{j,m}, \Sigma_{j,m})\right\} \prod_{x_i \notin \underline{x_G}} \left\{\sum_{j=1}^{G} \pi_j \phi(x_i; \mu_{j,m}, \Sigma_{j,m})\right\}, \tag{11}$$

and Remark 1, the first product is of order $O(\lambda_{\min}(\theta_m)^{-p/2})^G$, and the second one of order $o(\lambda_{\min}(\theta_m)^q)$ for any fixed q, which implies that $\mathcal{L}_n(\theta_m) \to 0$ and $l_n(\theta_m) \to -\infty$. ■

Lemma 3 *Assume A0, $\delta > 0$. $(\theta_m)_{m\in N}$ is a sequence in Θ_n. Assume also that for some $j = 1, 2, \ldots, G$ and $k = 1, 2, \ldots, p$, $\lambda_{k,j,m} \searrow 0$ as $m \to \infty$. Then $l_n(\theta_m) \to -\infty$.*

Proof Using the definitions of the proof of Lemma 2, instead of (11) now

$$\mathcal{L}_n(\theta_m) = \prod_{x_i \in \underline{x_G}} \left\{ \pi_{0,m}\delta + \sum_{j=1}^{G} \pi_j \phi(x_i; \mu_{j,m}, \Sigma_{j,m}) \right\} \prod_{x_i \notin \underline{x_G}} \left\{ \pi_{0,m}\delta + \sum_{j=1}^{G} \pi_j \phi(x_i; \mu_{j,m}, \Sigma_{j,m}) \right\} \tag{12}$$

has to be considered, so that the limit behaviour of $(\pi_{0,m})_{m\in\mathbb{N}}$ is relevant. (8) implies

$$\frac{1}{n} \sum_{x_i \in \underline{x_G}} \left(1 + \sum_{j=1}^{G} \frac{\pi_j \phi(x_i, \mu_{j,m}, \Sigma_{j,m})}{\pi_{0,m}\delta} \right)^{-1} + \frac{1}{n} \sum_{x_i \notin \underline{x_G}} \left(1 + \sum_{j=1}^{G} \frac{\pi_j \phi(x_i, \mu_{j,m}, \Sigma_{j,m})}{\pi_{0,m}\delta} \right)^{-1} \leq \pi_{\max}. \tag{13}$$

Suppose that $(\pi_{0,m})_{m\in\mathbb{N}}$ does not converge to zero as $O(\phi(x_i, \mu_{j,m}, \Sigma_{j,m}))$ for at least one $x_i \notin \underline{x_G}$. For $m \to \infty$, the left term of (13) is ≥ 0, and the right term (at least a subsequence) converges to $\frac{\#(x_n \setminus \underline{x_G})}{n}$, which A0 requires to be $> \pi_{\max}$ with contradiction, thus $\pi_{0,m} = O(\phi(x_i, x_j, \Sigma_{j,m}))$. Therefore, by the same argument as in the proof of Lemma 2, the right product in (12) vanishes fast enough so that $l_n(\theta_m) \to -\infty$. ■

From these Lemmas:

Theorem 4 (Finite Sample Existence) *Assume A0. Then $\theta_n(\delta)$ exists for all $\delta \geq 0$.*

Proof Θ_n depends on δ via (8). Θ_n is not empty for any δ, because for any fixed values of the other parameters, small enough π_0 will fulfil (8). Next show that there exists a compact set $K_n \subset \Theta_n$ such that $\sup_{\theta\in K_n} l_n(\theta) = \sup_{\theta\in\Theta_n} l_n(\theta)$.
Step A: consider θ such that $\pi_1 = 1$, $\mu_1 = x_1$, $\Sigma_j = I_p$ for all $j = 1, 2, \ldots, G$, arbitrary μ_j and $\pi_j = 0$ for all $j \neq 1$. For this, $l_n(\theta) = \sum_{i=1}^{n} \log \phi(x_i; x_1, \Sigma_1) > -\infty$, thus $\sup_{\theta\in\Theta_n} l_n(\theta) > -\infty$.
Step B: consider a sequence $(\dot{\theta}_m)_{m\in N}$. It needs to be proved that if $(\dot{\theta}_m)_{m\in N}$ leaves a suitably chosen compact set K_n, it cannot achieve as large values of l_n as one could find within K_n. Lemma 3 (Lemma 2 for $\delta = 0$) rules out the possibility of any $\lambda_{k,j,m} \searrow 0$.
Step C: (5) implies that $l_n(\theta)$ can be bounded from above in terms of $\pi_0, \lambda_{\min}$ and δ:

$$l_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log \left(\pi_0\delta + \sum_{j=1}^{G} \pi_j \phi(x_i; \mu_j, \Sigma_j) \right) \leq \log \left(\pi_0\delta + (1-\pi_0)(2\pi\lambda_{\min}(\theta))^{-\frac{p}{2}} \right).$$

Consider $\dot{\theta} \in \Theta_n$ such that $\dot{\lambda}_{k,j} < +\infty$ for all $k = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, G$ (using the obvious notation of components of "dotted" parameter vectors). Also consider a sequence $(\ddot{\theta}_m)_{m\in\mathbb{N}}$ such that $\ddot{\theta}_m \to \ddot{\theta}$ where $\ddot{\theta}$ is equal to $\dot{\theta}$ except that $\ddot{\lambda}_{k,j,m} \to +\infty$ for some $k \in \{1, \ldots, p\}$ and $j \in \{1, \ldots, G\}$. By (6), $\lambda_{\min}(\ddot{\theta}_m) \to +\infty$ and thus $l_n(\ddot{\theta}_m) \to \log(\ddot{\pi}_0\delta)$. Clearly $\ddot{\pi}_0 < 1$ because otherwise $n^{-1}\sum_{i=1}^{n} \tau_0(x_i, \ddot{\theta}) = 1$, violating (8). Therefore $\lim_{m\to\infty} l_n(\ddot{\theta}_m) \leq l_n(\dot{\theta})$.
Step D: now consider $||\dot{\mu}_{j,m}|| \to +\infty$, for $j = 1$, w.l.o.g. Choose $\ddot{\theta}_m$ equal to $\dot{\theta}_m$ except now $\ddot{\mu}_{1,m} = 0$ for all m. Note that $\phi(x_i; \dot{\mu}_{1,m}, \dot{\Sigma}_1) \to 0$ for all $i = 1, 2, \ldots, n$, which implies that $\psi_\delta(x_i, \dot{\theta}_m) < \psi_\delta(x_i, \ddot{\theta}_m)$ for large enough m, for which then $l_n(\dot{\theta}_m) < l_n(\ddot{\theta}_m)$. Applying this argument to all j with $||\dot{\mu}_{j,m}|| \to +\infty$ shows that better l_n can be achieved inside a compact set.

Continuity of l_n now guarantees existence of $\theta_n(\delta)$. ■

We now derive consistency for the sequence $\{\theta_n(\delta)\}_{n\in\mathbb{N}}$ as estimator of L_G. Consistency of the RIMLE can be achieved only if L_G exists. In order to ease the notation we define $\eta(x, \theta) = \sum_{j=1}^{G} \pi_j \phi(x; \mu_j, \Sigma_j)$. Consider the following assumptions on P:

A1 For every $x_1, \ldots, x_G \in \mathbb{R}^p : P\{x_1, \ldots, x_G\} < 1 - \pi_{max}$.

A2 $L_G > L_{G-1}$, where for $G \in \mathbb{N}$ let $L_G = \sup_{\theta\in\Theta_G(P)} L(\theta)$.

A3 There exist $\epsilon_1, \epsilon_2 > 0$ so that for every θ with $\pi_0 < \epsilon_1 : L(\theta) \leq L_G - \epsilon_2$.

Remark 5 *Assumption A1 requires that no set of G points carries probability $1 - \pi_{\max}$ or more. Otherwise the log-likelihood can be driven to ∞ by fitting G mixture components to G points with all covariance matrix eigenvalues converging to zero. The improper noise component could take care of all other points.*

Note that assumptions A2 and A3 are not both required, but only any single one of them. A2 states that G mixture components fit the data better than $G-1$ components. If this is not the case, there is at least one redundant component, and one cannot make sure that $L(\theta)$ is bounded away from L_G for large n in some distance from the "true" RIMLE-functional as the redundant component can be moved around, see Theorem 11. In case that A2 is not fulfilled, a weaker result can still be achieved, namely the existence of a not necessarily unique consistent sequence of local maximizers of l_n. This requires A3, which states that a noise proportion bounded away from zero is required for maximizing L. If neither A2 nor A3 are fulfilled, P can be fitted perfectly by fewer than G mixture components and no noise. In this case one cannot stop the remaining mixture components from leaving every compact set, and therefore one cannot expect consistency of all components for any method; as long as there is still noise bounded away from zero, those mixture components still can contribute to fitting what otherwise would be noise, and fits become worse if these degenerate.

Note that this is less often the case than one might expect; for example, a plain Gaussian mixture with $G-1$ components may still fulfill A3: if the density of one of the components is uniformly smaller than δ, a better pseudo-likelihood can obviously be achieved by assigning its proportion to the noise component than by choosing $\pi_0 = \pi_G = 0$ and otherwise the true parameters. A Gaussian component that "looks like noise" rather than like a "cluster" will be treated as noise.

Lemma 6 *For any probability measure P on $\mathbb{R}^p$, $L_G > -\infty$.*

Proof Choose compact $K \subset \mathbb{R}^p$ with $P(K) > 0$. Let $q = 1-P(K)$ and choose K big enough that $\pi_{\max} > q$. Choose $\mu_1 = \mathrm{E}_P(x|x \in K)$, $\Sigma_1 = \mathrm{Cov}_P(x|x \in K)$, $\pi_2 = \ldots = \pi_G = 0$. If all eigenvalues of Σ_1 are zero, choose $\Sigma_1 = I_p$. Let $\lambda_{\max,1}$ be the largest eigenvalue of Σ_1. If $\lambda_{\max,1}/\lambda_{i,1} > \gamma$ for any eigenvalue $\lambda_{i,1}$ of Σ_1, modify Σ_1 by replacing all eigenvalues smaller

than $\gamma\lambda_{\max,1}$ by $\gamma\lambda_{\max,1}$ in its spectral decomposition. Let $\phi_{\min} = \min_{x\in K}\phi(x,\mu_1,\Sigma_1) > 0$. Choose

$$\pi_0 = \frac{(\pi_{\max}-q)\phi_{\min}}{2((1-\pi_{\max})\delta + (\pi_{\max}-q)\phi_{\min})} > 0,\ \pi_1 = 1-\pi_0.$$

Observe that the resulting $\theta \in \Theta_G(P)$ (with all other parameters chosen arbitrarily) by

$$\mathrm{E}_P\frac{\pi_0\delta}{\psi_\delta(x,\theta)} = \int_K \frac{\pi_0\delta}{\pi_0\delta+\eta(x,\theta)}dP(x) + \int_{K^c}\frac{\pi_0\delta}{\pi_0\delta+\eta(x,\theta)}dP(x) \leq$$
$$\leq (1-q)\frac{\pi_0\delta}{\pi_0\delta+(1-\pi_0)\phi_{\min}} + q.$$

This is smaller than $\pi_{\max}$ if $\pi_0 < \frac{(\pi_{\max}-q)\phi_{\min}}{(1-\pi_{\max})c+(\pi_{\max}-q)\phi_{\min}}$. Furthermore, $L(\theta) \geq (1-q)\log(\pi_0\delta+(1-\pi_0)\phi_{\min}) + q\log(\pi_0\delta) > -\infty$. ■

Lemma 7 *Assume A1. There are $\lambda^*_{\min} > 0,\ \lambda^*_{\max} < \infty,\ \epsilon > 0$, so that*

(a) *$L(\theta) \leq L_G - \epsilon$ for every θ with $\lambda_{\min}(\theta) < \lambda^*_{\min}$ or $\lambda_{\max}(\theta) > \lambda^*_{\max}$,*

(b) *for $x_1, x_2, \ldots$ i.i.d. with $\mathcal{L}(x_1) = P$, for sequences $(\theta_n)_{n\in\mathbb{N}}$ with $\lambda_{\min}(\theta_n) < \lambda^*_{\min}$ or $\lambda_{\max}(\theta_n) > \lambda^*_{\max}$ for large enough n: $l_n(\theta_n) \leq l_n(\theta_n(\delta)) - \epsilon$ a.s.*

Proof Start with part (a). First consider a sequence $\{\theta_m\}_{m\in\mathbb{N}} \in \Theta_G(P)^{\mathbb{N}}$ with $\lambda_{\max}(\theta_m) \to \infty$. The eigenvalue ratio constraint forces all covariance matrix eigenvalues to infinity, and therefore $\sup_x \phi(x,\mu_{j,m},\Sigma_{j,m}) \searrow 0$. But this means that $\mathrm{E}_P\frac{\pi_0\delta}{\psi_\delta(x,\theta)} \to 1 > \pi_{\max}$ and $\theta_m \notin \Theta_G(P)$ eventually, unless $\pi_{0,m} \searrow 0$, too. If the latter is the case, $\psi_\delta(x,\theta) \searrow 0$ uniformly over all x and $L(\theta_m) \searrow -\infty$, which together with Lemma 6 makes it impossible that $L(\theta_m)$ is close to L_G for m large enough and $\lambda_{\max}(\theta_m)$ too large, proving the existence of the upper bound $\lambda^*_{\max} < \infty$ as required.
Now consider a sequence $\{\theta_m\}_{m\in\mathbb{N}} \in \Theta_G(P)^{\mathbb{N}}$ with $\lambda_{\min}(\theta_m) \to 0$. Define

$$A_{m,\epsilon} = \left\{x : \min_{j=1,2,\ldots,G}\|x-\mu_{j,m}\| > \epsilon\right\}.$$

A1 ensures that for $0 < \epsilon_3$ there exists $\epsilon > 0$ so that for all $m \in \mathbb{N} : P(A_{m,\epsilon}) \geq \pi_{\max} + \epsilon_3$. Based on (5) derive an upper bound for $\pi_{0,m}$ from the constraint $\int \frac{\pi_{0,m}\delta}{\pi_{0,m}\delta+\eta(x,\theta_m)}dP(x) \leq \pi_{\max}$:

$$\int \frac{\pi_{0,m}\delta}{\pi_{0,m}\delta+\eta(x,\theta_m)}dP(x) = \int_{A_{m,\epsilon}}\frac{\pi_{0,m}\delta}{\pi_{0,m}\delta+\eta(x,\theta_m)}dP(x) + \int_{A^c_{m,\epsilon}}\frac{\pi_{0,m}\delta}{\pi_{0,m}\delta+\eta(x,\theta_m)}dP(x) \geq$$
$$\geq P(A_{m,\epsilon})\frac{\pi_{0,m}\delta}{\pi_{0,m}\delta+\max_{x\in A_{m,\epsilon}}\eta(x,\theta_m)},$$

which by (5) implies

$$\pi_{0,m} \leq \frac{\pi_{\max}\max_{x\in A_{m,\epsilon}}\eta(x,\theta_m)}{\delta(P(A_{m,\epsilon})-\pi_{\max})} \leq \frac{\pi_{\max}\exp(-\frac{\epsilon^2}{2\gamma\lambda_{\min}(\theta_m)})}{\delta\epsilon_3(2\pi)^{p/2}\lambda_{\min}(\theta_m)^{p/2}}.$$

For the log-likelihood,

$$L(\theta_m) = \int_{A_{m,\epsilon}}\log(\pi_{0,m}\delta+\eta(x,\theta_m))dP(x) + \int_{A^c_{m,\epsilon}}\log(\pi_{0,m}\delta+\eta(x,\theta_m))dP(x) \leq$$
$$\leq \int_{A_{m,\epsilon}}\log\left(\frac{\delta\pi_{\max}\exp(-\frac{\epsilon^2}{2\gamma\lambda_{\min}(\theta_m)})}{\delta\epsilon_3(2\pi)^{p/2}\lambda_{\min}(\theta_m)^{p/2}} + \frac{\exp(-\frac{\epsilon^2}{2\gamma\lambda_{\min}(\theta_m)})}{(2\pi)^{p/2}\lambda_{\min}(\theta_m)^{p/2}}\right)dP(x) +$$
$$+ \int_{A^c_{m,\epsilon}}\log\left(\frac{\delta\pi_{\max}\exp(-\frac{\epsilon^2}{2\gamma\lambda_{\min}(\theta_m)})}{\delta\epsilon_3(2\pi)^{p/2}\lambda_{\min}(\theta_m)^{p/2}} + \frac{1}{(2\pi)^{p/2}\lambda_{\min}(\theta_m)^{p/2}}\right)dP(x) \leq$$
$$\leq P(A_{m,\epsilon})\left(-\frac{c_1}{\lambda_{\min}(\theta_m)} - c_2\log(\lambda_{\min}(\theta_m)) + c_3\right) +$$
$$+ P(A^c_{m,\epsilon})\left(o(1) - c_4\log(\lambda_{\min}(\theta_m)) + c_5\right) =$$
$$= -\frac{c_6}{\lambda_{\min}(\theta_m)} - c_7\log(\lambda_{\min}(\theta_m)) + c_8$$

for positive constants c_1, c_2, c_4, c_6, c_7 and constants c_3, c_5, c_8, all independent of θ_m. If $\lambda_{\min}(\theta_m) \searrow 0$, this implies $L(\theta_m) \searrow -\infty$, proving together with Lemma 6 the existence of the lower bound $\lambda^*_{\min} > 0$.

Part (b) holds because if $(\theta_m)_{m\in\mathbb{N}}$ is chosen as above for $m = n \to \infty$ and P is replaced by the empirical distribution P_n, Glivenko-Cantelli enforces $P_n(A_{n,\epsilon}) - P(A_{n,\epsilon}) \to 0$ a.s. Glivenko-Cantelli applies because the class of all $A_{n,\epsilon}$ is a subset of the class of intersections of the complements of all closed balls, and therefore a Vapnik-Chervonenkis class, see van der Vaart and Wellner (1996). The argument carries over using all other integrals in the finite sample-form, i.e., w.r.t. P_n. Lemma 6 carries over because $l_n(\theta) \to L(\theta)$ a.s. by the strong law of large numbers for θ with $L(\theta) > -\infty$. ■

Remark 8 *Lemma 7 (a) and (5) imply that for all θ with $L(\theta) > L_G - \epsilon$, $j = 1, 2, \ldots, G$ and all x:*

$$\phi(x,\mu_j,\Sigma_j) \leq \phi_{\max} = \frac{1}{(2\pi)^{p/2}(\lambda^*_{\min})^{p/2}}.$$

This implies $L_G < \infty$.

The same holds because of Lemma 7 (b) for $x_1, x_2, \ldots$ i.i.d. with $\mathcal{L}(x_1) = P$ for large enough n a.s. for all θ with $l_n(\theta) > l_n(\theta_n(\delta)) - \epsilon$.

Lemma 9 *Assume A1 and A2. There is a compact set $K \subset \mathbb{R}^p$ so that*

(a) *L reaches its supremum for $\mu_1, \ldots, \mu_G \in K$ and is bounded away from the supremum if not all of $\mu_1, \ldots, \mu_G \in K$ (i.e., $\exists\epsilon_4 > 0$ so that L in this case is bounded from above by $\sup L - \epsilon_4$),*

(b) *for $x_1, x_2, \ldots$ i.i.d. with $\mathcal{L}(x_1) = P$ for large enough n, l_n reaches its supremum for $\mu_1, \ldots, \mu_G \in K$ and is bounded away from the supremum if not all of $\mu_1, \ldots, \mu_G \in K$, a.s.*

Proof Start with part (a). Consider a sequence $\{\theta_m\}_{m\in\mathbb{N}} \in \Theta_G(P)^{\mathbb{N}}$ with $\|\mu_{jm}\| \to \infty$ for $j = 1,\ldots,k,\ 1 \le k < G$ and a compact set K with $\mu_{jm} \in K$ for $j > k$. Let

$$A_m = \left\{ x:\ \forall j \in \{1,\ldots,k\} : \phi(x,\mu_{j,m},\Sigma_{j,m}) \le \epsilon_m \sum_{l=k+1}^{G} \pi_{l,m}\phi(x,\mu_{l,m},\Sigma_{l,m}) \right\},$$

where $\epsilon_m \searrow 0$ slowly enough that $P(A_m) \to 1$. Let $\pi_m^* = \sum_{j=1}^{k}\pi_{j,m}$. Let $\theta_{(G-k),m}$ for $m \in \mathbb{N}$ be defined by $\pi_{0,(G-k),m} = \pi_{0,m},\ \pi_{(j-k),(G-k),m} = \pi_{j,m}(1-\pi_{0,m})(1-\pi_m^*-\pi_{0,m})^{-1}$ for $j = k+1,\ldots,G$ accompanied by the μ_j,Σ_j-parameters belonging to the components $k+1,\ldots,G$ of θ_m. Observe, using Remark 8,

$$\begin{aligned} L(\theta_m) &= \int_{A_m} \log(\psi_\delta(x,\theta_m))dP(x) + \int_{A_m^c} \log(\psi_\delta(x,\theta_m))dP(x) \le \\ &\le \int_{A_m} \log((1+\epsilon_m)\psi_\delta(x,\theta_{(G-k)m}))dP(x) + P(A_m^c)\log(\delta+\phi_{max}) \end{aligned}$$

implying $L(\theta_{(G-k)m}) - L(\theta_m) \to 0$. $\mathrm{E}_P[\pi_{0,(G-k),m}\delta(\psi_\delta(x,\theta_{(G-k)m}))^{-1}] \le \pi_{\max}$ will be fulfilled for m large enough because it is fulfilled for θ_m by definition and $\psi_\delta(x,\theta_{(G-k),m}) > \psi_\delta(x,\theta_m)$ on A_m with $P(A_m) \to 1$. $L(\theta_{(G-k)m}) < L_{G-1}$ implies that, because of A2, θ_m is bounded away from L_G.

Regarding existence of a maximum with $\mu_1,\ldots,\mu_G \in K$, observe that with Remark 8, $\psi_\delta(x,\theta)$ can be bounded by $\delta+\phi_{max}$ for all θ for which $L(\theta) > L_G - \epsilon$. Now consider a sequence $(\theta_m)_{m\in\mathbb{N}}$ so that $\forall m:\ \mu_{1m},\ldots,\mu_{Gm} \in K$, with the notation of Lemma 7, $\lambda_{min}^* \le \lambda_{min}(\theta_m) \le \lambda_{max}(\theta_m) \le \lambda_{max}^*$ and $L(\theta_m) \to L_G$. Because of compactness, w.l.o.g., $\theta_m \to \theta_+$ and, using Fatou's Lemma, $L_G = \lim_{m\to\infty} L(\theta_m) \le E_P \limsup_m \psi_\delta(x,\theta_m) = L(\theta_+) \le L_G$.

Part (b) holds because if $(\theta_m)_{m\in\mathbb{N}}$ is chosen as above for $m = n \to \infty$ and P is replaced by the empirical distribution P_n, Glivenko-Cantelli enforces $P_n(A_n) - P(A_n) \to 0$ a.s. Glivenko-Cantelli applies here because a sequence of closed balls $(B_n)_{n\in\mathbb{N}}$ can be constructed so that $B_n \subseteq A_n,\ P(B_n) \to 1$ a.s.; the closed balls are a Vapnik-Chervonenkis class, and $P_n(A_n) \ge P_n(B_n) \to 1$ a.s. Furthermore, for $\theta \in \Theta_G(P)$ with $L(\theta) > L_{G-1}$: $l_n(\theta) \to L(\theta)$ a.s. by the strong law of large numbers, so that for large enough n: $\sup_{\theta\in\Theta_G(P)} l_n(\theta) > L_{G-1}$ a.s. On the other hand, $\theta_{(G-k),n}$ can be chosen optimally in a compact set K because of Lemma 7, within which l_n converges uniformly to L a.s. (Theorem 2 in Jennrich (1969)), and therefore, $\limsup_{n\to\infty} l_n(\theta_{(G-k),n}) \le L_{G-1}$. With these ingredients, the argument of part (a) carries over. ■

Lemma 10 *Assume A1 and A3. There is a compact set $K \subset \mathbb{R}^p$ so that*

(a) *L reaches its supremum for $\mu_1,\ldots,\mu_G \in K$,*

(b) *for $x_1,x_2,\ldots$ i.i.d. with $\mathcal{L}(x_1) = P$, there exists a sequence $(\tilde{\theta}_n)_{n\in\mathbb{N}}$ maximizing l_n locally for $\mu_1,\ldots,\mu_G \in K$ so that $l_n(\tilde{\theta}_n) \to L_G$ a.s., and a.s. there is no sequence $\theta_n \in \Theta_G(P)$ so that $\limsup_{n\to\infty} l_n(\theta_n) > L_G$.*

Proof Start with part (a). Consider a sequence $\{\theta_m\}_{m\in\mathbb{N}} \in \Theta_G(P)^{\mathbb{N}}$ with $\|\mu_{j,m}\| \to \infty$ for $j = 1,\ldots,k,\ 1 \le k < G$ (the case $k = G$ is treated at the end), and a compact set K with $\mu_{j,m} \in K$ for $j > k$. By selecting a subsequence if necessary, assume that there exists $\mu_j = \lim_{m\to\infty}\mu_{j,m}$, $\Sigma_j = \lim_{m\to\infty}\Sigma_{j,m}$ for $k+1 \le j \le G$ and that $\pi_{j,m}$ converge for $j = 0,\ldots,G$. Let $j^* = \arg\max_{k+1\le j\le G} \mathrm{E}_P\phi(x,\mu_j,\Sigma_j)$. Suppose $L(\theta_m) \nearrow L_G$ monotonically and, by A3, assume $\pi_{0,m} > \epsilon_1$.

Consider first the case $\sum_{j=1}^{k}\pi_{j,m} \to \epsilon_3 > 0$. Construct another sequence $\{\theta_{*m}\}_{m\in\mathbb{N}} \in \Theta_G(P)^{\mathbb{N}}$ for which $\mu_{1,*m} = \ldots = \mu_{k,*m} = \mu_{j^*} \in K$, $\Sigma_{1,*m} = \ldots = \Sigma_{k,*m} = \Sigma_{j^*}$. All other parameters are the same as in θ_m. Let $A_m = \{x:\ \forall j \in \{1,\ldots,k\} : 2\phi(x,\mu_{j,m},\Sigma_{j,m}) \le \phi(x,\mu_{j^*},\Sigma_{j^*})\}$. Observe $P(A_m) \to 1$. Now

$$L(\theta_{*m}) - L(\theta_m) = \int_{A_m} \log\frac{\psi_\delta(x,\theta_{*m})}{\psi_\delta(x,\theta_m)}dP(x) + \int_{A_m^c} \log\frac{\psi_\delta(x,\theta_{*m})}{\psi_\delta(x,\theta_m)}dP(x). \tag{14}$$

For large enough m,

$$\int_{A_m} \log\frac{\psi_\delta(x,\theta_{*m})}{\psi_\delta(x,\theta_m)}dP(x) \ge \epsilon_5 > 0,$$

whereas (using Remark 8)

$$\int_{A_m^c} \log\frac{\psi_\delta(x,\theta_{*m})}{\psi_\delta(x,\theta_m)}dP(x) \ge P(A_m^c)\log\frac{\epsilon_1\delta}{\delta+\phi_{\max}} \to 0. \tag{15}$$

Therefore $L(\theta_{*m}) - L(\theta_m) > 0$ for large enough m so that $L(\theta_m)$ is improved by a θ with $\mu_j \in K$ for $j = 1,2,\ldots,G$.

Consider now $\sum_{j=1}^{k}\pi_{jm} \to 0$. Construct another sequence $\{\theta_{*m}\}_{m\in\mathbb{N}} \in \Theta_G(P)^{\mathbb{N}}$ for which $\mu_{1,*m} = \ldots = \mu_{k,*m} = \mu_{j^*} \in K$, $\Sigma_{1,*m} = \ldots = \Sigma_{k,*m} = \Sigma_{j^*}$, $\pi_{1*m} = \ldots = \pi_{k*m} = 0$, $\pi_{0*m} = \sum_{j=0}^{k}\pi_{jm}$, all other parameters taken from θ_m. Set $A_m = \{x:\ \forall j \in \{1,\ldots,k\} : \phi(x,\mu_{jm},\Sigma_{jm}) < \delta\}$. Again $P(A_m) \to 1$. With this, (14) holds again. This time

$$\int_{A_m} \log\frac{\psi_\delta(x,\theta_{*m})}{\psi_\delta(x,\theta_m)}dP(x) > 0$$

and again (15).

Let $\theta_* = \lim_{m\to\infty}\theta_{*m}$ (this exists by construction). Continuity of L implies that $L(\theta_*) = L_G$ and therefore for all $m: L(\theta_*) \ge L(\theta_m)$. $\sum_{j=1}^{k}\pi_{j,m} \to 0$ is required here because $\pi_{0,*m} \ge \pi_{0,m}$ does not necessarily fulfill $\mathrm{E}_P\frac{\pi_{0,*m}c}{\psi_\delta(x,\theta_{*m})} \le \pi_{\max}$, but $\lim_{m\to\infty}\pi_{0,*m} = \lim_{m\to\infty}\pi_{0,m}$ does.

Finally, consider $k = G$. With $A_{m,\epsilon} = \{x:\ \forall j \in \{1,\ldots,k\} : \phi(x,\mu_{j,m},\Sigma_{j,m}) < \delta\epsilon\}$, observe

$$\mathrm{E}_P\frac{\pi_{0,m}\delta}{\psi_\delta(x,\theta_m)} \ge P(A_{m,\epsilon})\frac{\pi_{0,m}\delta}{\pi_{0,m}(c+\epsilon)} > \pi_{\max},$$

for small enough ϵ and large enough m, violating for large m the corresponding constraint in $\Theta_G(P)$ as long as $\pi_{0,m}$ is bounded from below, as was assumed. Existence follows in the same way as in the proof of Lemma 9.

For part (b) let θ^* have $\mu_1^*,\ldots,\mu_G^* \in K$ and $L(\theta^*) = L_G$, which exists because of part (a) and Lemma 7, which ensures further that θ^* is in a compact K^*. Then the strong law

of large numbers yields $l_n(\theta^*) \to L_G$ a.s., and Theorem 2 of Jennrich (1969) implies that for all sequences $(\theta_n)_{n\in\mathbb{N}} \in (K^*)^{\mathbb{N}} : \limsup_{n\to\infty} l_n(\theta_n) \leq L_G$. This also holds for sequences $(\theta_n)_{n\in\mathbb{N}}$ that are eventually outside K^* because of part (a) of Lemma 7 and the proof of part (a) above, because if $(\theta_m)_{m\in\mathbb{N}}$ is chosen as above for $m = n \to \infty$ and P is replaced by the empirical distribution P_n, Glivenko-Cantelli (which applies by the same argument as used in the proof of Lemma 9) enforces $P_n(A_n) - P(A_n) \to 0$ a.s., which means that as in part (a), a.s., eventually $l_n(\theta_n)$ cannot converge to anything larger than L_G. ■

Theorem 11 (RIMLE existence) *Assume A1 and any one of A2 or A3. There is a compact subset $K \subset \Theta_G(P)$ so that there exists $\theta \in K : \ \infty > L(\theta) = L_G > -\infty$. Assuming A2, for $\theta \notin K$, $L(\theta)$ is bounded away from L_G.*

Proof Pieced together from Lemmas 6-9 parts (a) and Remark 8. ■

Theorem 11 establishes existence of the RIMLE functional

$$\theta^\star(\delta) = \underset{\theta\in\Theta_G(P)}{\arg\max}\, L(\theta). \tag{16}$$

Unfortunately neither $L(\theta)$ nor $l_n(\theta)$ can be expected to have a unique maximum. If we take the vector θ and we permute some of the triples (π_j, μ_j, Σ_j) we still obtain the same value for $L(\theta)$ and $l_n(\theta)$. This known as "label switching" in the mixture literature. There could be other causes for multiple maxima. Without strong restrictions on P, we cannot identify any specific source of multiple optima in the target function. Instead we show that asymptotically the sequence of estimators is close to some maximum of the pseudo-loglikelihood, which amounts to consistency of the RIMLE with respect to a quotient space topology identifying all loglikelihood maxima, as done in Redner (1981). By $\theta^\star(\delta)$ in (16) we mean any of the maximizer of $L(\theta)$. Define the sets

$$S(\dot\theta) = \left\{\theta \in \Theta_G(P) : \int \log\psi_\delta(x;\theta)dP(x) = \int \log\psi_\delta(x;\dot\theta)dP(x)\right\},$$

$$\mathcal{K}(\dot\theta,\varepsilon) = \left\{\theta \in \Theta_G(P) : \|\theta - \ddot\theta\| < \varepsilon \ \forall\ \ddot\theta \in S(\dot\theta)\right\}, \quad \text{for any} \quad \varepsilon > 0.$$

The following theorem makes a stronger statement assuming A2 than A3, because if A2 does not hold, the Gth mixture component is asymptotically not needed and cannot be controlled for finite n outside a compact set.

Theorem 12 (Consistency) *Assume A1 and A2. Then for every $\varepsilon > 0$ and every sequence of maximizers $\theta_n(\delta)$ of l_n:*

$$\lim_{n\to\infty} P\left\{\theta_n(\delta) \in \mathcal{K}(\theta^\star(\delta),\varepsilon)\right\} = 1.$$

Assuming A3 instead of A2, for every compact $K \supset \mathcal{K}(\theta^\star(\delta),\varepsilon)$ there exists a sequence of θ_n that maximize l_n locally in K so that

$$\lim_{n\to\infty} P\left\{\theta_n \in \mathcal{K}(\theta^\star(\delta),\varepsilon)\right\} = 1.$$

Proof Under A2, because of the parts (b) of the Lemmas 7 and 9 it can be assumed that there is a compact set K so that all $\theta_n(\delta) \in K$ for large enough n a.s. Under A3, considerations are restricted to K anyway.

Based on Theorem 11 and related Lemmas $|\log\psi_\delta(x,\theta)| \leq C$ for some finite constant C for all $\theta \in K$. Sufficient conditions for Theorem 2 in Jennrich (1969) are satisfied, which implies uniform convergence of $l_n(\theta)$, that is $\sup_{\theta\in K}|l_n(\theta) - L(\theta)| \to 0$ P-a.s. Based on the latter, and applying the same argument as in proof of Theorem 5.7 in van der Vaart (2000), it holds true that $L(\theta_n(\delta)) \to L(\theta^\star(\delta))$ P-a.s. By continuity of $L(\theta)$ and Theorem 11 we have that for every $\varepsilon > 0$ there exists a $\beta > 0$ such that $L(\theta^\star(\delta)) - L(\theta) > \beta$ for all $\theta \in K \setminus \mathcal{K}(\theta^\star(\delta),\varepsilon)$. Denote $(\Omega, \mathcal{A}, P)$ the probability space where the sample random variables are defined and consider the following events

$$A_n = \{\omega \in \Omega : \ \theta_n(\delta) \in K \setminus \mathcal{K}(\theta^\star(\delta),\varepsilon)\},$$

and

$$B_n = \{\omega \in \Omega : \ L(\theta^\star(\delta)) - L(\theta_n(\delta)) > \beta\}.$$

Clearly $A_n \subseteq B_n$ for all n. $P(B_n) \to 0$ for $n \to \infty$ implies $P(A_n) \to 0$. The latter proves the result. ■

5. Algorithms and practical issues

Following the presentation of the proposed algorithm to compute the RIMLE, we discuss its initialization and tuning.

5.1 RIMLE computing

In this section we develop Expectation–Maximization type algorithms (EM) to compute the RIMLE (for a fixed δ). Let $s = 0, 1, \ldots$ be the iteration index. Let $a^{(s+1)}$ be the quantity a computed at the sth step of the EM algorithm. Define

$$\begin{aligned} Q(\theta,\theta^{(s)}) = &\sum_{i=1}^{n}\sum_{j=0}^{G}\tau_j(x_i,\theta^{(s)})\log\pi_j + \sum_{i=1}^{n}\tau_0(x_i,\theta^{(s)})\log\delta + \\ &+ \sum_{i=1}^{n}\sum_{j=1}^{G}\tau_j(x_i,\theta^{(s)})\log\phi(x_i;\mu_j,\Sigma_j). \end{aligned} \tag{17}$$

Increasing (17) by an appropriate choice of θ increases $l_n(\cdot)$. An approximate candidate maximum of $l_n(\theta)$ can be found by the EM Algorithm 1.

Proposition 13 *Assume A0. The sequence $\{\theta^{(s)}\}_{s\in\mathbb{N}}$ produced by Algorithm 1 converges to a point $\theta_n^{em} \in \Theta$, and $l_n(\theta^{(s)})$ is increased in every step.*

Proof Find a set $A_n \subset \mathbb{R}^p$ that contains all points in $\underline{x_n}$ with Lebesgue measure $M(A_n) = 1/\delta$. δ is then a proper uniform density function on A_n. Hence, for a given dataset the pseudo-density $\psi_\delta(\cdot)$ can be written as proper density function. Therefore the convergence

Algorithm 1: EM-algorithm

input : $\{x_1, x_2, \ldots, x_n\}$, δ, $\pi_{\max}$, γ, $\theta^{(0)}$, tol¿0
output : θ^{em}

while $|l_n(\theta^{(s+1)}) - l_n(\theta^{(s)})| >$tol **do**
  E–step: compute $\tau_j(x_i, \theta^{(s)})$, for all $i = 1, 2, \ldots, n$ and $j = 0, 1, \ldots, G$
  M–step: $\theta^{(s+1)} \leftarrow \arg\max_{\theta\in\Theta} Q(\theta, \theta^{(s)})$
end
$\theta^{\text{em}} \leftarrow \theta^{(s+1)}$

Theorem 4.1 in Redner and Walker (1984) holds, with $Q(\theta, \theta^{(s)})$ playing the role of their $Q(\cdot)$ function. ■

Algorithm 1 is the analog of the EM algorithm for plain Gaussian mixtures (see Redner and Walker, 1984) except that now the M-step is a constrained optimization. Wu (1983) showed that the EM algorithm converges to the global maximum if the likelihood function is unimodal and certain differentiability conditions are satisfied. In general the limit of the EM algorithm is not guaranteed to coincide with a global maximum of likelihood function. However, Proposition 13 guarantees that θ_n^{em} is a stationary point of $l_n(\cdot)$. Running the EM algorithm for a large number of starting values increases the chances of finding the optimal solution. For finite Gaussian mixtures models it is well known that the likelihood surface is difficult to explore even when p is not too large, and the main advantage of the EM algorithm is that the M-step can be divided in a number of simpler optimization problems each of which has a closed form solution. However, for the RIMLE the constraints add some complexity, and in particular the noise proportion constraint does not allow to separate the M-step in simpler subprograms. One possibility is to perform the M-step using numerical optimization packages, but the eigenratio constraints requires to parameterize each Σ_j terms of its spectral components. The latter has the drawback to add $G{\times}p(1-p)/2$ parameters. Furthermore, the eigenratio constraint has a non-smooth nature that would make numerical techniques hard to adapt.

In Coretto and Hennig (2016) computations are based on Algorithm 1 where the M-step is performed as if the problem would be unconstrained, and breaking the iteration when updates drive the parameters outside the constrained parameter space. Coretto and Hennig (2016) also propose a heuristic method to enforce the constraints at the end of the iterations if necessary. Of course in such situations there would be no guarantee that the delivered solution is a stationary point of $l_n(\cdot)$. Here, we propose Algorithm 2 where constraints are applied exactly in each iteration. The M-step in Algorithm 1 is replaced with two conditional maximization (CM) steps. This transforms Algorithm 1 into an Expectation-Conditional Maximization algorithm (ECM) as introduced by Meng and Rubin (1993). For ease of notation, for $j = 0, 1, \ldots, G$ define $\tau_{i,j}^{(s)} = \tau_j(x_i, \theta^{(s)})$ and $T_j^{(s)} = \sum_{i=1}^n \tau_j(x_i, \theta^{(s)})$. Rewrite (17), using $\theta_1 = (\mu_1, \ldots, \mu_G, \text{vect}(\Sigma_1), \ldots, \text{vect}(\Sigma_G))$, $\theta_2 = (\pi_0, \pi_1, \ldots, \pi_G)'$, and $\theta = (\theta_1, \theta_2)'$, as

$$Q(\theta_1, \theta_2, \theta^{(s)}) = Q_1(\theta_1, \theta^{(s)}) + Q_2(\theta_2, \theta^{(s)}) + \text{const}, \tag{18}$$

where

$$Q_1(\theta_1, \theta^{(s)}) = \sum_{i=1}^n \sum_{j=1}^G \tau_{i,j}^{(s)} \log \phi(x_i; \mu_j, \Sigma_j), \quad Q_2(\theta_2, \theta^{(s)}) = \sum_{j=0}^G T_j^{(s)} \log \pi_j,$$

and const $= T_0^{(s)} \log(\delta)$ which does not depend on θ. Consider the following programs:

$$\underset{\theta_1}{\text{maximize}}\, Q_1(\theta_1, \theta^{(s)}) \quad \text{subject to} \quad \frac{\lambda_{\max}(\theta_1)}{\lambda_{\min}(\theta_1)} \leq \gamma, \tag{CM1}$$

and

$$\underset{\theta_2}{\text{maximize}}\, Q_2(\theta_2, \theta^{(s)}) \quad \text{subject to} \quad \sum_{j=0}^G \pi_j = 1,\ \sum_{i=1}^n \tau_0(x_i, \theta_1^{(s+1)}, \theta_2) \leq n\pi_{\max}. \tag{CM2}$$

The ECM algorithm consists of solving (**CM1**) and then (**CM2**). The sequence of optimizations replaces the M-step in Algorithm 1. Notice that in (**CM2**), $\pi_j = 0$ for some $j = 0, 1, \ldots, G$ would drive the objective function toward $-\infty$, so we do not need to restrict the π_j as > 0. $T_j^{(s)} = 0$ will not happen, see Remark 17. Also notice that for δ=0 the noise proportion constraint is automatically fulfilled, and for more analysis on these cases see Remark 17.

Before presenting the ECM Algorithm 2 we introduce additional notations. For $a \in R^d$ let diag(a) be the $d{\times}d$ diagonal matrix with elements of a on the main diagonal. For a matrix A let $\text{Spec}(A) = \Gamma\Lambda\Gamma'$ be the spectral decomposition of A, that is, Γ contains the normalized eigenvectors of A corresponding to the eigenvalues contained in the diagonal matrix Λ. Moreover for $a, m \in \mathbb{R}$ define the shrinkage operator $\ell_\gamma(a, m) = \min\{\max\{m, a\}, \gamma m\}$.
In each step of Algorithm 2 closed form expressions are computed except that for computing m_* in (CM1), and ω_* in (CM2). m_* is the solution of a one-dimensional convex problem. The resulting updates for the eigenvalues almost coincide with those of TCLUST. For TCLUST, Fritz et al. (2013) show that their analogue of m_* can be computed by $2pG + 1$ evaluations of the objective function. A similar result may hold here, however we do not consider it because in numerical experiments we found that the simple golden section search algorithm of Kiefer (1953) requires on average few objective function evaluations independently of p and G. Computation of ω_* can be performed by a one-dimensional root finder algorithm. Both are simple problems that do not require much computational effort.

Some additional results are given to show how the CM1–step and the CM2–step solve (**CM1**) and (**CM2**) respectively.

Lemma 14 *Assume Algorithm 2 has been run for s iterations. The vector $\theta_1^{(s+1)} = (\mu_1^{(s+1)}, \ldots, \mu_G^{(s+1)}, \text{vect}(\Sigma_1^{(s+1)}), \ldots, \text{vect}(\Sigma_G^{(s+1)}))'$ computed in the CM1–step is the global optimal solution to* (**CM1**). *Moreover, m_* exists and it is unique.*

Proof Based on standard normal likelihood theory, one can see that the unique maximum of $Q_1(\theta_1, \theta^{(s)})$ with respect to mean parameters is $\mu_j^{(s+1)}$ for all $j = 1, 2, \ldots, G$. Substituting $\mu_j^{(s+1)}$ into $Q_1(\cdot)$, and rearranging the exponent of the Gaussian density by using the cyclic

Algorithm 2: ECM

input : $\{x_1, x_2, \ldots, x_n\}$, δ, $\pi_{\max}$, γ, $\theta^{(0)} \in \Theta$, tol¿0
output : θ^{ecm}

while $|l_n(\theta^{(s+1)}) - l_n(\theta^{(s)})| >$tol **do**

E–step

compute $\tau_{i,j}^{(s)}$ for all $i = 1, 2, \ldots, n$ and $j = 0, 1, \ldots, G$

CM1–step

for $j = 1, 2, \ldots, G$ **do**

$$\mu_j^{(s+1)} \leftarrow \frac{1}{T_j^{(s)}} \sum_{i=1}^{n} \tau_{i,j}^{(s)} x_i$$

$$S_j^{(s+1)} \leftarrow \frac{1}{T_j^{(s)}} \sum_{i=1}^{n} \tau_{i,j}^{(s)} (x_i - \mu_j^{(s+1)})(x_i - \mu_j^{(s+1)})',$$

$$V_j^{(s+1)} \operatorname{diag}(e_{j,1}, \ldots, e_{j,p}) V_j^{(s+1)\prime} \quad \leftarrow \quad \operatorname{Spec}(S_j^{(s+1)})$$

end

if $\max\{e_{j,k}/e_{t,k}; \quad t, j = 1, 2, \ldots, G,\ j \neq t,\ k = 1, 2, \ldots, p\} \leq \gamma$ **then**

$$\Sigma_j^{(s+1)} \leftarrow S_j^{(s+1)}$$

else

$$m_* \leftarrow \arg\min_{m>0} \quad \sum_{j=1}^{G} T_j^{(s)} \sum_{k=1}^{p} \left(\log(\ell_\gamma(e_{j,k}, m)) + \frac{e_{j,k}}{\ell_\gamma(e_{j,k}, m)} \right)$$

$$\Sigma_j^{(s+1)} \leftarrow V_j^{(s+1)} \operatorname{diag}(\ell_\gamma(e_{j,1}, m_*), \ldots, \ell_\gamma(e_{j,p}, m_*))\, V_j^{(s+1)\prime} \text{ for all } j = 1, 2, \ldots, G$$

end

CM2–step

if $\sum_{i=1}^{n} \tau_0(x_i, (\theta_1^{(s+1)}, \dot{\theta}_2)) \leq n\pi_{\max}$, *where* $\dot{\theta}_2 = \left(T_0^{(s)}/n, \ldots, T_G^{(s)}/n \right)'$ **then**

$$\pi_j^{(s+1)} \leftarrow T_j^{(s)}/n \text{ for all } j = 0, 1, \ldots, G$$

else

$$\text{compute } \omega_* : \quad \left(\sum_{i=1}^{n} \frac{\omega_* \delta}{\omega_* \delta + \frac{1-\omega_*}{n - T_0^{(s)}} \sum_{j=1}^{G} T_j^{(s)} \phi(x_i; \mu_j^{(s+1)}, \Sigma_j^{(s+1)})} \right) - n\pi_{\max} = 0$$

$$\pi_0^{(s+1)} \leftarrow \quad \omega_*$$

$$\pi_j^{(s+1)} \leftarrow \quad \frac{1-\omega_*}{n - T_0^{(s)}} T_j^{(s)}$$

end

end

$$\theta^{\text{ecm}} \leftarrow \theta^{(s+1)}$$

property of the matrix trace (see Anderson and Olkin, 1985), program (**CM1**) is completed by choosing $\Sigma_1, \ldots, \Sigma_G$ maximizing

$$Q_1(\dot{\theta}_1, \theta^{(s)}) = \text{const} - \frac{1}{2} \sum_{j=1}^{G} T_j^{(s)} \left(\operatorname{tr}(\Sigma_j^{-1} S_j^{(s+1)}) - \log\det(\Sigma_j^{-1}) \right) \tag{19}$$

under the eigenratio constraint, where $\dot{\theta}_1 = (\mu_1^{(s+1)}, \ldots, \mu_j^{(s+1)}, \operatorname{vect}(\Sigma_1), \ldots, \operatorname{vect}(\Sigma_G))$. Consider the spectral decompositions

$$\operatorname{Spec}(\Sigma_j) = \Gamma_j \Lambda_j \Gamma_j', \quad \text{and} \quad \operatorname{Spec}(S_j^{(s+1)}) = V_j^{(s+1)} E_j V_j^{(s+1)\prime},$$

where $\Lambda_j = \operatorname{diag}(\lambda_{j,1}, \ldots, \lambda_{j,p})$ and $E_j = \operatorname{diag}(e_{j,1}, \ldots, e_{j,p})$. Theorem 1 and Corollary 1 in Theobald (1975) imply that

$$\operatorname{tr}\left(\Lambda_j^{-1} E_j\right) - \log\det(\Lambda_j^{-1}) \leq \operatorname{tr}\left(\Sigma_j^{-1} S_j^{(s+1)}\right) - \log\det(\Sigma_j^{-1}), \tag{20}$$

with the previous holding with equality if and only if $\Gamma_j = V_j^{(s+1)}$. Therefore, $\Sigma_j^{(s+1)} = V_j^{(s+1)} \Lambda_j V_j^{(s+1)\prime}$ is plugged into (19), and (**CM1**) reduces to

$$\begin{aligned} &\underset{\lambda_{1,1},\ldots,\lambda_{G,p}}{\text{minimize}} \quad \sum_{j=1}^{G} T_j^{(s)} \sum_{k=1}^{p} \left(\log(\lambda_{j,k}) + \frac{e_{j,k}}{\lambda_{j,k}} \right), \\ &\text{subject to} \quad 0 < m \leq \lambda_{j,1}, \ldots, \leq \lambda_{j,p} \leq m\gamma \quad \forall j = 1, 2, \ldots, G. \end{aligned} \tag{21}$$

Program (21) is separable in the optimization variables, and therefore the summands of (21) can be minimized separately for a given m. Fix $m > 0$, then $\ell_\gamma(e_{j,k}, m)$ is the unique optimal solution to the minimization of $\log(\lambda_{j,k}) + e_{j,k}\, \lambda_{j,k}^{-1}$. Notice that $e_{j,k} \leq e_{j,t} \Longrightarrow \ell_\gamma(e_{j,k}, m) \leq \ell_\gamma(e_{j,t}, m)$ for any $m > 0$ and $t = 1, 2 \ldots, p$. This means that the relative ordering of the elements on the diagonal of E_j remains unchanged after having applied the shrinkage operator $\ell(\cdot)$. Replace $\lambda_{j,k}$ with $\ell_\gamma(e_{j,k}, m)$ and (21) is transformed into

$$\begin{aligned} &\underset{m}{\text{minimize}} \quad \sum_{j=1}^{G} T_j^{(s)} \sum_{k=1}^{p} \left(\log(\ell_\gamma(e_{j,k}, m)) + \frac{e_{j,k}}{\ell_\gamma(e_{j,k}, m)} \right), \\ &\text{subject to} \quad m > 0 \end{aligned} \tag{22}$$

(22) is now a convex program in m. Therefore (**CM1**) is solved by the unique m_* that solves (22). This implies that the CM1–step is solved by taking $\Sigma_j^{(s+1)} = V_j^{(s+1)} E_j^* V_j^{(s+1)\prime}$ where $E_j^* = \operatorname{diag}(\ell_\gamma(e_{j,1}, m_*), \ldots, \ell_\gamma(e_{j,p}, m_*))$. Notice that uniqueness of m_* implies the uniqueness of the solution to **CM1**. Observe that when the eigenvalues of $S_1^{(s+1)}, \ldots, S_G^{(s+1)}$ fulfill the eigenratio constraint, then $E_j^* = E_j$ and $\Sigma_j^{(s+1)} = S_j^{(s+1)}$ for all $j = 1, \ldots, G$. The latter completes the proof. (The result is connected to Lemma 1 in Won et al. (2013), by which the last part of the proof is inspired.)

■

Based on the previous Lemma, the constrained eigenvalues can be found by simply solving a convex one-dimensional problem. The optimal choice of the covariances is a form of Steinian-type nonlinear shrinkage (see Gavish and Donoho, 2017).

Lemma 15 *Assume Algorithm 2 has been run for s iterations. The vector* $\theta_2^{(s+1)} = (\pi_0^{(s+1)}, \ldots, \pi_G^{(s+1)})'$ *computed in the CM2–step is the global optimal solution to* (**CM2**)*. Moreover,* ω_* *exists and it is unique.*

Proof The objective function in (**CM2**) is strictly concave and the equality constraint is linear. Take $\theta_2', \theta_2'' \in \{\theta_2 : \ \pi_0 + \ldots, +\pi_G = 1, \ \pi_j \in [0,1] \ \forall j = 0, 1, \ldots, G\}$, if $\pi_0' \leq \pi_0''$ then $\tau_0(\cdot, \cdot, \theta_2') \leq \tau_0(\cdot, \cdot, \theta_2'')$. Therefore, for $\beta \in (0,1)$

$$\sum_{i=1}^{n} \tau_0(x_i, \theta_1^{(s+1)}, \ \beta\theta_2' + (1-\beta)\theta_2'') \leq \max\left\{\sum_{i=1}^{n} \tau_0(x_i, \theta_1^{(s+1)}, \theta_2'), \sum_{i=1}^{n} \tau_0(x_i, \theta_1^{(s+1)}, \theta_2'')\right\},$$

which implies that $\sum_{i=1}^{n} \tau_0(x_i, \theta_1^{(s+1)}, \theta_2)$ is quasiconvex in the optimization variable θ_2. It is concluded that Karush–Kuhn–Tucker (KKT) conditions are necessary for a global optimal solution (see Bertsekas, 1999). Such a solution will be a stationary point of the Langrangean function

$$H(\theta_2, h_1, h_2) \coloneqq Q_2(\theta_2, \theta^{(s)}) + h_1\left(1 - \sum_{j=0}^{G} \pi_j\right) + h_2\left(n\pi_{\max} - \sum_{i=1}^{n} \tau_0(x_i, \theta_1^{(s+1)}, \theta_2)\right),$$

where h_1 and h_2 are the dual variables. Let ∇_j denote derivatives with respect to the j component of θ_2. Let $\theta_2^\star$ the optimal solution, then based on KKT conditions there exists $h_1^\star, h_2^\star$ such that the following hold

$$\nabla_j Q_2(\theta_2^\star, \theta^{(s)}) - h_1^\star - h_2^\star \nabla_j \sum_{i=1}^{n} \tau_0(x_i, \theta_1^{(s+1)}, \theta_2^\star) = 0 \qquad \text{for all} \quad j = 0, 1, \ldots, G, \tag{23}$$

$$h_2^\star\left(\sum_{i=1}^{n} \tau_0(x_i, \theta_1^{(s+1)}, \theta_2^\star) - n\pi_{\max}\right) = 0, \quad h_2^\star \geq 0. \tag{24}$$

First consider the case when the noise proportion constraint does not bind, that is $h_2^\star = 0$. Than (23) becomes $T_j^{(s)}/\pi_j^\star - h_1^\star = 0$ for all $j = 0, 1, \ldots, G$. Solving the latter for $\pi_j^\star$, using the equality constraints and that $\sum_{j=0}^{G} T_j^{(s)} = n$, it results that $h_1^\star = n$ and $\pi_j^\star = T_j^{(s)}/n$ for all $j = 0, 1, \ldots, G$.
Now assume that the noise proportion constraints binds, hence $h_2^\star > 0$. Let $\pi_0^\star = \omega$ and rewrite the equality constraint as $\pi_1^\star +, \ldots, +\pi_G^\star = 1 - \omega$. Stationary points of $H(\cdot)$ satisfy $T_j^{(s)}/\pi_j^\star - h_1^\star = 0$. Solving the latter for $\pi_j^\star$ and using the equality constraints it results that $h_1^\star = \sum_{j=1}^{G} T_j^{(s)}/(1-\omega)$. Since $\sum_{j=1}^{G} T_j^{(s)} = n - T_0^{(s)}$, then

$$\pi_j^\star = \frac{1-\omega}{n - T_0^{(s)}} T_j^{(s)} \quad \text{for all} \quad j = 1, 2, \ldots, G.$$

Now the solution for $j = 1, 2, \ldots, G$ is a function of ω, which can be determined by using the fact that the inequality constraints binds. Define

$$g(\omega) = \left(\sum_{i=1}^{n} \frac{\omega\delta}{\omega\delta + \frac{1-\omega}{n-T_0^{(s)}} \sum_{j=1}^{G} T_j^{(s)} \phi(x_i; \mu_j^{(s+1)}, \Sigma_j^{(s+1)})}\right) - n\pi_{\max}.$$

$g(\omega)$ is bracketed on the interval $[0,1]$, in fact $g(0) = -n\pi_{\max} < 0$ and $g(1) = n(1-\pi_{\max}) > 0$. Moreover $g(\omega)$ is continuous, and it can be easily verified that it's derivative is continuous and positive at any $\omega \in (0,1)$. This implies that there exists a unique ω_* such that $g(\omega_*) = 0$. Setting $\pi_0^\star = \omega_*$ and replacing ω_* into $\pi_j^\star$ gives the optimal solution. We now compare the two solutions in terms of objective function, and we show that there is hierarchy between them. Define

$$\dot{\theta}_2 = \left(\frac{T_0^{(s)}}{n}, \frac{T_1^{(s)}}{n}, \ldots, \frac{T_1^{(s)}}{n}\right)', \qquad \ddot{\theta}_2 = \left(\omega_*, \frac{1-\omega_*}{n-T_0^{(s)}} T_1^{(s)}, \ldots, \frac{1-\omega_*}{n-T_0^{(s)}} T_G^{(s)}\right)'.$$

Using Wald's information inequality it can be shown that

$$\frac{T_0^{(s)}}{n} \log(\omega_*) + \sum_{j=1}^{G} \frac{T_j^{(s)}}{n} \log\left(\frac{1-\omega_*}{n-T_0^{(s)}} T_j^{(s)}\right) \ \leq \ \sum_{j=0}^{G} \frac{T_j^{(s)}}{n} \log\left(\frac{T_j^{(s)}}{n}\right),$$

with the previous holding with equality if and only if $\omega_* = T_0^{(s)}/n$. The latter implies that $Q_2(\ddot{\theta}_2, \theta^{(s)}) < Q_2(\dot{\theta}_2, \theta^{(s)})$ whenever $\ddot{\theta}_2 \neq \dot{\theta}_2$. Hence $\dot{\theta}_2$ is the global optimal solution whenever it is feasible, otherwise the global optimal solution is $\ddot{\theta}_2$. The latter proves that the updating in CM2-step selects the global optimal solution to (**CM2**). ■

Theorem 16 *Assume A0. The* $\{\theta^{(s)}\} s \in \mathbb{N}$ *produced by Algorithm 2 converges to a point* $\theta_n^{ecm} \in \Theta$*, and* $l_n(\theta^{(s)})$ *is increased in every step.*

Proof As consequence of Lemma 14 and 15, $Q(\theta, \theta^{(s)})$ is never decreased, in fact for all $s = 0, 1, \ldots$

$$Q(\theta_1^{(s+1)}, \theta_2^{(s+1)}, \theta^{(s)}) \geq Q(\theta_1^{(s+1)}, \theta_2^{(s)}, \theta^{(s)}) \geq Q(\theta_1^{(s)}, \theta_2^{(s)}, \theta^{(s)}),$$

A0 ensures existence of $\theta_n(\delta)$, and the convergence Theorem 4.1 in Redner and Walker (1984) holds with $Q(\theta, \theta^{(s)})$ playing the role of their $Q(\cdot)$ function. ■

Remark 17 *The eigenratio constraint together with the noise proportion constraint rule out the possibility that at some point along the iteration* $T_j^{(s)} = 0$ *for some* $j = 1, 2, \ldots, G$ *and updates in CM1-step are guaranteed to exist. In fact* $T_j^{(s)} = 0$ *means that according to* $\theta^{(s-1)}$ *none of points contributes to the* j*th Gaussian component. In theory this can only happen if the* j*th component has an infinite dispersion according to* $\theta^{(s-1)}$*. However, in*

that case the eigenratio constraint would force all eigenvalues in $\theta^{(s-1)}$ to diverge to $+\infty$ at the same rate so that $T_j^{(s)} \searrow 0$ for all $j = 1, 2, \ldots, G$, which is not possible because of the noise proportion constraint. Although in theory an appropriate choice of $\theta^{(0)} \in \Theta$ should not produce such a degeneracy, it may well be that in practice this is caused because of limited numerical resolution. Notice also that for δ=0 the noise proportion constraint is automatically fulfilled, and this would take the problem back to the EM algorithm for the MLE of a finite Gaussian mixture model with the additional eigenratio constraint. Therefore Algorithm 2 would become the EM Algorithm 1 where the M-step would coincide with CM1-step of Algorithm 2 plus the usual updating for the proportion parameters: $\pi_j^{(s+1)} \leftarrow T_j^{(s)}/n$ for all $j = 0, 1, \ldots, G$.

Remark 18 *There are substantial differences between Algorithm 2 and the one proposed in Coretto and Hennig (2016). Algorithm 2 implements the constraints of the RIMLE exactly and it is shown to converge under mild conditions on the data set. The algorithm presented in Coretto and Hennig (2016) handles the constraint heuristically as follows: (i) if the EM iteration converges, the eigenratio constraint is checked at the end of the iteration, and it is enforced simply increasing the eigenvalues smaller than $\lambda_{\max}(\theta)/\gamma$; (ii) if at some point along the EM iteration the estimated π_0 hits $\pi_{\max}$, the iteration is stopped and the solution discarded. Of course, this approximate algorithm is on average faster than Algorithm 2 for low values of γ and $\pi_{\max}$. In fact, in such a situation it is likely that Algorithm 2 needs to go through the additional computation of m_* and ω_* for most of the iterations. In order to give an idea to the reader, we estimated the relative speed for the AsyNoise data presented in Section 2 with $\gamma = 1, \pi_{\max} = 1\%$, and we found that on average the algorithm of Coretto and Hennig (2016) runs ten times faster. However, setting $\gamma = 100, \pi_{\max} = 50\%$ the difference becomes negligible. Furthermore, for low settings of $\pi_{\max}$ the algorithm of Coretto and Hennig (2016) is likely to record premature stops not leading to a solution (this happened in the previous experiment with $\pi_{\max} = 1\%$).*

5.2 Choice of initial values and input parameters

Algorithms 1 and 2 require the initial value $\theta^{(0)}$, and the input parameters $\pi_{\max}, \gamma$ and δ. The initial value $\theta^{(0)}$ can be set by randomly assigning points to G clusters and then computing cluster parameters. Initialization like this needs to be performed a number of times so that the solution with the largest pseudo-likelihood is selected. Implementation of the RIMLE given in the otrimle software of Coretto and Hennig (2017) relies on a more refined initialization strategy which consist in the following steps.

Initial denoising: for each data point compute its kth-nearest neighbors distance (k-NND), for some k. All points with k-NND larger than the $(1 - \pi_{\max})$-quantile of the k-NND are initialized as noise. The interpretation of k is that $(k-1)$, but not k, points close together may still be interpreted as noise/outliers, whereas k such points would constitute a cluster. The default value in the otrimle package is $k = 3$.

Initial clusters: agglomerative hierarchical clustering based on ML criteria for Gaussian mixture models as in Fraley (1998) is performed on the remaining $\lfloor n(1 - \pi_{\max}) \rfloor$ regular points to find the initial clusters. The sample mean and covariance matrix of points belonging to each cluster are computed to define $\theta^{(0)}$. This step is performed based on the `hc()` function from the mclust package.

The constraint defining quantities $\pi_{\max}$ and γ are regularization parameters that allow solving an otherwise ill-posed optimization problem. $\pi_{\max}$ also controls robustness because it specifies the maximum proportion of points assignable to the noise component. In order to be as robust as possible $\pi_{\max} = 1/2$ is a convenient choice that guarantees maximum protection. This implements a familiar condition in robust statistics that at most half of the data should be classified as "outliers/noise". A choice of $\pi_{\max}$ lower than the actual noise/outlier proportion will enforce some outliers to be assigned to clusters with potentially problematic implications. Hence, unless one has prior knowledge about the contamination process, we suggested to stick to $\pi_{\max} = 1/2$.

The role of the eigenratio can be twofold. If γ is set to a low-value, strong restrictions on clusters' shape are imposed. In this respect, the eigenratio constraint acts as a model selector. Unless one knows precisely the implications of a low choice of γ, it is suggested to use the eigenratio constraint as a regularization parameter. In fact, a large value of γ will regularize the covariance matrices without affecting clusters' shape too much. For example, a large γ would allow discovering an elongated concentrated cluster along with clusters having widespread spherical scatters. Ritter (2014) contains an in-depth analysis of constraints in model-based clustering. In Section 7 we present Monte Carlo experiments where the effect of different γ values is investigated.

Although through the presence of the product $\pi_0\delta$ in (1) the parameters π_0 and δ may seem confounded, they actually play a very different role in the RIMLE. δ is not treated as a model parameter to be estimated, but rather as a tuning device to enable a good robust clustering. The interpretation is that δ is the density value below which groups of observations should rather be treated as "noise" than as "cluster". This means that a larger value of δ will normally yield a larger estimate of π_0 because more observations will be classified as noise, as opposed to the intuition suggested by having the product $\pi_0\delta$ in (1). Whether small groups of observations of a certain size and with a certain density peak should rather count as "cluster" or rather as "group of outliers" cannot be identified from the data alone, but is rather a matter of interpretation. RIMLE may be sensitive to the choice of δ, and a good choice of δ is therefore important in practice. For instance, in the example of Figure 2 it has been shown that outside a certain interval of δ values the RIMLE does not perform well. Occasionally, subject matter knowledge may be available aiding the choice of a fixed value of δ, but often such knowledge may not exist. The OTRIMLE, a data dependent method ("optimally tuned RIMLE") to choose δ is presented in Coretto and Hennig (2016). The basic idea is to find a δ that optimizes a weighted Kolmogorov-type distance measure between the Mahalanobis distances of all objects to their corresponding cluster centers and the χ^2-distribution, which the Mahalanobis distances should follow if the clusters were indeed Gaussian. The current implementation of the OTRIMLE in the otrimle package selects the best RIMLE solution computed with algorithm 2 on a selected grid of 50 values of $\log(\delta)$. The default grid includes $\log(\delta) = -\infty$ so that a pure Gaussian mixture is always included in the competition (see the otrimle manual for more details).

6. Breakdown robustness of the RIMLE

Although robustness results for some clustering methods can be found in the literature, robustness theory in cluster analysis remains a tricky issue. Some work exists on breakdown points (García-Escudero and Gordaliza, 1999; Hennig, 2004; Gallegos and Ritter, 2005), addressing whether parameters can diverge to infinity (or zero, for covariance eigenvalues and mixture proportions) under small modifications of the data. An addition breakdown point of $r/(n+r)$ means that r, but not $r-1$, points can be added to a data set of size n so that at least one of the parameters "breaks down" in the above sense.

It is well known (García-Escudero and Gordaliza, 1999; Hennig, 2008), assuming the fitted number of clusters to be fixed, that robustness in cluster analysis has to be data dependent, for the following reasons:

- If there are two not well separated clusters in the data set, a very small amount of "contamination" can merge them, freeing up a cluster to fit outliers converging to infinity.
- Very small clusters cannot be robust because a group of outlying points can legitimately be seen as a "cluster" and will compete for fit with non-outlying clusters of the same size. Noise component-based and trimming methods are prone to trimming whole clusters if they are small enough.

Therefore, all nontrivial breakdown results (i.e., with breakdown point larger than the minimum $1/(n+1)$) in clustering require a condition that makes sure that the clusters in the data set are strongly clustered in some sense, which usually means that the clusters are homogeneous and strongly separated.

The theory for the RIMLE given here generalizes the argument given in Hennig (2004), Theorem 4.11, to the multivariate setup. We consider fixed datasets $\underline{x_n} = (x_1, x_2, \ldots, x_n)$ and sequences of estimators $(E_n)_{n\in\mathbb{N}}$ mapping observations from $(\mathbb{R}^p)^n$ to Θ. Denote the components of $E_n(\underline{x_n})$ by
$(\pi_{En0}, \pi_{En1}, \ldots, \pi_{EnG}, \mu_{En1}, \ldots, \mu_{EnG}, \Sigma_{En1}, \ldots, \Sigma_{EnG})$, G being the number of mixture components as usually.

The following assumption in the definition of the breakdown point makes sure that $E_n(\underline{x_n})$ indeed parametrizes G different mixture components; if there was a mixture component with proportion zero or two equal ones, one mixture component would be free to be driven to breakdown.

A4 For $j = 1, \ldots, G : \pi_{Enj} > 0$, and all $(\mu_{Enj}, \Sigma_{Enj})$ are pairwise different.

Definition 19 *Assume that $(E_n)_{n\in\mathbb{N}}$ and $\underline{x_n}$ fulfil A4. Then,*

$$
\begin{aligned}
B(E_n, \underline{x_n}) \;=\; & \min_r \left\{ \frac{r}{n+r} : \; \exists 1 \le j \le G \right. \\
& \forall D = [\pi_{min}, 1] \times C \text{ for which } \pi_{min} > 0, C \subset \mathbb{R}^p \times \mathcal{S}_p \text{ compact} \\
& \exists \underline{x_{n+r}} = (x_1, x_2, \ldots, x_n, x_{n+1}, \ldots x_{n+r}) \text{ so that for} \\
& \left. E_{n+g}(\underline{x_{n+g}}) : \; (\pi_{E(n+g)j}, \mu_{E(n+g)j}, \Sigma_{E(n+g)j}) \notin D \right\},
\end{aligned}
$$

where $\mathcal{S}_p$ is the set of all positive definite real valued $p\times p$-matrices, is called the **breakdown point** *of E_n at dataset $\underline{x_n}$.*

Denote the sequence of RIMLE estimators defined in (3) as $(\theta_{mH})_{m\in\mathbb{N}}$, write $l_{mH}(\underline{x_m}, \theta)$ for $l_m(\theta)$ with any $m \in \mathbb{N}$ and number of components H in (2), $l^o_{mH} = l_{mH}(\underline{x_m}, \theta_{mH}(\underline{x_m}))$. Let $\theta^* = \theta_{nG}(\underline{x_n})$ for the specific $\underline{x_n}$ and G considered here. Components of θ^* and later θ^+ are denoted with upper index "$*$" and "$+$", respectively. For $j = 1, \ldots, G$, let $\phi^*_j(x) = \phi(x, \mu^*_j, \Sigma^*_j)$, same with upper index "$+$". Assume $\delta > 0$ fixed throughout this section. We start with a straightforward extension of Lemma 3.

Lemma 20 *Assume A0 for $\underline{x_n}$. If $(\theta_m)_{m\in N}$ is any sequence in Θ so that for some $j = 1, 2, \ldots, G$ and $k = 1, 2, \ldots, p$, $\lambda_{k,j,m} \searrow 0$ as $m \to \infty$. For $\underline{x_{n+r}} = (x_1, x_2, \ldots, x_n, x_{n+1}, \ldots x_{n+r})$: $\sup_{(x_{n+1},\ldots,x_{n+r})\in(\mathbb{R}^p)^r} l_{(n+r)G}(\underline{x_{n+r}}, \theta_m) \to -\infty$.*

Proof The proof of Lemma 3 still applies because adding further observations only adds further positive terms to the sum in (13). ■

Corollary 21 *Assume that $\underline{x_n}$ is a fixed dataset fulfilling A0 and A4 for $E_n = \theta_{nG}$. Then there is a $\lambda_0 > 0$ bounding from below all λ_{min} for $\theta = \theta_{(n+r)G}(\underline{x_{n+r}})$ where $\underline{x_{n+r}} = (x_1, x_2, \ldots, x_n, x_{n+1}, \ldots x_{n+r})$ for any $(x_{n+1}, \ldots, x_{n+r}) \in (\mathbb{R}^p)^r$. Consequently $\phi_{max} = (2\pi)^{-\frac{p}{2}}\lambda_0^{-\frac{p}{2}}$ is an upper bound for all $\phi(x; \mu, \Sigma)$ with (μ, Σ) occurring as component parameters in any such θ.*

Proof Observe

$$
l_{(n+r)G}(\underline{x_{n+r}}, \theta^*) \ge \frac{1}{n+r}\left(\sum_{i=1}^{n} \log \psi_\delta(x_i, \theta^*) + r \log(\pi^*_0 \delta)\right) > -\infty. \tag{25}
$$

If the Corollary was wrong, it would be possible to construct a sequence $(\theta_m)_{m\in N}$ with $\lambda_{k,j,m} \searrow 0$ for some $j = 1, 2, \ldots, G$ and $k = 1, 2, \ldots, p$ so that each $\theta_m = \theta_{(n+r)G}(\underline{x_{n+r}})$ for an admissible $\underline{x_{n+r}}$. But (25) implies that there is a lower bound for $l_{(n+r)G}(\underline{x_{n+r}}, \theta_m)$, contradicting Lemma 20. ■

The following theorem gives conditions under which the RIMLE estimator is breakdown robust against adding r observation to $\underline{x_n}$. (26) states that the dataset needs to be fitted by G Gaussian components considerably better than by $G-1$ components, because otherwise the remaining mixture component would be available for fitting the added observations without doing much damage to the original fit. (27) makes sure that the noise proportion in $\underline{x_n}$ is low enough that the added observations can still be fitted by the noise component without exceeding π_{max}.

Theorem 22 *Assume that $\underline{x_n}$ fulfils A0 and A4 for $E_n = \theta_{nG}$. If*

$$
\begin{aligned}
l^o_{n(G-1)} \;<\; & \sum_{i=1}^{n} \log\left(\sum_{j=1}^{G} \pi^*_j \phi^*_j(x_i) + \left(\pi^*_0 + \frac{r}{n}\right)\delta\right) + \\
& r \log\left(\left(\pi^*_0 + \frac{r}{n}\right)\delta\right) + (n+r)\log\frac{n}{n+r} - r\log\phi_{max},
\end{aligned} \tag{26}
$$

ϕ_{max} defined in Corollary 21, and

$$\frac{1}{n+r}\left(\sum_{i=1}^{n}\frac{(n\pi_0^*+r)\delta}{(n+r)\psi_\delta(x_i,\theta^*)}+r\right)<\pi_{max}, \tag{27}$$

then $B(\theta_{nG},\underline{x_n})>\frac{r}{n+r}$.

Proof For $\underline{x_{n+r}}=(x_1,\dots x_{n+r})$, let $\theta^+=\theta_{(n+r)G}(\underline{x_{n+r}})$. Let $H<G$. Then,

$$l^o_{(n+r)G}\leq\sum_{i=1}^{n}\log\left(\sum_{j=1}^{H}\pi_j^+\phi_j^+(x_i)+\sum_{j=H+1}^{G}\pi_j^+\phi_j^+(x_i)+\pi_0^+\delta\right)+r\log\phi_{max}.$$

Assume w.l.o.g. that the parameter estimators of the mixture components $H+1,\dots,G$ leave a compact set D of the form $D=[\pi_{min},1]\times C,\ C\subset\mathbb{R}^p\times\mathcal{S}_p$ compact, $\pi_{min}>0$. Then there exists ϕ_{min} bounding $\phi_j^+(x_i)$ from below for $j=1,\dots,H$ and $i=1,\dots,n$, so $\sum_{j=1}^{H}\pi_j^+\phi_j^+(x_i)\geq H\pi_{min}\phi_{min}$.

Consider sequences $(\theta_m)_{m\in\mathbb{N}}\in\Theta$ with $l_{(n+r)G}(\underline{x_{n+r}},\theta_m)\to l^o_{(n+r)G}$ and leaving any D for $j=H+1\dots,G$, i.e., $\|\mu_{mj}\|\to\infty$ or $\lambda_{k,j,m}\to\infty$ or $\pi_{mj}\to 0$, but with all $\lambda_{k,j,m}\geq\lambda_0$ as established in Corollary 21. Observe that for such sequences $\sum_{j=H+1}^{G}\pi_{mj}\phi_{mj}(x_i)$ becomes arbitrarily small for $i=1,\dots,n$. Thus, for arbitrary $\epsilon>0$ and D large enough:

$$\begin{aligned}l^o_{(n+r)G}&\leq\sum_{i=1}^{n}\log\left(\sum_{j=1}^{H}\pi_j^+\phi_j^+(x_i)+\pi_0^+\delta\right)+r\log\phi_{max}+\epsilon\\&\leq\max_{H<G}l^o_{nH}+r\log\phi_{max}+\epsilon\leq l^o_{n(G-1)}+r\log\phi_{max}+\epsilon.\end{aligned}$$

But a potential estimator $\hat\theta$ could be defined by $\hat\pi_0=\frac{n\pi_0^*+r}{n+r},\ \hat\pi_j=\frac{n}{n+r}\pi_j^*,\ \hat\mu_j=\mu_j^*,\ \hat\Sigma_j=\Sigma_j^*,\ j=1,\dots,G$. Note that $\hat\theta\in\Theta$ because of (27). Therefore,

$$\begin{aligned}l^o_{(n+r)G}&\geq\sum_{i=1}^{n}\log\left(\sum_{j=1}^{G}\pi_j^*\phi_j^*(x_i)+\left(\pi_0^*+\frac{r}{n}\right)\delta\right)\\&\quad+r\log\left[\left(\pi_0^*+\frac{r}{n}\right)\delta\right]+(n+r)\log\frac{n}{n+r}\\\Rightarrow l^o_{n(G-1)}&\geq\sum_{i=1}^{n}\log\left(\sum_{j=1}^{G}\pi_j^*\phi_j^*(x_i)+\left(\pi_0^*+\frac{r}{n}\right)\delta\right)\\&\quad+r\log\left[\left(\pi_0^*+\frac{r}{n}\right)\delta\right]+(n+r)\log\frac{n}{n+r}-r\log\phi_{max}-\epsilon.\end{aligned}$$

This contradicts (26) by $\epsilon\to 0$. ■

Table 1: Parameters of the AsyNoise sampling design. Let π and ν be the expected proportion and the degrees of freedom. m_1,v_1 and m_2,v_2 are the mean parameters (m) and variance parameters (v) along dimensions 1 and 2 respectively. $c_{1,2}$ denotes the covariance between marginals 1 and 2. All remaining variances are set equal to 1, while all remaining mean and covariance parameters are set equal to 0.

Parameter	Cluster				
	1	2	3	4	5
π	10.05%	20.10%	6.70%	10.05%	20.10%
ν	10	11	12	13	14
m_1	0	7	5	-11	-7
m_2	3	1	9	11	5
v_1	1	2	2	0.5	2.5
v_2	1	2	2	0.5	2.5
c_{12}	0.5	-1.5	1.3	0	0

7. Numerical experiments

In this section, we perform Monte Carlo experiments to compare robust clustering methods on the two sampling designs introduced in Section 2. There is already a comprehensive simulation study involving OTRIMLE and competitors in Coretto and Hennig (2016), so here we use different setups. Below, apart from involving competing methods from the literature, the two OTRIMLE algorithms are compared (see Remark 18).

The AsyNoise design of Figure 1 generates $G=5$ clusters in $p=20$ dimensions and an expected noise proportion of 33%. The five clusters are generated from a mixture of t-distributions with parameters given in Table 1. The five clusters show a combination of structures that are often difficult to handle together. Some of them are not well separated, they are of different size, and although they are all elliptically shaped, there are strong differences in cluster scatters, and deviations from normality. The noise originates from a distribution obtained as the product of two independent one-dimensional uniform distributions with support on the interval $[-25,25]$, and 18 independent one-dimensional χ^2-distributions with 1 degree of freedom. The first and the third marginal are distributed uniformly, producing background noise on both clustered and non-clustered dimensions, and the χ^2-distribution adds a strong dose of asymmetry.

The second sampling design is called GEM (see Figure 3). In this case, the sampling design is a mixture of two Gaussian distributions in $p=20$ dimensions, with the addition of a few potential outliers. In this design, the first cluster has strongly correlated marginals, whereas the second one is spherical, and this produces a large discrepancy between the clusters' shapes. Define the $p\times p$ correlation matrix $C(\rho):=(\rho^{|l-k|})_{l,k}$ for $l,k=1,\dots,p$ (also called AR(1) correlation model). The parameters of the GEM design are specified in Table 2. An expected 2% of points are generated from a 20-dimensional t-distribution with 3 degrees of freedom, centered at $(0,0,-7,\dots,-7)'$, with unit variances and correlation matrix $C(0.9999)$. This produces a few points far from both clusters, although these outliers

Table 2: Parameters of the GEM sampling design. Let π be the expected proportion. m is the mean parameter constant across all marginals for the same cluster. Each clusters has unit variance across all marginals and correlation matrix given by $C(\rho)$.

Parameter	Cluster	
	1	2
π	29.4%	68.6%
m	0	4
ρ	0.99	0

are not extremely separated from the regular data. While non-robust methods can cope with weakly separated outliers at the expense of large estimation bias, some robust methods capable of handling extreme outliers might get in trouble if the separation gap between regular and nonregular points is modest.

In this experiment, ML for Gaussian mixtures with uniform noise and TCLUST are compared with the RIMLE optimally tuned according to the OTRIMLE method introduced in Coretto and Hennig (2016). Methods under comparison are set up as follow:

OtrimleECM: RIMLE is computed based on the ECM algorithm 2 on a grid of 50 $\log(\delta))$ values as described in Section 5.2. The OTRIMLE criterion proposed in Coretto and Hennig (2016) selects the best solution. The input parameter $\pi_{\max}$ is always set to the conventional 50%. The eigenratio constraint is varied between the strongest restriction ($\gamma = 0$), and no restriction at all ($\gamma = +\infty$). In particular, $\log_{10}(\gamma) = \{0, 0.5, 1, 2, 3, 6, +\infty\}$. The initial partition is computed as described in Section 5.2. OtrimleECM is computed using the otrimle package of Coretto and Hennig (2017).

OtrimleAEM: RIMLE is computed using the approximate EM-algorithm introduced in Coretto and Hennig (2016). Both $\pi_{\max}$ and γ are set for OtrimleECM as well as initial values. Software for OtrimleAEM is available as part of the supplementary materials in Coretto and Hennig (2016).

TclustOracle: TCLUST with trimming rate set to the true underlying noise proportion. The eigenratio constraint is treated as for OtrimleECM. TclustOracle is computed using the tclust package of Fritz et al. (2012) which does not allow the user to choose an initial partition. TCLUST initialization is random, and we increased the default number of random starts to the sample size. The default maximum number of iterations is also increased to 500 because several convergence problems were recorded.

TclustFix: same as TclustOracle but with trimming rate fixed to a low 5% for the GEM design, and a high 50% for the AsyNoise design. The package tclust also includes the "ctlcurves"-tool for guiding the user toward the choice of a suitable trimming rate. Unfortunately, this graphical tool does not give any clear indication for the data sets generated in this experiment.

MCLUSTn: ML for Gaussian mixtures with uniform noise as implemented in the mclust package of Fraley et al. (2012). Regularization of the covariance matrices is done by choosing an appropriate covariance parameterization based on the BIC (Bayesian Information Criterion). Mclust requires noise initialization, and this is initialized as for the OtrimleECM. Note that the otrimle package uses mclust initialization for the regular points, hence OtrimleECM and MCLUSTn both start from the same partition.

There exist other methods capable of handling noise not considered here. The true clusters can be characterized by having a considerably higher density than the noise region, so density based clustering would seem to be another promising approach, but it suffers from the high dimensionality of the data, too. The DBSCAN algorithm of Ester et al. (1996) can handle noise, however its performance strongly depends on a pair of tunings that need to be carefully selected based on the dataset, and this makes it hardly comparable in a Monte Carlo experiment like the present one. ML based on t-mixtures of Peel and McLachlan (2000) would also be appropriate, however it requires discretionary decisions on how to define noise in terms of the tails of the estimated student-t distributions.

The sample size is set to $n = 500$ for AsyNoise, and $n = 100$ for GEM. With these relatively low sample sizes, the regularization of the covariance matrices becomes crucial because often small clusters are found compared to the dimensionality p. For both data sets 1000 Monte Carlo replicates have been considered. The true cluster label of a point is defined based on the component of the sampling distribution that generates it. Misclassification rates are computed with respect to the minimizing permutation of clusters' indexes not involving the estimated noise, which is always matched to the true noise. The underlying eigenratio behavior of these designs is largely varying. The true γ is 7 for AsyNoise, and it is 3704.7 for GEM. However, if one computes the eigenratio of sample clusters' covariances based on true labels the figure can be completely different. In fact, we computed the (5%, 95%)-quantiles of the Monte Carlo distribution of these quantities, and we obtained (44.5, 273.4) for AsyNoise, and (19899.3, 246826.8) for GEM. In the examples given in Section 2 we fixed $\gamma = 100$, because in real world applications one typically does not have information on it, and we used the central value adopted in these experiments.

Results are summarized in Tables 3 and 4, and Figures 5 and 6. Since MCLUSTn does not enforce an eigenratio constraint, results are recorded at $\gamma = +\infty$, although MCLUSTn has its own covariance regularization. MCLUSTn is seriously affected by contamination in both designs. Its performance is better for the AsyNoise design for which the boxplot of Figure 5 shows that in some replica it can produce misclassification rates below 10%. Regarding the Otrimle and Tclust versions, the performance depends on the setting of the eigenratio constraint. However, OtrimleECM offers the most stable performance in both designs. OtrimleECM achieves the best misclassification performance in all situations except for few cases where TclustOracle does better, but in fact, TclustOracle is run with the assumption that one knows the expected amount of noise exactly, which is never true in reality. Note that TclustOracle seems to not tolerate large values of γ in both designs. This is counterintuitive at least for GEM, where the true $\log_{10}(\gamma)$ is between 3 and 6, but in this range both TclustOracle and TclustFix have serious problems. OtrimleAEM is the second best overall, although it shows a large positive skewness in the distribution of the misclassification rates for AsyNoise, and in both designs it is not able to enforce low γ values appropriately. This is because in OtrimleAEM an approximate eigenratio constraint

Table 3: Monte Carlo averages, with standard errors in parenthesis, of misclassification rates (%) for the AsyNoise sampling design. "na" is reported if the software did not produce a valid answer in more than 50% of the replicates.

$\log_{10}(\gamma)$	OtrimleECM	OtrimleAEM	TclustOracle	TclustFix	MCLUSTn
0	15.31(0.00)	31.96(0.03)	4.33(0.00)	18.63(0.00)	—
0.5	11.25(0.01)	25.57(0.03)	5.31(0.00)	18.65(0.00)	—
1	9.46(0.00)	22.14(0.02)	16.52(0.00)	23.30(0.00)	—
2	11.48(0.00)	19.40(0.01)	50.26(0.00)	48.66(0.00)	—
3	12.37(0.01)	19.71(0.01)	57.88(0.00)	56.72(0.00)	—
6	12.05(0.01)	19.90(0.01)	57.26(0.00)	56.89(0.00)	—
$+\infty$	12.08(0.00)	19.92(0.01)	na	na	27.05(0.02)

Table 4: Monte Carlo averages, with standard errors in parenthesis, of misclassification rates (%) for the GEM sampling design. "na" is reported if the software did not produce a valid answer in more than 50% of the replicates.

$\log_{10}(\gamma)$	OtrimleECM	OtrimleAEM	TclustOracle	TclustFix	MCLUSTn
0	1.10(0.00)	63.97(0.04)	1.78(0.00)	3.32(0.00)	—
0.5	0.87(0.00)	11.96(0.03)	1.50(0.00)	3.17(0.00)	—
1	1.57(0.01)	3.49(0.01)	1.25(0.00)	3.11(0.00)	—
2	0.52(0.00)	2.25(0.00)	8.10(0.01)	9.68(0.01)	—
3	0.50(0.00)	0.71(0.00)	16.41(0.00)	18.08(0.00)	—
6	3.82(0.01)	1.18(0.01)	14.07(0.00)	16.80(0.00)	—
$+\infty$	4.66(0.01)	1.17(0.01)	na	na	41.60(0.01)

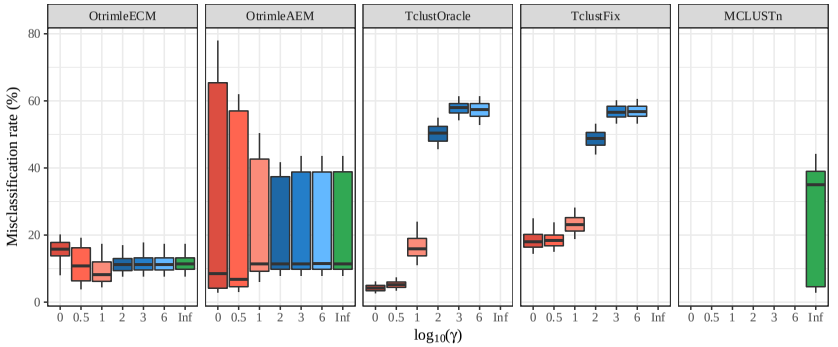


Figure 5: Modified boxplots of the Monte Carlo distribution of misclassification rates for the AsyNoise design: whiskers coincide with (5%, 95%)–quantiles of the distribution.

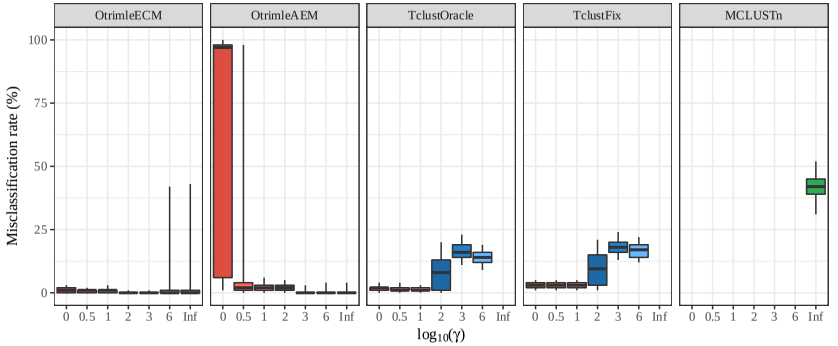


Figure 6: Modified boxplots of the Monte Carlo distribution of misclassification rates for the GEM design: whiskers coincide with (5%, 95%)–quantiles of the distribution.

is applied at the end of the EM iteration (see Remark 18). These results show a remarkable improvement of the ECM algorithm 2 over the approximate solution proposed in Coretto and Hennig (2016).

In practice the user has to specify γ. According to the results shown here, OtrimleECM is not very sensitive to this choice. Also the results show that good misclassification rates can be achieved in GEM, with a true $\gamma > 3000$, using a much lower γ for OTRIMLE; actually for TCLUST a much lower γ is even required to achieve good results. Choosing a lower γ in such situations may provide some welcome regularization. $\gamma = 100$ often seems to be a sensible choice. However, the user needs to have in mind that a straight interpretation of γ requires that a variance of 1 (say) along a one-dimensional projection has the same meaning in all directions in data space, which is particularly doubtful if variables have different measurement units or variable-wise variations are not meaningfully comparable. In such cases sphering or at least variable standardization may be advisable.

8. Concluding Remarks

The RIMLE robustifies the MLE in the Gaussian mixture model by adding an improper constant mixture component to catch outliers and points that cannot appropriately assigned to any cluster. Characteristics of the method compared to other robust clustering methods aiming for approximately Gaussian clusters are a smooth mixture-type transition between clusters and noise, and the fact that noise and outliers are not modelled by a specific and usually misspecified distribution, but rather as anything where the estimated mixture density is so low that the observation is rather classified to the constant noise than to any mixture component. If needed, the density value of the improper constant noise component can be chosen in a data-adaptive based on the OTRIMLE criterion developed in (Coretto and Hennig, 2016). The RIMLE/OTRIMLE has shown competitive performance when compared with state of the art methods for robust model-based clustering methods. In this paper we investigated theoretical properties of the RIMLE, and it is shown existence, consistency, breakdown behaviour, and convergence of algorithms. Since the RIMLE coincides with the MLE for Gaussian finite mixture models (when $\delta = 0$), the present paper also gives a comprehensive treatment for it which was missing in the literature.

References

Grigory Alexandrovich. A note on the article 'Inference for multivariate normal mixtures' by J. Chen and X. Tan. *Journal of Multivariate Analysis*, 129:245–248, 2014.

Theodore Wilbur Anderson and Ingram Olkin. Maximum-likelihood estimation of the parameters of a multivariate normal distribution. *Linear Algebra and its Applications*, 70: 147–171, 1985.

Jeffrey D. Banfield and Adrian E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):803, sep 1993. doi: 10.2307/2532201.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

Gilles Celeux and Gérard Govaert. Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793, 1995.

Jiahua Chen and Xianming Tan. Inference for multivariate normal mixtures. *J. Multivariate Anal.*, 100(7):1367–1383, 2009.

Pietro Coretto and Christian Hennig. A simulation study to compare robust clustering methods based on mixtures. *Advances in Data Analysis and Classification*, 4(2):111–135, 2010.

Pietro Coretto and Christian Hennig. Maximum likelihood estimation of heterogeneous mixtures of gaussian and uniform distributions. *Journal of Statistical Planning and Inference*, 141(1):462–473, 2011.

Pietro Coretto and Christian Hennig. Robust improper maximum likelihood: tuning, computation, and a comparison with other methods for robust Gaussian clustering. *Journal of the American Statistical Association*, 111:1648–1659, 2016.

Pietro Coretto and Christian Hennig. otrimle: Robust model-based clustering, 2017. R package version 1.1. Available at: https://CRAN.R-project.org/package=otrimle.

Juan Antonio Cuesta-Albertos, Alfonso Gordaliza, and Carlos Matrán. Trimmed k-means: an attempt to robustify quantizers. *The Annals of Statistics*, 25(2):553–576, 1997.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. Institute for Computer Science, University of Munich, 1996.

Chris Fraley. Algorithms for model-based gaussian hierarchical clustering. *SIAM Journal on Scientific Computing*, 20(1):270–281, 1998.

Chris Fraley and Adrian E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

Chris Fraley, Adrian E. Raftery, Thomas Brendan Murphy, and Luca Scrucca. mclust version 4 for r: Normal mixture modeling for model-based clustering, classification, and density estimation. Technical Report 597, University of Washington, Department of Statistics, 2012.

Heinrich Fritz, Luis A. García-Escudero, and Agustín Mayo-Iscar. tclust: An R package for a trimming approach to cluster analysis. *Journal of Statistical Software*, 47(12):1–26, 2012.

Heinrich Fritz, Luis A. García-Escudero, and Agustín Mayo-Iscar. A fast algorithm for robust constrained clustering. *Comput. Statist. Data Anal.*, 61:124–136, 2013. ISSN 0167-9473.

María Teresa Gallegos and Gunter Ritter. A robust method for cluster analysis. *The Annals of Statistics*, 33(1):347–380, 2005.

María Teresa Gallegos and Gunter Ritter. Trimmed ML estimation of contaminated mixtures. *Sankhyā. The Indian Journal of Statistics*, 71(2, Ser. A):164–220, 2009. ISSN 0972-7671.

María Teresa Gallegos. Maximum likelihood clustering with outliers. In *Classification, Clustering, and Data Analysis*, pages 247–255. Springer, 2002.

María Teresa Gallegos Gallegos and Gunter Ritter. Strong consistency of k-parameters clustering. *Journal of Multivariate Analysis*, 117:14–31, 2013.

Luis Angel García-Escudero and Alfonso Gordaliza. Robustness properties of k means and trimmed k means. *Journal of the American Statistical Association*, 94(447):956–969, 1999.

Luis Angel García-Escudero, Alfonso Gordaliza, Carlos Matrán, and Agustin Mayo-Iscar. A general trimming approach to robust cluster analysis. *The Annals of Statistics*, 36(3): 1324–1345, 2008.

Luis Angel García-Escudero, Alfonso Gordaliza, Carlos Matrán, and Agustin Mayo-Iscar. Avoiding spurious local maximizers in mixture modeling. *Statistics and Computing*, 25: 1–15, 2014.

Luis Angel García-Escudero, Alfonso Gordaliza, Carlos Matrán, Agustin Mayo-Iscar, and Christian Hennig. Robustness and outliers. In Christian Hennig, Marina Meila, Fionn Murtagh, and Roberto Rocci, editors, *Handbook of Cluster Analysis*, pages 653–678. CRC Press, Boca Raton FL, 2015.

Matan Gavish and David L. Donoho. Optimal shrinkage of singular values. *IEEE Transactions on Information Theory*, 63(4):2137–2152, 2017.

Richard J. Hathaway. A constrained formulation of maximum-likelihood estimation for normal mixture distributions. *The Annals of Statistics*, 13(2):795–800, 1985.

Christian Hennig. Breakdown points for maximum likelihood estimators of location?scale mixtures. *The Annals of Statistics*, 32(4):1313–1340, 2004.

Christian Hennig. Dissolution point and isolation robustness: Robustness criteria for general cluster analysis methods. *Journal of Multivariate Analysis*, 99(6):1154–1176, 2008.

Christian Hennig. Clustering strategy and method selection. In C. Hennig, M. Meila, F. Murtagh, and R. Rocci, editors, *Handbook of Cluster Analysis*, chapter 31, pages 703–730. Chapman & Hall/CRC, Boca Raton FL, 2015a.

Christian Hennig. What are the true clusters? *Pattern Recognition Letters*, 64:53–62, 2015b.

Christian Hennig and Tim F. Liao. How to find an appropriate clustering for mixed-type variables with application to socio-economic stratification. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(3):309–369, 2013.

Salvatore Ingrassia. A likelihood-based constrained algorithm for multivariate normal mixture models. *Statistical Methods & Applications*, 13(2):151–166, 2004.

Salvatore Ingrassia and Roberto Rocci. Constrained monotone EM algorithms for finite mixture of multivariate gaussians. *Computational Statistics & Data Analysis*, 51(11): 5339–5351, 2007.

Salvatore Ingrassia and Roberto Rocci. Degeneracy of the EM algorithm for the MLE of multivariate gaussian mixtures and dynamic constraints. *Computational Statistics & Data Analysis*, 55(4):1715–1725, 2011.

Robert I. Jennrich. Asymptotic properties of non-linear least squares estimators. *The Annals of Mathematical Statistics*, 40(2):633–643, 1969.

John E. Dennis Jr. Algorithms for nonlinear fitting. In M. J. D. Powell, editor, *Proceedings of the NATO Advanced Research Institute on "Nonlinear Optimization", held at Trinity Hall, Cambridge*, NATO Conference Series. Series II: Systems Science, London, 1981. Academic Press in cooperation with NATO Scientific Affairs Division.

Jack Kiefer. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4(3):502–502, 1953.

Jack Kiefer and Jacob Wolfowitz. Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. *The Annals of Mathematical Statistics*, 27(4):887–906, 1956.

Xiao-Li Meng and Donald B. Rubin. Maximum likelihood estimation via the ECM algorithm: a general framework. *Biometrika*, 80(2):267–278, 1993.

David Peel and Geoffrey John McLachlan. Robust mixture modelling using the t–distribution. *Statistics and Computing*, 10(4):339–348, 2000.

Richard Redner. Note on the consistency of the maximum likelihood estimate for nonidentifiable distributions. *The Annals of Statistics*, 9(1):225–228, 1981.

Richard Redner and Homer F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26(2):195–239, 1984.

Gunter Ritter. *Robust Cluster Analysis and Variable Selection*. Monographs on Statistics and Applied Probability. Chapman and Hall/CRC, 2014.

Chris M. Theobald. An inequality with application to multivariate analysis. *Biometrika*, 62(2):461–466, 1975.

Aad van der Vaart. *Asymptotic statistics*. Cambridge University Press, 2000.

Aad van der Vaart and Jon A. Wellner. *Weak convergence and empirical processes*. Springer Series in Statistics. Springer-Verlag, New York, 1996.

Joong-Ho Won, Johan Lim, Seung-Jean Kim, and Bala Rajaratnam. Condition-number-regularized covariance estimation. *Journal of the Royal Statistical Society, Series B*, 75 (3):427–450, 2013.

Chien-Fu Jeff Wu. On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

Journal of Machine Learning Research 18 (2017) 1-41 Submitted 3/17; Revised 9/17; Published 12/17

On Computationally Tractable Selection of Experiments in Measurement-Constrained Regression Models

Yining Wang YININGWA@CS.CMU.EDU
Adams Wei Yu WEIYU@CS.CMU.EDU
Aarti Singh AARTISINGH@CMU.EDU
Machine Learning Department, School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Michael Mahoney

Abstract

We derive computationally tractable methods to select a small subset of experiment settings from a large pool of given design points. The primary focus is on linear regression models, while the technique extends to generalized linear models and Delta's method (estimating functions of linear regression models) as well. The algorithms are based on a continuous relaxation of an otherwise intractable combinatorial optimization problem, with sampling or greedy procedures as post-processing steps. Formal approximation guarantees are established for both algorithms, and numerical results on both synthetic and real-world data confirm the effectiveness of the proposed methods.

Keywords: optimal selection of experiments, A-optimality, computationally tractable methods, minimax analysis

1. Introduction

Despite the availability of large datasets, in many applications, collecting labels for all data points is not possible due to measurement constraints. We consider the problem of measurement-constrained regression where we are given a large pool of n data points but can only observe a small set of $k \ll n$ labels. Classical *experimental design* approaches in statistical literature Pukelsheim (1993) have investigated this problem, but the proposed solutions tend to be often combinatorial. In this work, we investigate computationally tractable methods for selecting data points to label from a given pool of design points in measurement-constrained regression.

Despite the simplicity and wide applicability of OLS, in practice it may not be possible to obtain the full n-dimensional response vector y due to measurement constraints. It is then a common practice to select a small subset of rows (e.g., $k \ll n$ rows) in X so that the statistical efficiency of regression on the selected subset of design points is maximized. Compared to the classical *experimental design* problem (Pukelsheim, 1993) where X can be freely designed, in this work we consider the setting where the selected design points must come from an existing (finite) design pool X.

Below we list three example applications where such measurement constraints are relevant:

©2017 Yining Wang, Adams Wei Yu and Aarti Singh.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-175.html.

Example 1 (Material synthesis) Material synthesis experiments are time-consuming and expensive, whose results are sensitive to experimental setting features such as temperature, duration and reactant ratio. Given hundreds, or even thousands of possible experimental settings, it is important to select a handful of representative settings such that a model can be built with maximized statistical efficiency to predict quality of the outcome material from experimental features. In this paper we consider such an application of low-temperature microwave-assisted thin film crystallization (Reeja-Jayan et al., 2012; Nakamura et al., 2017) and demonstrate the effectiveness of our proposed algorithms.

Example 2 (CPU benchmarking) Central processing units (CPU) are vital to the performance of a computer system. It is an interesting statistical question to understand how known manufacturing parameters (clock period, cache size, etc.) of a CPU affect its execution time (performance) on benchmark computing tasks (Ein-Dor and Feldmesser, 1987). As the evaluation of real benchmark execution time is time-consuming and costly, it is desirable to select a subset of CPUs available in the market with diverse range of manufacturing parameters so that the statistical efficiency is maximized by benchmarking for the selected CPUs.

Example 3 (Wind speed prediction) In Chen et al. (2015) a data set is created to record wind speed across a year at established measurement locations on highways in Minnesota, US. Due to instrumentation constraints, wind speed can only be measured at intersections of high ways, and a small subset of such intersections is selected for wind speed measurement in order to reduce data gathering costs.

In this work, we primarily focus on the linear regression model (though extensions to generalized linear models and functions of linear models are also considered later)

$$y = X\beta_0 + \varepsilon,$$

where $X \in \mathbb{R}^{n\times p}$ is the design matrix, $y \in \mathbb{R}^n$ is the response and $\varepsilon \sim \mathcal{N}_n(0, \sigma^2 I_n)$ are homoscedastic Gaussian noise with variance σ^2. β_0 is a p-dimensional regression model that one wishes to estimate. We consider the "large-scale, low-dimensional" setting where both $n, p \to \infty$ and $p < n$, and X has full column rank. A common estimator is the *Ordinary Least Squares* (OLS) estimator:

$$\widehat{\beta}^{\text{ols}} = \text{argmin}_{\beta\in\mathbb{R}^p} \|y - X\beta\|_2^2 = (X^\top X)^{-1}X^\top y.$$

The mean square error of the estimated regression coefficients $\mathbb{E}\|\widehat{\beta}^{\text{ols}} - \beta_0\|_2^2 = \sigma^2 \text{tr}\left[\left(X^\top X\right)^{-1}\right]$. Under measurement constraints, it is well-known that the statistically optimal subset S^* for estimating the regression coefficients is given by the *A-optimality* criterion (Pukelsheim, 1993):

$$S^* = \text{argmin}_{S\subseteq[n],|S|\le k} \text{tr}\left[\left(X_S^\top X_S\right)^{-1}\right]. \tag{1}$$

Despite the statistical optimality of Eq. (1), the optimization problem is combinatorial in nature and the optimal subset S^* is difficult to compute. A brute-force search over all possible subsets of size k requires $O(n^k k^3)$ operations, which is infeasible for even moderately sized designs X.

In this work, we focus on **computationally tractable** methods for experiment selection that achieve near-optimal statistical efficiency in linear regression models. We consider two experiment selection models: the *with replacement* model where each design point (row of X) can be selected more than once with independent noise involved for each selection, and the *without replacement* model where distinct row subsets are required. We propose two computationally tractable algorithms: one sampling based algorithm that achieves $O(1)$ approximation of the statistically optimal solution for the with replacement model and, when Σ^* is well-conditioned, the algorithm also works for the without replacement model. In the "soft budget" setting $|S| = O_{\mathbb{P}}(k)$, the approximation ratio can be further improved to $1+O(\epsilon)$. We also propose a greedy method that achieves $O(p^2/k)$ approximation for the without replacement model regardless of the conditioning of the design pool X.

2. Problem formulation and backgrounds

We first give a formal definition of the experiment selection problem in linear regression models:

Definition 1 (experiment selection problem in linear regression models) *Let X be a known $n\times p$ design matrix with full column rank and k be the subset budget, with $p \le k \le n$. An experiment selection problem aims to find a subset $S \subseteq [n]$ of size k, either deterministically or randomly, then observes $y_S = X_S\beta_0+\widetilde{\varepsilon}$, where each coordinate of $\widetilde{\varepsilon}$ is i.i.d. Gaussian random variable with zero mean and equal covariance, and then generates an estimate $\widehat{\beta}$ of the regression coefficients based on (X_S, y_S). Two types of experiment selection algorithms are considered:*

1. With replacement*: S is a multi-set which allows duplicates of indices. Note that fresh (independent) noise is imposed after, and independent of, experiment selection, and hence duplicate design points have independent noise. We use $\mathcal{A}_1(k)$ to denote the class of all with replacement experiment selection algorithms.*

2. Without replacement*: S is a standard set that may not have duplicate indices. We use $\mathcal{A}_2(k)$ to denote the class of all without replacement experiment selection algorithms.*

As evaluation criterion, we consider the mean square error $\mathbb{E}\|\widehat{\beta}-\beta_0\|_2^2$, where $\widehat{\beta}$ is an estimator of β_0 with X_S and y_S as inputs. We study *computationally tractable* algorithms that approximately achieves the *minimax* rate of convergence over $\mathcal{A}_1(k)$ or $\mathcal{A}_2(k)$:

$$\inf_{A'\in\mathcal{A}_b(k)} \sup_{\beta_0\in\mathbb{R}^p} \mathbb{E}\left[\|\widehat{\beta}_{A'}-\beta_0\|_2^2\right]. \tag{2}$$

Formally, we give the following definition:

1. "Additive" means that the statistical error of the resulting estimator cannot be bounded by a multiplicative factor of the minimax optimal error.
2. The leverage score sampling method in Ma et al. (2015) does not have rigorous approximation guarantees in terms of $\|\widehat{\beta}-\beta_0\|_2^2$ or $\|X\widehat{\beta}-X\beta_0\|_2^2$. However, the bounds in that paper establish that leverage score sampling can be worse or better than uniform sampling under different settings.

Table 1: Summary of approximation results. $\Sigma_* = X^\top \mathrm{diag}(\pi^*)X$, where π^* is the optimal solution of Eq. (4). Sampling based algorithms can be with or without replacement and are randomized, which succeed with at least 0.8 probability.

Algorithm	Model	Constraint	$C(n,p,k)$	Assumptions
Ma et al. (2015)	with rep.	additive[1]	-[2]	asymptotic
Avron and Boutsidis (2013)	with rep.	additive	$O(n/k)$	$k=\Omega(p)$
sampling	with rep.	$\|S\|=O_{\mathbb{P}}(k)$	$1+O(\epsilon)$	$p\log k/k = O(\epsilon^2)$
sampling	without rep.	$\|S\|=O_{\mathbb{P}}(k)$	$1+O(\epsilon)$	$\|\Sigma_*^{-1}\|_2\kappa(\Sigma_*)\|X\|_\infty \log p = O(\epsilon^2)$
sampling	with rep.	$\|S\|\le k$	$O(1)$	$p\log k/k = O(1)$
sampling	without rep.	$\|S\|\le k$	$O(1)$	$\|\Sigma_*^{-1}\|_2\kappa(\Sigma_*)\|X\|_\infty \log p = O(1)$ $k>p$, or $k=\Omega(p^2/\epsilon)$
greedy	without rep.	$\|S\|\le k$	**Rigorous**: $1+\frac{p(p+1)}{2(k-p+1)}$ **Conjecture**: $1+O(\frac{p}{k-p})$	if $(1+\epsilon)$-approximation desired

Definition 2 ($C(n,p,k)$-approximate algorithm) *Fix $n \ge k \ge p$ and $b \in \{1,2\}$. We say an algorithm (either deterministic or randomized) $A \in \mathcal{A}_b(k)$ is a $C(n,p,k)$-approximate algorithm if for any $X \in \mathbb{R}^{n\times p}$ with full column rank, A produces a subset $S \subseteq [n]$ with size k and an estimate $\widehat{\beta}_A$ in polynomial time such that, with probability at least 0.8,*

$$\sup_{\beta_0\in\mathbb{R}^p} \mathbb{E}\left[\|\widehat{\beta}_A-\beta_0\|_2^2 \middle| X_S\right] \le C(n,p,k)\cdot \inf_{A'\in\mathcal{A}_b(k)} \sup_{\beta_0\in\mathbb{R}^p} \mathbb{E}\left[\|\widehat{\beta}_{A'}-\beta_0\|_2^2\right]. \tag{3}$$

Here both expectations are taken over the randomness in the noise variables $\widetilde{\varepsilon}$ and the inherent randomness in A.

Table 1 gives an overview of approximation ratio $C(n,p,k)$ for algorithms proposed in this paper. We remark that the combinatorial A-optimality solution of Eq. (1) upper bounds the minimax risk (since minimaxity is defined over deterministic algorithms as well), hence the approximation guarantees also hold with respect to the combinatorial A-optimality objective.

2.1 Related work

There has been an increasing amount of work on fast solvers for the general least-square problem $\min_\beta \|y-X\beta\|_2^2$. Most of existing work along this direction (Woodruff, 2014; Dhillon et al., 2013; Drineas et al., 2011; Raskutti and Mahoney, 2015) focuses solely on the computational aspects and do not consider statistical constraints such as limited measurements of y. A convex optimization formulation was proposed in Davenport et al. (2015) for a constrained adaptive sensing problem, which is a special case of our setting, but without finite sample guarantees with respect to the combinatorial problem. In Horel et al. (2014) a computationally tractable approximation algorithm was proposed for the D-optimality criterion of the experimental design problem. However, the core idea in Horel et al. (2014) of pipage rounding an SDP solution (Ageev and Sviridenko, 2004) is not applicable in our problem because the objective function we consider in Eq. (4) is not submodular.

Popular subsampling techniques such as leverage score sampling (Drineas et al., 2008) were studied in least square and linear regression problems (Zhu et al., 2015; Ma et al., 2015; Chen et al., 2015). While computation remains the primary subject, measurement

constraints and statistical properties were also analyzed within the linear regression model (Zhu et al., 2015). However, none of the above-mentioned (computationally efficient) methods achieve near minimax optimal statistical efficiency in terms of estimating the underlying linear model β_0, since the methods can be worse than uniform sampling which has a fairly large approximation constant for general X. One exception is Avron and Boutsidis (2013), which proposed a greedy algorithm that achieves an error bound of estimation of β_0 that is within a multiplicative factor of the minimax optimal statistical efficiency. The multiplicative factor is however large and depends on the size of the full sample pool X, as we remark in Table 1 and also in Sec. 3.4.

Another related area is *active learning* (Chaudhuri et al., 2015; Hazan and Karnin, 2015; Sabato and Munos, 2014), which is a stronger setting where feedback from prior measurements can be used to guide subsequent data selection. Chaudhuri et al. (2015) analyzes an SDP relaxation in the context of active maximum likelihood estimation. However, the analysis in Chaudhuri et al. (2015) only works for the with replacement model and the two-stage feedback-driven strategy proposed in Chaudhuri et al. (2015) is not available under the experiment selection model defined in Definition 1 where no feedback is assumed.

2.2 Notations

For a matrix $A \in \mathbb{R}^{n\times m}$, we use $\|A\|_p = \sup_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p}$ to denote the induced p-norm of A. In particular, $\|A\|_1 = \max_{1\leq j\leq m} \sum_{i=1}^n |A_{ij}|$ and $\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^m |A_{ij}|$. $\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$ denotes the Frobenius norm of A. Let $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_{\min(n,m)}(A) \geq 0$ be the singular values of A, sorted in descending order. The condition number $\kappa_2(A)$ is defined as $\kappa(A) = \sigma_1(A)/\sigma_{\min(n,m)}(A)$. For sequences of random variables X_n and Y_n, we use $X_n \xrightarrow{p} Y_n$ to denote X_n converges in probability to Y_n. We say $a_n \lesssim b_n$ if $\lim_{n\to\infty} |a_n/b_n| \leq 1$ and $a_n \gtrsim b_n$ if $\lim_{n\to\infty} |b_n/a_n| \leq 1$. For two d-dimensional symmetric matrices A and B, we write $A \preceq B$ if $u^\top(A-B)u \leq 0$ for all $u \in \mathbb{R}^d$, and $A \succeq B$ if $u^\top(A-B)u \geq 0$ for all $u \in \mathbb{R}^d$.

3. Methods and main results

We describe two computationally feasible algorithms for the experiment selection problem, both based on a continuous convex optimization problem. Statistical efficiency bounds are presented for both algorithms, with detailed proofs given in Sec. 7.

3.1 Continuous optimization and minimax lower bounds

We consider the following continuous optimization problem, which is a convex relaxation of the combinatorial A-optimality criterion of Eq. (1):

$$\pi^* = \operatorname{argmin}_{\pi\in\mathbb{R}^n} f(\pi; X) = \operatorname{argmin}_{\pi\in\mathbb{R}^p} \operatorname{tr}\left[\left(X^\top \operatorname{diag}(\pi) X\right)^{-1}\right], \tag{4}$$
$$s.t.\quad \pi \geq 0,\ \ \|\pi\|_1 \leq k,$$
$$\|\pi\|_\infty \leq 1 \quad \text{(only for the without replacement model)}.$$

Note that the $\|\pi\|_\infty \leq 1$ constraint is only relevant for the without replacement model and for the with replacement model we drop this constraint in the optimization problem. It is easy to verify that both the objective $f(\pi; X)$ and the feasible set in Eq. (4) are convex, and hence the global optimal solution π^* of Eq. (4) can be obtained using computationally tractable algorithms. In particular, we describe an SDP formulation and a practical projected gradient descent algorithm in Appendix B and D, both provably converge to the global optimal solution of Eq. (4) with running time scaling polynomially in n, p and k.

We first present two facts, which are proved in Sec. 7.

Fact 3.1 *Let π and π' be feasible solutions of Eq. (4) such that $\pi_i \leq \pi'_i$ for all $i = 1, \cdots, n$. Then $f(\pi; X) \geq f(\pi'; X)$, with equality if and only if $\pi = \pi'$.*

Fact 3.2 *$\|\pi^*\|_1 = k$.*

We remark that the inverse monotonicity of f in π implies second fact, which can potentially be used to understand sparsity of π^* in later sections.

The following theorem shows that the optimal solution of Eq. (4) lower bounds the minimax risk defined in Eq. (2). Its proof is placed in Sec. 7.

Theorem 3 *Let $f_1^*(k; X)$ and $f_2^*(k; X)$ be the optimal objective values of Eq. (4) for with replacement and without replacement, respectively. Then for $b \in \{1, 2\}$,*

$$\inf_{A\in\mathcal{A}_b(k)} \sup_{\beta_0\in\mathbb{R}^p} \mathbb{E}\left[\|\widehat{\beta}_A - \beta_0\|_2^2\right] \geq \sigma^2 \cdot f_b^*(k; X). \tag{5}$$

Despite the fact that Eq. (4) is computationally tractable, its solution π^* is *not* a valid experiment selection algorithm under a measurement budget of k because there can be much more than k components in π^* that are not zero. In the following sections, we discuss strategies for choosing a subset $\widehat{S}$ of rows with $|\widehat{S}| = O_{\mathbb{P}}(k)$ (i.e., soft constraint) or $|\widehat{S}| \leq k$ (i.e., hard constraint), using the solution π^* to the above.

3.2 Sampling based experiment selection: soft size constraint

We first consider a weaker setting where *soft* constraint is imposed on the size of the selected subset $\widehat{S}$; in particular, it is allowed that $|\widehat{S}| = O(k)$ with high probability, meaning that a constant fraction of over-selection is allowed. The more restrictive setting of hard constraint is treated in the next section.

A natural idea of obtaining a valid subset S of size k is by *sampling* from a weighted row distribution specified by π^*. Let $\Sigma_* = X^\top \operatorname{diag}(\pi^*) X$ and define distributions $p_j^{(1)}$ and $p_j^{(2)}$ for $j = 1, \cdots, n$ as

$$P^{(1)}:\ \ p_j^{(1)} = \pi_j^* x_j^\top \Sigma_*^{-1} x_j / p, \qquad \text{with replacement;}$$
$$P^{(2)}:\ \ p_j^{(2)} = \pi_j^* / k, \qquad \text{without replacement.}$$

Note that both $\{p_j^{(1)}\}_{j=1}^n$ and $\{p_j^{(2)}\}_{j=1}^n$ sum to one because $\sum_{j=1}^n \pi_j^* = k$ and $\sum_{j=1}^n \pi_j^* x_j^\top \Sigma_*^{-1} x_j = \operatorname{tr}((\sum_{j=1}^n \pi_j^* x_j x_j^\top)\Sigma_*^{-1}) = \operatorname{tr}(\Sigma_* \Sigma_*^{-1}) = p$.

input : $X \in \mathbb{R}^{n\times p}$, optimal solution π^*, target subset size k.
output: $\widehat{S} \subseteq [n]$, a selected subset of size at most $O_{\mathbb{P}}(k)$.
Initialization: $t = 0$, $S_0 = \emptyset$.
With replacement: for $t = 1, \cdots, k$ do:
- sample $i_t \sim P^{(1)}$ and set $w_t = \lceil \pi^*_{i_t}/(kp^{(1)}_{i_t})\rceil$;
- update: $S_{t+1} = S_t \cup \{w_t \text{ repetitions of } x_{i_t}\}$.

Without replacement: for $i = 1, \cdots, n$ do:
- sample $w_i \sim \text{Bernoulli}(kp^{(2)}_j)$;
- update: $S_{i+1} = S_i \cup \{w_i \text{ repetitions of } x_i\}$.

Finally, output $\widehat{S} = S_k$ for with replacement and $\widehat{S} = S_n$ for without replacement.

Figure 1: Sampling based experiment selection (expected size constraint).

Under the without replacement setting the distribution $P^{(2)}$ is straightforward: $p^{(2)}_j$ is proportional to the optimal continuous weights π^*_j; under the with replacement setting, the sampling distribution takes into account leverage scores (effective resistance) of each data point in the conditioned covariance as well. Later analysis (Theorem 5) shows that it helps with the finite-sample condition on k. Figure 1 gives details of the sampling based algorithms for both with and without replacement settings.

The following proposition bounds the size of $\widehat{S}$ in high probability:

Proposition 4 *For any $\delta \in (0, 1/2)$ with probability at least $1 - \delta$ it holds that $|\widehat{S}| \le 2(1 + 1/\delta)k$. That is, $|\widehat{S}| \le O_{\mathbb{P}}(k)$.*

Proof Apply Markov's inequality and note that an additional k samples need to be added due to the ceiling operator in with replacement sampling. ■

The sampling procedure is easy to understand in an asymptotic sense: it is easy to verify that $\mathbb{E}X^\top_{i_t} X_{i_t} = X^\top \text{diag}(\pi^*/k)X$ and $\mathbb{E}[w_{i_t}] = 1$, for both with and without replacement settings. Note that $\|p^{(2)}_{i_t}\|_\infty \le 1/k$ by feasibility constraints and hence $\text{Bernoulli}(kp^{(2)}_{i_t})$ is a valid distribution for all $i_t \in [n]$. For the with replacement setting, by weak law of large numbers, $X^\top_{\widehat{S}} X_{\widehat{S}} \xrightarrow{p} X^\top \text{diag}(\pi^*)X$ as $k \to \infty$ and hence $\text{tr}[(X^\top_S X_S)^{-1}] \xrightarrow{p} f(\pi^*; X)$ by continuous mapping theorem. A more refined analysis is presented in Theorem 5 to provide explicit conditions under which the asymptotic approximations are valid and on the statistical efficiency of $\widehat{S}$ as well as analysis under the more restrictive without replacement regime.

Theorem 5 *Fix $\epsilon > 0$ as an arbitrarily small accuracy parameter. Suppose the following conditions hold:*

$$\textit{With replacement}: \quad p \log k/k = O(\epsilon^2);$$
$$\textit{Without replacement}: \quad \|\Sigma_*^{-1}\|_2 \kappa(\Sigma_*) \|X\|^2_\infty \log p = O(\epsilon^2).$$

Here $\Sigma_* = X^\top \text{diag}(\pi^*)X$ and $\kappa(\Sigma_*)$ denotes the conditional number of Σ_*. Then with probability at least 0.9 the subset OLS estimator $\widehat{\beta} = (X^\top_{\widehat{S}} X_{\widehat{S}})^{-1} X^\top_{\widehat{S}} y_{\widehat{S}}$ satisfies

$$\mathbb{E}\left[\|\widehat{\beta} - \beta_0\|_2^2 \Big| X_{\widehat{S}}\right] = \sigma^2 \text{tr}\left[(X^\top_{\widehat{S}} X_{\widehat{S}})^{-1}\right] \le (1 + O(\epsilon)) \cdot \sigma^2 f^*_b(X; k), \qquad b \in \{1, 2\}.$$

We adapt the proof technique of Spielman and Srivastava in their seminal work on spectral sparsification of graphs (Spielman and Srivastava, 2011). More specifically, we prove the following stronger "two-sided" result which shows that $X^\top_{\widehat{S}} X_{\widehat{S}}$ is a *spectral approximation* of Σ_* with high probability, under suitable conditions.

Lemma 6 *Under the same conditions in Theorem 5, it holds that with probability at least 0.9 that*

$$(1 - \epsilon) z^\top \Sigma_* z \le z^\top \widehat{\Sigma}_{\widehat{S}} z \le (1 + \epsilon) z^\top \Sigma_* z, \quad \forall z \in \mathbb{R}^p,$$

where $\Sigma_* = X^\top \text{diag}(\pi^*)X$ and $\Sigma_{\widehat{S}} = X^\top_{\widehat{S}} X_{\widehat{S}}$.

Lemma 6 implies that with high probability $\sigma_j(\widehat{\Sigma}_{\widehat{S}}) \ge (1 - \epsilon)\sigma_j(\Sigma_*)$ for all $j = 1, \cdots, p$. Recall that $\text{tr}[(X^\top_S X_S)^{-1}] = \sum_{j=1}^p \sigma_j(\widehat{\Sigma}_{\widehat{S}})^{-1}$ and $f(\pi^*; X) = \sum_{j=1}^p \sigma_j(\Sigma_*)$. Subsequently, for $\epsilon \in (0, 1 - c]$ for some constant $c > 0$, $\text{tr}[(X^\top_{\widehat{S}} X_{\widehat{S}})^{-1}] \le (1 + O(\epsilon)) f(\pi^*; X)$. Theorem 5 is thus proved.

3.3 Sampling based experiment selection: hard size constraint

In some applications it is mandatory to respect a hard subset size constraint; that is, a randomized algorithm is expected to output $\widehat{S}$ that satisfies $|\widehat{S}| \le k$ almost surely, and no over-sampling is allowed. To handle such hard constraints, we revise the algorithm in Figure 1 as follows:

input : $X \in \mathbb{R}^{n\times p}$, optimal solution π^*, target subset size k.
output: $\widehat{S} \subseteq [n]$, a selected subset of size at most k.
Initialization: $t = 0$, $S_0 = \emptyset$, $R_0 = \emptyset$.
1. *With replacement*: sample $i_t \sim P^{(1)}$; set $w_t = \lceil \pi^*_{i_t}/(kp^{(1)}_{i_t})\rceil$;
   Without replacement: pick random $i_t \notin R_{t-1}$; sample $w_t \sim \text{Bernoulli}(kp^{(2)}_{i_t})$.
2. *Update*: $S_t = S_{t-1} \cup \{w_t \text{ repetitions of } i_t\}$, $R_t = R_{t-1} \cup \{i_t\}$.
3. Repeat steps 1 and 2 until at some $t = T$, $|S_{T+1}| > k$ or $R_{T+1} = [n]$. Output $\widehat{S} = S_T$.

Figure 2: Sampling based experiment selection (deterministic size constraint).

We have the following theorem, which mimics Theorem 5 but with weaker approximation bounds:

Theorem 7 *Suppose the following conditions hold:*

$$\textit{with replacement}: \quad p \log k/k = O(1);$$
$$\textit{without replacement}: \quad \|\Sigma_*^{-1}\|_2 \kappa(\Sigma_*) \|X\|^2_\infty \log p = O(1).$$

Then with probability at least 0.8 the subset estimator $\widehat{\beta} = (X_{\widehat{S}}^\top X_{\widehat{S}})^{-1} X_{\widehat{S}}^\top y_S$ *satisfies*

$$\mathbb{E}\left[\|\widehat{\beta} - \beta_0\|_2^2 \Big| X_{\widehat{S}}\right] = \sigma^2 \mathrm{tr}\left[(X_{\widehat{S}}^\top X_{\widehat{S}})^{-1}\right] \le O(1) \cdot \sigma^2 f_b^*(X;k), \qquad b \in \{1,2\}.$$

The following lemma is key to the proof of Theorem 7. Unlike Lemma 6, in Lemma 8 we only prove one side of the spectral approximation relation, which suffices for our purposes. To handle without replacement, we cite matrix Bernstein for combinatorial matrix sums in (Mackey et al., 2014).

Lemma 8 *Define* $\widehat{\Sigma}_{\widehat{S}} = X_{\widehat{S}}^\top X_{\widehat{S}}$. *Suppose the following conditions hold:*

$$\begin{aligned} &\textit{With replacement}: \quad p \log T/T = O(1); \\ &\textit{Without replacement}: \quad \|\Sigma_*^{-1}\|_2 \kappa(\Sigma_*) \|X\|_\infty^2 \log p = O(T/n). \end{aligned}$$

Then with probability at least 0.9 *the following holds:*

$$z^\top \widehat{\Sigma}_{\widehat{S}} z \ge K_T z^\top \Sigma_* z, \qquad \forall z \in \mathbb{R}^p, \tag{6}$$

where $K_T = \Omega(T/k)$ *for with replacement and* $K_T = \Omega(T/n)$ *for without replacement.*

Finally, we need to relate conditions on T in Lemma 8 to interpretable conditions on subset budget k:

Lemma 9 *Let* $\delta > 0$ *be an arbitrarily small fixed failure probability. The with probability at least* $1 - \delta$ *we have that* $T \ge \delta k$ *for with replacement and* $T \ge \delta n$ *for without replacement.*

Proof For with replacement we have $\mathbb{E}[\sum_{t=1}^T w_t] = T$ and for without replacement we have $\mathbb{E}[\sum_{t=1}^T w_t] = Tk/n$. Applying Markov's inequality on $\Pr[\sum_{t=1}^T w_t > k]$ for $T = \delta k$ and/or $T = \delta n$ we complete the proof of Lemma 9. ■

Combining Lemmas 8 and 9 with $\delta = 0.1$ and note that $T \le k$ almost surely (because $w_t \ge 1$), we prove Theorem 7.

3.4 Greedy experiment selection

input : $X \in \mathbb{R}^{n \times p}$, Initial subset $S_0 \subseteq [n]$, target size $k \le |S_0|$.
output: $\widehat{S} \subseteq [n]$, a selected subset of size k.
Initialization: $t = 0$.
1. Find $j^* \in S_t$ such that $\mathrm{tr}[(X_{S_t \backslash \{j^*\}}^\top X_{S_t \backslash \{j^*\}})^{-1}]$ is minimized.
2. Remove j^* from S_t: $S_{t+1} = S_t \backslash \{j^*\}$.
3. Repeat steps 1 and 2 until $|S_t| = k$. Output $\widehat{S} = S_t$.

Figure 3: Greedy experiment selection.

Avron and Boutsidis (2013) proposed an interesting greedy removal algorithm (outlined in Figure 3) and established the following result:

Lemma 10 *Suppose* $\widehat{S} \subseteq [n]$ *of size* k *is obtained by running algorithm in Figure 3 with an initial subset* $S_0 \subseteq [n]$, $|S_0| \ge k$. *Both* $\widehat{S}$ *and* S_0 *are standard sets (i.e., without replacement). Then*

$$\mathrm{tr}\left[(X_{\widehat{S}}^\top X_{\widehat{S}})^{-1}\right] \le \frac{|S_0| - p + 1}{k - p + 1} \mathrm{tr}\left[(X_{S_0}^\top X_{S_0})^{-1}\right].$$

In Avron and Boutsidis (2013) the greedy removal procedure in Figure 3 is applied to the entire design set $S_0 = [n]$, which gives approximation guarantee $\mathrm{tr}[(X_{\widehat{S}}^\top X_{\widehat{S}})^{-1}] \le \frac{n-p+1}{k-p+1} \mathrm{tr}[(X^\top X)^{-1}]$. This results in an approximation ratio of $C(n,p,k) = \frac{n-p+1}{k-p+1}$ as defined in Eq. (2), by applying the trivial bound $\mathrm{tr}[(X^\top X)^{-1}] \le f_2^*(k;X)$, which is tight for a design that has exactly k non-zero rows.

To further improve the approximation ratio, we consider applying the greedy removal procedure with S_0 equal to the support of π^*; that is, $S_0 = \{j \in [n] : \pi_j^* > 0\}$. Because $\|\pi^*\|_\infty \le 1$ under the without replacement setting, we have the following corollary:

Corollary 11 *Let* S_0 *be the support of* π^* *and suppose* $\|\pi^*\|_\infty \le 1$. *Then*

$$\mathrm{tr}[(X_{\widehat{S}}^\top X_{\widehat{S}})^{-1}] \le \frac{\|\pi^*\|_0 - p + 1}{k - p + 1} f(\pi^*; X) = \frac{\|\pi^*\|_0 - p + 1}{k - p + 1} f_2^*(k; X).$$

It is thus important to upper bound the support size $\|\pi^*\|_0$. With the trivial bound of $\|\pi^*\|_0 \le n$ we recover the $\frac{n-p+1}{k-p+1}$ approximation ratio by applying Figure 3 to $S_0 = [n]$. In order to bound $\|\pi^*\|_0$ away from n, we consider the following assumption imposed on X:

Assumption 3.1 *Define mapping* $\phi : \mathbb{R}^p \to \mathbb{R}^{\frac{p(p+1)}{2}}$ *as* $\phi(x) = (\xi_{ij} x(i) x(j))_{1 \le i \le j \le p}$, *where* $x(i)$ *denotes the* i*th coordinate of a* p*-dimensional vector* x *and* $\xi_{ij} = 1$ *if* $i = j$ *and* $\xi_{ij} = 2$ *otherwise. Denote* $\widetilde{\phi}(x) = (\phi(x), 1) \in \mathbb{R}^{\frac{p(p+1)}{2}+1}$ *as the affine version of* $\phi(x)$. *For any* $\frac{p(p+1)}{2} + 1$ *distinct rows of* X, *their mappings under* $\widetilde{\phi}$ *are linear independent.*

Assumption 3.1 is essentially a general-position assumption, which assumes that no $\frac{p(p+1)}{2} + 1$ design points in X lie on a degenerate affine subspace after a specific quadratic mapping. Like other similar assumptions in the literature (Tibshirani, 2013), Assumption 3.1 is very mild and almost always satisfied in practice, for example, if each row of X is independently sampled from absolutely continuous distributions.

We are now ready to state the main lemma bounding the support size of π^*.

Lemma 12 $\|\pi^*\|_0 \le k + \frac{p(p+1)}{2}$ *if Assumption 3.1 holds.*

Lemma 12 is established by an interesting observation into the properties of Karush-Kuhn-Tucker (KKT) conditions of the optimization problem Eq. (4), which involves a linear system with $\frac{p(p+1)}{2} + 1$ variables. The complete proof of Lemma 12 is given in Sec. 7.5. To contrast the results in Lemma 12 with classical rank/support bounds in SDP and/or linear programming (e.g. the Pataki's bound (Pataki, 1998)), note that the number of constraints in the SDP formulation of Eq. (4) (see also Appendix B) is linear in n, and hence analysis similar to (Pataki, 1998) would result in an upper bound of $\|\pi^*\|_0$ that scales with n, which is less useful for our analytical purpose.

Combining results from both Lemma 12 and Corollary 11 we arrive at the following theorem, which upper bounds the approximation ratio of the greedy removal procedure in Figure 3 initialized by the support of π^*.

Theorem 13 *Let π^* be the optimal solution of the without replacement version of Eq. (4) and $\widehat{S}$ be the output of the greedy removal procedure in Figure 3 initialized with $S_0 = \{i \in [n] : \pi_i^* > 0\}$. If $k > p$ and Assumption 3.1 holds then the subset OLS estimator $\widehat{\beta} = (X_{\widehat{S}}^\top X_{\widehat{S}})^{-1} X_{\widehat{S}}^\top y_{\widehat{S}}$ satisfies*

$$\mathbb{E}\left[\|\widehat{\beta} - \beta_0\| \big| X_{\widehat{S}}\right] = \sigma^2 \mathrm{tr}\left[(X_{\widehat{S}}^\top X_{\widehat{S}})^{-1}\right] \le \left(1 + \frac{p(p+1)}{2(k-p+1)}\right) f_2^*(k; X).$$

Under a slightly stronger condition that $k > 2p$, the approximation ratio $C(n,k,p) = 1 + \frac{p(p+1)}{2(k-p+1)}$ can be simplified to $C(n,k,p) = 1 + O(p^2/k)$. In addition, $C(n,k,p) = 1 + o(1)$ if $p^2/k \to 0$, meaning that near-optimal experiment selection is achievable with computationally tractable methods if $O(p^2)$ design points are allowed in the selected subset.

3.5 Interpretable subsampling example: anisotropic Gaussian design

Even though we consider a fixed pool of design points so far, here we use an anisotropic Gaussian design example to demonstrate that a non-uniform sampling can outperform uniform sampling even under random designs, and to interpret the conditions required in previous analysis. Let $x_1, \cdots, x_n$ be i.i.d. distributed according to an isotropic Gaussian distribution $\mathcal{N}_p(0, \Sigma_0)$.

We first show that non-uniform weights $\pi_i \neq k/n$ could improve the objective $f(\pi; X)$. Let π^{unif} be the uniformly weighted solution of $\pi_i^{\mathrm{unif}} = k/n$, corresponding to selecting each row of X uniformly at random. We then have

$$f(\pi^{\mathrm{unif}}; X) = \frac{1}{k}\mathrm{tr}\left[\left(\frac{1}{n}X^\top X\right)^{-1}\right] \xrightarrow{p} \frac{1}{k}\mathrm{tr}(\Sigma_0^{-1}).$$

On the other hand, let $B^2 = \gamma \mathrm{tr}(\Sigma_0)$ for some universal constant $\gamma > 1$ and $\mathbb{B} = \{x \in \mathbb{R}^p : \|x\|_2^2 \le B^2\}$, $\mathbb{X} = \{x_1, \cdots, x_n\}$. By Markov inequality, $|\mathbb{X} \cap \mathbb{B}| \gtrsim \frac{\gamma n}{1-\gamma}$. Define weighted solution π^w as $\pi_i^w \propto 1/p(x_i|\mathbb{B}) \cdot I[x_i \in \mathbb{B}]$ normalized such that $\|\pi_i^w\|_1 = k$. [1] Then

$$f(\pi^w; X) = \frac{1}{k} \frac{\mathrm{tr}\left[\left(\frac{1}{n}\sum_{x_i \in \mathbb{X}\cap\mathbb{B}} \frac{x_i x_i^\top}{p(x_i|\mathbb{B})}\right)^{-1}\right]}{\frac{1}{n}\sum_{x_i \in \mathbb{X}\cap\mathbb{B}} 1/p(x_i|\mathbb{B})} \xrightarrow{p} \frac{1}{k}\frac{n}{|\mathbb{X}\cap\mathbb{B}|} \frac{\mathrm{tr}\left[\left(\int_{\mathbb{B}} x x^\top \mathrm{d}x\right)^{-1}\right]}{\int_{\mathbb{B}} 1 \mathrm{d}x} \lesssim \frac{p^2}{(1-\gamma)\mathrm{tr}(\Sigma_0)}.$$

1. Note that π^* may *not* be the optimal solution of Eq. (4); however, it suffices for the purpose of the demonstration of improvement resulting from non-uniform weights.

Here in the last inequality we apply Lemma 17. Because $\frac{\mathrm{tr}(\Sigma_0^{-1})}{p} = \frac{1}{p}\sum_{i=1}^p \frac{1}{\sigma_i(\Sigma_0)} \ge \left(\frac{1}{p}\sum_{i=1}^p \sigma_i(\Sigma_0)\right)^{-1} = \frac{p}{\mathrm{tr}(\Sigma_0)}$ by Jensen's inequality, we conclude that in general $f(\pi^w; X) < f(\pi^{\mathrm{unif}}; X)$, and the gap is larger for ill-conditioned covariance Σ_0. This example shows that uneven weights in π helps reducing the trace of inverse of the weighted covariance $X^\top \mathrm{diag}(\pi) X$.

Under this model, we also simplify the conditions for the without replacement model in theorem 5 and 7. Because $x_1, \cdots, x_n \overset{i.i.d.}{\sim} \mathcal{N}_p(0, \Sigma_0)$, it holds that $\|X\|_\infty^2 \le O_{\mathbb{P}}(\|\Sigma_0\|_2^2 p \log n)$. In addition, by simple algebra $\|\Sigma_*^{-1}\|_2 \le p^{-1}\kappa(\Sigma_*)\mathrm{tr}(\Sigma_*^{-1}) \le p^{-1}\kappa(\Sigma_*) f_1^*(k; X)$. Using a very conservative upper bound of $f_1^*(k; X)$ by sampling rows in X uniformly at random and apply weak law of large numbers and the continuous mapping theorem, we have that $f_1^*(k; X) \lesssim \frac{1}{k}\mathrm{tr}(\Sigma_0^{-1})$. In addition, $\mathrm{tr}(\Sigma_0^{-1})\|\Sigma_0\|_2 \le p\|\Sigma_0^{-1}\|_2\|\Sigma_0\|_2 = p\kappa(\Sigma_0)$. Subsequently, the condition $\|\Sigma_*^{-1}\|_2\kappa(\Sigma_*)\|X\|_\infty^2 \log p = O(\epsilon^2)$ is implied by

$$\frac{p\kappa(\Sigma_*)^2\kappa(\Sigma_0)\log p \log n}{k} = O(\epsilon^2). \tag{7}$$

Essentially, the condition is reduced to $k \gtrsim \kappa(\Sigma_0)\kappa(\Sigma_*) \cdot p \log n \log p$. The linear dependency on p is necessary, as we consider the low-dimensional linear regression problem and $k < p$ would imply an infinite mean-square error in estimation of β_0. We also remark that the condition is scale-invariant, as $X' = \xi X$ and X share the same quantity $\|\Sigma_*^{-1}\|_2\kappa(\Sigma_*)\|X\|_\infty^2 \log k$.

3.6 Extensions

We discuss possible extension of our results beyond estimation of β_0 in the linear regression model.

3.6.1 Generalized linear models

In a generalized linear model $\mu(x) = \mathbb{E}[Y|x]$ satisfies $g(\mu(x)) = \eta = x^\top \beta_0$ for some known link function $g : \mathbb{R} \to \mathbb{R}$. Under regularity conditions (Van der Vaart, 2000), the maximum-likelihood estimator $\widehat{\beta}_n \in \mathrm{argmax}_\beta\{\sum_{i=1}^n \log p(y_i|x_i;\beta)\}$ satisfies $\mathbb{E}\|\widehat{\beta}_n - \beta_0\|_2^2 = (1+o(1))\mathrm{tr}(I(X,\beta_0)^{-1})$, where $I(X,\beta_0)$ is the Fisher's information matrix:

$$I(X,\beta_0) = -\sum_{i=1}^n \mathbb{E}\frac{\partial^2 \log p(y_i|x_i;\beta_0)}{\partial\beta\partial\beta^\top} = -\sum_{i=1}^n \left(\mathbb{E}\frac{\partial^2 \log p(y_i;\eta_i)}{\partial\eta_i^2}\right) x_i x_i^\top. \tag{8}$$

Here both expectations are taken over y conditioned on X and the last equality is due to the sufficiency of $\eta_i = x_i^\top \beta_0$. The experiment selection problem is then formulated to select a subset $S \subseteq [n]$ of size k, either with or without duplicates, that minimizes $\mathrm{tr}(I(X_S, \beta_0)^{-1})$.

It is clear from Eq. (8) that the optimal subset S^* depends on the unknown parameter β_0, which itself is to be estimated. This issue is known as the *design dependence* problem for generalized linear models (Khuri et al., 2006). One approach is to consider *locally optimal designs* (Khuri et al., 2006; Chernoff, 1953), where a consistent estimate $\check{\beta}$ of β_0 is first obtained on an initial design subset [2] and then $\check{\eta}_i = x_i^\top \check{\beta}$ is supplied to compute a more

2. Notice that a consistent estimate can be obtained using much fewer points than an estimate with finite approximation guarantee.

refined design subset to get the final estimate $\widehat{\beta}$. With the initial estimate $\check{\beta}$ available, one may apply transform $x_i \mapsto \widetilde{x}_i$ defined as

$$\widetilde{x}_i = \sqrt{-\mathbb{E}\frac{\partial^2 \log p(y_i;\check{\eta}_i)}{\partial \eta^2}} x_i.$$

Note that under regularity conditions $-\mathbb{E}\frac{\partial^2 \log p(y_i;\check{\eta}_i)}{\partial \eta_i^2} = \mathbb{E}\left(\frac{\partial \log(y_i;\check{x}_i)}{\partial \eta_i}\right)^2$ is non-negative and hence the square-root is well-defined. All results in Theorems 5, 7 and 13 are valid with $X = [x_1, \cdots, x_n]^\top$ replaced by $\widetilde{X} = [\widetilde{x}_1, \cdots, \widetilde{x}_n]^\top$ for generalized linear models. Below we consider two generalized linear model examples and derive explicit forms of $\widetilde{X}$.

Example 1: Logistic regression In a logistic regression model responses $y_i \in \{0, 1\}$ are binary and the likelihood model is

$$p(y_i; \eta_i) = \psi(\eta_i)^{y_i}(1 - \psi(\eta_i))^{1-y_i}, \qquad \text{where} \quad \psi(\eta_i) = \frac{e^{\eta_i}}{1 + e^{\eta_i}}.$$

Simple algebra yields

$$\widetilde{x}_i = \sqrt{\frac{e^{\check{\eta}_i}}{(1 + e^{\check{\eta}_i})^2}} x_i,$$

where $\check{\eta}_i = x_i^\top \check{\beta}$.

Example 2: Poisson count model In a Poisson count model the response variable y_i takes values of non-negative integers and follows a Poisson distribution with parameter $\lambda = e^{\eta_i} = e^{x_i^\top \beta_0}$. The likelihood model is formally defined as

$$p(y_i = r; \eta_i) = \frac{e^{\eta_i r} e^{-e^{\eta_i}}}{r!}, \qquad r = 0, 1, 2, \cdots.$$

Simple algebra yields

$$\widetilde{x}_i = \sqrt{e^{\check{\eta}_i}} x_i,$$

where $\check{\eta}_i = x_i^\top \check{\beta}$.

3.6.2 Delta's method

Suppose $g(\beta_0)$ is the quantity of interest, where $\beta_0 \in \mathbb{R}^p$ is the parameter in a linear regression model and $g : \mathbb{R}^p \to \mathbb{R}^m$ is some known function. Let $\widehat{\beta}_n = (X^\top X)^{-1} X^\top y$ be the OLS estimate of β_0. If ∇g is continuously differentiable and $\widehat{\beta}_n$ is consistent, then by the classical delta's method (Van der Vaart, 2000) $\mathbb{E}\|g(\widehat{\beta}_n) - g(\beta_0)\|_2^2 = (1 + o(1))\sigma^2 \mathrm{tr}(\nabla g(\beta_0)(X^\top X)^{-1} \nabla g(\beta_0)^\top) = (1 + o(1))\sigma^2 \mathrm{tr}(G_0 (X^\top X)^{-1})$, where $G_0 = \nabla g(\beta_0)^\top \nabla g(\beta_0)$. If G_0 depends on the unknown parameter β_0 then the design dependence problem again exists, and a locally optimal solution can be obtained by replacing G_0 in the objective function with $\check{G} = \nabla g(\check{\beta})^\top \nabla g(\check{\beta})$ for some initial estimate $\check{\beta}$ of β_0.

If $\check{G}$ is invertible, then there exists invertible $p \times p$ matrix $\check{P}$ such that $\check{G} = \check{P}\check{P}^\top$ because $\check{G}$ is positive definite. Applying the linear transform

$$x_i \mapsto \widetilde{x}_i = \check{P}^{-1} x_i$$

we have that $\mathrm{tr}[G_0 (X^\top X)^{-1}] = \mathrm{tr}[(\widetilde{X}^\top \widetilde{X})^{-1}]$, where $\widetilde{X} = [\widetilde{x}_1, \cdots, \widetilde{x}_n]^\top$. Our results in Theorems 5, 7 and 13 remain valid by operating on the transformed matrix $\widetilde{X} = X\check{P}^{-\top}$.

Table 2: Running time (seconds) / no. of iterations for L_5^* and $L_6^\dagger$.

Isotropic design (t distribution), df = 1.					
$k =$	$2p$	$4p$	$6p$	$8p$	$10p$
L_5^*	1.8/31	0.8/19	1.7/26	0.9/14	0.3/9
$L_6^\dagger$	140.1/90	425.9/156	767.5/216	993.7/245	1077/253
Isotropic design (t distribution), df = 3.					
$k =$	$2p$	$4p$	$6p$	$8p$	$10p$
L_5^*	0.6/14	0.4/8	0.3/7	0.2/5	0.2/5
$L_6^\dagger$	158.3/104	438.0/161	802.5/223	985.2/242	1105/252
Skewed design (multivariate Gaussian), $\alpha = 3$.					
$k =$	$2p$	$4p$	$6p$	$8p$	$10p$
L_5^*	0.8/16	0.7/12	0.5/9	0.4/8	0.4/8
$L_6^\dagger$	182.9/120	487.1/180	753.4/212	935.8/230	1057/250

Example: prediction error. In some application scenarios the *prediction error* $\|Z\widehat{\beta} - Z\beta_0\|_2^2$ rather than the estimation error $\|\widehat{\beta} - \beta_0\|_2^2$ is of interesting, either because the linear model is used mostly for prediction or component of the underlying model β_0 lack physical interpretations. Another interesting application is the *transfer learning* (Pan and Yang, 2010), in which the training and testing data have different designs (e.g., Z instead of X) but share the same conditional distribution of labels, parameterized by the linear model β_0.

Suppose $Z \in \mathbb{R}^{m \times p}$ is a known full-rank data matrix upon which predictions are seeked, and define $\widehat{\Sigma}_Z = \frac{1}{m} Z^\top Z \succ 0$ to be the sample covariance of Z. Our algorithmic framework as well as its corresponding analysis remain valid for such prediction problems with transform $x_i \mapsto \widehat{\Sigma}_Z^{-1/2} x_i$. In particular, the guarantees for the greedy algorithm and the with replacement sampling algorithm remain unchanged, and the guarantee for the without replacement sampling algorithm is valid as well, except that the $\|\Sigma_*^{-1}\|_2$ and $\kappa(\Sigma_*)$ terms have to be replaced by the (relaxed) optimal sample covariance after the linear transform $x_i \mapsto \widehat{\Sigma}_Z^{-1/2} x_i$.

4. Numerical results on synthetic data

We report selection performance (measured in terms of $F(S; X) := \mathrm{tr}[(X_S^\top X_S)^{-1}]$ which is the mean squared error of the ordinary least squares estimator based on (X_S, y_S) in estimating the regression coefficients) on synthetic data. Only the without replacement setting is considered, and results for the with replacement setting are similar. In all simulations, the experimental pool (number of given design points) n is set to 1000, number of variables p is set to 50, number of selected design points k ranges from $2p$ to $10p$. For randomized methods, we run them for 20 independent trials under each setting and report the median. Though Fedorov's exchange algorithm is randomized in nature (outcome depending on the initialization), we only run it once for each setting because of its expensive computational requirement. It is observed that in practice, the algorithm's outcome is not sensitive to initializations.

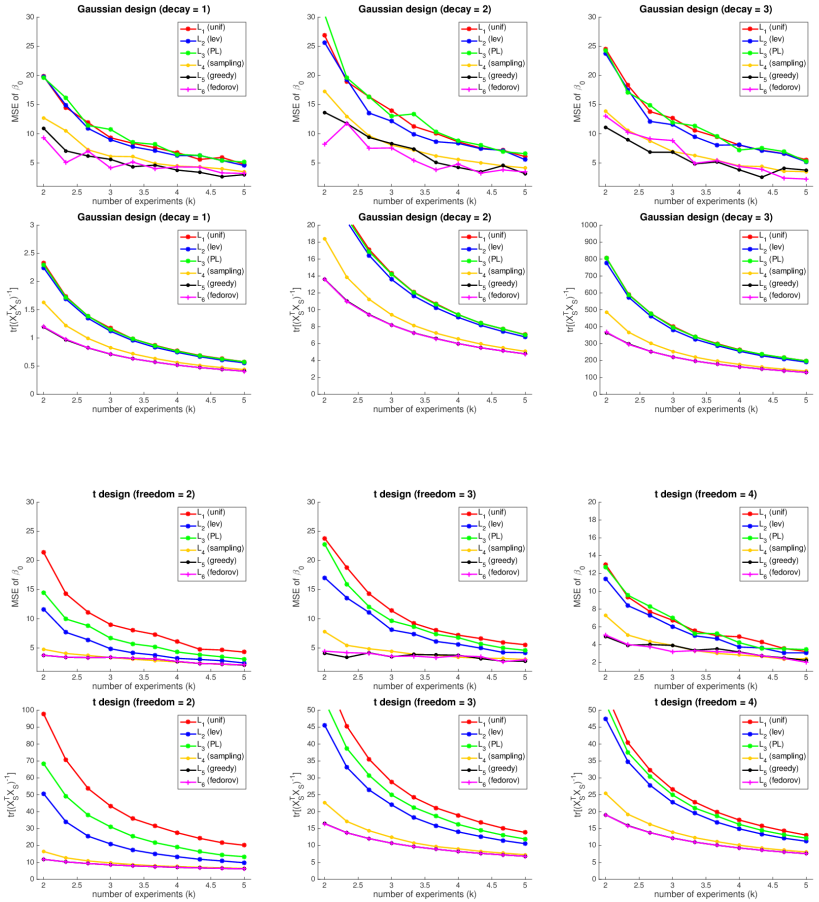


Figure 1: The ratio of MSE $\|\widehat{\beta} - \beta_0\|_2^2$ compared against the MSE of the OLS estimator on the full data pool $\|\widehat{\beta}^{\mathrm{ols}} - \beta_0\|_2^2$, $\widehat{\beta}^{\mathrm{ols}} = (X^\top X)^{-1} X^\top y$, and objective values $F(\widehat{S}; X) = \mathrm{tr}[(X_{\widehat{S}}^\top X_{\widehat{S}})^{-1}]$ on synthetic data sets. Note that the L_5 (greedy) curve is under the L_6 (federov) curve when not visible. See Sec. 4 for details of experimental settings.

4.1 Data generation

The design pool X is generated so that each row of X is sampled from some underlying distribution. We consider two distributional settings for generating X. Similar settings were considered in (Ma et al., 2015) for subsampling purposes.

1. *Distributions with skewed covariance*: each row of X is sampled i.i.d. from a multivariate Gaussian distribution $\mathcal{N}_p(0, \Sigma_0)$ with $\Sigma_0 = U \Lambda U^\top$, where U is a random orthogonal matrix and $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_p)$ controls the skewness or conditioning of X. A power-law decay of λ, $\lambda_j = j^{-\alpha}$, is imposed, with small α corresponding to "flat" distribution and large α corresponding to "skewed" distribution. Rows 1-2 in Figure 1 correspond to this setting.

2. *Distributions with heavy tails*: We use t distribution as the underlying distribution for generating each entry in X, with degrees of freedom ranging from 2 to 4. These distributions are heavy tailed and high-order moments of X typically do not exist. They test the robustness of experiment selection methods. Rows 3-4 in Figure 1 correspond to this setting.

4.2 Methods

The methods that we compare are listed below:

- L_1 (uniform sampling): each row of X is sampled uniformly at random, without replacement.

- L_2 (leverage score sampling): each row of X, x_i, is sampled without replacement, with probability proportional to its leverage score $x_i^\top (X^\top X)^{-1} x_i$. This strategy is considered in Ma et al. (2015) for subsampling in linear regression models.

- L_3 (predictive length sampling): each row of X, x_i, is sampled without replacement, with probability proportional to its ℓ_2 norm $\|x_i\|_2$. This strategy is derived in Zhu et al. (2015).

- L_4^* (sampling based selection with π^*): the sampling based method that is described in Sec. 3.3, based on π^*, the optimal solution of Eq. (4). We consider the hard size constraint algorithm only.

- L_5^* (greedy based selection with π^*): the greedy based method that is described in Sec. 3.4.

- $L_6^\dagger$ (Fedorov's exchange algorithm): the Fedorov's exchange algorithm (Miller and Nguyen, 1994) with the A-optimality objective. Note that this method does not have guarantees, nor does it provably terminate in polynomial number of exchanges. Description of the algorithm is placed in Appendix C.

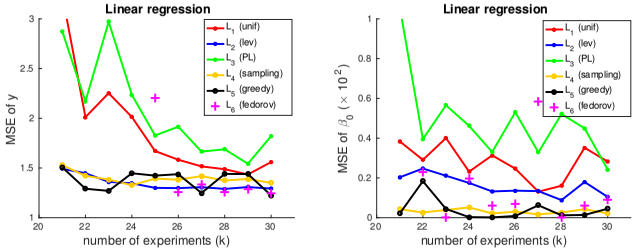


Figure 2: Results on real material synthesis data ($n = 133, p = 9, 20 < k \leq 30$). Left: prediction error $\|y - X\widehat{\beta}\|_2^2$, normalized by dividing by the prediction error $\|y - X\widehat{\beta}^{\text{ols}}\|_2^2$ of the OLS estimator on the full data set; right: estimation error $\|\widehat{\beta} - \widehat{\beta}^{\text{ols}}\|_2^2$. All randomized algorithms ($L_1, L_2, L_3, L_4^*, L_6^\dagger$) are run for 50 times and the median performance is reported.

4.3 Performance

The ratio of the mean square error $\|\widehat{\beta} - \beta_0\|_2^2$ compared to the OLS estimator on the full data set $\|\widehat{\beta}^{\text{ols}} - \beta_0\|_2^2$, $\widehat{\beta}^{\text{ols}} = (X^\top X)^{-1} X^\top y$, and objective values $F(\widehat{S}; X) = \text{tr}((X_{\widehat{S}}^\top X_{\widehat{S}})^{-1})$ are reported in Figure 1.

Table 2 reports the running time and number of iterations of L_5^* (greedy based selection) and $L_6^\dagger$ (Fedorov's exchange algorithm). In general, the greedy method is 100 to 1000 times faster than the exchange algorithm, and also converges in much fewer iterations.

For both synthetic settings, our proposed methods (L_4^* and L_5^*) significantly outperform existing approaches (L_1, L_2, L_3), and their performance is on par with the Fedorov's exchange algorithm, which is much more computationally expensive (cf. Table 2).

5. Numerical results on real data

5.1 The material synthesis dataset

We apply our proposed methods to an experimental design problem of low-temperature microwave-assisted crystallization of ceramic thin films (Reeja-Jayan et al., 2012; Nakamura et al., 2017). The microwave-assisted experiments were controlled by four experimental parameters: temperature T ($120°C$ to $170°C$), hold time H (0 to 60 minutes), ramp power P ($0W$ to $60W$) and tri-ethyl-gallium (TEG) volume ratio R (from 0 to 0.93). A generalized quadratic regression model was employed in (Nakamura et al., 2017) to estimate the coverage percentage of the crystallization CP:

$$CP = \beta_1 + \beta_2 T + \beta_3 H + \beta_4 P + \beta_5 R + \beta_6 T^2 + \beta_7 H^2 + \beta_8 P^2 + \beta_9 R^2. \quad (9)$$

As a proof of concept, we test our proposed algorithms and compare with baseline methods on a data set consisting of 133 experiments, with the coverage percentage fully collected and measured. We consider selection of subsets of k experiments, with k ranging

Table 3: Selected subsets of uniform sampling (Uniform), leverage score sampling (Levscore) and greedy method (Greedy) on the material synthesis dataset.

Uniform (L_1)				Levscore (L_2)				Greedy (L_5^*)			
T	P	H	R	T	P	H	R	T	P	H	R
150	30	7.5	.07	120	30	5	.07	140	15	7.5	.80
165	30	3	.73	140	15	2.5	.80	140	15	2.5	.80
140	15	7.6	.80	170	30	7.5	.93	140	60	2.5	.80
170	30	2.5	.07	140	60	7.5	.80	160	15	7.5	.80
160	15	7.5	.80	150	15	2.5	.80	160	60	7.5	.80
150	15	2.5	.80	150	30	60	.67	140	60	2.5	.80
160	30	2.5	.80	160	30	30	.87	160	60	7.5	.80
150	30	6.1	.87	170	30	2.5	.07	170	30	7.5	.93
160	30	6.1	.87	170	30	0	.93	165	30	7.5	0
150	30	4.1	.67	140	30	5	.80	150	30	5	0
165	30	7.5	.13	140	30	2.5	.87	135	30	2.5	.5
140	30	15	.87	140	30	6.2	.67	150	30	4.1	.87
160	60	7.5	.80	160	30	60	.87	150	30	2.5	.87
150	30	5	.50	140	30	60	.87	135	30	3	.50
140	30	7.5	.80	160	30	7.5	.80	135	30	3	0
150	30	5	.80	160	30	2.5	.87	140	30	30	.67
140	30	6.1	.67	140	30	4.1	.67	140	30	30	.87
140	30	2.5	.67	135	30	3	.50	150	30	30	.87
150	60	5	.80	150	30	30	.67	160	30	30	.87
160	30	5	.80	165	30	5	.67	120	30	5	.07
135	30	3	.07	160	60	7.5	.80	160	30	60	.87
165	30	5	0	165	30	3	.67	140	30	60	.87
120	30	7.5	.93	135	30	3	.67	120	30	7	.07
160	30	5	.80	160	15	7.5	.80	120	30	0	.93
165	30	0	.73	150	60	7.5	.80	170	30	0	.93
165	30	3	.67	160	30	4.1	.67	170	30	0	.07
150	15	7.5	.80	165	30	0	.07	150	30	0	.5
140	30	5	.80	135	30	2.5	.73	150	30	0	.6
150	60	2.5	.80	150	30	15	.87	165	30	0	.5
135	30	3	.67	160	60	2.5	.80	160	30	60	.07

Table 4: Comparison of the "true" model $\beta_0 = [0.49; 0.30; 0.19; 3.78]$ and the subsampled estimators $\widehat{\beta}$ for different components of β. k is the number of subsamples. Random algorithms (L_1 through L_4^*) are repeated for 100 times and the median difference is reported.

	$\Delta\beta_1$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\beta_4$	$\|\Delta\beta\|_2$		$\Delta\beta_1$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\beta_4$	$\|\Delta\beta\|_2$
$k = 20$						$k = 30$					
L_1	.043	.071	.074	.274	.295	L_1	.038	.064	.070	.171	.199
L_2	.017	.026	.018	.260	.263	L_2	.011	.024	.014	.199	.201
L_3	.017	.043	.026	.243	.249	L_3	.015	.031	.018	.236	.239
L_4^*	.014	.039	.114	.215	.247	L_4^*	.014	.023	.030	.044	.060
L_5^*	.010	.024	.032	.209	.213	L_5^*	.022	.000	.050	.088	.104
$L_6^\dagger$	.010	.024	.032	.209	.213	$L_6^\dagger$	.022	.000	.050	.088	.104
$k = 50$						$k = 75$					
L_1	.025	.035	.037	.130	.142	L_1	.016	.032	.024	.133	.139
L_2	.009	.027	.012	.130	.134	L_2	.006	.034	.010	.126	.131
L_3	.011	.035	.013	.166	.170	L_3	.009	.029	.009	.124	.128
L_4^*	.022	.009	.036	.097	.106	L_4^*	.012	.001	.015	.036	.041
L_5^*	.025	.003	.040	.009	.048	L_5^*	.005	.013	.009	.016	.023
$L_6^\dagger$	.025	.003	.040	.009	.048	$L_6^\dagger$	.005	.012	.010	.011	.020

from 21 to 30, and report in Figure 2 the mean-square error (MSE) of both the prediction error $\|y - X\widehat{\beta}\|_2^2$, normalized by dividing by the prediction error $\|y - X\widehat{\beta}_{\text{ols}}\|_2^2$ of the OLS estimator on the full data set, on the full 133 experiments and the estimation error $\|\widehat{\beta} - \widehat{\beta}^{\text{ols}}\|_2^2$. Figure 2 shows that our proposed methods consistently achieve the best performance and are more stable compared to uniform sampling and Fedorov's exchange methods, even though the linear model assumption may not hold.

We also report the actual subset ($k = 30$) design points selected by uniform sampling (L_1), leverage score sampling (L_3) and our greedy method (L_5^*) in Table 3. Table 3 shows that the greedy algorithm (L_5^*) picked very diverse experimental settings, including several settings of very lower temperature ($120°C$) and very short hold time (0). In contrast, the experiments picked by both uniform sampling (L_1) and leverage score sampling (L_3) are less diverse.

5.2 The CPU performance dataset

CPU relative performance refers to the relative performance of a particular CPU model in terms of a base machine - the IBM 370/158 model. Comprehensive benchmark tests are required to accurately measure the relative performance of a particular CPU, which might be time-consuming or even impossible if the actual CPU has not been on the market yet. Ein-Dor and Feldmesser (1987) considered a linear regression model to characterize the relationship between CPU relative performance and several CPU capacity parameters such as main memory size, cache size, channel units and machine clock cycle time. These parameters are fixed for any specific CPU model and could be known even before the manufacturing process. The learned model can also be used to predict the relative performance of a new CPU model based on its model parameters, without running extensive benchmark tests.

Using domain knowledge, Ein-Dor and Feldmesser (1987) narrow down to three parameters of interest: *average memory size* (X_1), *cache size* (X_2) and *channel capacity* (X_3), all being explicitly computable functions from CPU model parameters. An offset parameter is also involved in the linear regression model, making the number of variables $p = 4$. A total of $n = 209$ CPU models are considered, with all of the model parameters and relative performance collected and no missing data. Stepwise linear regression was applied to obtain the following linear model:

$$0.49X_1 + 0.30X_2 + 0.19X_3 + 3.78. \tag{10}$$

To use this data set as a benchmark for evaluating the experiment selection methods, we synthesize labels Y using model Eq. (10) with standard Gaussian noise and measure the difference between fit $\widehat{\beta}$ and the true model $\beta_0 = [0.49; 0.30; 0.19; 3.78]$. Table 4 shows that under various measurement budget (k) constraints, the proposed methods L_4^* and L_5^* consistently achieve small estimation error, and is comparable to the Fedorov's exchange algorithm ($L_6^\dagger$). As the data set is small (209×4), all algorithms are computationally efficient.

5.3 The Minnesota Wind Dataset

The Minnesota wind dataset collects wind speed information across $n = 2642$ locations in Minnesota, USA for a period of 24 months (for the purpose of this experiment, we only use wind speed data for one month). The 2642 locations are connected with 3304 bi-directional roads, which form an $n \times n$ sparse unweighted undirected graph G. Let $L = \text{diag}(d) - G$ be the $n \times n$ Laplacian of G, where d is a vector of node degrees and let $V \in \mathbb{R}^{n \times p}$ be an orthonormal eigenbasis corresponding to the smallest p eigenvalues of L. As the wind speed signal $y \in \mathbb{R}^n$ is relatively smooth, it can be well-approximated as $y = V\theta + \varepsilon$, where $\theta \in \mathbb{R}^p$ corresponds to the coefficients of the graph Laplacian basis. The speed signal y has a fast decay on the Laplacian basis V; in particular, OLS on the first $p = 50$ basis accounts for over 99% of the variation in y. (That is, $\|V\widehat{\theta}^{\text{ols}} - y\|_2^2 \le 0.01\|y\|_2^2$, where $\widehat{\theta}^{\text{ols}} = (V^\top V)^{-1}V^\top y$.)

In our experiments we compare the 6 algorithms (L_1 through L_6) when a very small portion of the full samples is selected. The ratio of the mean-square prediction error $\frac{1}{n}\|V\widehat{\theta} - y\|_2^2$ compared to the MLE of the full-sample OLS estimator $\frac{1}{n}\|V\widehat{\theta}^{\text{ols}} - y\|_2^2$ is reported. Apart from the real speed data, we also report results under a "semi-synthetic" setting similar to Sec. 5.2, where the full OLS estimate $\widehat{\theta}^{\text{ols}}$ is first computed and then y is synthesized as $y = V\widehat{\theta}^{\text{ols}} + \varepsilon$ where ε are i.i.d. standard Normal random variables.

From Figure 3, greedy methods ($L_5^*, L_6^\dagger$) achieve consistently the lowest MSE and are robust to subset size k. However, the Fedorov's exchange algorithm is very slow and requires more than 10 times the running time than our proposed methods (L_4^*, L_5^*). On the other hand, sampling based methods (L_1 through L_4^*) behave quite badly when subset size k is small and close to the problem dimension p. This is because when k is close to p, even very small changes resulted from randomization could lead to highly singular designs and hence significantly increases the mean-square prediction error. We also observe that for

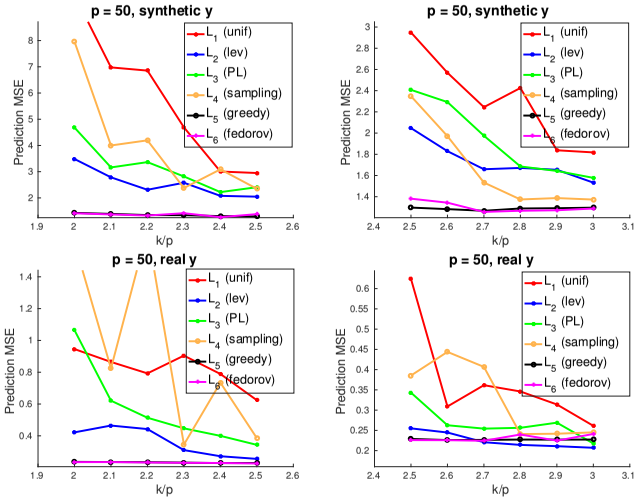


Figure 3: Plots of the ratio of mean square prediction error $\frac{1}{n}\|V\widehat{\theta} - y\|_2^2$ compared against the MSE of the full-sample OLS estimator $\frac{1}{n}\|V\widehat{\theta}^{\text{ols}} - y\|_2^2$ on the Minnesota wind dataset. In the top panel response variables y are synthesized as $y = V\widehat{\theta}^{\text{ols}} + \varepsilon$, where $\widehat{\theta}^{\text{ols}}$ is the full-sample OLS estimator and ε are i.i.d. standard Normal random variables. In the bottom panel the real wind speed response y is used. For randomized algorithms (L_1 to L_4) the experiments are repeated for 50 times and the median MSE ratio is reported.

sufficiently large subset size (e.g., $k = 3p$), the performance gap between all methods is smaller on real signal compared to the synthetic signals. We conjecture that this is because the linear model $y = V\theta + \varepsilon$ only approximately holds in real data.

6. Discussion

We discuss potential improvements in the analysis presented in this paper.

6.1 Sampling based method

To fully understand the finite-sample behavior of the sampling method introduced in Secs. 3.2 and 3.3, it is instructive to relate it to the *graph spectral sparsification* problem (Spielman and Srivastava, 2011) in theoretical computer science: Given a directed weighted graph $G = (V, E, W)$, find a subset $\widetilde{E} \subseteq E$ and new weights $\widetilde{W}$ such that $\widetilde{G} = (V, \widetilde{E}, \widetilde{W})$ is a *spectral sparsification* of G, which means there exists $\epsilon \in (0, 1)$ such that for any vector

$z \in \mathbb{R}^{|V|}$:

$$(1-\epsilon)\sum_{(u,v)\in E} w_{uv}(z_u - z_v)^2 \le \sum_{(u,v)\in \widetilde{E}} \widetilde{w}_{uv}(z_u - z_v)^2 \le (1+\epsilon)\sum_{u,v\in E} w_{uv}(z_u - z_v)^2.$$

Define $B_G \in \mathbb{R}^{|E|\times|V|}$ to be the *signed edge-vertex incidence matrix*, where each row of B_G corresponds to an edge in E, each column of B_G corresponds to a vertex in G, and $[B_G]_{ij} = 1$ if vertex j is the head of edge i, $[B_G]_{ij} = -1$ if vertex j is the tail of edge i, and $[B_G]_{ij} = 0$ otherwise. The spectral sparsification requirement can then be equivalently written as

$$(1-\epsilon)z^\top(B_G^\top W B_G)z \le z^\top(B_{\widetilde{G}}^\top \widetilde{W} B_{\widetilde{G}})z \le (1+\epsilon)z^\top(B_G^\top W B_G)z.$$

The similarity of graph sparsification and the experimental selection problem is clear: $B_G \in \mathbb{R}^{n\times p}$ would be the known pool of design points and $W = \text{diag}(\pi^*)$ is a diagonal matrix with the optimal weights π^* obtained by solving Eq. (4). The objective is to seek a small subset of rows in B_G (i.e., the sparsified edge set $\widetilde{E}$) which is a spectral approximation of the original $B_G^\top \text{diag}(\pi^*) B_G$. Approximation of the A-optimality criterion or any other eigen-related quantity immediately follows. One difference is that in linear regression each row of B_G is no longer a $\{\pm 1, 0\}$ vector. Also, the subsampled weight matrix $\widetilde{W}$ needs to correspond to an unweighted graph for linear regression, i.e. diagonal entries of $\widetilde{W}$ must be in $\{0, 1\}$. However, we consider this to be a minor difference as it does not interfere with the spectral properties of B_G.

The spectral sparsification problem where $\widetilde{W}$ can be arbitrarily designed (i.e. not restricted to have $\{0, 1\}$ diagonal entries) is completely solved (Spielman and Srivastava, 2011; Batson et al., 2012), where the size of the selected edge subset is allowed to be *linear* to the number of vertices, or in the terminology of our problem, $k \asymp p$. Unfortunately, both methods require the power of arbitrary designing the weights in $\widetilde{W}$, which is generally not available in experiment selection problems (i.e., cannot set noise variance or signal strength arbitrarily for individual design points). Recently, it was proved that when the original graph is unweighted ($W = I$), it is also possible to find unweighted linear-sized edge sparsifiers ($\widetilde{W} \propto I$) (Marcus et al., 2015a,b; Anderson et al., 2014). This remarkable result leads to the solution of the long-standing Kardison-Singer problem. However, the condition that the *original* weights W are uniform is not satisfied in the linear regression problem, where the optimal solution π^* may be far from uniform. The experiment selection problem somehow falls in between, where an *unweighted* sparsifier is desired for a *weighted* graph. This leads us to the following question:

Question 6.1 *Given a* weighted *graph $G = (V, E, W)$, under what conditions are there small edge subset $\widetilde{E} \subseteq E$ with* uniform *weights $\widetilde{W} \propto I$ such that $\widetilde{G} = (V, \widetilde{E}, \widetilde{W})$ is a (one-sided) spectral approximation of G?*

The answer to the above question, especially the smallest possible edge size $|\widetilde{E}|$, would have immediate consequences on finite-sample conditions of k and p in the experiment selection problem.

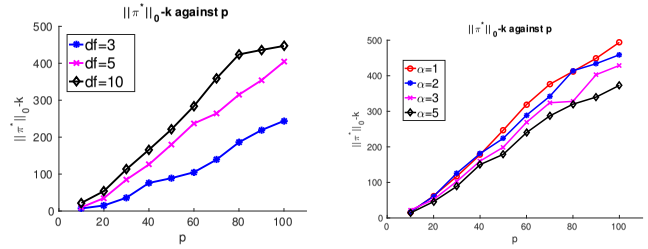


Figure 4: Empirical verification of the rate of $\|\pi^*\|_0$. Left: isotropic design (t distribution); Right: skewed design (transformed multivariate Gaussian) with spectral decay $\lambda_j \propto j^{-\alpha}$.

6.2 Greedy method

Corollary 11 shows that the approximation quality of the greedy based method depends crucially upon $\|\pi^*\|_0$, the support size of the optimal solution π^*. In Lemma 12 we formally established that $\|\pi^*\|_0 \le k + p(p+1)/2$ under mild conditions; however, we conjecture the $p(p+1)/2$ term is loose and could be improved to $O(p)$ in general cases.

In Fig 4 we plot $\|\pi^*\|_0 - k$ against number of variables p, where p ranges from 10 to 100 and k is set to $3p$. The other simulation settings are kept unchanged. We observe that in all settings $\|\pi^*\|_0 - k$ scales linearly with p, suggesting that $\|\pi^*\|_0 \le k + O(p)$. Furthermore, the slope of the scalings does not seem to depend on the conditioning of X, as shown in the right panel of Fig 4 where the conditioning of X is controlled by the spectral decay rate α. This is in contrast to the analysis of the sampling based method (Theorem 5), in which the finite sample bound depends crucially upon the conditioning of X for the without replacement setting.

6.3 High-dimensional settings

This paper focuses solely on the so-called *large-sample, low-dimensional* regression setting, where the number of variables p is assumed to be much smaller than both design pool size n and subsampling size k. It is an interesting open question to extend our results to the *high-dimensional* setting, where p is much larger than both n and k, and an assumption like sparsity of β_0 is made to make the problem feasible, meaning that very few components in β_0 are non-zero. In particular, it is desirable to find a sub-sampling algorithm that attains the following minimax estimation rate over high-dimensional sparse models:

$$\inf_{A \in \mathcal{A}_b(k)} \sup_{\|\beta_0\|_0 \le s} \mathbb{E}_{A,\beta_0} \left[\|\widehat{\beta} - \beta_0\|_2^2 \right].$$

One major obstacle of designing subsampling algorithms for high-dimensional regression is the difficulty of evaluating and optimizing the *restricted eigenvalue* (or other similar criteria) of the subsampled covariance matrix. Unlike the ordinary spectrum, the restricted eigenvalue of a matrix is NP-hard to compute (Dobriban and Fan, 2016) and heavily influences the statistical efficiency of a Lasso-type estimator (Bickel et al., 2009). The question

of optimizing such restricted eigenvalues could be even harder and remains largely open. We also mention that Davenport et al. (2015) suggests a two-step approach where half the budget is used to identify the sparse locations from randomly sampled points and the remaining budget is used to generate a better estimate of regression coefficients using the same convex programming formulation proposed in our paper. Their paper provides some experimental support for this idea, however, no theoretical guarantees are established for the (sub)optimality of such a procedure. Analyzing such a two-step approach could be an interesting future direction.

6.4 Approximate linear models

In cases when the linear model $y = X\beta_0 + \varepsilon$ only approximately holds, we describe here a method that takes into consideration both bias and variance of OLS estimates on subsampled data in order to find good sub-samples. Suppose $y = f_0(X) + \varepsilon$ for some unknown underlying function f_0 that might not be linear, and let $\beta^* = (X^\top X)^{-1} X^\top f_0(X)$ be the optimal linear predictor on the full sample X. Suppose $X_S = \Psi X$ is the subsampled data, where $\Psi \in \mathbb{R}^{|S| \times n}$ is the subsampling matrix where each row of Ψ is $e_i = (0, \cdots, 0, 1, 0, \cdots, 0) \in \mathbb{R}^n$, with $i \in [n]$ being the subsampled rows in X. Let $\widehat{\beta} = (X_S^\top X_S)^{-1} X_S^\top y_S$ be the OLS on the subsampled data. The error of $\widehat{\beta}$ can then be decomposed and upper bounded as

$$\mathbb{E}\|\widehat{\beta} - \beta^*\|_2^2 = f_0(X)^\top \left[(X^\top \Psi^\top \Psi X)^{-1} X^\top \Psi^\top \Psi - (X^\top X)^{-1} X^\top \right] f_0(X) + \sigma^2 \mathrm{tr}\left[(X^\top \Psi^\top \Psi X)^{-1} \right] \tag{11}$$

$$\le \left\| (X^\top \Psi^\top \Psi X)^{-1} X^\top \Psi^\top \Psi - (X^\top X)^{-1} X^\top \right\|_{\mathrm{op}} \cdot \|f_0(X)\|_2^2 + \sigma^2 \mathrm{tr}\left[(X^\top \Psi^\top \Psi X)^{-1} \right]. \tag{12}$$

In the high noise setting, the first term can be ignored and the solution is close to the one considered in this paper. In the low-noise setting, the second term can be ignored and a relaxation similar to Eq. (4) can be derived; however a linear approximation may be undesirable in this case. In general, when $\|f_0(X)\|_2^2 \approx \lambda$ is known or can be estimated, the following continuous optimization problem serves as an approximate objective of subsampled linear regression with approximate linear models:

$$\min_{\pi \in \mathbb{R}^n} \quad \lambda \left\| \left(\sum_{i=1}^n \pi_i x_i x_i^\top \right)^{-1} X^\top \mathrm{diag}(\pi) - (X^\top X)^{-1} X^\top \right\|_{\mathrm{op}} + \sigma^2 \mathrm{tr}\left[\left(\sum_{i=1}^n \pi_i x_i x_i^\top \right)^{-1} \right]$$

$$s.t. \quad \sum_{i=1}^n \pi_i \le k, \quad 0 \le \pi_i \le 1.$$

Unfortunately, the relaxation in Eq. (12) may be loose and hence the optimization problem above fails to serve as a good objective for the optimal subsampling problem with approximate linear models. Exact characterization of the bias term in Eq. (11) without strong assumptions on f_0 is an interesting open question.

7. Proofs

7.1 Proof of facts in Sec. 3.1

Proof [Proof of Fact 3.1] Let $A = X^\top \mathrm{diag}(\pi) X$ and $B = X^\top \mathrm{diag}(\pi') X$. By definition, $B = A + \Delta$ where $\Delta = X^\top \mathrm{diag}(\pi' - \pi) X$ is positive semi-definite. Subsequently, $\sigma_\ell(B) \geq \sigma_\ell(A)$ for all $\ell = 1, \cdots, p$. We then have that

$$f(\pi'; X) = \mathrm{tr}(B^{-1}) = \sum_{\ell=1}^{p} \sigma_\ell(B)^{-1} \leq \sum_{\ell=1}^{p} \sigma_\ell(A)^{-1} = \mathrm{tr}(A^{-1}) = f(\pi; X).$$

Note also that if $\pi' \neq \pi$ then $\Delta \neq 0$ and hence there exists at least one ℓ with $\sigma_\ell(B) < \sigma_\ell(A)$. Therefore, the equality holds if and only if $\pi = \pi'$. ■

Proof [Proof of Fact 3.2] Suppose $\|\pi^*\|_1 < k$. Then there exists some coordinate $j \in \{1, \cdots, n\}$ such that $\pi^*_j < 1$. Define π' as $\pi'_i = \pi^*_i$ for $i \neq j$ and $\pi'_j = \min\{1, \pi^*_j + k - \|\pi^*\|_1\}$. Then π' is also a feasible solution. On the other hand, By Fact 3.1 we have that $f(\pi'; X) < f(\pi^*; X)$, contradicting the optimality of π^*. Therefore, $\|\pi^*\|_1 = k$. ■

7.2 Proof of Theorem 3

We only prove Theorem 3 for the with replacement setting ($b = 1$). The proof for the without replacement setting is almost identical.

Define $\widetilde{\mathcal{A}}_1(k)$ as the class of *deterministic* algorithms that proceed as follows:

1. The algorithm deterministically outputs pairs $\{(w_i, x_i)\}_{i=1}^M$, where x_i is one of the rows in X and $\{w_i\}_{i=1}^n$ satisfies $w_i \geq 0$, $\sum_{i=1}^M w_i \leq k$. Here M is an arbitrary finite integer.
2. The algorithm observes $\{y_i\}_{i=1}^M$ with $y_i = \sqrt{w_i} x_i^\top \beta + \varepsilon_i$. Here β is a fixed but unknown regression model and $\varepsilon_i \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$ is the noise.
3. The algorithm outputs $\widehat{\beta}$ as an estimation of β, based on the observations $\{y_i\}_{i=1}^M$.

Because all algorithms in $\widetilde{\mathcal{A}}_1(k)$ are deterministic and the design matrix still has full column rank, the optimal estimator of $\widehat{\beta}$ *given* $\{(w_i, x_i)\}_{i=1}^M$ is the OLS estimator. For a specific data set weighted through $\sqrt{w_1}, \cdots, \sqrt{w_M}$, the minimax estimation error (which is achieved by OLS) is given by

$$\inf_{\widehat{\beta}} \sup_{\beta} \mathbb{E}\left[\|\widehat{\beta} - \beta\|_2^2\right] = \sup_{\beta} \mathbb{E}\left[\|\widehat{\beta}^{\mathrm{ols}} - \beta\|_2^2\right] = \sigma^2 \mathrm{tr}\left[(\widetilde{X}^\top \widetilde{X})^{-1}\right] = \sigma^2 \mathrm{tr}\left[\left(\sum_{i=1}^n \widetilde{w}_i x_i x_i^\top\right)^{-1}\right],$$

where $\widetilde{w}_i$ is the aggregated weight of data point x_i in all the M weighted pairs. Subsequently,

$$\inf_{\widetilde{A} \in \widetilde{\mathcal{A}}_1(k)} \sup_{\beta} \mathbb{E}\left[\|\widehat{\beta}_{\widetilde{A}} - \beta\|_2^2\right] = \inf_{w_1 + \cdots + w_M \leq k} \sigma^2 \mathrm{tr}\left[\left(\sum_{i=1}^M w_i x_i x_i^\top\right)^{-1}\right] = \sigma^2 f_1^*(k; X).$$

It remains to prove that

$$\inf_{\widetilde{A} \in \widetilde{\mathcal{A}}_1(k)} \sup_{\beta} \mathbb{E}\left[\|\widehat{\beta}_{\widetilde{A}} - \beta\|_2^2\right] \leq \inf_{A \in \mathcal{A}_1(k)} \sup_{\beta} \mathbb{E}\left[\|\widehat{\beta}_A - \beta\|_2^2\right].$$

We prove this inequality by showing that for every (possibly random) algorithm $A \in \mathcal{A}_1(k)$, there exists $\widetilde{A} \in \widetilde{\mathcal{A}}_1(k)$ such that $\sup_\beta \mathbb{E}[\|\widehat{\beta}_A - \beta\|_2^2] \geq \sup_\beta \mathbb{E}[\|\widehat{\beta}_{\widetilde{A}} - \beta\|_2^2]$. To see this, we construct $\widetilde{A}$ based on A as follows:

1. For every k-subset (duplicates allowed) of all possible outputs of A (which by definition are all subsets of X) and its corresponding weight vector w, add $(w'_i, \widetilde{x}_i)$ to the design set of $\widetilde{A}$, where $w'_i = w_i \Pr_A(\widetilde{X})$.
2. The algorithm $\widetilde{A}$ observes all responses $\{y_i\}$ for $\{(w'_i, \widetilde{x}_i)\}$.
3. $\widetilde{A}$ outputs the *expected* estimation of A; that is, $\widetilde{A}(X, y) = \mathbb{E}_{\widetilde{X}}[\mathbb{E}_y[\widehat{\beta}_A | \widetilde{X}]]$. Note that by definition of the estimator class $\mathcal{A}_1(k)$, all estimators $\widehat{\beta}_A$ *conditioned on subsampled data points* $\widetilde{X}$ are deterministic.

We claim that $\widetilde{A} \in \widetilde{\mathcal{A}}_1(k)$ because

$$\sum_i w'_i = \sum_{\widetilde{X}} \Pr(\widetilde{X}) \sum_{i=1}^k w_i \leq k.$$

Furthermore, by Jensen's inequality we have

$$\mathbb{E}_{\widetilde{X}, y}\left[\|\widehat{\beta}_A - \beta\|_2^2\right] \geq \mathbb{E}_y\left[\|\mathbb{E}_{\widetilde{X}}[\widehat{\beta}_A] - \beta\|_2^2\right] = \mathbb{E}_y\left[\|\widehat{\beta}_{\widetilde{A}} - \beta\|_2^2\right].$$

Taking supreme over β we complete the proof.

7.3 Proof of Lemma 6

With replacement setting Define $\Phi = \mathrm{diag}(\pi^*)$ and $\Pi = \Phi^{1/2} X \Sigma_*^{-1} X^\top \Phi^{1/2} \in \mathbb{R}^{n \times n}$. The following proposition lists properties of Π:

Proposition 14 (Properties of projection matrix) *The following properties for* Π *hold:*

1. Π *is a projection matrix. That is,* $\Pi^2 = \Pi$.
2. $\mathrm{Range}(\Pi) = \mathrm{Range}(\Phi^{1/2} X)$.
3. *The eigenvalues of* Π *are 1 with multiplicity* p *and 0 with multiplicity* $n - p$.
4. $\Pi_{ii} = \|\Pi_{i,\cdot}\|_2^2 = \pi_i^* x_i^\top \Sigma_*^{-1} x_i$.

Proof Proof of 1: By definition, $\Sigma_* = X^\top \Phi X$ and subsequently

$$\begin{aligned} \Pi^2 &= \Phi^{1/2} X \Sigma_*^{-1} X^\top \Phi^{1/2} \Phi^{1/2} X \Sigma_*^{-1} X^\top \Phi^{1/2} \\ &= \Phi^{1/2} X (X^\top \Phi X)^{-1} X^\top \Phi X (X^\top \Phi X)^{-1} \Phi^{1/2} \\ &= \Phi^{1/2} X (X^\top \Phi X)^{-1} X^\top \Phi^{1/2} = \Pi. \end{aligned}$$

Proof of 2: First note that $\text{Range}(\Pi) = \text{Range}(\Phi^{1/2}X\Sigma_*^{-1}X^\top\Phi^{1/2}) \subseteq \text{Range}(\Phi^{1/2}X)$. For the other direction, take arbirary $u \in \text{Range}(\Phi^{1/2}X)$ and express u as $u = \Phi^{1/2}Xv$ for some $v \in \mathbb{R}^p$. We then have

$$\begin{aligned}\Pi u &= \Phi^{1/2}X\Sigma_*^{-1}X^\top\Phi^{1/2}u \\ &= \Phi^{1/2}X(X^\top\Phi X)^{-1}X^\top\Phi^{1/2}\Phi^{1/2}Xv \\ &= \Phi^{1/2}Xv = u\end{aligned}$$

and hence $u \in \text{Range}(\Pi)$.

Proof of 3: Because $\Sigma_* = X^\top\Phi X$ is invertible, the $n\times p$ matrix $\Phi^{1/2}X$ must have full column rank and hence $\ker(\Phi^{1/2}X) = \{0\}$. Consequently, $\dim(\text{Range}(\Pi)) = \dim(\text{Range}(\Phi^{1/2}X)) = p - \dim(\ker(\Phi^{1/2}X)) = p$. On the other hand, the eigenvalues of Π must be either 0 or 1 because Π is a projection matrix. So the eigenvalues of Π are 1 with multiplicity p and 0 with multiplicity $n-p$.

Proof of 4: By definition,

$$\Pi_{ii} = \sqrt{\pi_i^*}x_i^\top\Sigma_*^{-1}x_i\sqrt{\pi_i^*} = \pi_i^* x_i^\top\Sigma_*^{-1}x_i.$$

In addition, Π is a symmetric projection matrix. Therefore,

$$\Pi_{ii} = [\Pi^2]_{ii} = \Pi_{i,\cdot}^\top\Pi_{i,\cdot} = \|\Pi_{i,\cdot}\|_2^2.$$

■

The following lemma shows that a spectral norm bound over deviation of the projection matrix implies spectral approximation of the underlying (weighted) covariance matrix.

Lemma 15 (Spielman and Srivastava (2011), Lemma 4) *Let $\Pi = \Phi^{1/2}X\Sigma_*^{-1}X^\top\Phi^{1/2}$ and W be an $n\times n$ non-negative diagonal matrix. If $\|\Pi W\Pi - \Pi\|_2 \le \epsilon$ for some $\epsilon \in (0, 1/2)$ then*

$$(1-\epsilon)u^\top\Sigma_* u \le u^\top\widetilde{\Sigma}_* u \le (1+\epsilon)u^\top\Sigma_* u, \quad \forall u \in \mathbb{R}^p,$$

where $\Sigma_* = X^\top\Phi X$ and $\widetilde{\Sigma}_* = X^\top W^{1/2}\Phi W^{1/2}X$.

We next proceed to find an appropriate diagonal matrix W and validate Lemma 15. Define $w_j^* = \pi_j^*/(kp_j^{(2)})$ and $\widehat{\Sigma}_{\widehat{W}} = \sum_{t=1}^k w_{i_t}^* x_{i_t}x_{i_t}^\top$. It is obvious that $\widehat{\Sigma}_{\widehat{W}} \preceq \widehat{\Sigma}_{\widehat{S}}$ because $w_i = \lceil w_i^* \rceil \ge w_i^*$. It then suffices to lower bound the spectrum of $\widehat{\Sigma}_{\widehat{W}}$ by the spectrum of $\widehat{\Sigma}_*$. Define random diagonal matrix $W^{(1)}$ as ($\mathbb{I}[\cdot]$ is the indicator function)

$$W_{jj}^{(1)} = \frac{\sum_{t=1}^k w_{i_t}^*\mathbb{I}[i_t = j]}{\pi_j^*}, \quad j = 1, \cdots, n.$$

Then by definition, $\widetilde{\Sigma}_* = X^\top W^{1/2}\Phi W^{1/2}X = \sum_{t=1}^k w_{i_t}^* x_{i_t}x_{i_t}^\top = \widehat{\Sigma}_{\widehat{R}}$. The following lemma bounds the perturbation $\|\Pi W\Pi - \Pi\|_2$ for this particular choice of W.

Lemma 16 *For any $\epsilon > 0$,*

$$\Pr\left[\|\Pi W^{(1)}\Pi - \Pi\|_2 > \epsilon\right] \le 2\exp\left\{-C\cdot\frac{k\epsilon^2}{p\log k}\right\},$$

where $C > 0$ is an absolute constant.

Proof Define n-dimensional random vector v as [3]

$$\Pr\left[v = \sqrt{\frac{kw_j^*}{\pi_j^*}}\Pi_{j\cdot}\right] = p_j^{(2)}, \quad j = 1, \cdots, n.$$

Let $v_1, \cdots, v_k$ be i.i.d. copies of v and define $A_t = v_t v_t^\top$. By definition, $\Pi W^{(1)}\Pi$ is equally distributed with $\frac{1}{k}\sum_{t=1}^k A_t$. In addition,

$$\mathbb{E}A_t = \sum_{j=1}^n \frac{kw_j^*}{\pi_j^*}p_j^{(2)}\Pi_{j\cdot}\Pi_{j\cdot}^\top = \Pi^2 = \Pi,$$

which satisfies $\|\mathbb{E}A_t\|_2 = 1$, and

$$\|A_t\|_2 = \|v_t\|_2^2 \le \sum_{1\le j\le n}\frac{kw_j^*}{\pi_j^*}\|\Pi_{j\cdot}\|_2^2 \le \sup_{1\le j\le n}\frac{p}{x_j^\top\Sigma_*^{-1}x_j}\cdot x_j^\top\Sigma_*^{-1}x_j = p.$$

Applying Lemma 18 we have that

$$\Pr\left[\|\Pi W^{(1)}\Pi - \Pi\|_2 > \epsilon\right] \le 2\exp\left\{-C\cdot\frac{k\epsilon^2}{p\log k}\right\}.$$

■

With Lemma 16, we know that $\|\Pi W\Pi - \Pi\|_2 \le \epsilon$ holds with probability at least 0.9 if $\frac{p\log k}{k} = O(\epsilon^2)$. Eq. (6) then holds with high probability by Lemma 15.

Without replacement setting Define independently distributed random matrices $A_1, \cdots, A_n$ as

$$A_j = (w_j - \pi_j^*)x_j x_j^\top, \quad j = 1, \cdots, n.$$

Note that w_j is a random Bernoulli variable with $\Pr[w_j = 1] = kp_j^{(2)} = \pi_j^*$. Therefore, $\mathbb{E}A_j = 0$. In addition,

$$\sup_{1\le j\le n}\|A_j\|_2 \le \sup_{1\le j\le n}\|x_j\|_2^2 \le \|X\|_\infty^2 \quad a.s.$$

and

$$\left\|\sum_{j=1}^n \mathbb{E}A_j^2\right\|_2 = \left\|\sum_{j=1}^n \pi_j^*(1-\pi_j^*)\|x_j\|_2^2 x_j x_j^\top\right\|_2 \le \|X\|_\infty^2\|\Sigma_*\|_2.$$

Noting that $\sum_{j=1}^n A_j = X_{\widehat{S}}^\top X_{\widehat{S}} - X^\top\text{diag}(\pi^*)X = \widehat{\Sigma}_{\widehat{S}} - \Sigma_*$ and invoking Lemma 19 with $t = \epsilon\lambda_{\min}(\Sigma_*)$ we have that

$$\Pr\left[\|\widehat{\Sigma}_{\widehat{S}} - \Sigma_*\|_2 > \epsilon\lambda_{\min}(\Sigma_*)\right] \le 2p\exp\left\{-\frac{\epsilon^2\lambda_{\min}(\Sigma_*)^2}{3\|\Sigma_*\|_2\|X\|_\infty^2 + 2\|X\|_\infty^2\cdot\epsilon\lambda_{\min}(\Sigma_*)}\right\}.$$

3. For those j with $\pi_j^* = 0$, we have by definition that $p_j^{(2)} = 0$.

Equating the right-hand side with $O(1)$ we have that

$$\|\Sigma_*^{-1}\|_2\kappa(\Sigma_*)\|X\|_\infty^2 \log p = O(\epsilon^2).$$

Finally, by Weyl's theorem we have that

$$\left|\lambda_\ell(\widehat{\Sigma}_{\widehat{S}}) - \lambda_\ell(\Sigma_*)\right| \le \|\widehat{\Sigma}_{\widehat{S}} - \Sigma_*\|_2 \le \epsilon\lambda_{\min}(\Sigma_*) \le \epsilon\lambda_\ell(\Sigma_*),$$

and hence the proof of Lemma 6.

7.4 Proof of Lemma 8

With replacement setting Define $\widetilde{w}_j^* = k/T \cdot w_j^* = \pi_j^*/(Tp_j^{(1)})$ and let $\widehat{\Sigma}_T = \sum_{t=1}^T \widetilde{w}_{i_t}^* x_{i_t} x_{i_t}^\top$. Because $\widehat{\Sigma}_T = \frac{k}{T}\widehat{\Sigma}_{\widehat{\Sigma}_W} \preceq \widehat{\Sigma}_{\widehat{S}}$, we have that $\widehat{\Sigma}_{\widehat{S}} \succeq \frac{T}{k}\widehat{\Sigma}_T$ and hence $z^\top\widehat{\Sigma}_{\widehat{S}}z \ge \frac{T}{k}z^\top\widehat{\Sigma}_T z$ for all $z \in \mathbb{R}^p$. Therefore, to lower bound the spectrum of $\widehat{\Sigma}_{\widehat{S}}$ it suffices to lower bound the spectrum of $\widehat{\Sigma}_T$.

Define diagonal matrix $W^{(2)}$ as

$$W_{jj}^{(2)} = \frac{\sum_{t=1}^T \widetilde{w}_{i_t}^* \mathbb{I}[i_t = j]}{\pi_j^*}, \qquad j = 1, \cdots, n.$$

We have that $\widetilde{\Sigma}_* = \widehat{\Sigma}_T$ for this particular choice of W. Following the same analysis in the proof of Lemma 16, we have that for every $t > 0$

$$\Pr\left[\|\Pi W^{(2)}\Pi - \Pi\|_2 > t\right] \le 2\exp\left\{-C \cdot \frac{T\epsilon^2}{p\log T}\right\}.$$

Set $t = O(1)$ and equate the right-hand side of the above inequality with $O(1)$. We then have

$$\Pr\left[\widehat{\Sigma}_T \succeq \Omega(1) \cdot \widehat{\Sigma}_*\right] = \Omega(1) \qquad \text{if } p\log T/T = O(1).$$

Subsequently, under the condition that $p\log T/T = O(1)$, with probability at least 0.9 it holds that

$$\widehat{\Sigma}_{\widehat{S}} \succeq \Omega(T/k) \cdot \Sigma_*,$$

which completes the proof of Lemma 8 for the without replacement setting.

Without replacement setting Define $\widehat{\Sigma}_{\widehat{R}} = X_{R_T}^\top X_{R_T} = \sum_{t=1}^T \pi_{i_t} x_{i_t} x_{i_t}^\top$. Conditioned on R_T, the subset $\widehat{S} = S_T$ is selected using the same procedure of the soft-constraint algorithm Figure 1 on $X_{R_T} = \{x_i\}_{i\in R_T}$. Subsequently, following analysis in the proof of Lemma 6 we have

$$\Pr\left[\|\widehat{\Sigma}_{\widehat{S}} - \widehat{\Sigma}_T\|_2 > t \middle| R_T\right] \le 2p \cdot \exp\left\{-\frac{t^2}{3\|\widehat{\Sigma}_T\|_2\|X\|_\infty^2 + 2\|X\|_\infty^2 t}\right\}.$$

Setting $t = O(1) \cdot \lambda_{\min}(\Sigma_T)$ we have that, if $\|\widehat{\Sigma}_T^{-1}\|_2\kappa(\widehat{\Sigma}_T)\|X\|_\infty^2 \log p = O(1)$, then with probability at least 0.95 conditioned on $\widehat{\Sigma}_T$

$$\Omega(1) \cdot \widehat{\Sigma}_T \preceq \widehat{\Sigma}_{\widehat{S}} \preceq O(1) \cdot \widehat{\Sigma}_T. \tag{13}$$

It remains to establish spectral similarity between $\widehat{\Sigma}_T$ and $\frac{T}{n}\Sigma_*$, a scaled version of Σ_*. Define deterministic matrices $A_1, \cdots, A_n$ as

$$A_j = \pi_j^* x_j x_j^\top - \frac{1}{n}\Sigma_*, \qquad j = 1, \cdots, n.$$

By definition, $\sum_{j=1}^n A_j = 0$ and $\sum_{t=1}^T A_{\sigma(t)} = \widehat{\Sigma}_T - \frac{T}{n}\Sigma_*$, where σ is a random permutation from $[n]$ to $[n]$. In addition,

$$\sup_{1\le j\le n} \|A_j\|_2 \le \frac{1}{n}\|\Sigma_*\|_2 + \sup_{1\le j\le n}\|x_j\|_2^2 \le 2\|X\|_\infty^2$$

and

$$\begin{aligned}
\frac{T}{n}\left\|\sum_{j=1}^n A_j^2\right\|_2 &\le \frac{2T}{n}\left(\left\|\sum_{j=1}^n (\pi_j^*)^2\|x_i\|_2^2 x_i x_i^\top\right\|_2 + \frac{1}{n^2}\|\Sigma_*\|_2^2\right) \\
&\le \frac{2T}{n}\left(\|X\|_\infty^2\left\|\sum_{j=1}^n \pi_j^* x_i x_i^\top\right\|_2 + \frac{1}{n^2}\|\Sigma_*\|_2^2\right) \\
&\le \frac{2T}{n}\left(\|X\|_\infty^2\|\Sigma_*\|_2 + \frac{1}{n^2}\|\Sigma_*\|_2^2\right) \\
&\le \frac{4T}{n}\|X\|_\infty^2\|\Sigma_*\|_2.
\end{aligned}$$

Invoking Lemma 20, we have that

$$\Pr\left[\left\|\widehat{\Sigma}_T - \frac{T}{n}\Sigma_*\right\|_2 > t\right] \le p\exp\left\{-t^2\left[\frac{48T}{n}\|X\|_\infty^2\|\Sigma_*\|_2 + 8\sqrt{2}\|X\|_\infty^2 t\right]^{-1}\right\}.$$

Set $t = O(T/n) \cdot \lambda_{\min}(\Sigma_*)$. We then have that, if $\|\Sigma_*^{-1}\kappa(\Sigma_*)\|X\|_\infty^2 \log p = O(T/n)$ holds, then with probability at least 0.95

$$\Omega(T/n) \cdot \Sigma_* \preceq \widehat{\Sigma}_T \preceq O(T/n) \cdot \Sigma_*. \tag{14}$$

Combining Eqs. (13,14) and noting that $\|\widehat{\Sigma}_T^{-1}\|_2 \le O(\frac{n}{T})\|\Sigma_*^{-1}\|_2$, $\kappa(\Sigma_T) \le O(1)\kappa(\Sigma_*)$, we complete the proof of Lemma 8 under the without replacement setting.

7.5 Proof of Lemma 12

Let $f(\pi; \lambda, \widetilde{\lambda}, \mu)$ be the Lagrangian muliplier function of the without replacement formulation of Eq. (4):

$$f(\pi;\lambda,\widetilde{\lambda},\mu) = f(\pi;X) - \sum_{i=1}^n \lambda_i\pi_i + \sum_{i=1}^n \widetilde{\lambda}_i\left(\pi_i - \frac{1}{k}\right) + \mu\left(\sum_{i=1}^n \pi_i - 1\right).$$

Here $\{\lambda_i\}_{i=1}^n \ge 0$, $\{\widetilde{\lambda}_i\}_{i=1}^n \ge 0$ and $\mu \ge 0$ are Lagrangian multipliers for constraints $\pi_i \ge 0$, $\pi_i \le 1$ and $\sum_i \pi_i \le k$, respectively. By KKT condition, $\frac{\partial f}{\partial \pi_i}\big|_{\pi^*} = 0$ and hence

$$-\left.\frac{\partial f}{\partial \pi_i}\right|_{\pi^*} = x_i^\top\Sigma_*^{-2}x_i = \widetilde{\lambda}_i - \lambda_i + \mu, \qquad i = 1, \cdots, n,$$

where $\Sigma_* = X^\top \mathrm{diag}(\pi^*) X$ is a $p \times p$ positive definite matrix.

Split the index set $[n]$ into three disjoint sets defined as $A = \{i \in [n] : \pi_i^* = 1\}$, $B = \{i \in [n] : 0 < \pi_i^* < 1\}$ and $C = \{i \in [n] : \pi_i^* = 0\}$. Note that $\|\pi^*\|_0 = |A| + |B|$ and $|A| \le k$. Therefore, to upper bound $\|\pi^*\|_0$ it suffices to upper bound $|B|$. By complementary slackness, for all $i \in B$ we have that $\widetilde{\lambda}_i = \lambda_i = 0$; that is,

$$x_i^\top \Sigma_*^{-2} x_i = \langle \phi(x_i), \psi(\Sigma_*^{-2}) \rangle = \mu, \qquad \forall i \in B, \tag{15}$$

where $\phi : \mathbb{R}^p \to \mathbb{R}^{p(p+1)/2}$ is the mapping defined in Assumption 3.1 and $\psi(\cdot)$ takes the upper triangle of a symmetric matrix and vectorizes it into a $\frac{p(p-1)}{2}$-dimensional vector. Assume by way of contradiction that $|B| > p(p+1)/2$ and let $x_1, \cdots, x_{p(p+1)/2+1}$ be arbitrary distinct $\frac{p(p+1)}{2} + 1$ rows whose indices belong to B. Eq. (15) can then be cast as a homogenous linear system with $\frac{p(p+1)}{2} + 1$ variables and equations as follows:

$$\begin{bmatrix} \widetilde{\phi}(x_1) \\ \widetilde{\phi}(x_2) \\ \vdots \\ \widetilde{\phi}(x_{p(p+1)/2+1}) \end{bmatrix} \begin{bmatrix} \psi(\Sigma_*^{-2}) \\ -\mu \end{bmatrix} = 0.$$

Under Assumption 3.1, $\widetilde{\Phi} = [\widetilde{\phi}(x_1); \cdots; \widetilde{\phi}(x_{p(p+1)/2+1})]^\top$ is invertible and hence both $\psi(\Sigma_*^{-2})$ and μ must be zero. This contradicts the fact that Σ_*^{-2} is positive definite.

Appendix A. Technical lemmas

Lemma 17 *Let $\mathbb{B} = \{x \in \mathbb{R}^p : \|x\|_2^2 \le B^2\}$ and $\mathrm{vol}(\mathbb{B}) = \int_{\mathbb{B}} 1 \mathrm{d}x$ be the volume of $\mathbb{B}$. Then $\int_{\mathbb{B}} x x^\top \mathrm{d}x = \frac{B^2}{p+2} \mathrm{vol}(\mathbb{B}) I_{p\times p}$.*

Proof Let U be the uniform distribution in the p-dimensional ball of radius B. By definition, $\int_{\mathbb{B}} x x^\top \mathrm{d}x = \mathrm{vol}(\mathbb{B}) \mathbb{E}_{x\sim U}[x x^\top]$. By symmetry, $\mathbb{E}_{x\sim U}[x x^\top] = c \cdot I_{p\times p}$ for some constant c that depends on p and B. To determine the constant c, note that

$$c = \mathbb{E}[x_1^2] = \frac{1}{p}\mathbb{E}\|x\|_2^2 = \frac{1}{p} \frac{\int_0^B r^{p-1} \cdot r^2 \mathrm{d}r}{\int_0^B r^{p-1} \mathrm{d}r} = \frac{B^2}{p+2}.$$

■

Lemma 18 (Rudelson and Vershynin (2007)) *Let x be a p-dimensional random vector such that $\|x\|_2 \le M$ almost surely and $\|\mathbb{E}x x^\top\|_2 \le 1$. Let $x_1, \cdots, x_n$ be i.i.d. copies of x. Then for every $t \in (0,1)$*

$$\Pr\left[\left\|\frac{1}{n}\sum_{i=1}^n x_i x_i^\top - \mathbb{E}x x^\top\right\|_2 > t\right] \le 2\exp\left\{-C \cdot \frac{nt^2}{M^2 \log n}\right\},$$

where $C > 0$ is some universal constant.

Lemma 19 (Corollary 5.2 of (Mackey et al., 2014), Matrix Bernstein) *Let $(Y_k)_{k\ge1}$ be a sequence of random d-dimensional Hermitian matrices that satisfy*

$$\mathbb{E}Y_k = 0 \quad \text{and} \quad \|Y_k\|_2 \le R \quad a.s.$$

Define $X = \sum_{k\ge1} Y_k$. The for any $t > 0$,

$$\Pr\left[\|X\|_2 \ge t\right] \le d \cdot \exp\left\{-\frac{t^2}{3\sigma^2 + 2Rt}\right\} \quad \text{for } \sigma^2 = \left\|\sum_{k\ge1} \mathbb{E}Y_k^2\right\|_2.$$

Lemma 20 (Corollary 10.3 of (Mackey et al., 2014)) *Let $A_1, \cdots, A_n$ be a sequence of deterministic d-dimensional Hermitian matrices that satisfy*

$$\sum_{k=1}^n A_k = 0 \quad \text{and} \quad \sup_{1\le k\le n} \|A_k\|_2 \le R.$$

Define random matrix $X = \sum_{j=1}^m A_{\sigma(j)}$ for $m \le n$, where σ is a random permutation from $[n]$ to $[n]$. Then for all $t > 0$,

$$\Pr\left[\|X\|_2 \ge t\right] \le d\exp\left\{-\frac{t^2}{12\sigma^2 + 4\sqrt{2}Rt}\right\} \quad \text{for } \sigma^2 = \frac{m}{n}\left\|\sum_{k=1}^n A_k^2\right\|_2.$$

Appendix B. Optimization methods

Two algorithms for optimizing Eq. (4) are described. The SDP formulation is of theoretical interest only and the projected gradient descent algorithm is practical, which also enjoys theoretical convergence guarantees.

SDP formulation For $\pi \in \mathbb{R}^n$ define $A(\pi) = \sum_{i=1}^n \pi_i x_i x_i^\top$, which is a $p \times p$ positive semidefinite matrix. By definition, $f(\pi; X) = \sum_{j=1}^p e_j^\top A(\pi)^{-1} e_j$, where e_j is the p-dimensional vector with only pth coordinate being 1. Subsequently, Eq. (4) is equivalent to the following SDP problem:

$$\min_{\pi,t} \sum_{j=1}^p t_j \quad \text{subject to} \quad 0 \le \pi_i \le 1, \ \sum_{i=1}^n \pi_i \le k, \ \ \mathrm{diag}(B_1, \cdots, B_p) \succeq 0,$$

where

$$B_j = \begin{bmatrix} A(\pi) & e_j \\ e_j^\top & t_j \end{bmatrix}, \qquad j = 1, \cdots, p.$$

Global optimal solution of an SDP can be computed in polynomial time (Vandenberghe and Boyd, 1996). However, this formulation is not intended for practical computation because of the large number of variables in the SDP system. First-order methods such as projected gradient descent is a more appropriate choice for practical computation.

Projected gradient descent For any convex set S and point x let $\mathcal{P}_S(x) = \text{argmin}_{y\in S}\|x-y\|_2$ denote the ℓ_2 projection of x onto S. The projected gradient descent algorithm is a general purpose method to solve convex constrained smooth convex optimization problems of the form

$$\min_x f(x) \quad \text{subject to} \quad x \in S.$$

The algorithm (with step size selected via backtracking line search) iterates until desired optimization accuracy is reached:

input : backtracking parameters $\alpha \in (0, 1/2]$, $\beta \in (0, 1)$.
output: $\widehat{x}$, approximate solution of the optimization problem.
Initialization: x_0, $t = 0$.
1. Compute gradient $g_t = \nabla f(x_t)$.
2. Find the smallest integer $s \geq 0$ such that $f(x') - f(x_t) \leq \alpha g_t^\top (x' - x_t)$, where $x' = \mathcal{P}_S(x_t - \beta^s g_t)$.
3. Set $x_{t+1} = \mathcal{P}_S(x_t - \beta^s g_t)$, $t \leftarrow t + 1$ and repeat steps 1 and 2, until the desired accuracy is achieved. Output $\widehat{x} = x_t$.

Figure 4: The projected gradient descent algorithm.

The gradient $\nabla_\pi f(\pi; X)$ in Eq. (4) is easy to compute:

$$\frac{\partial f(\pi; X)}{\partial \pi_i} = -x_i^\top (X^\top \text{diag}(\pi) X)^{-1} x_i, \quad i = 1, \cdots, n.$$

Because $X^\top \text{diag}(\pi) X$ is a shared term, computing $\nabla_\pi f(\pi; X)$ takes $O(np^2 + p^3)$ operations. The projection step onto the intersection of ℓ_1 and ℓ_∞ balls is complicated and non-trivial, which we describe in details in Appendix D. In general, the projection step can be done in $O(n \log \|\pi\|_\infty)$ time, where π is the point to be projected.

The following proposition establishes convergence guarantee for the projected gradient descent algorithm. Its proof is given in the appendix.

Proposition 21 *Let $\pi^{(0)}$ be the "flat" initialization (i.e., $\pi_i^{(0)} = k/n$) and $\pi^{(t)}$ be the solution after t projected gradient iterations. Then*

$$f(\pi^{(t)}; X) - f(\pi^*; X) \leq \frac{\|X\|_2^4 [\text{tr}(\widehat{\Sigma}^{-1})]^3 \|\pi^{(0)} - \pi^*\|_2^2}{\beta t},$$

where $\widehat{\Sigma} = \frac{1}{n} X^\top X$.

We also remark that the provided convergence speed is very conservative, especially in the cases when k is large where practical evidence suggests that the algorithm converges in very few iterations (cf. Sec. 4).

Appendix C. Fedorov's exchange algorithm

The algorithm starts with a initial subset $S \subseteq [n]$, $|S| \leq k$, usually initialized with random indices. A "best" pair of exchanging indices are computed as

$$i^*, j^* = \text{argmin}_{i\in S, j\notin S} \ell(S\backslash\{i\} \cup \{j\}),$$

where $\ell(S)$ is the objective function to be minimized. In our case it would be the A-optimality objective $\ell(S) = F(S; X) = \text{tr}((X_S^\top X_S)^{-1})$. The algorithm then "exchanges" i^* and j^* by setting $S' \leftarrow S\backslash\{i^*\} \cup \{j^*\}$ and continues such exchanges until no exchange can lower the objective value. Under without replacement settings, special care needs to be taken to ensure that S consists of distinct indices.

Computing the objective function $F(S; X) = \text{tr}((X_S^\top X_S)^{-1})$ requires inverting a $p \times p$ matrix, which could be computationally slow. A more computationally efficient approach is to perform rank-1 update of the inverse of $X_S^\top X_S$ after each iteration, via the Sherman-Morrison formula:

$$(A + uv^\top)^{-1} = A^{-1} - \frac{A^{-1} u v^\top A^{-1}}{1 + u^\top A^{-1} v}.$$

Each exchange would then take $O(nkp^2)$ operations. The total number of exchanges, however, is unbounded and could be as large as $\binom{n}{k}$ in theory. In practice we do observe that a large number of exchanges are required in order to find a local optimal solution.

Appendix D. Projection onto the intersection of ℓ_1 and ℓ_∞ norm balls

Similar to (Yu et al., 2012), we only need to consider the case that π lies in the first quadrant, i.e., $\pi_i \geq 0, i = 1, 2, ..., n$. Then the projection x also lies in the first quadrant. Furthermore, we assume the point to be project lies in the area which is out of both norm balls and projection purely onto either ℓ_1 or ℓ_∞ ball is not the intersection of both. Otherwise, it is trivial to conduct the projection. The projection problem is formulated as follows:

$$\begin{aligned} \min_x \quad & \frac{1}{2}\|\pi - x\|_2^2 \\ \text{s.t.} \quad & \|x\|_1 \leq c_1, \quad \|x\|_\infty \leq c_2. \end{aligned}$$

By introducing an auxiliary variable d, the problem above has the following equivalent form:

$$\begin{aligned} \min_{x,d} \quad & \frac{1}{2}\|\pi - x\|_2^2 \\ \text{s.t.} \quad & \|x\|_1 \leq c_1, \\ & x_i \leq d, \ i = 1, 2, ...n, \\ & d \leq c_2. \end{aligned} \tag{16}$$

The Lagrangian of problem (16) is

$$\mathcal{L}(x, d, \lambda_1, \lambda_2) = \frac{1}{2}||\pi - x||_2^2 + \sum_i \omega_i (x_i - d) + \lambda_1 (\|x\|_1 - c_1) + \lambda_2 (d - c_2) \tag{17}$$

input : π, c_1, c_2 and precision parameter δ.
output: x, the projection of π onto $\{x : \|x\|_1 \le c_1\} \cap \{x : \|x\|_\infty \le c_2\}$.
Initialization: $\lambda_1 = \|\pi\|_\infty$.
while $|h(\lambda_1)| > \delta$ **do**
  if $h(\lambda_1) > \delta$ **then**
    $l = \lambda_1$;
  else
    $r = \lambda_1$;
  end
  $\lambda_1 = (l + r)/2$;
  for each i **do** $x_i = \min(\max(\pi_i - \lambda_1, 0), c_2)$;
end

Algorithm 5: Projecting π onto $\{x : \|x\|_1 \le c_1\} \cap \{x : \|x\|_\infty \le c_2\}$.

Let x^*, d^* and λ_1^*, λ_2^* respectively be the primal and dual solution of (16), then its KKT condition is:

$$0 \in \partial \mathcal{L}_x(x^*, d^*, \lambda_1^*, \lambda_2^*), \tag{18}$$
$$0 \in \partial \mathcal{L}_d(x^*, d^*, \lambda_1^*, \lambda_2^*), \tag{19}$$
$$w_i^*(x_i^* - d^*) = 0, \quad i = 1, 2, ..., n \tag{20}$$
$$\lambda_2^*(d^* - c_2) = 0, \tag{21}$$
$$\lambda_1^*(\|x^*\|_1 - c_1) = 0, \tag{22}$$
$$x_i^* \le d^*, \quad i = 1, 2, ..., n \tag{23}$$
$$d^* \le c_2, \tag{24}$$
$$\|x^*\|_1 \le c_1, \tag{25}$$
$$x_i^*, d^*, \lambda_1^*, \lambda_2^*, w_i^* \ge 0. \tag{26}$$

The following lemmas illustrate the relation between the primal and dual solution of problem (16):

Lemma 22 *Either one of the following holds: 1) $d^* > 0$ and $\sum_i (\max(\pi_i - \lambda_1^*, 0) - x_i^*) = \lambda_2^*$; 2) $d^* = 0$ and $\sum_i \max(\pi_i - \lambda_1^*, 0) \le \lambda_2^*$.*

Lemma 23 *For $i = 1, 2, ..., n$, the optimal x_i^* satisfies*

$$x_i^* = \min(\max(\pi_i - \lambda_1^*, 0), d^*). \tag{27}$$

Proof Lemma 22 and Lemma 23 can be both viewed as the special cases of Lemma 6 and 7 in Yu et al. (2012). ■

Now the KKT conditions can be reduced to finding $\lambda_1^*, \lambda_2^*, d^*$ that satisfy the following equations:

$$\sum_i \min(\max(\pi_i - \lambda_1^*, 0), d^*) - c_1 = 0, \tag{28}$$
$$d^* - c_2 = 0, \tag{29}$$
$$\sum_i \max(\max(\pi_i - \lambda_1^*, 0) - d^*, 0) = \lambda_2^* \text{ iff } d^* > 0, \tag{30}$$
$$\sum_i \max(\pi_i - \lambda_1^*, 0) \le \lambda_2^* \text{ iff } d^* = 0, \tag{31}$$
$$d^* \ge 0, \lambda_2^* \ge 0, \lambda_1^* \ge 0. \tag{32}$$

Further simplification induces the following result:

Lemma 24 *Suppose d^* and λ_1^* are the primal and dual solution respectively, then*

$$d^* = c_2, \tag{33}$$
$$x_i^* = \min(\max(\pi_i - \lambda_1^*, 0), c_2). \tag{34}$$

Proof Direct result of (27) and (29). ■

(34) shows that the solution x^* is determined once the optimal λ_1^* is found. Given a $\lambda_1 \in [0, \max_i(\pi_i)]$, we only need to check whether (28) holds by looking into the function

$$h(\lambda_1) = \sum_i \min(\max(\pi_i - \lambda_1, 0), c_2) - c_1, \tag{35}$$

and λ_1^* is simply the zero point of $h(\cdot)$. The following theorem shows that $h(\cdot)$ is a strictly monotonically decreasing function, so a binary search is sufficient to find λ_1^*, and x_i^* can be determined accordingly.

Theorem 25 *1) $h(\lambda_1)$ is a continuous piecewise linear function in $[0, \max_i\{\pi_i\}]$; 2) $h(\lambda_1)$ is strictly monotonically decreasing and it has a unique root in $[0, \max_i\{\pi_i\}]$.*

Proof 1) is obviously true. It is easy to check that $h'(\lambda_1) < 0$ in each piece, $h(\max_i\{\pi_i\}) < 0$ and $h(0) > 0$, so 2) also holds. ■

Complexity Analysis: Algorithm 5 is proposed based on the observation above to solve the projection problem, which is essentially a bisection to search λ_1. Given a precision δ, the iteration complexity of bisection is $O(\log(\max_i(\pi_i)/\delta))$. Besides, the time complexity of evaluating $h(\cdot)$ is $O(n)$, so the total complexity of Algorithm 5 is $O(n \log(\max_i(\pi_i)/\delta))$.

Appendix E. Convergence analysis of PGD

We provide a convergence analysis of the projected gradient descent algorithm used in optimizing Eq. (4). The analysis shows that the PGD algorithm approximately computes the global optimum of Eq. (4) in polynomial time. In simulation studies, much fewer iterations are required for convergence than predicted by the theoretical results.

Because we're using exact projected gradient descent algorithms, the objective function shall decay monotonically and hence convergence of such algorithms can be established by showing Lipschitz continuity of ∇f on a specific *level set*; that is, for some Lipschitz constant $L > 0$ the following holds for all π, π' such that $f(\pi), f(\pi') \leq f(\pi^{(0)})$:

$$\|\nabla f(\pi) - \nabla f(\pi')\|_2 \leq L\|\pi - \pi'\|_2, \tag{36}$$

where $\pi^{(0)} = (1/n, 1/n, \cdots, 1/n)$ is the initialization point. Once Eq. (36) holds, linear convergence (i.e., $f(\pi^{(t)}) - f(\pi^*) = O(1/t)$) can be established via standard projected gradient analysis.

The main idea of establishing Eq. (36) is to upper bound the spectral norm of the Hessian matrix $H = \nabla^2 f(\pi)$ *uniformly* over all points π that satisfies $f(\pi) \leq f(\pi^{(0)})$. As a first step, we derive analytic forms of H in the following proposition:

Proposition 26 *Let $\widetilde{\Sigma} = X^\top \mathrm{diag}(\pi) X$. We then have that*

$$H = 2(X^\top \widetilde{\Sigma}^{-2} X) \circ (X^\top \widetilde{\Sigma}^{-1} X),$$

where $\circ$ denotes the element-wise Hadamard product between two matrices of same dimensions.

Proof We first derive the gradient of f. Fix arbitrary $i \in [n]$. The partial derivative can be computed as

$$\frac{\partial f}{\partial \pi_i} = \frac{\partial \mathrm{tr}(\widetilde{\Sigma}^{-1})}{\partial \pi_i} = \left\langle \frac{\partial \mathrm{tr}(\widetilde{\Sigma}^{-1})}{\partial \widetilde{\Sigma}}, \frac{\partial \widetilde{\Sigma}}{\partial \pi_i} \right\rangle = \left\langle -\widetilde{\Sigma}^{-2}, x_i x_i^\top \right\rangle = -x_i^\top \widetilde{\Sigma}^{-2} x_i.$$

Here $\langle A, B \rangle = \mathrm{tr}(B^\top A)$ is the element-wise multiplication inner product between two matrices. The second-order partial derivatives can then be computed as

$$\begin{aligned}\frac{\partial^2 f}{\partial \pi_i \partial \pi_j} = -\frac{\partial (x_i^\top \widetilde{\Sigma}^{-2} x_i)}{\partial \pi_j} &= -x_i^\top \frac{\partial \widetilde{\Sigma}^{-2}}{\partial \pi_j} x_i = x_i^\top \widetilde{\Sigma}^{-2} \frac{\partial \widetilde{\Sigma}^2}{\partial \pi_j} \widetilde{\Sigma}^{-2} x_i \\ &= x_i^\top \widetilde{\Sigma}^{-2} (x_j x_j^\top \widetilde{\Sigma} + \widetilde{\Sigma} x_j x_j^\top) \widetilde{\Sigma}^{-2} x_i = 2(x_i^\top \widetilde{\Sigma}^{-2} x_j) \cdot (x_i^\top \widetilde{\Sigma}^{-1} x_j).\end{aligned}$$

Subsequently,

$$H = \nabla^2 f = \left[\frac{\partial^2 f}{\partial \pi_i \partial \pi_j} \right]_{i,j=1}^n = 2(X^\top \widetilde{\Sigma}^{-2} X) \circ (X^\top \widetilde{\Sigma}^{-1} X).$$

■

Corollary 27 *Suppose π satisfies $f(\pi) \leq f(\pi^{(0)}) = \mathrm{tr}(\Sigma_0^{-1})$, where $\Sigma_0 = \frac{1}{n} X^\top X$. We then have*

$$\|H\|_2 = \|\nabla^2 f(\pi)\|_2 \leq 2\|X\|_2^4 \cdot [\mathrm{tr}(\Sigma_0^{-1})]^3.$$

Proof Let $\rho(A) = \max_i |\sigma_i(A)|$ denote the *spectral range* of matrix A. Clearly, $\|A\|_2 \leq \rho(A)$ and $\|A\|_2 = \rho(A)$ for positive semi-definite matrices. In (Hom and Johnson, 1991) it is established that $\rho(A \circ B) \leq \rho(A)\rho(B)$. Subsequently,

$$\|H\|_2 \leq 2\|X\|_2^4 \|\widetilde{\Sigma}^{-1}\|_2^3 \leq 2\|X\|_2^4 \cdot [\mathrm{tr}(\widetilde{\Sigma}^{-1})]^3 \leq 2\|X\|_2^4 \cdot [\mathrm{tr}(\Sigma_0^{-1})]^3,$$

where the last inequality is due to the condition that $\mathrm{tr}(\widetilde{\Sigma}^{-1}) = f(\pi) \leq f(\pi^{(0)}) = \mathrm{tr}(\Sigma_0^{-1})$. ■

From Corollary 27, we can prove the convergence of the optimization procedures outlined in Appendix. B following standard analysis of projected gradient descent on objective functions with Lipschitz continuous gradient:

Theorem 28 *Suppose $\pi^{(t)}$ is the solution at the tth iteration of the projected gradient descent algorithm and π^* is the optimal solution. We then have*

$$f(\pi^{(t)}) - f(\pi^*) \leq \frac{\|X\|_2^4 \cdot [\mathrm{tr}(\Sigma_0^{-1})]^3 \cdot \|\pi^{(0)} - \pi^*\|_2^2}{\beta t},$$

where $\beta \in (0, 1)$ is the backtracking parameter in backtracking line search.

Acknowledgments

We thank Reeja Jayan and Chiqun Zhang for sharing with us data on the material synthesis experiments, and Siheng Chen for a pre-processed version of the Minnesota wind speed data set. This work is supported by NSF CCF-1563918, NSF CAREER IIS-1252412 and AFRL FA87501720212.

References

Alexander A Ageev and Maxim I Sviridenko. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*, 8(3):307–328, 2004.

David G Anderson, Ming Gu, and Christopher Melgaard. An efficient algorithm for unweighted spectral graph sparsification. *arXiv preprint arXiv:1410.4273*, 2014.

Haim Avron and Christos Boutsidis. Faster subset selection for matrices and applications. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1464–1499, 2013.

Joshua Batson, Daniel A Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012.

Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

Kamalika Chaudhuri, Sham Kakade, Praneeth Netrapalli, and Sujay Sanghavi. Convergence rates of active learning for maximum likelihood estimation. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2015.

Siheng Chen, Rohan Varma, Aarti Singh, and Jelena Kovačević. Signal representations on graphs: Tools and applications. *arXiv:1512.05406*, 2015.

Herman Chernoff. Locally optimal designs for estimating parameters. *The Annals of Statistics*, pages 586–602, 1953.

Mark A Davenport, Andrew K Massimino, Deanna Needell, and Tina Woolf. Constrained adaptive sensing. *IEEE Transactions on Signal Processing*, 64(20):5437–5449, 2015.

Paramveer Dhillon, Yichao Lu, Dean Foster, and Lyle Ungar. New sampling algorithms for fast least squares regression. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2013.

Edgar Dobriban and Jianqing Fan. Regularity properties for sparse regression. *Communications in Mathematics and Statistics*, 4(1):1–19, 2016.

Petros Drineas, Michael W Mahoney, and S Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844–881, 2008.

Petros Drineas, Michael W Mahoney, S Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, 2011.

Phillip Ein-Dor and Jacob Feldmesser. Attributes of the performance of central processing units: a relative performance prediction model. *Communications of the ACM*, 30(4): 308–317, 1987.

Elad Hazan and Zohar Karnin. Hard margin active linear regression. In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

RA Hom and CR Johnson. *Topics in Matrix Analysis*. Cambridge UP, New York, 1991.

Thibaut Horel, Stratis Ioannidis, and S Muthukrishnan. Budget feasible mechanisms for experimental design. In *Latin American Symposium on Theoretical Informatics (LATIN)*, pages 719–730. Springer, 2014.

Andre Khuri, Bhramar Mukherjee, Bikas Sinha, and Malay Ghosh. Design issues for generalized linear models: a review. *Statistical Science*, 21(3):376–399, 2006.

Ping Ma, Michael W Mahoney, and Bin Yu. A statistical perspective on algorithmic leveraging. *Journal of Machine Learning Research*, 16(1):861–911, 2015.

Lester Mackey, Michael I Jordan, Richard Y Chen, Brendan Farrell, and Joel A Tropp. Matrix concentration inequalities via the method of exchangeable pairs. *The Annals of Probability*, 42(3):906–945, 2014.

Adam Marcus, Daniel Spielman, and Nikhil Srivastava. Interlacing families I: bipartite ramanujan graphs of all degrees. *Annals of Mathematics*, 182:307–325, 2015a.

Adam Marcus, Daniel Spielman, and Nikhil Srivastava. Interlacing families II: Mixed characteristic polynomials and the kadison-singer problem. *Annals of Mathematics*, 182: 327–350, 2015b.

Alan Miller and Nam-Ky Nguyen. A fedorov exchange algorithm for d-optimal design. *Journal of the Royal Statistical Society, Series C (Applied Stiatistics)*, 43(4):669–677, 1994.

Nathan Nakamura, Jason Seepaul, Joseph B Kadane, and B Reeja-Jayan. Design for low-temperature microwave-assisted crystallization of ceramic thin films. *Applied Stochastic Models in Business and Industry*, 2017.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

Gabor Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23(2):339–358, 1998.

Friedrich Pukelsheim. *Optimal design of experiments*, volume 50. SIAM, 1993.

Garvesh Raskutti and Michael Mahoney. Statistical and algorithmic perspectives on randomized sketching for ordinary least-squares. In *Proceedings of International Conference on Machine Learning (ICML)*, 2015.

B Reeja-Jayan, Katharine L Harrison, K Yang, Chih-Liang Wang, AE Yilmaz, and Arumugam Manthiram. Microwave-assisted low-temperature growth of thin films in solution. *Scientific reports*, 2(1003):1–8, 2012.

Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, 54(4):21, 2007.

Sivan Sabato and Remi Munos. Active regression by stratification. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2014.

Daniel A Spielman and Nikhil Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011.

Ryan Tibshirani. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7: 1456–1490, 2013.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Review*, 38(1): 49–95, 1996.

David Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Adams Wei Yu, Hao Su, and Fei-Fei Li. Efficient euclidean projections onto the intersection of norm balls. In *Proceedings of International Conference on Machine Learning (ICML)*, 2012.

Rong Zhu, Ping Ma, Michael Mahoney, and Bin Yu. Optimal subsampling approaches for large sample linear regression. *arXiv:1509.05111*, 2015.

Journal of Machine Learning Research 18 (2017) 1-46 Submitted 8/14; Revised 10/17; Published 11/17

Generalized Conditional Gradient for Sparse Estimation

Yaoliang Yu YAOLIANG.YU@UWATERLOO.CA
School of Computer Science
University of Waterloo
Waterloo, ON, N2L 3G1, Canada

Xinhua Zhang ZHANGX@UIC.EDU
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607

Dale Schuurmans DAES@UALBERTA.CA
Department of Computing Science
University of Alberta
Edmonton, Alberta T6G 2E8, Canada

Editor: Koby Crammer

Abstract

Sparsity is an important modeling tool that expands the applicability of convex formulations for data analysis, however it also creates significant challenges for efficient algorithm design. In this paper we investigate the generalized conditional gradient (GCG) algorithm for solving sparse optimization problems—demonstrating that, with some enhancements, it can provide a more efficient alternative to current state of the art approaches. After studying the convergence properties of GCG for general convex composite problems, we develop efficient methods for evaluating polar operators, a subroutine that is required in each GCG iteration. In particular, we show how the polar operator can be efficiently evaluated in learning low-rank matrices, instantiated with detailed examples on matrix completion and dictionary learning. A further improvement is achieved by interleaving GCG with fixed-rank local subspace optimization. A series of experiments on matrix completion, multi-class classification, and multi-view dictionary learning shows that the proposed method can significantly reduce the training cost of current alternatives.

Keywords: generalized conditional gradient, frank-wolfe, dictionary learning, matrix completion, multi-view learning, sparse estimation

1. Introduction

Sparsity is an important concept in high-dimensional statistics (Bühlmann and van de Geer, 2011) and signal processing (Eldar and Kutyniok, 2012), which has led to important application successes by reducing model complexity and improving interpretability of the results. Although it is common to promote sparsity by adding appropriate regularizers, such as the ℓ_1 norm, sparsity can also present itself in more sophisticated forms, such as group sparsity (Yuan and Lin, 2006), graph sparsity (Kim and Xing, 2009; Jacob et al., 2009), matrix rank sparsity (Candès and Recht, 2009), and other combinatorial sparsity patterns (Obozinski and Bach, 2012). These recent notions of *structured* sparsity (Bach et al., 2012; Micchelli et al., 2013) have greatly enhanced the ability to model structural relationships in data; however, they also create significant computational challenges.

A currently popular optimization scheme for training sparse models is *accelerated proximal gradient* (APG) (Beck and Teboulle, 2009; Nesterov, 2013), which enjoys an optimal rate of convergence among black-box first-order procedures (Nesterov, 2013). An advantage of APG is that each iteration consists solely of computing a *proximal update* (PU), which for simple regularizers can have a very low complexity. For example, under ℓ_1 norm regularization each iterate of APG reduces to a soft-shrinkage operator that can be computed in linear time, which partly explains its popularity for such problems. Unfortunately, for matrix low-rank regularizers the PU soon becomes a computational bottleneck. For example, the trace norm is often used to promote low rank solutions in matrix variable problems such as matrix completion (Candès and Recht, 2009); but here the associated PU requires a *full* singular value decomposition (SVD) of the gradient matrix on each iteration, which prevents APG from being applied to large problems. Not only does the PU require nontrivial computation for matrix regularizers, it also yields *dense* intermediate iterates.

To address the primary shortcomings with APG, we investigate the *generalized conditional gradient* (GCG) strategy for sparse optimization, which has been motivated by the promise of recent sparse approximation methods (Hazan, 2008; Clarkson, 2010). A special case of GCG, known as the conditional gradient (CG), was originally proposed by Frank and Wolfe (1956) and has received significant renewed interest (Bach, 2015; Clarkson, 2010; Freund and Grigas, 2016; Hazan, 2008; Jaggi and Sulovsky, 2010; Jaggi, 2013; Shalev-Shwartz et al., 2010; Tewari et al., 2011; Yuan and Yan, 2013). The key advantage of GCG for sparse estimation is that it need only compute the *polar* of the regularizer in each iteration, which sometimes admits a much more efficient update than the PU in APG. For example, under trace norm regularization, GCG only requires the *spectral norm* of the gradient to be computed in each iteration, which is an order of magnitude cheaper than evaluating the full SVD as required by APG. Furthermore, the greedy nature of GCG affords explicit control over the sparsity of intermediate iterates, which is not available in APG. Although existing work on GCG has generally been restricted to *constrained* optimization (i.e. enforcing an upper bound on the sparsity-inducing regularizer), in this paper we consider the more general *regularized* version of the problem. Despite their theoretical equivalence, the regularized form allows more efficient local optimization to be interleaved with the primary update, which provides a significant acceleration in practice.

This paper is divided into two major parts: first we present a general treatment of GCG and establish its convergence properties for convex optimization in Section 3, with a special focus on convex gauge regularizers; then we apply the method to important case studies to demonstrate its effectiveness on large-scale, sparse estimation problems in Section 4, with experiments presented in Section 5.

In particular, in the first part, after establishing notation and briefly discussing relevant optimization schemes (Section 2), we present our first main contribution (Section 3): the convergence properties of GCG for general convex composite optimization. The algorithm is then specialized to convex gauge regularizers with refined analysis (Section 3.2). Indeed we prove that, under standard assumptions, GCG converges to the global optimum at the rate of $O(1/t)$—which is a significant improvement over previous bounds (Dudik et al., 2012). Although the convergence rate for GCG remains inferior to that of APG, these two

©2017 Yaoliang Yu, Xinhua Zhang, and Dale Schuurmans.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/14-348.html.

algorithms demonstrate alternative trade-offs between the number of iterations required and the per-step complexity. As shown in our experiments, GCG can be significantly faster than APG in terms of *overall* computation time for large problems. The results in Section 3 are new.

Equipped with the technical results of GCG on convex optimization, the second part of the paper then applies the algorithm to the important application of low rank learning (Section 4). Since imposing a hard bound on matrix rank generally leads to an intractable problem, in Section 4.2 we first present a generic relaxation strategy based on convex gauges (Chandrasekaran et al., 2012; Tewari et al., 2011) that yields a convex program (Bach et al., 2008; Bradley and Bagnell, 2009; Zhang et al., 2012). Conveniently, the resulting problem can be easily optimized using the GCG algorithm from Section 3.2. To further reduce computation time, we introduce in Section 4.3 an auxiliary fixed rank subspace optimization within each iteration of GCG. Although similar hybrid approaches have been previously suggested (Burer and Monteiro, 2005; Laue, 2012; Mishra et al., 2013), we propose an efficient new alternative that, instead of locally optimizing a *constrained* fixed rank problem, optimizes an *unconstrained* surrogate objective based on a variational representation of matrix norms. This alternative strategy allows a far more efficient local optimization without compromising GCG's convergence properties. In Section 4.4 we show that the approach can be applied to dictionary learning and matrix completion problems, as well as a non-trivial multi-view form of dictionary learning (White et al., 2012).

Finally, in Section 5 we provide an extensive experimental evaluation that compares the performance of GCG (augmented with interleaved local search) to state-of-the-art optimization strategies, across a range of problems, including matrix completion, multi-class classification, and multi-view dictionary learning.

1.1 Extensions over Previously Published Work

Some of the results in this article have appeared in a preliminary form in two conference papers (Zhang et al., 2012; White et al., 2012); however, several specific extensions have been added in this paper, including:

- A more extensive and unified treatment of GCG in general Banach spaces.
- A more refined analysis of the convergence properties of GCG.
- A new application of GCG to multi-view dictionary learning.
- New experimental evaluations on larger data sets, including new results for matrix completion and image multi-class classification.
- Open source Matlab code released at `https://www.cs.uic.edu/~zhangx/GCG`.

2. Preliminaries

We first establish some of the definitions and notation that will be used throughout the paper, before providing a brief background on the relevant optimization methods that establish a context for discussing GCG.

2.1 Notation and Definitions

Throughout this paper we assume that the underlying space, unless otherwise stated, is a real Banach space $\mathcal{B}$ with norm $\|\cdot\|$; for example, $\mathcal{B}$ could be the Euclidean space $\mathbb{R}^d$ with any norm. The dual space (the set of all real-valued continuous linear functions on $\mathcal{B}$) is denoted $\mathcal{B}^*$ and is equipped with the dual norm $\|\mathbf{g}\|^\circ = \sup\{\mathbf{g}(\mathbf{w}) : \|\mathbf{w}\| \leq 1\}$. Note that for a differentiable function $f : \mathcal{B} \to \overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$, its (Frechét) derivative ∇f is in the dual space $\mathcal{B}^*$.

We use bold lowercase letters to denote vectors, where the i-th component of a vector $\mathbf{w}$ is denoted w_i, while $\mathbf{w}_t$ denotes some other vector. We use the shorthand $\langle \mathbf{w}, \mathbf{g} \rangle := \mathbf{g}(\mathbf{w})$ for any $\mathbf{g} \in \mathcal{B}^*$ and $\mathbf{w} \in \mathcal{B}$. We call a function $f : \mathcal{B} \to \overline{\mathbb{R}}$ σ-strongly convex (w.r.t. the norm $\|\cdot\|$) if there exists some $\sigma \geq 0$ such that for all $0 \leq \lambda \leq 1$ and $\mathbf{w}, \mathbf{z} \in \mathcal{B}$,

$$f(\lambda \mathbf{w} + (1-\lambda)\mathbf{z}) + \tfrac{1}{2}\sigma\lambda(1-\lambda)\|\mathbf{w}-\mathbf{z}\|^2 \leq \lambda f(\mathbf{w}) + (1-\lambda) f(\mathbf{z}). \tag{1}$$

In the case where $\sigma = 0$ we simply say f is convex. A set $C \subseteq \mathcal{B}$ is called convex if for all $\mathbf{w}, \mathbf{z} \in C \implies \lambda \mathbf{w} + (1-\lambda)\mathbf{z} \in C$ for all $0 \leq \lambda \leq 1$. It follows from the definition that the domain of a convex function f, denoted as $\operatorname{dom} f := \{\mathbf{w} : f(\mathbf{w}) < \infty\}$, is a convex set. The function f is called proper if $\operatorname{dom} f \neq \emptyset$, and closed if all sublevel sets $\{\mathbf{w} : f(\mathbf{w}) \leq \alpha\}$ are closed for all $\alpha \in \mathbb{R}$. The subdifferential of a convex function $f : \mathcal{B} \to \overline{\mathbb{R}}$ at $\mathbf{w}$ is defined as the set $\partial f(\mathbf{w}) = \{\mathbf{g} \in \mathcal{B}^* : f(\mathbf{z}) \geq f(\mathbf{w}) + \langle \mathbf{z} - \mathbf{w}, \mathbf{g} \rangle, \forall \mathbf{z} \in \mathcal{B}\}$.

Key to our subsequent development is two particular functions associated with a nonempty set in $\mathcal{B}$: the indicator function

$$\iota_C(\mathbf{w}) = \begin{cases} 0, & \text{if } \mathbf{w} \in C \\ \infty, & \text{otherwise} \end{cases}, \tag{2}$$

and the gauge function

$$\kappa(\mathbf{w}) = \kappa_A(\mathbf{w}) := \inf\{\rho \geq 0 : \mathbf{w} \in \rho A\}, \tag{3}$$

where $\rho A := \{\rho \mathbf{w} : \mathbf{w} \in A\}$, and the infimum is taken to be infinity over an empty set (i.e. when the condition $\mathbf{w} \in \rho A$ is not satisfied for any $\rho \geq 0$). Intuitively, $\kappa(\mathbf{w})$ is the smallest rescaling of the set A to "barely" contain $\mathbf{w}$; see Figure 1. Note that we always have $\kappa(\mathbf{0}) = 0$, and the gauge function is always positive homogeneous (i.e., $\kappa(t\mathbf{w}) = t\kappa(\mathbf{w})$ for all $t \geq 0$). Let $\mathsf{B} = \mathsf{B}_\kappa := \{\mathbf{w} : \kappa(\mathbf{w}) \leq 1\}$ denote the "unit ball" of κ, and then $\kappa_{\mathsf{B}} = \kappa_A$. B_κ is obviously intrinsic to κ, while different choices of A may yield the same gauge κ_A. Clearly, the indicator function ι_C is convex iff C is convex, while the gauge function κ is convex iff its unit ball is convex. Moreover, the gauge κ is a semi-norm iff its unit ball B is convex and symmetric (i.e. $\mathsf{B} = -\mathsf{B}$). Specifically, all semi-norms are convex gauge functions.

In practice, it is natural to engineer κ_A from a general "atomic set" A, which does not have to be convex (and can even be discrete) or contain the origin. Examples include the set of permutation matrices (Chandrasekaran et al., 2012), and the set of $\mathbf{x}\mathbf{x}^\top$ with $\|\mathbf{x}\|_2 = 1$ and $x_i \geq 0$ for low-rank nonnegative matrix factorization ($\mathbf{x}^\top$ is the transpose of $\mathbf{x}$). In this paper, we will work only with *convex closed* gauge functions, for which a sufficient *but not necessary* condition is that A is convex, closed, and containing the origin (an assumption

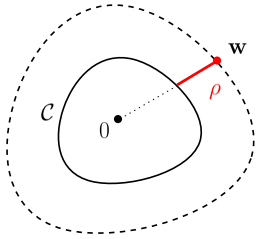


Figure 1: Defining a convex gauge via the Minkowski functional of a convex set $\mathcal{C}$.

made by many textbooks). For practical flexibility, we intentionally avoid making the latter assumptions in the first place, and the resulting technical complication is minimal.

A related function that underpins our algorithm is the *polar* of a gauge function κ_A, defined for all $\mathbf{g} \in \mathcal{B}^*$ as

$$\kappa^\circ(\mathbf{g}) = \kappa_A^\circ(\mathbf{g}) := \inf\{\mu \geq 0 : \langle \mathbf{w}, \mathbf{g} \rangle \leq \mu\kappa(\mathbf{w}),\ \forall \mathbf{w} \in \mathcal{B}\} \tag{4}$$

$$= \sup_{\mathbf{w} \in \mathsf{B}_\kappa} \langle \mathbf{w}, \mathbf{g} \rangle = \sup_{\mathbf{w} \in A \cup \{\mathbf{0}\}} \langle \mathbf{w}, \mathbf{g} \rangle . \tag{5}$$

As the notation suggests, the polar of a norm (a *bona fide* convex gauge) is its dual norm.

Lastly, a differential function $\ell : \mathcal{B} \to \mathbb{R}$ is said to be L-smooth (w.r.t. the norm $\|\cdot\|$) if for all $\mathbf{w}, \mathbf{z} \in \mathcal{B}$,

$$\ell(\mathbf{w}) \leq \tilde{\ell}_L(\mathbf{w}; \mathbf{z}) := \ell(\mathbf{z}) + \langle \mathbf{w} - \mathbf{z}, \nabla\ell(\mathbf{z}) \rangle + \tfrac{L}{2} \|\mathbf{w} - \mathbf{z}\|^2 , \tag{6}$$

i.e., ℓ is upper bounded by the quadratic approximation around $\mathbf{z}$. It is well-known that if the gradient $\nabla\ell$ is L-Lipschitz continuous, i.e., $\|\nabla\ell(\mathbf{w}) - \nabla\ell(\mathbf{z})\|^\circ \leq L\|\mathbf{w} - \mathbf{z}\|$ for all $\mathbf{w}, \mathbf{z}$, then ℓ is L-smooth. The converse is also true if ℓ is additionally convex. We will make repeated use of this quadratic upper bound in our analysis below.

2.2 Composite Minimization Problem

Many machine learning problems, particularly those formulated in terms of regularized risk minimization, can be formulated as:

$$\inf_{\mathbf{w} \in \mathcal{B}} F(\mathbf{w}),\ \text{such that}\ F(\mathbf{w}) := \ell(\mathbf{w}) + f(\mathbf{w}), \tag{7}$$

where f is convex and ℓ is convex and continuously differentiable. Typically, ℓ is a loss function and f is a regularizer although their roles can be reversed in some common examples, such as support vector machines and Lasso.

Due to the importance of this problem in machine learning, significant effort has been devoted to designing efficient algorithms for solving (7). Standard methods are generally based on computing an update sequence $\{\mathbf{w}_t\}$ that converges to a global minimizer of F when F is convex, or converges to a stationary point otherwise. For example, a popular

example is the mirror descent (MD) algorithm (Beck and Teboulle, 2003), which takes an arbitrary subgradient $\mathbf{g}_t \in \partial F(\mathbf{w}_t)$ and successively updates by

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left\{ \langle \mathbf{w}, \mathbf{g}_t \rangle + \frac{1}{2\eta_t} \mathsf{D}(\mathbf{w}, \mathbf{w}_t) \right\} \tag{8}$$

with a suitable step size $\eta_t \geq 0$; here $\mathsf{D}(\mathbf{w}, \mathbf{z}) := \mathsf{d}(\mathbf{w}) - \mathsf{d}(\mathbf{z}) - \langle \mathbf{w} - \mathbf{z}, \nabla\mathsf{d}(\mathbf{z}) \rangle$ denotes a Bregman divergence induced by some differentiable 1-strongly convex function $\mathsf{d} : \mathcal{B} \to \overline{\mathbb{R}}$. Under mild assumptions, MD converges (in terms of the function value) at a rate of $O(1/\sqrt{t})$ (Beck and Teboulle, 2003). However, this algorithm ignores the composite structure in (7) and can be impractically slow for many applications.

Another widely used algorithm is the proximal gradient (PG) (e.g. Fukushima and Mine, 1981), also known as the forward-backward splitting procedure, which iteratively performs the proximal update (PU)

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}} \left\{ \langle \mathbf{w}, \nabla\ell(\mathbf{w}_t) \rangle + f(\mathbf{w}) + \frac{1}{2\eta_t} \mathsf{D}(\mathbf{w}, \mathbf{w}_t) \right\}. \tag{9}$$

PU differs from MD updates in that it does not linearize the regularizer f. The downside is that (9) is usually harder to solve than (8) for each iteration, with an upside that a faster $O(1/t)$ rate of convergence can be achieved (Tseng, 2010). This rate can be further improved to $O(1/t^2)$ with the extrapolation variant known as accelerated proximal gradient (APG) (Beck and Teboulle, 2009; Nesterov, 2013). In particular, suppose $\mathcal{B} = \mathbb{R}^d$, $\mathsf{D}(\mathbf{w}, \mathbf{z}) = \frac{1}{2}\|\mathbf{w} - \mathbf{z}\|_2^2$, and $f(\mathbf{w}) = \|\mathbf{w}\|_1$, where the ℓ_p norm is defined as $\|\mathbf{w}\|_\mathsf{p} := (\sum_{i=1}^d |w_i|^\mathsf{p})^{1/\mathsf{p}}$ for $\mathsf{p} \geq 1$. Then PU recovers the soft-shrinkage operator, which plays an important role in sparse estimation:

$$\mathbf{z}_t = \mathbf{w}_t - \eta_t \nabla\ell(\mathbf{w}_t), \qquad \text{and} \qquad \mathbf{w}_{t+1} = (1 - \eta_t / |\mathbf{z}_t|)_+ \, \mathbf{z}_t. \tag{10}$$

Here the algebraic operations in the latter formula are performed componentwise. Similarly, when $\mathcal{B} = \mathbb{R}^{m \times n}$, $\mathsf{D}(W, Z) = \frac{1}{2}\|W - Z\|_\mathrm{F}^2$ (Frobenius norm) and $f(W) = \|W\|_\mathrm{tr}$ (trace norm, i.e. sum of singular values), one recovers the matrix analogue of soft-shrinkage used for matrix completion (Candès and Recht, 2009): here the same gradient update of $\mathbf{z}_t$ in (10) is applied, but the shrinkage operator on $\mathbf{w}_{t+1}$ in (10) is applied to the singular values of Z_t while keeping the singular vectors intact. Unfortunately, such a PU can become prohibitively expensive for large matrices, since it requires a *full* singular value decomposition (SVD) at each iteration, at a cubic time cost. We will see in Section 3.2 that an alternative algorithm only requires computing the dual of the trace norm (namely the spectral norm $\|\cdot\|_\mathrm{sp}$, i.e. the greatest singular value), which requires only quadratic time per iteration.

3. Generalized Conditional Gradient for Convex Optimization

In this section we present the generalized conditional gradient (GCG) algorithm, which has recently been receiving renewed attention (Bredies and Lorenz, 2008; Hazan, 2008; Jaggi, 2013; Shalev-Shwartz et al., 2010; Yuan and Yan, 2013; Zhang et al., 2012). Our goal in this section will be to develop efficient solution methods for (7), where f is convex and ℓ

is convex and L-smooth (cf. (6)). In particular, we will refine our algorithms to convex gauge regularized problems where special computational savings can be achieved through its polar. These are clearly among the most important settings in practice, and after some enhancements the algorithm and convergence guarantees we establish in this section will be applied to dictionary learning in Section 4.

3.1 General Setting with Convex and Smooth ℓ

We first develop the framework of the GCG algorithm and its analysis by focusing on general convex functions f. Formally, we make the following assumption throughout this Section 3.

Assumption 1 *ℓ is convex and L-smooth, and f is convex.*

Protocol: To avoid unnecessary pathologies, we tacitly assume all functions considered in this work are proper and closed.

To derive the optimization algorithm, first observe that given the above assumption, $F = \ell + f$ is convex and so any $\mathbf{w} \in \mathcal{B}$ is globally optimal for (7) if, and only if,

$$\mathbf{0} \in \partial F(\mathbf{w}) = \nabla \ell(\mathbf{w}) + \partial f(\mathbf{w}). \tag{11}$$

Let $f^*(\mathbf{g}) := \sup_{\mathbf{w}} \langle \mathbf{w}, \mathbf{g} \rangle - f(\mathbf{w})$ denote the Fenchel conjugate of f, and using the fact that $\mathbf{g} \in \partial f(\mathbf{w})$ iff $\mathbf{w} \in \partial f^*(\mathbf{g})$ when f is closed (Borwein and Vanderwerff, 2010), one can observe that the necessary and sufficient condition for $\mathbf{w}$ to be globally optimal can also be expressed as

$$(11) \quad \Longleftrightarrow \quad \mathbf{w} \in \partial f^*(-\nabla \ell(\mathbf{w})) \quad \Longleftrightarrow \quad \mathbf{w} \in (1-\eta)\mathbf{w} + \eta \partial f^*(-\nabla \ell(\mathbf{w})), \tag{12}$$

where $\eta \in (0, 1)$. Then, by the definition of f^*, we have

$$\mathbf{d} \in \partial f^*(-\nabla \ell(\mathbf{w})) \quad \Longleftrightarrow \quad \mathbf{d} \in \arg\min_{\mathbf{d}} \left\{ \langle \mathbf{d}, \nabla \ell(\mathbf{w}) \rangle + f(\mathbf{d}) \right\}. \tag{13}$$

Thus, the condition for a global minimum can be characterized by the fixed-point inclusion

$$\mathbf{w} \in (1-\eta)\mathbf{w} + \eta \mathbf{d} \quad \text{ for some } \quad \mathbf{d} \in \arg\min_{\mathbf{d}} \left\{ \langle \mathbf{d}, \nabla \ell(\mathbf{w}) \rangle + f(\mathbf{d}) \right\}. \tag{14}$$

This particular fixed-point condition immediately suggests an update that provides the foundation for the generalized conditional gradient algorithm (GCG) outlined in Algorithm 1. Each iteration of this procedure involves linearizing the smooth loss ℓ, solving the subproblem (13), selecting a step size, taking a convex combination, then conducting some form of local improvement. The GCG update based on (14) naturally generalizes the original conditional gradient update, which was first studied by (Frank and Wolfe, 1956) for the case when $f = \iota_C$ for a polyhedral set C.

Note that the subproblem (13) shares some similarity with the proximal update (9): both choose to leave the potentially nonsmooth function f intact, but here the smooth loss ℓ is replaced by its plain linearization rather than its linearization *plus* a (strictly convex) proximal regularizer such as the quadratic upper bound $\tilde{\ell}_L(\cdot)$ in (6). Consequently, there might be *multiple* solutions, in which case we simply adopt any one, or no solution (e.g.

Algorithm 1 Generalized Conditional Gradient (GCG).

1: Initialize: $\mathbf{w}_0 \in \operatorname{dom} f$.
2: **for** $t = 0, 1, \ldots$ **do**
3: Compute $\mathbf{d}_t \in \arg\min_{\mathbf{d}} \{\langle \mathbf{d}, \mathbf{g}_t \rangle + f(\mathbf{d})\}$, where $\mathbf{g}_t = \nabla \ell(\mathbf{w}_t)$.
4: Choose step size $\eta_t \in [0, 1]$ and perform update $\tilde{\mathbf{w}}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t \mathbf{d}_t$.
5: $\mathbf{w}_{t+1} = \texttt{Improve}(\tilde{\mathbf{w}}_{t+1}, \ell, f)$ ▷ Subroutine, see Definition 1
6: **end for**

divergence), in which case we will impose extra assumptions to ensure boundedness (details below, especially Assumption 2 and Theorem 2).

An important component of Algorithm 1 is the final step, `Improve`, in Line 5, which is particularly important in practice: it allows for heuristic local improvement to be conducted on the iterates without sacrificing any theoretical convergence properties of the overall procedure (see Section 4.3). A precursor of `Improve` appeared in (Meyer, 1974, Algorithm 2). Since `Improve` has access to both ℓ and f it can be very powerful in principle—we will consider the following variants that have respective consequences for practice and subsequent analysis.

Definition 1 *The subroutine `Improve` in Algorithm 1 is called `Null` if for all t, $\mathbf{w}_{t+1} = \tilde{\mathbf{w}}_{t+1}$; `Descent` if $F(\mathbf{w}_{t+1}) \leq F(\tilde{\mathbf{w}}_{t+1})$; `Monotone` if $F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}_t)$; and `Relaxed` if*

$$F(\mathbf{w}_{t+1}) \leq \tilde{\ell}_L(\tilde{\mathbf{w}}_{t+1}; \mathbf{w}_t) + (1 - \eta_t) f(\mathbf{w}_t) + \eta_t f(\mathbf{d}_t). \tag{15}$$

Obviously, `Relaxed` can be easily morphed to further satisfy `Descent` and/or `Monotone` by comparing $F(\mathbf{w}_{t+1})$ with $F(\tilde{\mathbf{w}}_{t+1})$ and/or $F(\mathbf{w}_t)$, whereas `Null` and `Descent` must be `Relaxed` (since ℓ is L-smooth).

Our main goal remains to establish that Algorithm 1 indeed converges to a global optimum under general conditions given in Assumption 1. In order to address the aforementioned issue of bounding $\mathbf{d}_t$, we next introduce one more assumption.

Assumption 2 *The sequences $\{\mathbf{d}_t\}$ and $\{\mathbf{w}_t\}$ generated by the algorithm are bounded for any choice of $\eta_t \in [0, 1]$.*

Although this assumption is more stringent, it can fortunately be achieved under various conditions, which we summarize as follows. The proof is given in Appendix A.1.

Proposition 2 *Let Assumption 1 hold. Then Assumption 2 is satisfied if any of the following holds.*

(a) The subroutine `Improve` is `Monotone`; the sublevel set $\{\mathbf{w} \in \operatorname{dom} f : F(\mathbf{w}) \leq F(\mathbf{w}_0)\}$ is compact; and f is cofinite, i.e., its Fenchel conjugate f^* has full domain.

(b) The subroutine `Improve` is `Monotone`; the sublevel set $\{\mathbf{w} \in \operatorname{dom} f : F(\mathbf{w}) \leq F(\mathbf{w}_0)\}$ is bounded; and f is super-coercive, i.e., $\lim_{\|\mathbf{w}\| \to \infty} f(\mathbf{w}) / \|\mathbf{w}\| \to \infty$.

(c) $\operatorname{dom} f$ is bounded.

In particular, under any of the conditions in Theorem 2, Line 3 of Algorithm 1 is well-defined. Note that if $\mathcal{B} = \mathbb{R}^d$, then condition (a) is in fact equivalent to condition (b). On the other hand, if f is a convex gauge, then none of the three conditions above holds, and we will deal with this in Section 3.2.

Finally, for each $\mathbf{w} \in \mathcal{B}$ we define a quantity, referred to as the *duality gap*, that will be useful in understanding the convergence properties of GCG:

$$\begin{aligned}\mathsf{G}(\mathbf{w}) &:= F(\mathbf{w}) - \inf_{\mathbf{d}} \{\ell(\mathbf{w}) + \langle \mathbf{d} - \mathbf{w}, \nabla \ell(\mathbf{w})\rangle + f(\mathbf{d})\} \quad \text{(linearizing } \ell \text{ at } \mathbf{w}) \qquad (16)\\ &= \langle \mathbf{w}, \nabla \ell(\mathbf{w})\rangle + f(\mathbf{w}) - \inf_{\mathbf{d}} \{\langle \mathbf{d}, \nabla \ell(\mathbf{w})\rangle + f(\mathbf{d})\}\\ &= \langle \mathbf{w}, \nabla \ell(\mathbf{w})\rangle + f(\mathbf{w}) + f^*(-\nabla \ell(\mathbf{w})). \qquad \text{(definition of Fenchel dual)} \qquad (17)\end{aligned}$$

Since f and ℓ are convex in Assumption 1, the duality gap provides an upper bound on the suboptimality of any search point, as established in the following proposition.

Proposition 3 *Let Assumption 1 hold. Then* $\mathsf{G}(\mathbf{w}) \geq F(\mathbf{w}) - \inf_{\mathbf{d}} F(\mathbf{d}) \geq 0$ *for all* $\mathbf{w} \in \mathcal{B}$. *Furthermore,* $\mathsf{G}(\mathbf{w}) = 0$ *iff* $\mathbf{w}$ *is globally optimal, i.e.* $\mathbf{w}$ *satisfies the condition* (11).

Proof As ℓ is convex, (16) implies $\mathsf{G}(\mathbf{w}) \geq F(\mathbf{w}) - \inf_{\mathbf{d}}\{\ell(\mathbf{d}) + f(\mathbf{d})\} = F(\mathbf{w}) - \inf_{\mathbf{d}} F(\mathbf{d}) \geq 0$. By the Fenchel-Young inequality, (17) implies $\mathsf{G}(\mathbf{w}) = 0$ iff $\mathbf{w} \in \partial f^*(-\nabla \ell(\mathbf{w}))$, i.e. (11) holds. ■

Since $\mathsf{G}(\mathbf{w}_t)$ is an upper bound on the suboptimality of $\mathbf{w}_t$, the duality gap gives a natural stopping criterion for (7) that can be easily computed in each iteration of Algorithm 1.

We are now ready for the first convergence result regarding Algorithm 1.

Theorem 4 *Let Assumptions 1 and 2 hold. Also assume the subroutine is* `Relaxed`*, and the subproblem* (13) *is solved up to some additive error* $\varepsilon_t \geq 0$. *Then, for any* $\mathbf{w} \in \operatorname{dom} F$ *and* $t \geq 0$, *Algorithm 1 yields*

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}) + \pi_t(1 - \eta_0)(F(\mathbf{w}_0) - F(\mathbf{w})) + \sum_{s=0}^{t} \frac{\pi_t}{\pi_s} \eta_s^2 (\varepsilon_s/\eta_s + \tfrac{L}{2} \|\mathbf{d}_s - \mathbf{w}_s\|^2), \quad (18)$$

where $\pi_t := \prod_{s=1}^{t}(1 - \eta_s)$ *with* $\pi_0 = 1$. *Furthermore, for all* $t \geq k \geq 0$, *the minimal duality gap* $\tilde{\mathsf{G}}_k^t := \min_{k \leq s \leq t} \mathsf{G}(\mathbf{w}_s)$ *satisfies*

$$\tilde{\mathsf{G}}_k^t \leq \frac{1}{\sum_{s=k}^{t} \eta_s} \left[F(\mathbf{w}_k) - F(\mathbf{w}_{t+1}) + \sum_{s=k}^{t} \eta_s^2 \left(\varepsilon_s/\eta_s + \tfrac{L}{2} \|\mathbf{d}_s - \mathbf{w}_s\|^2 \right) \right]. \quad (19)$$

From Theorem 4 we derive the following concrete rates of convergence.

Corollary 5 *Under the same assumptions as Theorem 4, also let* $\eta_t = 2/(t+2)$, $\varepsilon_t \leq \delta \eta_t / 2$ *for some* $\delta \geq 0$, *and* $L_F := \sup_t L \|\mathbf{d}_t - \mathbf{w}_t\|^2$. *Then, for all* $t \geq 1$ *and* $\mathbf{w}$, *Algorithm 1 yields*

$$F(\mathbf{w}_t) \leq F(\mathbf{w}) + \frac{2(\delta + L_F)}{t+3}, \qquad \text{and} \qquad \tilde{\mathsf{G}}_1^t \leq \frac{3(\delta + L_F)}{t \ln 2} \leq \frac{4.5(\delta + L_F)}{t}. \quad (20)$$

The proofs of Theorem 4 and Theorem 5 can be found in Appendix A.2 and Appendix A.3 respectively. Note that the simple step size rule $\eta_t = 2/(t+2)$ already leads to an $O(1/t)$ bound on the objective value attained. Of course, it is possible to use other step size rules. For instance, both $\eta_s = 1/(s+1)$ and the constant rule $\eta_s = 1 - (t+1)^{1/t}$ lead to an $O(\frac{1+\log t}{t+1})$ rate; see (Freund and Grigas, 2016) for detailed calculations.

We emphasize that GCG with the open-loop step size rule $\eta_t = 2/(t+2)$ requires neither the Lipschitz constant L nor the norm $\|\cdot\|$ to be known. We can therefore freely adopt the best choices for the problem at hand in the analysis. Note that, even though the rate does not depend on the initial point $\mathbf{w}_0$ provided $\eta_0 = 1$, by letting $\eta_0 \neq 1$ the bound can be slightly improved (Freund and Grigas, 2016).

Remark 6 *The observation that an* $O(1/t)$ *rate can be obtained by an open-loop step size rule such as* $\eta_t = O(1/t)$ *appears to have been first made by Dunn and Harshbarger (1978). The same rate can be achieved by the closed-loop step size rule where* η_t *is selected to minimize* $\ell((1-\eta_t)\mathbf{w}_t + \eta_t \mathbf{d}_t)$ *or its quadratic upper bound* $\tilde{\ell}((1-\eta_t)\mathbf{w}_t + \eta_t \mathbf{d}_t; \mathbf{w}_t)$ *(Frank and Wolfe, 1956; Levitin and Polyak, 1966; Dem'yanov and Rubinov, 1967). A similar rate on the minimal duality gap was also given in (Clarkson, 2010), and extended by (Jaggi, 2013). However, all these previous works focused on the special case where* $f = \iota_C$ *for some compact set* C. *Fukushima and Mine (1981) was the first to consider a general convex function* f *(and possibly a nonconvex* ℓ*), but they only established convergence for the algorithm. Bredies and Lorenz (2008) proved the* $O(1/t)$ *rate by using a closed-loop step size rule. More recently, Bach (2015) considered the special case where* f *is strongly convex, and identified the equivalence between GCG and MD, hence also establishing the* $O(1/t)$ *rate for this special case.*

3.2 Improved Algorithm and Refined Analysis when f is a Convex Gauge

Although the results in Theorem 4 and Theorem 5 hold for general convex functions f, they require $\{\mathbf{d}_t\}$ and $\{\mathbf{w}_t\}$ to be bounded as in Assumption 2. Some sufficient conditions were provided by Theorem 2. Unfortunately, these conditions cannot be satisfied by an important class of regularizers in machine learning: norms, semi-norms, and more generally convex *gauge* functions, cf. Section 2 for the detailed definition. Examples include the ℓ_p norms ($p \geq 1$), the total variation semi-norm, the trace norm for matrices, etc. Indeed, the subproblem (13) in Line 3 of Algorithm 1 might diverge to $-\infty$, leaving the update undefined. In this subsection, we develop a specialized form of GCG that circumvents this problem and exploits properties of convex gauges to significantly reduce computational overhead. The resulting GCG formulation, which is one of our main contributions, will be used to achieve efficient algorithms in important case studies in the second part of the paper, such as matrix completion under trace norm regularization (Section 4.2.1 below).

In the literature, two main approaches have been used to bypass the unboundedness in solving (13); unfortunately, both remain unsatisfactory in different ways. First, one obvious way to restore boundedness to (13) is to reformulate the regularized problem (7) in its equivalent constrained form

$$\inf_{\mathbf{w}} \ \ell(\mathbf{w}) \ \text{ s.t. } \ f(\mathbf{w}) \leq \zeta, \quad (21)$$

where it is well-known that a correspondence between (7) and (21) can be achieved if the constant ζ is chosen appropriately. In this case, Algorithm 1 can be directly applied as long as (21) has a bounded domain. Much recent work in machine learning has investigated this variant (Hazan, 2008; Jaggi and Sulovsky, 2010; Jaggi, 2013; Shalev-Shwartz et al., 2010; Tewari et al., 2011; Yuan and Yan, 2013). However, finding the appropriate value of ζ is costly in general, and often one cannot avoid considering the penalized formulation (e.g. when it is nested in another problem). Furthermore, the constraints in (21) preclude the application of many efficient local `Improve` techniques designed for *unconstrained* problems.

A second, alternative, approach is to square f, which ensures that it becomes super-coercive (Bradley and Bagnell, 2009). However, this modification is somewhat arbitrary, in the sense that super-coerciveness can be achieved by raising the norm to the pth power for any $p > 1$ (when $\mathcal{B} = \mathbb{R}^d$). Moreover, the insertion of a local `Improve` step in such an approach requires the regularizer f to be evaluated at all iterates, which is expensive for applications such as matrix completion.

3.2.1 Generalized Convex Gauge Regularization

To address the issue of ensuring well defined iterates, while also reducing their cost, we develop an alternative modification of GCG. The algorithm we develop also allows a mild generalization of convex gauge regularization by introducing a composition with an increasing convex function. In particular, let κ denote a convex gauge induced by a convex, closed set $\mathcal{C}$ containing the origin (cf. (3)) and let $h : \mathbb{R}_+ \to \overline{\mathbb{R}}$ denote an *increasing convex* function over $\mathbb{R}_+ := [0, \infty]$. Then regularizing by $f(\cdot) = h(\kappa(\cdot))$ yields the particular form of (7) that we will address with this modified approach:

$$\inf_{\mathbf{w}} \ \ell(\mathbf{w}) + h(\kappa(\mathbf{w})). \tag{22}$$

For the structured sparse regularizers typically considered in machine learning, the set $\mathcal{C}$ used to construct the gauge function κ via (3) has additional structure that admits efficient computation (Chandrasekaran et al., 2012). In particular, $\mathcal{C}$ is often constructed by taking the convex hull of some *compact* set of "atoms" $\mathcal{A}$ specific to the problem; i.e., $\mathcal{C} = \operatorname{conv}\mathcal{A}$. Such a structured formulation allows one to re-express the gauge as the "$\mathfrak{l}_1$ norm" over an appropriate "atomic decomposition." In details:

$$\kappa_{\mathcal{C}}(\mathbf{w}) = \inf\left\{\sum_{i=1}^{r} \lambda_i : \mathbf{w} = \sum_{i=1}^{r} \lambda_i \mathbf{a}_i, \mathbf{a}_i \in \mathcal{A}, \lambda_i \geq 0\right\} \tag{23}$$

$$\kappa^{\circ}_{\mathcal{C}}(\mathbf{g}) = \sup_{\mathbf{w} \in \mathcal{A} \cup \{\mathbf{0}\}} \langle \mathbf{w}, \mathbf{g} \rangle . \tag{24}$$

That is, if we decompose $\mathbf{w}$ as a conic combination of the "atoms" in $\mathcal{A}$, then $\kappa(\mathbf{w})$ is the least $\mathfrak{l}_1$ norm of the decomposition coefficients. Determining the convex gauge $\kappa_{\mathcal{C}}$ directly from the set $\mathcal{C}$ is not always easy, and the following duality result may come into aid (Rockafellar and Wets, 1998, p. 491):

$$\mathcal{C} \text{ is closed convex and } \mathbf{0} \in \mathcal{C} \implies \kappa_{\mathcal{C}} = (\kappa^{\circ}_{\mathcal{C}})^{\circ}. \tag{25}$$

Since $\mathcal{A}$ is compact, the polar $\kappa^{\circ}_{\mathcal{C}}$ has full domain. Intuitively, κ° has a much simpler form than κ if the structure of $\mathcal{A}$ is "simple." In such cases, it is advantageous to design

an optimization algorithm that circumvents evaluation of κ, by instead using κ° which is generally far less expensive to evaluate. This will be the underpinning motivation of our algorithm design.

3.2.2 Modified GCG Algorithm for Generalized Gauge Regularization

The key to the modified algorithm is to avoid the possibly unbounded solution to (13) while also avoiding explicit evaluation of the gauge κ. First, to avoid unboundedness, we adopt the technique of moving the regularizer to the constraint, as in (21), but rather than applying the modification directly to (22) we only use it in the subproblem (13) via:

$$\mathbf{d}_t \in \arg\min_{\mathbf{d}: h(\kappa(\mathbf{d})) \leq \zeta} \langle \mathbf{d}, \nabla \ell(\mathbf{w}_t) \rangle . \tag{26}$$

Then, to bypass the complication of dealing with h and the unknown bound ζ, we decompose $\mathbf{d}_t$ into its normalized direction $\mathbf{a}_t$ and scale θ_t (i.e., such that $\mathbf{d}_t = \mathbf{a}_t \theta_t / \eta_t$), determining each separately. In particular, from this decomposition, the update from Line 4 of Algorithm 1 can be equivalently expressed as

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t \mathbf{d}_t = (1 - \eta_t)\mathbf{w}_t + \theta_t \mathbf{a}_t, \tag{27}$$

hence the modified algorithm need only work with $\mathbf{a}_t$ and θ_t directly.

Determining the direction. First, the normalized direction, $\mathbf{a}_t$, can be recovered via

$$\mathbf{a}_t \in \arg\min_{\mathbf{a}: \kappa(\mathbf{a}) \leq 1} \langle \mathbf{a}, \nabla \ell(\mathbf{w}_t) \rangle \iff \mathbf{a}_t \in \arg\min_{\mathbf{a} \in \mathcal{A} \cup \{\mathbf{0}\}} \langle \mathbf{a}, \nabla \ell(\mathbf{w}_t) \rangle \quad \text{(by (4))} \tag{28}$$

$$\iff \mathbf{a}_t \in \arg\max_{\mathbf{a} \in \mathcal{A} \cup \{\mathbf{0}\}} \langle \mathbf{a}, -\nabla \ell(\mathbf{w}_t) \rangle . \tag{29}$$

Observe that (29) effectively involves computation of the polar $\kappa^{\circ}(-\nabla \ell(\mathbf{w}))$ only, by (4). Since this optimization might still be challenging, we further allow it to be solved only *approximately* to within an additive error $\varepsilon_t \geq 0$ and a multiplicative factor $\alpha_t \in (0, 1]$; that is, by relaxing (28) the modified algorithm only requires an $\mathbf{a}_t \in \mathcal{A}$ to be found that satisfies

$$\langle \mathbf{a}_t, \nabla \ell(\mathbf{w}_t) \rangle \leq \alpha_t \left(\varepsilon_t + \min_{\mathbf{a} \in \mathcal{A} \cup \{\mathbf{0}\}} \langle \mathbf{a}, \nabla \ell(\mathbf{w}_t) \rangle \right) = \alpha_t \left(\varepsilon_t - \kappa^{\circ}(-\nabla \ell(\mathbf{w}_t)) \right). \tag{30}$$

Intuitively, this formulation computes the normalized direction that has (approximately) the greatest negative "correlation" with the gradient $\nabla \ell$, yielding the steepest local decrease in ℓ. Importantly, this formulation does *not* require evaluation of the gauge function κ.

Determining the scale. Next, to choose the scale θ_t, it is natural to consider minimizing a simple upper bound on the objective that is obtained by replacing $\ell(\cdot)$ with its quadratic upper approximation $\tilde{\ell}_L(\cdot; \mathbf{w}_t)$ given in (6):

$$\min_{\theta \geq 0} \tilde{\ell}_L((1 - \eta_t)\mathbf{w}_t + \theta \mathbf{a}_t; \mathbf{w}_t) + h(\kappa((1 - \eta_t)\mathbf{w}_t + \theta \mathbf{a}_t)). \tag{31}$$

Unfortunately, κ still participates in (31), so a further approximation is required. Here is where the decomposition of $\mathbf{d}_t$ into $\mathbf{a}_t$ and θ_t is particularly advantageous. Observe that

$$\begin{aligned} h(\kappa((1-\eta_t)\mathbf{w}_t + \theta \mathbf{a}_t)) &\leq h((1-\eta_t)\kappa(\mathbf{w}_t) + \eta_t \kappa(\theta \mathbf{a}_t / \eta_t)) \\ &\leq (1-\eta_t) h(\kappa(\mathbf{w}_t)) + \eta_t h(\kappa(\theta \mathbf{a}_t / \eta_t)) \qquad (32) \\ &\leq (1 - \eta_t) h(\kappa(\mathbf{w}_t)) + \eta_t h(\theta / \eta_t)), \qquad (33) \end{aligned}$$

Algorithm 2 GCG for positively homogeneous regularizers.

Require: The set $\mathcal{A}$ whose convex hull $\mathcal{C}$ defines the gauge κ.

Initialize $\mathbf{w}_0$ and $\rho_0 \geq \kappa(\mathbf{w}_0)$.
for $t = 0, 1, \ldots$ **do**
  Choose normalized direction $\mathbf{a}_t$ that satisfies (30).
  Choose step size $\eta_t \in [0, 1]$ and set the scaling $\theta_t \geq 0$ by (34)
  $\tilde{\mathbf{w}}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \theta_t \mathbf{a}_t$, and $\tilde{\rho}_{t+1} = (1 - \eta_t)\rho_t + \theta_t$
  $(\mathbf{w}_{t+1}, \rho_{t+1}) = \texttt{Improve}(\tilde{\mathbf{w}}_{t+1}, \tilde{\rho}_{t+1}, \ell, f)$ ▷ Subroutine, see Theorem 8
end for

where the first inequality follows from the convexity of κ and the fact that h is increasing, the second inequality follows from the convexity of h, and (33) follows from the fact that $\kappa(\mathbf{a}_t) \leq 1$ by construction ($\mathbf{a}_t \in \mathcal{A}$ in (4)). Although κ still participates in (33) we can now bypass evaluation of κ by maintaining a simple upper bound $\rho_t \geq \kappa(\mathbf{w}_t)$, yielding the relaxed update

$$\theta_t = \arg\min_{\theta \geq 0} \left\{ \tilde{\ell}_L((1 - \eta_t)\mathbf{w}_t + \theta \mathbf{a}_t; \mathbf{w}_t) + (1 - \eta_t)h(\rho_t) + \eta_t h(\theta/\eta_t) \right\}. \tag{34}$$

Crucially, the upper bound $\rho_t \geq \kappa(\mathbf{w}_t)$ can be maintained by the simple update $\rho_{t+1} = (1 - \eta_t)\rho_t + \theta_t$, which is sufficient to ensure that ρ_{t+1} upper bounds $\kappa(\mathbf{w}_{t+1})$:

$$\rho_{t+1} = (1 - \eta_t)\rho_t + \theta_t \geq (1 - \eta_t)\kappa(\mathbf{w}_t) + \theta_t \kappa(\mathbf{a}_t) \geq \kappa((1 - \eta_t)\mathbf{w}_t + \theta_t \mathbf{a}_t) \geq \kappa(\mathbf{w}_{t+1}). \tag{35}$$

From these components we obtain the final modified procedure, Algorithm 2: a variant of GCG (Algorithm 1) that avoids the potential unboundedness of the subproblem (13) by replacing it with (30), while also completely avoiding explicit evaluation of the gauge κ. The `Improve` subroutine can be adapted by mirroring Theorem 1.

Definition 7 *The subroutine* `Improve` *in Algorithm 2 is called* `Null` *if for all* t, $\mathbf{w}_{t+1} = \tilde{\mathbf{w}}_{t+1}$ *and* $\rho_{t+1} = \tilde{\rho}_{t+1}$. *Provided that* $\rho_{t+1} \geq \kappa(\mathbf{w}_{t+1})$, `Improve` *is called* `Descent` *if* $\ell(\mathbf{w}_{t+1}) + h(\rho_{t+1}) \leq \ell(\tilde{\mathbf{w}}_{t+1}) + h(\tilde{\rho}_{t+1})$; `Monotone` *if* $\ell(\mathbf{w}_{t+1}) + h(\rho_{t+1}) \leq \ell(\mathbf{w}_t) + h(\rho_t)$; *and* `Relaxed` *if*

$$\ell(\mathbf{w}_{t+1}) + h(\rho_{t+1}) \leq \tilde{\ell}_L(\tilde{\mathbf{w}}_{t+1}; \mathbf{w}_t) + (1 - \eta_t)h(\rho_t) + \eta_t h(\theta_t/\eta_t). \tag{36}$$

Similar to Theorem 1, `Relaxed` can be easily morphed to further satisfy `Descent` and/or `Monotone` by comparing $(\mathbf{w}_{t+1}, \rho_{t+1})$ with $(\tilde{\mathbf{w}}_{t+1}, \tilde{\rho}_{t+1})$ and/or $(\mathbf{w}_t, \rho_t)$ over $\ell(\mathbf{w}) + h(\rho)$. As before `Null` and `Descent` are specializations of `Relaxed`.

For problems of the form (22), Algorithm 2 can provide significantly more efficient updates than Algorithm 1. Despite the relaxations introduced in Algorithm 2 to achieve faster iterates, the following theorem establishes that the procedure is still sound, preserving essentially the same convergence guarantees as the more expensive Algorithm 1.

Theorem 8 *Let* $h : \mathbb{R}_+ \to \overline{\mathbb{R}}$ *be an increasing convex function,* κ *be a convex gauge induced by the convex hull of a compact atomic set* $\mathcal{A} \cup \{\mathbf{0}\}$, *and* ℓ *be* L*-smooth and convex. Let* $F = \ell + h \circ \kappa$ *and assume Algorithm 2 generates a bounded sequence* $\{\mathbf{w}_t\}$. *Let* $\alpha_t > 0$, $0 \leq \eta_t \leq 1$, *and the subroutine* `Improve` *be* `Relaxed`. *Then for any* $\mathbf{w} \in \operatorname{dom} F$ *and* $t \geq 0$, *we have*

$$F(\mathbf{w}_{t+1}) \leq F(\mathbf{w}) + \pi_t(1 - \eta_0)\big(F(\mathbf{w}_0) - F(\mathbf{w})\big) + \sum_{s=0}^{t} \frac{\pi_t}{\pi_s} \eta_s^2 \Big(\big(\rho\varepsilon_s + h(\rho/\alpha_s) - h(\rho)\big)/\eta_s + \tfrac{L}{2} \left\| \tfrac{\rho}{\alpha_s}\mathbf{a}_s - \mathbf{w}_s \right\|^2 \Big), \tag{37}$$

where $\rho := \kappa(\mathbf{w})$ *and* $\pi_t := \prod_{s=1}^{t}(1 - \eta_s)$ *with* $\pi_0 = 1$.

The proof is given in Appendix A.4. From this theorem, we obtain the following rate of convergence by pursuing a similar argument as in the proof of Theorem 5.

Corollary 9 *Under the same setting as in Theorem 8, let* $\eta_t = 2/(t + 2)$, $\varepsilon_t \leq \delta\eta_t/2$ *for some absolute constant* $\delta > 0$, $\alpha_t = \alpha > 0$, $\rho = \kappa(\mathbf{w})$, *and* $L_F := L \sup_t \left\| \frac{\rho}{\alpha}\mathbf{a}_t - \mathbf{w}_t \right\|^2$. *Then for all* $t \geq 1$ *and* $\mathbf{w}$, *we have*

$$F(\mathbf{w}_t) \leq F(\mathbf{w}) + \frac{2(\rho\delta + L_F)}{t + 3} + h(\rho/\alpha) - h(\rho). \tag{38}$$

Moreover, if $\ell \geq 0$ *and* h *is the identity map, then*

$$F(\mathbf{w}_t) \leq \frac{1}{\alpha} F(\mathbf{w}) + \frac{2(\rho\delta + L_F)}{t + 3}. \tag{39}$$

Some remarks concerning this algorithm and its convergence properties are in order.

Remark 10 *When* $\alpha_t = 1$ *for all* t, *and* h *is the indicator* $\iota_{[0,\zeta]}$ *for some* $\zeta > 0$, *Theorem 9 implies the same convergence rate for the constrained problem* (21), *which immediately recovers (some of) the more specialized results discussed in (Clarkson, 2010; Hazan, 2008; Jaggi, 2013; Shalev-Shwartz et al., 2010; Tewari et al., 2011; Yuan and Yan, 2013). For the special case where* h *is the identity map, a similar* $O(1/t)$ *rate achieved by a similar algorithm appeared in Harchaoui et al. (2015), whose workshop version appeared around the same time as our conference version (Zhang et al., 2012). Harchaoui et al. (2015) considered the slightly more restrictive setting where the convex gauge* κ *is the restriction of a norm into a convex cone and they did not consider (multiplicative) approximate polar computations. Their algorithm also needs to evaluate* κ *explicitly, while the key advantage of our method is to avoid this expensive computation by an upper bound. On the other hand, Harchaoui et al. (2015) investigated the memory based acceleration in (Holloway, 1974; Meyer, 1974) while we develop in the next section a fixed-rank local improvement step.*

Remark 11 *The additional factor* ρ *in the above bounds is necessary because the approximation in* (30) *is not invariant to scaling, requiring some compensation. Note also that* $\alpha \leq 1$, *since the right-hand side of* (30) *must become negative as* $\varepsilon_t \to 0$. *The result in* (39) *states roughly that an* α*-approximate subroutine (for computing the polar of* κ*) leads to an* α*-approximate "minimum", also at the rate of* $O(1/t)$. *This observation was independently made in (Bach, 2013), where the multiplicative factor* ρ *is introduced to solve approximately some matrix factorization problems. In a similar vein, Cheng et al. (2016) leveraged the multiplicative guarantee to solve some tensor problems.*

Remark 12 *The assumption that $\{\mathbf{w}_t\}$ is bounded can be satisfied easily, which in turn ensures $L_F < \infty$. The simplest way is to postulate that the sublevel set $\{\mathbf{w} \in \operatorname{dom} F : F(\mathbf{w}) \leq \ell(\mathbf{w}_0) + h(\rho_0)\}$ is bounded. In this case $\{\mathbf{w}_t\}$ is trivially bounded if* `Monotone` *is used. When* `Relaxed` *is used instead (which includes* `Null` *and* `Descent`*), we can strengthen it into* `Monotone` *via the roll-back trick in Theorem 7. At first sight, the algorithm appears to be stuck, but in fact the diminishing step size η_t will eventually lead to progresses as guaranteed by Theorem 9. Computationally, keeping track of $\ell(\mathbf{w}_t)+h(\rho_t)$ does not incur any overhead. We will provide concrete values/bounds of L_F for the case studies in Section 4.2 and Section 4.4.*

Remark 13 *The convergence results in Theorem 8 and Theorem 9 obviously hold when* `Null` *or* `Descent` *is employed for* `Improve`*, because they are both specializations of* `Relaxed`*. The update for θ in (34) requires knowledge of the Lipschitz constant L and the norm $\|\cdot\|$. If the norm $\|\cdot\|$ is Hilbertian and h is the identity map, we have the formula*

$$\theta_t = \tfrac{1}{L\|\mathbf{a}_t\|^2} \max\{\langle \mathbf{a}_t, L\eta_t\mathbf{w}_t - \nabla\ell(\mathbf{w}_t)\rangle - 1, 0\}.$$

However, it is also easy to devise other scaling and step size updates that do not require the knowledge of L or the norm $\|\cdot\|$, such as

$$(\eta_t^*, \theta_t^*) \in \arg\min_{1\geq\eta\geq 0, \theta\geq 0} \ell((1-\eta)\mathbf{w}_t + \theta\mathbf{a}_t) + (1-\eta)h(\rho_t) + \eta h(\theta/\eta), \tag{40}$$

followed by the `Null` *step. Note that (40) is jointly convex in η and θ, since $\eta h(\theta/\eta)$ is a perspective function. Evidently this update can be treated as a* `Relaxed` *subroutine, since for any (η_t, θ_t) chosen in step 4 of Algorithm 2, it follows that*

$$\begin{aligned}
\ell(\mathbf{w}_{t+1}) + h(\rho_{t+1}) &\leq \ell((1-\eta_t^*)\mathbf{w}_t + \theta_t^*\mathbf{a}_t) + (1-\eta_t^*)h(\rho_t) + \eta_t^* h(\theta_t^*/\eta_t^*) \\
&\leq \ell((1-\eta_t)\mathbf{w}_t + \theta_t\mathbf{a}_t) + (1-\eta_t)h(\rho_t) + \eta_t h(\theta_t/\eta_t) \quad \text{(by (40))} \\
&\leq \tilde{\ell}_L(\tilde{\mathbf{w}}_{t+1}; \mathbf{w}_t) + (1-\eta_t)h(\rho_t) + \eta_t h(\theta_t/\eta_t).
\end{aligned}$$

The upper bound $\rho_{t+1} \geq \kappa(\mathbf{w}_{t+1})$ can be verified in the same way as for (35); hence the update (40) enjoys the same convergence guarantee as in Theorem 9.

Remark 14 *An alternative workaround for solving (7) with gauge regularization would be to impose an upper bound $\zeta \geq \kappa(\mathbf{w})$ and consider $\min_{\mathbf{w}:\kappa(\mathbf{w})\leq\zeta}\{\ell(\mathbf{w}) + \kappa(\mathbf{w})\}$, while dynamically adjusting ζ. A standard GCG approach, similar to (Harchaoui et al., 2015), would then require solving*

$$\mathbf{d}_t \in \arg\min_{\mathbf{d}:\kappa(\mathbf{d})\leq\zeta} \{\langle \mathbf{d}, \nabla\ell(\mathbf{w}_t)\rangle + \kappa(\mathbf{d})\} \tag{41}$$

as a subproblem, which can be as easy to solve as (30). If, however, one wished to include a local improvement step `Improve` *(which is essential in practice), it is not clear how to efficiently maintain the constraint $\kappa(\mathbf{w}) \leq \zeta$. Moreover, solving (41) up to some* multiplicative *factor might* not *be as easy as (30).*

3.3 Additional Discussions and Related Work

In this section we discuss some further properties and extensions of the GCG algorithm.

Smoothing: The convergence rates we have established require the convex loss ℓ to be L-smooth. Although this holds for a variety of losses such as the square and logistic loss, it does not hold for the hinge loss used in support vector machines. However, one can always approximate a nonsmooth (Lipschitz) loss by a smooth function (Nesterov, 2005) before applying GCG, at the expense of getting a slower $O(1/\sqrt{t})$ rate of convergence.

Totally Corrective: As given, Algorithm 2 adds a single atom in each iteration, with a conic combination between the new atom and the previous aggregate. We will refer to it as *vanilla* GCG if `Improve` is further set to `Null`. If $f = \iota_C$ for some compact and convex set C, we simply call it *vanilla* CG for disambiguation. An even more aggressive scheme is to re-optimize the coefficients of all (or a subset of) existing atoms in each iteration. Such a procedure was first studied in (Meyer, 1974; Holloway, 1974), which is often known as the "totally (or fully) corrective update" in the boosting literature and as the "restricted simplicial decomposition" in optimization. More generally, for the convex gauge κ, each iteration of a totally corrective procedure would involve solving

$$\min_{\boldsymbol{\sigma}\geq 0} \ell\left(\sum_{\tau\in A_t} \sigma_\tau \mathbf{b}_\tau\right) + \sum_{\tau\in A_t} \sigma_\tau, \tag{42}$$

where A_t is a set of atoms accumulated up to the t-th iteration. When $f = \iota_C$, the totally corrective CG solves $\min_{\boldsymbol{\sigma}} \ell\left(\sum_{\tau\in A_t} \sigma_\tau \mathbf{b}_\tau\right)$ over $\boldsymbol{\sigma} \geq 0$ and $\mathbf{1}'\boldsymbol{\sigma} = 1$. As pointed out by Holloway (1974), as long as $\{\mathbf{w}_t, \mathbf{a}_t\} \subseteq A_t$, the totally corrective CG will converge at least as fast as the vanilla CG. Much faster convergence is usually observed in practice, although this advantage must be countered by the extra effort spent in solving (42) and alike, which itself need not be trivial. In a finite dimensional setting, provided that the atoms are linearly independent and some restricted strong convexity is present, it is possible to prove that the totally corrective CG converges at a linear rate; see (Shalev-Shwartz et al., 2010; Yuan and Yan, 2013). Recent work of Lacoste-Julien and Jaggi (2015) has further relaxed the requirement of linear independence by making a weaker assumption that the number of atoms is finite (i.e. the feasible region is a polytope).

GCG v.s. PG: The convergence rate established for GCG is on par with PG. When $f = \iota_C$, Canon and Cullum (1968) showed that the rate of *vanilla* CG cannot be improved in general even when ℓ is strongly convex.[1] In this sense, *vanilla* GCG (which subsumes *vanilla* CG) is slower than an "optimal" algorithm like APG. GCG's advantage, however, is that it only needs to solve a linear subproblem (13) in each iteration (i.e., computing the polar κ°), whereas APG (or PG) needs to compute the proximal update, which is a quadratic problem for the least square prox-function. The two approaches are complementary in the sense that the polar can be easier to compute in some cases, whereas in other cases the proximal update can be computed analytically. An additional advantage GCG has over APG, however, is its greedy sparse nature: Each iteration of GCG amounts to a single atom, hence the number of atoms never exceeds the total number of iterations for GCG. By contrast, APG can produce a dense update in each iteration, although in later stages the

1. Improvements in specific cases are still possible; see the next paragraph. Indeed the counter-example of Canon and Cullum (1968) only applies when the solution lies at the boundary of a polyhedral set.

estimates may become sparser due to the shrinkage effect of the proximal update. Moreover, GCG is "robust" with respect to α-approximate atom selection (cf. (39) in Theorem 9), whereas similar results do not appear to be available for PG or APG when the proximal update is computed up to a multiplicative factor.

Faster Rates: Faster rates of convergence for CG (with a closed-loop step size rule), under the restriction $f = \iota_C$, have been considered by a number of works. Levitin and Polyak (1966) first showed that if C is strongly convex and the gradient $\nabla\ell$ is bounded away from $\mathbf{0}$, then a linear rate of convergence can be achieved by CG. Guélat and Marcotte (1986) showed that if there is a minimizer in the interior of C, then again a linear rate of convergence can be obtained. Wolfe (1970) proposed an away step modification of CG, and a variant of it designed by Guélat and Marcotte (1986) attained linear rate of convergence when C is polyhedral. The analysis was further improved into a global linear rate in (Lacoste-Julien and Jaggi, 2015; Beck and Shtern, 2015). Dunn (1979) proved $o(1/t)$ rate, linear rate and finite convergence for CG, depending on how "continuous" the polar operation is. Lastly, Garber and Hazan (2015) showed that for polytope C, a modification of CG achieves the linear rate if ℓ is of quadratic growth (in particular, strongly convex), and Garber and Hazan (2016) proved the $O(1/t^2)$ rate if ℓ is of quadratic growth and C is strongly convex. Note that these faster rates are possible by putting restrictive assumptions on the constraint set C (polyhedral or strongly convex). Interestingly, results on faster rates for *general convex* functions f are scarce, although establishing it for GCG is conceivably not hard. For example, in (Zhang et al., 2012) we proved a linear rate for the totally corrective GCG when ℓ is strongly convex (in a slightly stronger sense), f is a convex gauge, and $\mathcal{B} = \mathbb{R}^n$.

4. Application to Low Rank Learning

As a first application of GCG, we consider the problem of optimizing low rank matrices; a problem that forms the cornerstone of many machine learning tasks such as dictionary learning and matrix completion. Since imposing a bound on matrix rank generally leads to an intractable problem, much recent work has investigated approaches that relax the hard rank constraint with carefully designed regularization penalties that indirectly encourage low rank structure (Bach et al., 2008; Bradley and Bagnell, 2009). In Section 4.2 we first formulate a generic convex relaxation scheme for low rank problems arising in matrix approximation tasks, including dictionary learning and matrix completion. Conveniently, the modified GCG algorithm developed in Section 3.2, Algorithm 2, provides a general solution method for the resulting problems. Next, in Section 4.3 we show how the local improvement component of the modified GCG algorithm can be provided in this case by a procedure that optimizes an unconstrained surrogate (instead of a constrained problem) to obtain greater efficiency. This modification significantly improves the practical efficiency of GCG without affecting its convergence properties. We illustrate these contributions by highlighting the example of matrix completion under trace norm regularization, where the improved GCG algorithm demonstrates state of the art performance. Finally, we demonstrate how the convex relaxation can be generalized to handle latent multi-view models (White et al., 2012) in Section 4.4, showing how the improved GCG algorithm can still be applied once the major challenge of efficiently computing the polar κ° has been solved.

4.1 Low Rank Learning Problems

Many machine learning tasks, such as dictionary learning and matrix completion, can be formulated as seeking low rank matrix approximations of a given data matrix. In these problems, one is presented with an $n \times m$ matrix X (perhaps only partially observed) where each column corresponds to a training example and each row corresponds to a feature across examples; the goal is to recover a low rank matrix W that approximates (the observed entries of) X. To achieve this, we assume we are given a convex loss function $\ell(W) := L(W, X)$ that measures how well W approximates X.[2] If we impose a hard bound, t, on the rank of W, the core training problem can then be expressed as

$$\inf_{W:\mathrm{rank}(W)\leq t} \ell(W) = \inf_{U\in\mathbb{R}^{n\times t}} \inf_{V\in\mathbb{R}^{t\times m}} \ell(UV). \tag{43}$$

Unfortunately, this problem is not convex, and except for special cases (such as SVD) is believed to be intractable.

The two forms of this problem that we will explicitly illustrate in this paper are dictionary learning and matrix completion. First, for dictionary learning, the $n \times m$ data matrix X is fully observed, the $n \times t$ matrix U is interpreted as a "dictionary" consisting of t basis vectors, and the $t \times m$ matrix V is interpreted as the "coefficients" consisting of m code vectors. For this problem, the dictionary and coefficient matrices need to be inferred *simultaneously*, in contrast to traditional signal approximation schemes where the dictionary (e.g., the Fourier or a wavelet basis) is fixed *a priori*. Unfortunately, learning the dictionary with the coefficients creates significant computational challenge since the formulation is no longer *jointly* convex in the variables U and V, even when the loss ℓ is convex. Indeed, for a fixed dictionary size t, the problem is known to be NP-hard (Vavasis, 2010) except for very special cases (Yu and Schuurmans, 2011).

The second learning problem we illustrate is low rank matrix completion. Here, one observes a small number of entries in $X \in \mathbb{R}^{n\times m}$ and the task is to infer the remaining entries. In this case, the optimization objective can be expressed as $\ell(W) := L(\mathcal{P}(W - X))$, where L is a reconstruction loss (such as Frobenius norm squared) and $\mathcal{P} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ is a "mask" operator that simply fills the entries corresponding to unobserved indices in X with zero. Various recommendation problems, such as the Netflix prize,[3] can be cast as matrix completion. Due to the ill-posed nature of the problem it is standard to assume that the matrix X can be well approximated by a low rank reconstruction W. Unfortunately, the rank function is not convex, hence hard to minimize in general. Conveniently, both the dictionary learning and matrix completion problems are subsumed by the general formulation we consider in (43).

4.2 General Convex Relaxation and Solution via GCG

To develop a convex relaxation of (43), we first consider the factored formulation expressed in terms of U and V on the right hand side. Although it only involves an unconstrained minimization which can be convenient, care is needed to handle the scale invariance introduced by the factored form: since U can always be scaled and V counter-scaled to preserve

2. Note that the matrix W need not reconstruct the entries of X directly: a nonlinear transfer could be interposed while maintaining convexity of ℓ; for example, by using an appropriate Bregman divergence.
3. `http://www.netflixprize.com//community/viewtopic.php?id=1537`

the product UV, some form of penalty or constraint is required to restore well-posedness. A standard strategy is to constrain U, for which a particularly common choice is to constrain its columns to have unit length; i.e. $\|U_{:i}\|_c \leq 1$ for all i, where $U_{:i}$ stands for the i-th column of U, and c simply denotes some norm on the columns. In dictionary learning, for example, this amounts to constraining the basis vectors in the dictionary U to lie within the unit ball of $\|\cdot\|_c$. Similar normalization can be used to represent the rows $V_{i:}$, with the magnitude encoded by $\sigma_i \geq 0$. In consequence, the unconstrained optimization over $\ell(UV)$ in (43) can be reformulated as

$$\min_{U\in\mathbb{R}^{n\times t}, V\in\mathbb{R}^{t\times m}, \boldsymbol{\sigma}\in\mathbb{R}^t} \ell(U\operatorname{diag}(\boldsymbol{\sigma})V), \quad s.t. \quad \|U_{:i}\|_c \leq 1,\ \|V_{i:}\|_r \leq 1, \text{ and } \sigma_i \geq 0. \tag{44}$$

Here $\boldsymbol{\sigma} := (\sigma_1, \sigma_2, \ldots, \sigma_t)^\top$, and $\operatorname{diag}(\boldsymbol{\sigma})$ is a diagonal matrix with the i-th diagonal entry being σ_i. The specific form of the row norm $\|\cdot\|_r$ provides additional flexibility in promoting different structures; for example, the ℓ_1 norm leads to sparse solutions, the ℓ_2 norm yields low rank solutions, and block structured norms generate group sparsity. The specific form of the column norm $\|\cdot\|_c$ also has an effect, as discussed in Section 4.4 below.

Of course the problem remains non-convex. However, it has recently been observed that a simple relaxation of the rank constraint in (43) allows a convex reformulation to be achieved (Argyriou et al., 2008; Bach et al., 2008; Bradley and Bagnell, 2009; Zhang et al., 2011). Observe that replacing the rank constraint with a regularization penalty on the magnitude σ_i of each basis pair $(U_{:i}, V_{i:})$ yields a relaxed form of (43):

$$\inf_{U:\|U_{:i}\|_c\leq 1} \inf_{V:\|V_{i:}\|_r\leq 1} \inf_{\boldsymbol{\sigma}:\sigma_i\geq 0} \ell(U\operatorname{diag}(\boldsymbol{\sigma})V) + \lambda\|\boldsymbol{\sigma}\|_1, \tag{45}$$

where the number of columns in U (and also the number of rows in V) is *not* restricted. Similar to Lasso, the ℓ_1 norm on the "singular values" (combination weights) $\boldsymbol{\sigma}$ is used as a penalty that serves as a convex proxy for rank. By the shrinkage effect of the ℓ_1 norm, many σ_i will become zero, implying that the corresponding columns of U (and rows of V) can be dropped. As a result, the rank of the solution—which is indeed unknown *a priori*—can be chosen adaptively with the trade-off parameter $\lambda \geq 0$.

Importantly, although the problem (45) is still not jointly convex in the factors U, V and $\boldsymbol{\sigma}$, it can be exactly reformulated as a convex optimization as long as ℓ is convex:

$$(45) = \min_W\ \ell(W) + \lambda\ \inf\left\{\sum_i \sigma_i : \boldsymbol{\sigma}\geq 0, W = \sum_i \sigma_i U_{:i}V_{i:}, \|U_{:i}\|_c \leq 1, \|V_{i:}\|_r \leq 1\right\} \tag{46}$$

$$= \min_W\ \ell(W) + \lambda\,\kappa(W), \tag{47}$$

where the gauge κ is defined by the following set $\mathcal{A}$ with uncountably many elements

$$\mathcal{A} := \{\mathbf{u}\mathbf{v}^\top : \mathbf{u}\in\mathbb{R}^n, \mathbf{v}\in\mathbb{R}^m, \|\mathbf{u}\|_c \leq 1, \|\mathbf{v}\|_r \leq 1\}. \tag{48}$$

The resulting convex formulation (47) has been observed, in various forms, by, e.g., Argyriou et al. (2008); Bach et al. (2008); Zhang et al. (2011); White et al. (2012). However our derivation, first presented in (Zhang et al., 2012), provides a much simpler and flexible view based on convex gauges, which also establishes a direct connection to the GCG method

outlined in Section 3.2, yielding an efficient solution method with no additional effort. The reformulation (47) reveals that the relaxed learning problem (45) is essentially considering a rank-one decomposition of W, which generalizes the singular value decomposition. Furthermore it provides a clearer understanding of the computational properties of the regularizer in (47) via its polar:

$$\kappa^\circ(G) = \sup_{A\in\mathcal{A}} \langle A, G\rangle = \sup_{\|\mathbf{u}\|_c\leq 1, \|\mathbf{v}\|_r\leq 1} \mathbf{u}^\top G\mathbf{v} \tag{49}$$

$$= \sup_{\|\mathbf{v}\|_r\leq 1} \|G\mathbf{v}\|_c^\circ = \sup_{\|\mathbf{u}\|_c\leq 1} \left\|G^\top\mathbf{u}\right\|_r^\circ. \tag{50}$$

Recall that for gauge regularized problems in the form (47), Algorithm 2 only requires approximate computation of the polar (30) in Line 3; hence an efficient method for (approximately) solving (49) immediately yields an efficient implementation. However, this simplicity must be countered by the fact that the polar is not always tractable, since it involves *maximizing* a norm subject to a different norm constraint.[4] Nevertheless, there exist important cases where the polar can be efficiently evaluated, which we illustrate below.

The polar also allows one to gain insight into the structure of the original gauge regularizer, since by duality $\kappa = (\kappa^\circ)^\circ$ yields an explicit formula for κ. For example, consider the important special case where $\|\cdot\|_r = \|\cdot\|_c = \|\cdot\|_2$. In this case, using (49) one can infer that the polar $\kappa^\circ(\cdot) = \|\cdot\|_{\text{sp}}$ is the spectral norm, and therefore the corresponding gauge regularizer $\kappa(\cdot) = \|\cdot\|_{\text{tr}}$ is the trace norm. In this way, the trace norm regularizer, which is known to provide a convex relaxation of the matrix rank (Chandrasekaran et al., 2012), can be viewed from the perspective of dictionary learning as an *induced* norm that arises from a simple 2-norm constraint on the magnitude of the dictionary entries $U_{:i}$ and a simple 2-norm penalty on the magnitudes of the code vectors $V_{i:}$.

This convex relaxation strategy based on convex gauges is quite flexible and has been recently studied for some structured sparse problems (Chandrasekaran et al., 2012; Tewari et al., 2011; Zhang et al., 2012; Bach, 2013). Importantly, the generality of the column and row norms in (45) proffer considerable flexibility to finely characterize the structures sought by low rank learning methods. We will exploit this flexibility in Section 4.4 below to achieve a novel extension of this framework to multi-view dictionary learning (White et al., 2012).

4.2.1 Application to Matrix Completion

We end this subsection with a brief illustration of how the framework can be applied to matrix completion. Recall that in this case one is given a partially observed data matrix X, and the goal is to find a low rank approximator W that minimizes the reconstruction error on observed entries. In particular, consider the following standard specialization of (47):

$$\min_{W\in\mathbb{R}^{n\times m}} \frac{1}{2\|\mathcal{P}\|_0}\|\mathcal{P}(X-W)\|_{\text{F}}^2 + \lambda\|W\|_{\text{tr}}, \tag{51}$$

where, as noted above, $\mathcal{P}: \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ is the mask operator that fills entries where X is unobserved with zero. $\|\mathcal{P}\|_0$ is the number of observed entries. Surprisingly, Candès and

4. In this regard, the α-approximate polar oracle introduced in (30) and Theorem 9 might be useful, although we shall not develop this idea further here. See the work of Bach (2013) for some applications of this idea to matrix factorization and the work of Cheng et al. (2016) to tensor problems.

Recht (2009) have proved that, assuming X is (approximately) low-rank, the solution of the convex relaxation (51) will recover the true matrix X with high probability, even when only a small *random* portion of X is observed.

Note that, since (51) is a specialization of (47), the modified GCG algorithm, Algorithm 2, can be immediately applied. Interestingly, such an approach contrasts with the most popular algorithm favored in the current literature: PG (or its accelerated variant APG; cf. Section 2). The two strategies, GCG versus PG, exhibit an interesting trade-off. Each iteration of PG (or APG) involves solving the PU (9), which in this case requires a *full* singular value decomposition (SVD) of the current approximant W. The modified GCG method, Algorithm 2, by contrast, only requires the polar of the gradient matrix $\nabla\ell(W)$ to be (approximately) computed in each iteration (i.e. dual of the trace norm; namely, the spectral norm), which is an order of magnitude cheaper to evaluate than a full SVD (cubic versus squared worst case time).[5] On the other hand, GCG has a slower theoretical rate than APG, $O(1/t)$ versus $O(1/t^2)$, in terms of the number of iterations required to achieve a given accuracy. Once the enhancements in the next section have been incorporated, our experiments in Section 5.1 will demonstrate that GCG is far more efficient than APG for solving the matrix completion problem (51) to tolerances used in practice.

Given the specific objective (51), we may derive a concrete bound on L_F used in Theorem 9. Obviously, the smoothness constant is $L = 1/\left\|\mathcal{P}\right\|_0$. Suppose we use `Monotone` with $W_0 = \mathbf{0}$. So all W_t (and the optimal W) need to satisfy $\lambda \left\|W_t\right\|_{\text{tr}} \leq C$, where $C = \left\|\mathcal{P}X\right\|_{\text{F}}^2 / (2\left\|\mathcal{P}\right\|_0)$. Due to the normalization, C can be considered a universal constant. All matrices $A \in \mathcal{A}$ in (48) satisfy $\left\|A\right\|_{\text{F}} \leq 1$. Obviously $\frac{1}{n}\left\|W_t\right\|_{\text{tr}}^2 \leq \left\|W_t\right\|_{\text{F}}^2 \leq \left\|W_t\right\|_{\text{tr}}^2$. So

$$L_F \leq 2L \sup_t \left\{ \frac{\rho^2}{\alpha^2}\left\|A_t\right\|_{\text{F}}^2 + \left\|W_t\right\|_{\text{F}}^2 \right\} \leq 2L \sup_t \left\{ \frac{\rho^2}{\alpha^2} + \left\|W_t\right\|_{\text{tr}}^2 \right\} \leq \frac{2}{\left\|\mathcal{P}\right\|_0}\left(\frac{1}{\alpha^2}+1\right)\frac{C^2}{\lambda^2}. \quad (52)$$

If λ needs to be less than $\frac{1}{\left\|\mathcal{P}\right\|_0}$, then the upper bound on L_F will be $O(\left\|\mathcal{P}\right\|_0)$. Notice that under `Monotone`, the expression of L_F in Theorem 5 is on par with the condition number $L_f D_*^2$ in (Harchaoui et al., 2015, Theorem 3).

4.3 Fixed-rank Local Optimization

Although the modified GCG algorithm, Algorithm 2, can be readily applied to the reformulated low rank learning problem (47), the sublinear rate of convergence established in Theorem 9 is still too slow in practice. Nevertheless, an effective local improvement scheme can be developed that satisfies the properties of a `Relaxed` subroutine in Algorithm 2, which allows empirical convergence to be greatly accelerated. Recall that in Algorithm 2, the intermediate iterate $\tilde{W}_{t+1}$ is determined by a linear combination of the previous iterate W_t and the newly added atom A_t (we changed the notation from $\tilde{\mathbf{w}}_{t+1}$ to $\tilde{W}_{t+1}$ since

5. The per-iteration cost of computing the spectral norm can be further reduced for the sparse matrices occuring naturally in matrix completion, since the gradient matrix in (51) contains at most k nonzero elements for k equal to the number of observed entries in X. In such cases, an ϵ accurate estimate of the squared spectral norm can be obtained in $O\left(\frac{k\log(n+m)}{\sqrt{\epsilon}}\right)$ time (Jaggi, 2013; Kuczyński and Woźniakowski, 1992), offering a significant speed up over the general case. Although some speed ups might also be possible when computing the SVD for sparse matrices, the gap between the computational cost for evaluating the spectral norm versus the trace norm for sparse matrices remains significant.

the optimization variables are now matrices). We will next demonstrate that $\tilde{W}_{t+1}$ can be further improved by solving a reformulated local problem, and then show how this can still preserve the convergence guarantees of Algorithm 2.

The key reformulation relies on the following fact.

Proposition 15 *The convex gauge κ induced by the set $\mathcal{A}$ in (48) can be re-expressed as*

$$\kappa(W) = \min\left\{ \sum_{i=1}^{t} \left\|U_{:i}\right\|_c \left\|V_{i:}\right\|_r : UV = W \right\} = \min\left\{ \frac{1}{2}\sum_{i=1}^{t}\left(\left\|U_{:i}\right\|_c^2 + \left\|V_{i:}\right\|_r^2 \right) : UV = W \right\},$$

for $U \in \mathbb{R}^{n\times t}$ and $V \in \mathbb{R}^{t\times m}$, as long as $t \geq mn$.

In Appendix B.1 we provide a direct proof that allows us to bound the number of terms involved in the summation. The above lower bound mn on t may not be sharp. For example, if $\left\|\cdot\right\|_r = \left\|\cdot\right\|_c = \left\|\cdot\right\|_2$, then κ is the trace norm, where $\sum_{i=1}^{t}(\left\|U_{:i}\right\|_c^2 + \left\|V_{i:}\right\|_r^2)$ is simply $\left\|U\right\|_{\text{F}}^2 + \left\|V\right\|_{\text{F}}^2$, the sum of the squared Frobenius norms; that is, Proposition 15 yields the well-known variational form of the trace norm in this case (Srebro et al., 2005). Observe that t in this case can be as low as the rank of W. Theorem 15 also appeared in (Bach et al., 2008; Bach, 2013), although without the explicit bound $t \geq mn$.

Based on Theorem 15, the objective in (47) can then be *approximated* at iteration t as

$$\mathcal{F}_{t+1}(U,V) := \ell(UV) + \frac{\lambda}{2}\sum_{i=1}^{t+1}\left(\left\|U_{:i}\right\|_c^2 + \left\|V_{i:}\right\|_r^2 \right). \quad (53)$$

Our key intuition is to first factorize $\tilde{W}_{t+1}$ into $U_{\text{init}}V_{\text{init}}$, and then find improvement to (U,V) with respect to $\mathcal{F}_{t+1}$ (*not* the original F). This results in U_{t+1} and V_{t+1} satisfying $\mathcal{F}_{t+1}(U_{t+1},V_{t+1}) \leq \mathcal{F}_{t+1}(U_{\text{init}},V_{\text{init}})$. Finally we restore the iterates based on Theorem 15

$$W_{t+1} = U_{t+1}V_{t+1} \qquad \text{and} \qquad \rho_{t+1} = \frac{1}{2}\sum_{i=1}^{t+1}\left(\left\|(U_{t+1})_{:i}\right\|_c^2 + \left\|(V_{t+1})_{i:}\right\|_r^2 \right). \quad (54)$$

Using $\mathcal{F}_{t+1}$ as a surrogate objective for local improvement is particularly advantageous in computation, because the problem is now unconstrained and the gauge function has been replaced by much simpler forms such as $\left\|U_{:i}\right\|_c^2$ and $\left\|V_{i:}\right\|_r^2$. This allows efficient unconstrained minimization routines (e.g. L-BFGS) to be applied directly, improving the quality of the current iterate *without* increasing the size of the dictionary. Note however that (53) is not *jointly* convex in (U,V) since the size $t+1$ (the iteration counter) is fixed. When t is sufficiently large, any local minimizer of (53) *globally* solves the original problem (47) (Burer and Monteiro, 2005)[6], but we locally reduce (53) in each iteration even for small t.

Once this modification is added to Algorithm 2 we obtain the final variant, Algorithm 3. In this algorithm, Line 3 remains equivalent to before, and Line 4 adopts the slightly more sophisticated update rule in (40), although one can also use (34) with $\eta_t = \frac{2}{t+2}$. Lines 5 to 8 *collectively* implement the `Improve` subroutine: Line 5 splits the iterate into two parts

6. The catch here is that seeking a local minimizer can still be highly non-trivial, sometimes it is even hard to certify the local optimality.

Algorithm 3 GCG variant for low rank learning.

Require: The atomic set $\mathcal{A}$.

1: Initialize $W_0 = \mathbf{0}, \quad \rho_0 = 0, \quad U_0 = V_0 = \Lambda_0 = \emptyset$.
2: **for** $t = 0, 1, \ldots$ **do**
3: $\quad (\mathbf{u}_t, \mathbf{v}_t) \leftarrow \arg\min_{\mathbf{u}\mathbf{v}^\top \in \mathcal{A}} \left\langle \nabla \ell(W_t), \mathbf{u}\mathbf{v}^\top \right\rangle$
4: $\quad (\eta_t, \theta_t) \leftarrow \arg\min_{0 \le \eta \le 1, \theta \ge 0} \ell((1-\eta)W_t + \theta\, \mathbf{u}_t\mathbf{v}_t^\top) + \lambda((1-\eta)\rho_t + \theta)$
5: $\quad U_{\text{init}} \leftarrow (\sqrt{1-\eta_t}U_t, \sqrt{\theta_t}\mathbf{u}_t)$, $V_{\text{init}} \leftarrow (\sqrt{1-\eta_t}V_t^\top, \sqrt{\theta_t}\mathbf{v}_t)^\top$
6: $\quad (U_{t+1}, V_{t+1}) \leftarrow \texttt{Reduce}(\mathcal{F}_{t+1}, U_{\text{init}}, V_{\text{init}})$ s.t. $\mathcal{F}_{t+1}(U_{t+1}, V_{t+1}) \le \mathcal{F}_{t+1}(U_{\text{init}}, V_{\text{init}})$
$\quad$ e.g. find a local minimum of $\mathcal{F}_{t+1}$ with U and V initialized to U_{init} and V_{init} resp.
7: $\quad W_{t+1} \leftarrow U_{t+1}V_{t+1}$ ▷ Nominal as we only maintain (U_t, V_t) in practice
8: $\quad \rho_{t+1} \leftarrow \frac{1}{2}\sum_{i=1}^{t+1}(\|(U_{t+1})_{:i}\|_c^2 + \|(V_{t+1})_{i:}\|_r^2)$
9: **end for**

U_{init} and V_{init}, while Line 6 further reduces the reformulated local objective (53) with the designated initialization (e.g. by finding a local minimum). We referred to it as `Reduce` in order to be distinct from `Improve`. The last two steps restore the GCG iterate by (54).

The only major challenge left for inserting a local optimization step in Algorithm 2 is that the `Relaxed` condition (36) in Theorem 7 must be maintained. This is the condition under which Algorithm 3 retains the $O(1/t)$ rate of convergence established in Theorem 9. Fortunately this holds true as stated in Theorem 16 and its proof is given in Appendix B.2.

Proposition 16 *(U_{init}, V_{init}) in Line 5 is a factorization of the intermediate iterate $\tilde{W}_{t+1} = (1-\eta_t)W_t + \theta_t \mathbf{u}_t\mathbf{v}_t^\top$. Let (U_{t+1}, V_{t+1}) improve upon it in the sense that $\mathcal{F}_{t+1}(U_{t+1}, V_{t+1}) \le \mathcal{F}_{t+1}(U_{init}, V_{init})$. Then the W_{t+1} and ρ_{t+1} recovered by (54) satisfy the `Relaxed` condition.*

Although the `Reduce` subroutine is not guaranteed to achieve *strict* improvement in every iteration, we observe in our experiments in Section 5 below that Algorithm 3 is almost always significantly faster than the `Null` version of Algorithm 2. In other words, local acceleration appears to make a big difference in practice.

Remark 17 *Interleaving local improvement with a globally convergent procedure has been considered by Mishra et al. (2013) and Laue (2012). In particular, Laue (2012) considered the constrained problem* (21)*, which unfortunately leads to considering a constrained minimization for local improvement that can be less efficient than the unconstrained approach considered in* (53)*. In addition, Laue (2012) used the original objective F directly in local optimization. This leads to significant challenge when the regularizer κ is complicated such as the trace norm, and we avoid this issue by using a smooth surrogate objective $\mathcal{F}_{t+1}$.*

Focusing on trace norm regularization only, Mishra et al. (2013) proposed a trust-region procedure that locally minimizes the original *objective on the Stiefel manifold and the positive semidefinite cone, which unfortunately also requires constrained minimization and dynamic maintenance of the singular value decomposition of a small matrix. The rate of convergence of this procedure has not been analyzed.*

Remark 18 *We remark that local minimization of* (53) *constitutes the primary bottleneck of Algorithm 3, consuming over 50% of the computation. Since* (54) *is the same objective considered by the standard fixed rank approach to matrix factorization, the overall performance of Algorithm 3 can be substantially improved by leveraging the techniques that have been developed for this specific objective. These improvements include system level optimizations such as stochastic solvers on distributed, parallel and asynchronous infrastructures (Zhuang et al., 2013; Yun et al., 2014). In the experiments conducted in Section 5 below, we merely solve* (54) *using a generic L-BFGS solver to highlight the efficiency of the GCG framework itself. Nevertheless, such a study leaves significant room for further straightforward acceleration.*

Remark 19 *Much of our development in this section was inspired by the work of Bach et al. (2008), who first demonstrated, although in a slightly less clear way, that the convex relaxation in* (47) *is possible, but did not observe the computational connection to GCG. Our contribution here is to present this result succinctly using the notion of convex gauges, connect it naturally to the GCG algorithm we developed in Section 3, and propose an efficient fixed-rank local improvement subroutine `Reduce` that does not affect theoretical convergence guarantees. After our conference version (Zhang et al., 2012), Bach (2013) also independently extended the previous work (Bach et al., 2008) using convex gauges, and discussed a multiplicative approximate GCG variant similar to our Algorithm 2. However, the possibility to intervene fixed-rank local subroutines based on Theorem 15 with GCG, and its practical computational consequences, were not observed in (Bach, 2013).*

4.4 Multi-view Dictionary Learning

Finally, in this section we show how low rank learning can be generalized to a multi-view setting where, once again, the modified GCG algorithm can efficiently provide optimal solutions. The key challenge in this case is to develop column and row norms that can capture the more complex problem structure.

Consider a two-view learning task where one is given m paired observations $\left\{\begin{bmatrix}\mathbf{x}_j \\ \mathbf{y}_j\end{bmatrix}\right\}$ consisting of an x-view and a y-view with lengths n_1 and n_2 respectively. The goal is to infer a set of latent representations, $\mathbf{h}_j$ (of dimension $t < \min(n_1, n_2)$), and reconstruction models parameterized by the matrices A and B, such that $A\mathbf{h}_j \approx \mathbf{x}_j$ and $B\mathbf{h}_j \approx \mathbf{y}_j$ for all j. In particular, let X denote the $n_1 \times m$ matrix of x-view observations, Y the $n_2 \times m$ matrix of y-view observations, and $Z = \begin{bmatrix}X \\ Y\end{bmatrix}$ the concatenated $(n_1 + n_2) \times m$ matrix. The problem can then be expressed as recovering an $(n_1 + n_2) \times t$ matrix of model parameters, $C = \begin{bmatrix}A \\ B\end{bmatrix}$, and a $t \times m$ matrix of latent representations, H, such that $Z \approx CH$.

The key assumption behind multi-view learning is that each of the two views, $\mathbf{x}_j$ and $\mathbf{y}_j$, is conditionally independent given the shared latent representation, $\mathbf{h}_j$. Although multi-view data can always be concatenated hence treated as a single view, explicitly representing multiple views enables more accurate identification of the latent representation (as we will see), when the conditional independence assumption holds. To better motivate this model, it is enlightening to first reinterpret the classical formulation of multi-view dictionary learning, which is given by *canonical correlation analysis* (CCA). Typically, it is expressed as the problem of projecting two views so that the correlation between them is maximized. Assuming the data is centered (i.e. $X\mathbf{1} = \mathbf{0}$ and $Y\mathbf{1} = \mathbf{0}$), the sample covariance of X and Y is given by $XX^\top/m$ and $YY^\top/m$ respectively. CCA can then be expressed as an

optimization over matrix variables

$$\max_{U,V} \operatorname{tr}(U^\top XY^\top V) \qquad \text{s.t.} \qquad U^\top XX^\top U = V^\top YY^\top V = I \tag{55}$$

for $U \in \mathbb{R}^{n_1 \times t}$, $V \in \mathbb{R}^{n_2 \times t}$ (De Bie et al., 2005). Here I is the identity matrix whose size is clear from the context. Although this classical formulation (55) does not make the shared latent representation explicit, CCA can be expressed by a generative model: given a latent representation, $\mathbf{h}_j$, the observations $\mathbf{x}_j = A\mathbf{h}_j + \boldsymbol{\epsilon}_j$ and $\mathbf{y}_j = B\mathbf{h}_j + \boldsymbol{\nu}_j$ are generated by a linear mapping plus independent zero mean Gaussian noise, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \Sigma_x)$, $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \Sigma_y)$ (Bach and Jordan, 2006). Interestingly, one can show that the classical CCA problem (55) is equivalent to the following multi-view dictionary learning problem.

Proposition 20 (White et al. 2012, Proposition 1) *Fix t, let*

$$(A, B, H) = \arg \min_{A:\|A_{:i}\|_2 \le 1} \min_{B:\|B_{:i}\|_2 \le 1} \min_{H} \left\| \begin{bmatrix} (XX^\top)^{-1/2} X \\ (YY^\top)^{-1/2} Y \end{bmatrix} - \begin{bmatrix} A \\ B \end{bmatrix} H \right\|_{\mathrm{F}}^2 . \tag{56}$$

Then $U = (XX^\top)^{-\frac{1}{2}} A$ and $V = (YY^\top)^{-\frac{1}{2}} B$ provide an optimal solution to (55).

Proposition 20 demonstrates how formulation (56) respects the conditional independence of the separate views: given a latent representation $\mathbf{h}_j$, the reconstruction losses on the two views, $\mathbf{x}_j$ and $\mathbf{y}_j$, cannot influence each other, since the reconstruction models A and B are *individually* constrained. By contrast, in single-view dictionary learning (i.e. *principal component analysis*) A and B are concatenated in the larger variable $C = \left[\begin{smallmatrix} A \\ B \end{smallmatrix}\right]$, where C as a whole is constrained but A and B are not. A and B must then compete against each other to acquire magnitude to explain their respective "views" given $\mathbf{h}_j$ (i.e. conditional independence is not enforced). Such sharing can be detrimental if the two views really are conditionally independent given $\mathbf{h}_j$.

This matrix factorization viewpoint has recently allowed dictionary learning to be extended from the single-view setting to the multi-view setting (Jia et al., 2010). However, despite its elegance, the computational tractability of CCA hinges on its restriction to squared loss under a particular normalization, precluding other important losses. To combat these limitations, we apply the same convex relaxation principle as in Section 4.2. In particular, we reformulate (56) by first incorporating an arbitrary loss function ℓ that is convex in its first argument, and then relaxing the rank constraint by replacing it with the $\mathfrak{l}_1$ norm of the "singular values". This amounts to the following training problem that extends (Jia et al., 2010):

$$\min_{A,B,H,\boldsymbol{\sigma}} \ell\left(\begin{bmatrix} A \\ B \end{bmatrix} \operatorname{diag}(\boldsymbol{\sigma}) H; Z \right) + \lambda \|\boldsymbol{\sigma}\|_1 , \quad \text{s.t.} \quad \|H_{i:}\|_2 \le 1, \quad \sigma_i \ge 0, \quad \begin{bmatrix} A_{:i} \\ B_{:i} \end{bmatrix} \in \mathcal{D} \; \forall \, i,$$
$$\text{where} \quad \mathcal{D} := \left\{ \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} : \|\mathbf{a}\|_2 \le \beta, \; \|\mathbf{b}\|_2 \le \gamma \right\}. \tag{57}$$

The $\mathfrak{l}_2$ norm on the *rows* of H is a regularizer that encourages the rows to become sparse, hence reducing the size of the learned representation (Argyriou et al., 2008). Since the size of the latent representation (the number of rows of H) is not fixed, our results in Section 4.2

immediately allows the problem to be solved globally and efficiently for A, B and H. This considerably improves the previous local solutions in (Jia et al., 2010) which fixed the latent dimension and approached the non-convex problem by alternating block coordinate descent between A, B and $\operatorname{diag}(\boldsymbol{\sigma})H$.

Indeed, (57) fits exactly in the framework of (45), with $\|\cdot\|_r$ being the $\mathfrak{l}_2$ norm and

$$\|\mathbf{c}\|_c = \max\left(\frac{1}{\beta} \|\mathbf{a}\|_2 , \frac{1}{\gamma} \|\mathbf{b}\|_2 \right) \quad \text{for} \quad \mathbf{c} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}. \tag{58}$$

Next we will show how to efficiently compute the polar operator (50) that underpins Algorithm 3 in this multi-view model.

4.4.1 Efficient Polar Operator

The polar operator (50) for problem (57) turns out to admit an efficient but nontrivial solution. For simplicity, we assume $\beta = \gamma = 1$ while the more general case can be dealt with in exactly the same way. The proof of the following key theorem is in Appendix B.3.

Proposition 21 *Let $I_1 := \operatorname{diag}(\left[\begin{smallmatrix} 1 \\ 0 \end{smallmatrix}\right])$ and $I_2 := \operatorname{diag}(\left[\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right])$ be the identity matrices on the subspaces spanned by x-view and y-view, respectively. For any $\mu > 0$, denote $D_\mu = \sqrt{1+\mu} I_1 + \sqrt{1+1/\mu} I_2$ as a diagonal scaling of the two identity matrices. Then, for the row norm $\|\cdot\|_r = \|\cdot\|_2$ and column norm $\|\cdot\|_c$ defined as in (58), the corresponding polar (49) is given by*

$$\kappa^\circ(G) = \min_{\mu \ge 0} \|D_\mu G\|_{\mathrm{sp}} . \tag{59}$$

Despite its variational form, the polar (59) can still be efficiently computed by considering its squared form and expanding:

$$\|D_\mu G\|_{\mathrm{sp}}^2 = \left\| G^\top D_\mu D_\mu G \right\|_{\mathrm{sp}} = \left\| G^\top \left[(1+\mu) I_1 + (1+1/\mu) I_2 \right] G \right\|_{\mathrm{sp}} .$$

Clearly, the term inside the spectral norm $\|\cdot\|_{\mathrm{sp}}$ is convex in μ on $\mathbb{R}_+$ while the spectral norm is convex and increasing on the cone of positive semidefinite matrices, hence by the usual rules of convex composition the left hand side above is a *convex* function of μ. Consequently we can use a subgradient algorithm, or even golden section search, to find the optimal μ in (59). Given an optimal μ, an optimal pair of "atoms" $\mathbf{a}$ and $\mathbf{b}$ can be recovered via the KKT conditions; see Appendix B.4 for details. Thus, by using (59) Algorithm 3 is able to solve the resulting optimization problem far more efficiently than the initial procedure offered by White et al. (2012).

Similar to the case of matrix completion for single view in Section 4.2.1, the L_F in Theorem 5 for this multi-view setting can be bounded in exactly the same way as in (52). The only difference is that instead of $\|W\|_{\mathrm{tr}} \ge \|W\|_{\mathrm{F}}$, we now have $\kappa(W) \ge \frac{1}{\sqrt{2}} \|W\|_{\mathrm{F}}$. To see it, just set $\mu = 1$ in (59), and we immediately obtain $\kappa^\circ(G) \le \sqrt{2} \|G\|_{\mathrm{sp}}$. Dualizing both sides leads to $\kappa(W) \ge \frac{1}{\sqrt{2}} \|W\|_{\mathrm{tr}} \ge \frac{1}{\sqrt{2}} \|W\|_{\mathrm{F}}$. Therefore we can proceed in analogy to (52):

$$L_F \le 2L \sup_t \left\{ \frac{\rho^2}{\alpha^2} \|A_t\|_{\mathrm{F}}^2 + \|W_t\|_{\mathrm{F}}^2 \right\} \le 2L \sup_t \left\{ \frac{2\rho^2}{\alpha^2} + 2\kappa(W_t)^2 \right\} \le \frac{4}{\|\mathcal{P}\|_0} \left(\frac{1}{\alpha^2} + 1 \right) \frac{C^2}{\lambda^2}. \tag{60}$$

4.4.2 A Modified Power Iteration

In practice, one can compute the polar (59) much more efficiently by deploying a heuristic variant of the power iteration method, whose motivation arises from the KKT conditions in the proof of Theorem 21 (see (76) in Appendix B.4). In particular, two normalized vectors $\mathbf{a}$ and $\mathbf{b}$ are optimal for the polar (50) if and only if there exist nonnegative scalars μ_1 and μ_2 such that

$$\begin{bmatrix} \mu_1 \mathbf{a} \\ \mu_2 \mathbf{b} \end{bmatrix} = GG^\top \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \qquad \text{and} \qquad \mu_1 I_1 + \mu_2 I_2 \succeq GG^\top. \tag{61}$$

Here $A \succeq B$ asserts that $A - B$ is positive semidefinite. Therefore, after randomly initializing $\mathbf{a}$ and $\mathbf{b}$ with unit norm, it is natural to extend the power iteration method by repeating the update:

$$\begin{bmatrix} \mathbf{s} \\ \mathbf{t} \end{bmatrix} \leftarrow GG^\top \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix}, \quad \text{then} \quad \begin{bmatrix} \mathbf{a}^{\text{new}} \\ \mathbf{b}^{\text{new}} \end{bmatrix} \leftarrow \begin{bmatrix} \frac{1}{\|\mathbf{s}\|_2}\mathbf{s} \\ \frac{1}{\|\mathbf{t}\|_2}\mathbf{t} \end{bmatrix}.$$

This procedure usually converges to a solution where the Lagrange multipliers μ_1 and μ_2, recovered from (61), satisfy (61) hence confirming their optimality. While we leave the investigation of the theoretical guarantees of this power iteration heuristic for future work, let us mention that it works very well in practice (as in the single view case) and when the occasional failure occurs we revert to (59) and follow the recovery rule detailed in Appendix B.4.

5. Experimental Evaluation

We evaluate the extended GCG approach developed above in three sparse learning problems: matrix completion, multi-class classification, and multi-view dictionary learning. All algorithms were implemented in Matlab unless otherwise noted.[7] All experiments were run on a single core of a cluster housed at National ICT Australia with AMD 6-Core Opteron 4184 (2.8GHz, 3M L2/6M L3 Cache, 95W TDP, 64 GB memory).

5.1 Matrix Completion with Trace Norm Regularization

Our first set of experiments is on matrix completion (51), where the division by $\|\mathcal{P}\|_0$ is omitted to be consistent with existing solvers. Five data sets are used and their statistics are given in Table 1. The data sets `Yahoo252M` (Dror et al., 2012) and `Yahoo60M`[8] provide music ratings, and the data sets `Netflix` (Dror et al., 2012), `MovieLens1M`, and `MovieLens10M`[9] all provide movie ratings. For `MovieLens1M` and `Yahoo60M`, we randomly sampled 75% ratings as training data, using the rest for testing. The other three data sets came with training and testing partition.

We compared the extended GCG approach (Algorithm 3) with three state-of-the-art solvers for trace norm regularized objectives: MMBS[10] (Mishra et al., 2013), DHM (Dudik et al., 2012), and JS (Jaggi and Sulovsky, 2010). The local optimization in GCG was performed by L-BFGS with the maximum number of iteration set to 20.[11] Traditional solvers such as proximal methods (Toh and Yun, 2010; Pong et al., 2010) were not included in this comparison because they are much slower.

7. All code is available for download at `https://www.cs.uic.edu/~zhangx/GCG`.
8. `Yahoo60M` is the R1 data set at `http://webscope.sandbox.yahoo.com/catalog.php?datatype=r`.
9. Both MovieLens1M and MovieLens10M were downloaded from `http://www.grouplens.org/datasets/movielens`.
10. Downloaded from `http://www.montefiore.ulg.ac.be/~mishra/softwares/traceNorm.html`.

data set	n (#user)	m (#product)	#train	#test	λ
`MovieLens1M`	6,040	3,952	750,070	250,024	50
`MovieLens10M`	71,567	65,133	9,301,174	698,599	100
`Netflix`	2,549,429	17,770	99,072,112	1,408,395	1
`Yahoo60M`	98,130	1,738,442	60,021,577	20,872,219	300
`Yahoo252M`	1,000,990	624,961	252,800,275	4,003,960	100

Table 1: Summary of the data sets used in matrix completion experiments. Here n is the total number of users, m is the number of products (movies or music), #train and #test are the number of ratings used for training and testing respectively.

MMBS is confined to trace norm regularized problems. At each iteration t, they solve

$$\min_{U \in \mathbb{R}^{n\times t},\ B \in \mathbb{R}^{t\times t},\ V \in \mathbb{R}^{t\times m}} \ell(UBV) + \lambda \operatorname{tr}(B), \quad \text{s.t. } U^\top U = VV^\top = I, \text{ and } B \succeq \mathbf{0}. \tag{62}$$

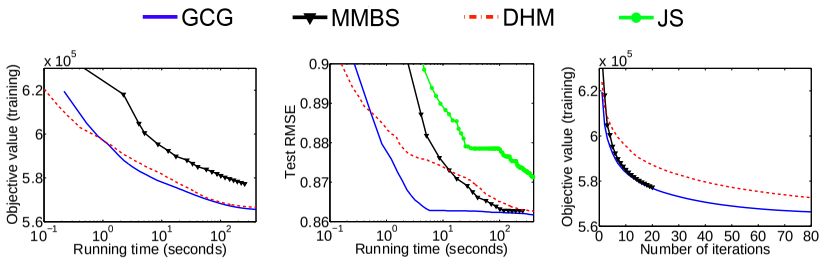
After obtaining a stationary point, the gradient of ℓ is computed, with its top left and right singular vectors appended to U and V respectively, and repeat. Unfortunately, the optimization in (62) is highly challenging on the Stiefel manifold, and their algorithm relies on the directional Hessian which becomes rather complicated in computation for general ℓ. By contrast, the local surrogate (53) used in our approach is substantially easier to optimize.

DHM is a special case of GCG, with `Improve` instantiated with a line search procedure (Dudik et al., 2012). The detail of line search is intricate, but in practice it is slower than totally corrective updates (42) solved by L-BFGS. To further streamline the optimization for (42), we buffered the quantities $\langle \mathcal{P}(\mathbf{u}_t\mathbf{v}_t'), \mathcal{P}(\mathbf{u}_s\mathbf{v}_s') \rangle$ and $\langle \mathcal{P}(\mathbf{u}_t\mathbf{v}_t'), \mathcal{P}(X) \rangle$, where $(\mathbf{u}_t, \mathbf{v}_t)$ is the result of polar operator at iteration t (Line 3 of Algorithm 3). This allowed the objective (42) to be evaluated efficiently without having to compute matrix inner product.

JS was proposed for solving the constrained problem: $\min_W \ell(W)$ s.t. $\|W\|_{\text{tr}} \leq \zeta$, which makes it hard to directly compare with solvers for the penalized problem (51). As a workaround, given λ we first found the optimal solution W^* for the penalized problem, and then we set $\zeta = \|W^*\|_{\text{tr}}$. To solve the constrained problem, JS lifts it to an SDP by observing that $\|W\|_{\text{tr}} \leq \frac{t}{2}$ iff there exist symmetric matrices A and B such that $Z := \begin{pmatrix} A & W \\ W^\top & B \end{pmatrix} \succeq \mathbf{0}$ and $\operatorname{tr}(A) + \operatorname{tr}(B) = t$. Then the objective $\ell(W)$ can be rewritten in terms of Z, solved by the conventional conditional gradient algorithm for constrained problems.

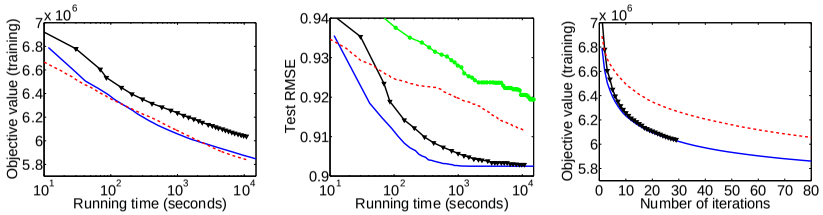
Figures 2 to 6 show how quickly the various algorithms drive the training objective down, while also providing the root mean square error (RMSE) on test data (i.e. comparing the reconstructed matrix each iteration to the ground truth). JS is comparable only on RMSE. The regularization constant λ, given in Table 1, was chosen from

11. Downloaded from `http://www.cs.ubc.ca/~pcarbo/lbfgsb-for-matlab.html`. Recall a suboptimal solution is sufficient for the fixed-rank local optimization in `Relaxed`.



(a) Objective value vs CPU time (b) Test RMSE vs CPU time (c) Objective value vs #iteration

Figure 2: Matrix completion on `MovieLens1M`



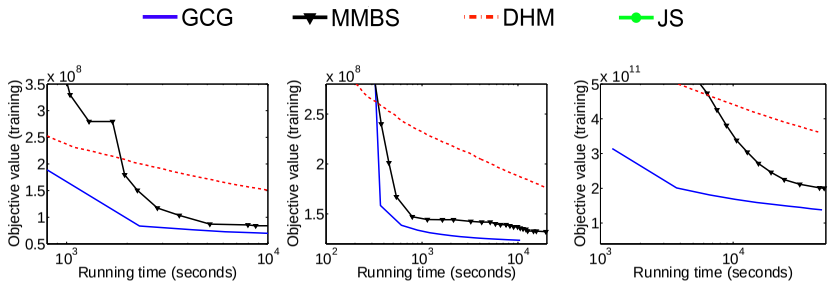
(a) Objective value vs CPU time (b) Test RMSE vs CPU time (c) Objective value vs #iteration

Figure 3: Matrix completion on `MovieLens10M`

$\{1, 10, 50, 100, 200, 300, 400, 500, 1000\}$ to minimize the test RMSE. On the `MovieLens1M` and `MovieLens10M` data sets, GCG and DHM exhibit similar efficiency in reducing the objective value with respect to CPU time (see Figure 2(a) and 3(a)). However, GCG achieves comparable test RMSE an order of magnitude faster than DHM, as shown in Figure 2(b) and 3(b). Interestingly, this discrepancy can be explained by investigating the rank of the solution, i.e. the number of outer iterations taken. In Figure 2(c) and 3(c), DHM clearly converges much more slowly in terms of the number of iterations, which is expected because it does not re-optimize the basis at each iteration, unlike GCG and MMBS. Therefore, although the overall computational cost for DHM appears on par with that for GCG, its solution has a much higher rank, resulting in a much higher test RMSE in these cases.

MMBS takes nearly the same number of iterations as GCG to achieve the same objective value (see Figure 2(c) and 3(c)). However, its local search at each iteration is conducted on a *constrained* manifold, making it much more expensive than locally optimizing the unconstrained smooth surrogate (53). As a result, the overall computational complexity of MMBS is an order of magnitude greater than that of GCG, with respect to optimizing both the objective value and the test RMSE.

The observed discrepancies between the relative efficiency of optimizing the objective value and test RMSE becomes much smaller on the other three data sets, which are much larger in size. In Figure 4 to 6, it is clear that GCG takes significantly less CPU time to

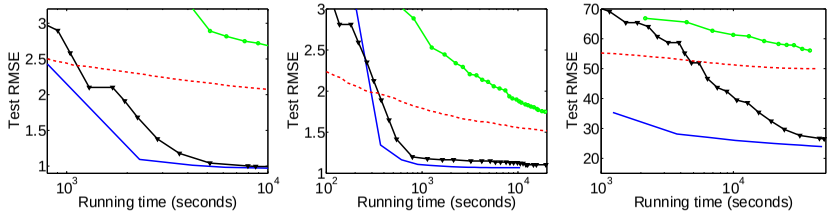


(a) Objective value vs CPU time

(a) Objective value vs CPU time

(a) Objective value vs CPU time

Figure 4: Matrix completion on `Netflix`



(b) Test RMSE vs CPU time

(b) Test RMSE vs CPU time

(b) Test RMSE vs CPU time

Figure 5: Matrix completion on `Yahoo100m`

Figure 6: Matrix completion on `Yahoo252m`

optimize both criteria. The declining trend of the objective value has a similar shape to that of the test RMSE. Here we observe that DHM is substantially slower than GCG and MMBS, confirming the importance of optimizing the bases at the same time as optimizing their weights. JS, which does not exploit the penalized form of the objective, is considerably slower with respect to reducing the test RMSE.

5.2 Multi-class Classification with Trace Norm Regularization

Next we compared the four algorithms on multi-class classification problems also in the context of trace norm regularization. Here the task is to predict the class membership of input instances, $\mathbf{x}_i \in \mathbb{R}^n$, where each instance is associated with a unique label $y_i \in \{1, \ldots, C\}$. In particular, we consider a standard linear classification model where each class c is associated with a weight vector in $\mathbb{R}^n$, stacked in a model matrix $W \in \mathbb{R}^{n \times C}$. Then for each training example $(\mathbf{x}_i, y_i)$, the individual logistic loss $L(W; \mathbf{x}_i, y_i)$ is given by

$$L(W; \mathbf{x}_i, y_i) = -\log \frac{\exp(\mathbf{x}_i' W_{:,y_i})}{\sum_{c=1}^{C} \exp(\mathbf{x}_i' W_{:c})},$$

data set	n	C	#train	#test	λ
Fungus10	4,096	10	10 per class	10 per class	10^{-2}
Fungus134	4,096	134	50 per class	50 per class	10^{-3}
k1024	131,072	50	50 per class	50 per class	10^{-3}

Table 2: Summary of the data sets used in multi-class classification experiments. Here n is the number of features, C is the number of classes, #train and #test are the number of training and test images respectively.

and the complete loss is given by averaging over the training set, $\ell(W) := \frac{1}{m}\sum_{i=1}^{m} L(W; \mathbf{x}_i, y_i)$. It is reasonable to assume that W has a low rank; i.e. the weight vector of all classes lie in a low dimensional subspace (Akata et al., 2014), which motivates the introduction of a trace norm regularizer on W, yielding a training problem in the form of (47).

We conducted experiments on three data sets extracted from the ImageNet repository (Deng et al., 2009), with characteristics shown in Table 2 (Akata et al., 2014).[12] k1024 is from the ILSVRC2011 benchmark, and Fungus10 and Fungus134 are from the Fungus group. All images are represented by the Fisher descriptor.

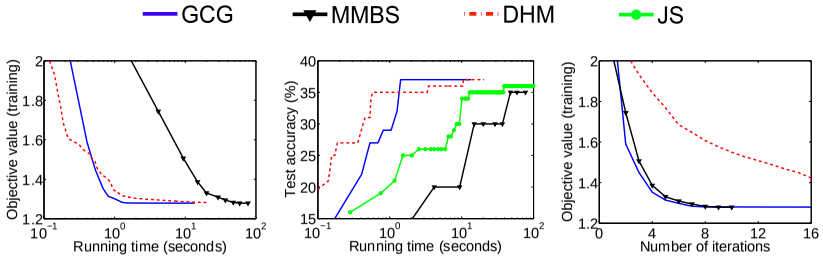
As in the matrix completion case above, we again compared how fast the algorithms reduced the objective value and test accuracy. Here the value of λ was chosen from $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ to maximize test accuracy. Figures 7 to 9 show that the models produced by GCG achieve comparable (or better) training objectives and test accuracies in far less time than MMBS, DHM and JS. Although DHM is often more efficient than GCG early on, it is soon overtaken. This observation could be related to the diminishing return of adding bases, as shown in Figure 7(c), 8(c), and 9(c), since the addition of the first few bases yields a much steeper decrease in objective value than that achieved by later bases. Hence, after the initial stage, the computational efficiency of a solver turns to be dominated by better exploitation of the bases. That is, using added bases as an initialization for local improvement (as in GCG) appears to be more effective than simply finding an optimal conic combination of added bases (as in DHM).

Also notice from Figures 7(c), 8(c) and 9(c) that MMBS and GCG are almost identical in terms of the number of iterations required to achieve a given objective value. However, the manifold based local optimization employed by MMBS at each iteration makes its overall computational time worse than that of DHM and JS. This is a different outcome from the matrix completion experiments we saw earlier, and seems to suggest that the manifold approach in MMBS is more effective for the least square loss than for the logistic loss.

5.3 Multi-view Dictionary Learning

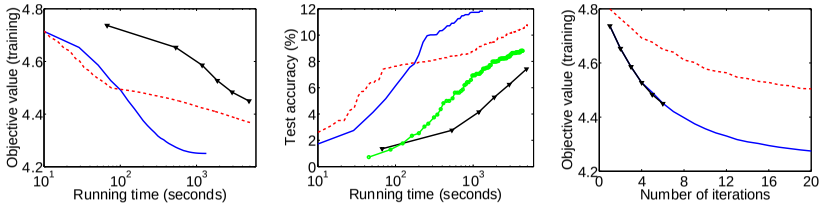
Our third set of experiments focuses on optimizing the objective for the multi-view dictionary learning model (57) presented in Section 4.4. Here we compared the GCG approach outlined in Section 4.4 with a straightforward *local* solver that is based on block coordinate descent (BCD), which alternates between: (a) fixing H and optimizing A and B (with norm

12. The three data sets were all downloaded from http://lear.inrialpes.fr/src/jsgd/.



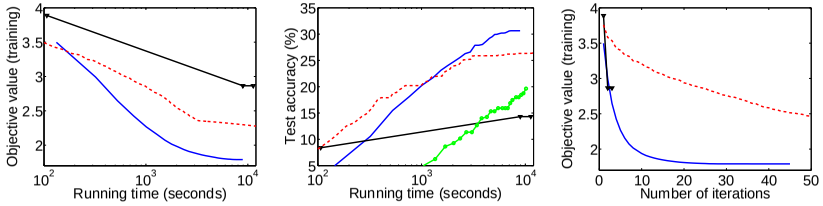
(a) Objective value vs CPU time (b) Test accuracy vs CPU time (c) Test accuracy vs #iteration

Figure 7: Multi-class classification on Fungus10



(a) Objective value vs CPU time (b) Test accuracy vs CPU time (c) Objective value vs #iteration

Figure 8: Multi-class classification on Fungus134



(a) Objective value vs CPU time (b) Test accuracy vs CPU time (c) Objective value vs #iteration

Figure 9: Multi-class classification on k1024

ball constraints), then (b) fixing A and B and optimizing H (with nonsmooth regularizer). Both steps were solved by FISTA (Beck and Teboulle, 2009), initialized with the solution from the previous iteration. We tested several values of t, the dimensionality of the latent subspace of the solution, in these experiments. The GCG approach used Algorithm 3 with the local objective (53) instantiated by the norm expression in (58) and solved by L-BFGS with max iteration set to 20. We initialized the U and V matrices with 20 columns and rows respectively, each drawn i.i.d. from a uniform distribution on the unit sphere. Based

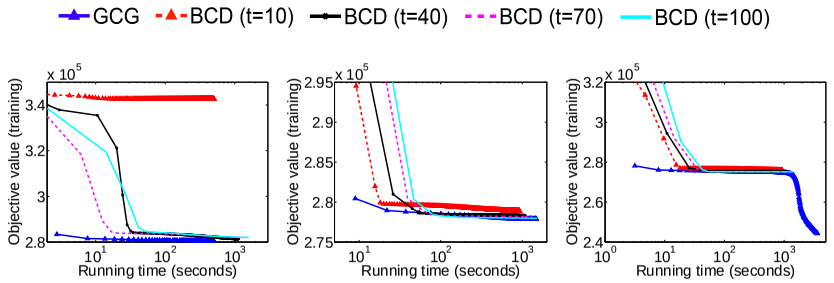
on the noise model we consider in these experiments, we started with a base loss ℓ set to $\mathfrak{l}_1(Z, Z^*) = \sum_{ij} |Z_{ij} - Z^*_{ij}|$ where Z^* is the true underlying matrix; however, following (Becker et al., 2009, Eq 3.1), we smoothed this $\mathfrak{l}_1$ loss by adding a quadratic prox-function with strong convexity modulus 10^{-4}. Under the resulting loss, the optimization over A and B is decoupled given H. We terminated all algorithms after a maximum of running time (3,600 seconds) is reached.

5.3.1 Synthetic Data

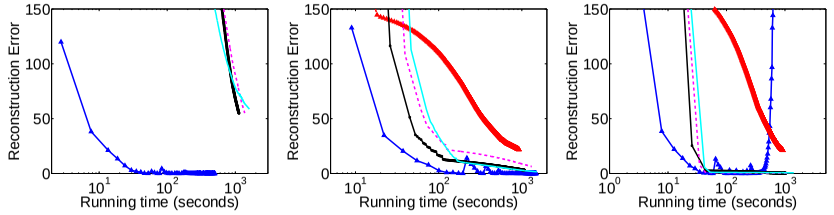
We first constructed experiments on a synthetic data set that fulfills the conditional independence assumption (57) of the multi-view model (as discussed in Section 4.4). Here the x- and y-views have basis matrices $A^* \in \mathbb{R}^{n_1 \times t^*}$ and $B^* \in \mathbb{R}^{n_2 \times t^*}$ respectively, where all columns of A^* were drawn i.i.d. from the $\mathfrak{l}_2$ sphere of radius $\beta = 1$, and all columns of B^* were drawn i.i.d. from the $\mathfrak{l}_2$ sphere of radius $\gamma = 5$. We set $n_1 = 800$, $n_2 = 1000$, and $t^* = 10$ (so that the dictionary size is indeed small). The latent representation matrix $H^* \in \mathbb{R}^{t^* \times m}$ has all elements drawn i.i.d. from the zero-mean unit-variance Gaussian distribution. The number of training examples is set to $m = 200$, which is much lower than the number of features n_1 and n_2. Then the clean versions of x-view and y-view were generated by $X^* = A^*H^*$ and $Y^* = B^*H^*$ respectively, while the noisy versions were obtained by adding i.i.d. noise to 15% entries of X^* and Y^* that were selected uniformly at random, yielding $\hat{X}$ and $\hat{Y}$ respectively. The noise is uniformly distributed in $[0, 10]$. Given the noisy observations, the denoising task is to find reconstructions X and Y for the two views, such that the error $\|X - \hat{X}\|_{\mathrm{F}}^2 + \|Y - \hat{Y}\|_{\mathrm{F}}^2$ is small. The composite training problem is formulated in (57).

The results of comparing the modified GCG method with local optimization are presented in Figures 10 to 12, where the regularization parameter λ has been varied among $\{80, 60, 40\}$. In each case, GCG optimizes the objective value significantly faster than BCD for all values of t chosen for BCD. The differences between methods are even more evident when considering the reconstruction error. In particular, although the local optimization in GCG already achieves a low objective value in the first outer iteration (see Figure 10(a) and 11(a)), the reconstruction error remains high, thus requiring further effort to reduce it.

When $t = 10$ which equals the true rank $t^* = 10$, BCD is more likely to get stuck in a poor local minimum when λ is large, for example as shown in Figure 10(a) for $\lambda = 80$. This occurs because when optimizing H, the strong shrinkage in the proximal update stifles major changes in H (and in A and B consequently). This problem is exacerbated when the rank of the solution is overly restricted, leading to even worse local optima. On the other hand, under-regularization does indeed cause overfitting in this case; for example, as shown in Figure 12 for $\lambda = 40$. Interestingly, BCD gets trapped in local optima for all t, while GCG eventually escapes suboptimal solutions and eventually finds a way to considerably reduce the objective value. However, this simply creates a dramatic *increase* in the test reconstruction error, a typical phenomenon of overfitting.



(a) Objective value vs CPU time (a) Objective value vs CPU time (a) Objective value vs CPU time



(b) Reconst. error vs CPU time (b) Reconst. error vs CPU time (b) Reconst. error vs CPU time

Figure 10: Denoising on synthetic data $\lambda = 80$

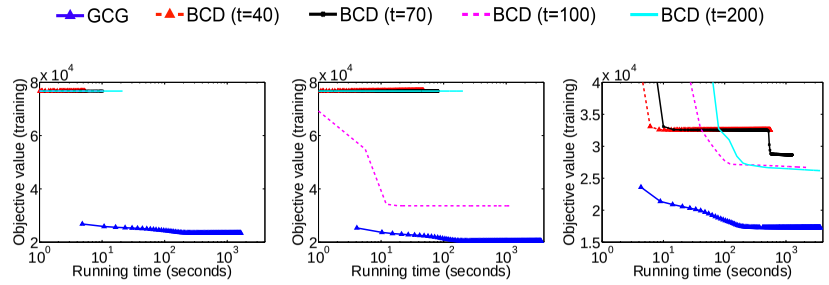
Figure 11: Denoising on synthetic data $\lambda = 60$

Figure 12: Denoising on synthetic data $\lambda = 40$

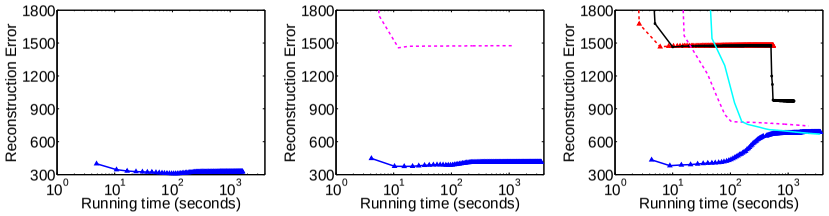
5.3.2 Image Denoising

We then conducted multi-view dictionary learning experiments on natural image data. The data set we considered is based on the Extended Yale Face Database B[13] (Georghiades et al., 2001). In particular, the data set we used consists of grey level face images of 28 human subjects, each with 9 poses and 64 lighting conditions. To construct the data set, we set the x-view to a fixed lighting (+000E+00) and the y-view to a different fixed lighting (+000E+20). We obtained paired views by randomly drawing a subject and a pose under the two lighting conditions. The underlying assumption is that each lighting has its own set of bases (A and B) and each (person, pose) pair has the same latent representation for the two lighting conditions. All images were down-sampled to 50-by-50, meaning $n_1 = n_2 = 2500$, and we randomly selected $t = 50$ pairs of (person, pose) for training. The x-view was kept clean, while pixel errors were added to the y-view, randomly setting 10% of the pixel values to 1. This noise model mimics natural phenomena such as occlusion and loss of pixel information from image transfer. The goal is again to enable appropriate reconstruction of

13. Data downloaded from `http://vision.ucsd.edu/content/extended-yale-face-database-b-b`.



(a) Objective value vs CPU time



(b) Reconst. error vs CPU time

Figure 13: Denoising on face data with $\lambda = 40$

(a) Objective value vs CPU time

(b) Reconst. error vs CPU time

Figure 14: Denoising on face data with $\lambda = 30$

(a) Objective value vs CPU time

(b) Reconst. error vs CPU time

Figure 15: Denoising on face data with $\lambda = 20$

a noisy image using other views. Naturally, we set $\beta = \gamma = 1$ since the data is in the $[0, 1]$ interval.

Figures 13 to 15 show the results comparing GCG with the local BCD for different choices of t for BCD. Clearly the value of λ that yields the lowest reconstruction error is 40. However, here BCD simply gets stuck at the trivial local minimum $H = \mathbf{0}$, as shown in Figure 13(a). This occurs because under the random initialization of A and B, the proximal update simply sets H to $\mathbf{0}$ when the regularization is too strong. When λ is decreased to 30, as shown in Figure 14(a), BCD can escape the trivial $H = \mathbf{0}$ solution for $t = 100$, but again it becomes trapped in another local minimum that remains much worse than the globally optimal solution found by GCG. When λ is further reduced to 20, the BCD algorithms finally escape the trivial $H = \mathbf{0}$ point for all t, but still fail to find an acceptable solution. Although GCG again converges to a much better global solution in terms of the objective value, overfitting can still occur: in Figure 15(b) the reconstruction error of GCG increases significantly after an initial decline.

6. Conclusion

We have presented a unified treatment of the generalized conditional gradient (GCG) algorithm that covers several useful extensions. After a thorough treatment of its convergence properties, we illustrated how an extended form of GCG can efficiently handle a convex gauge regularizer in particular. We illustrated the application of these GCG methods in learning low rank matrices, instantiated with examples on matrix completion, dictionary learning, and multi-view learning. In each case, we considered a generic convex relaxation procedure and focused on the efficient computation of the polar operator—one of the key steps to obtaining an efficient implementation of GCG. To further accelerate its convergence, we interleaved the standard GCG update with a fixed rank subspace optimization, which greatly improves performance in practice without affecting the theoretical convergence properties. Extensive experiments on matrix completion, multi-class classification, multi-view learning confirmed the superiority of the modified GCG approach in practice.

For future work, several interesting extensions are worth investigating. The current form of GCG is inherently framed as a batch method, where each iteration requires gradient of the loss to be computed over *all* training data. When the training data is large, it is preferable to only use a random subset, or distribute the training data across several processors. Another interesting direction is to interpolate between the polar operator and proximal updates; that is, one could incorporate a mini-batch of atoms at each iteration. For example, for trace norm regularization, one can consider adding k instead of just the 1 largest singular vectors. It remains an open question whether such a "mini-batch" of atoms can provably accelerate the overall optimization, but some recent work (Hsieh and Olsen, 2014) has demonstrated promise in practice.

Acknowledgments

This research was supported by the Alberta Innovates Center for Machine Learning (AICML), the Canada Research Chairs program, and NSERC. Most of this work was carried out when Y Yu and X Zhang were with the University of Alberta and AICML. X Zhang would like to thank the support of NICTA, where he worked as a researcher from Oct 2012 to Oct 2015 and conducted 50% of his work on this paper. We thank Csaba Szepesvári, the action editor, and the reviewers at various stages of this work for the many constructive comments.

Appendix A. Proofs for Section 3

A.1 Proof of Theorem 2

Since $\nabla \ell$ is Lipschitz continuous, it must be uniformly continuous on bounded set.

(a): Let C be the closure of the sequence $\{\mathbf{w}_t\}_t$. Due to the compactness assumption on the sublevel set and the monotonicity of $F(\mathbf{w}_t)$, C is compact. Moreover $C \subseteq \operatorname{dom} f$ since for all cluster points, say $\mathbf{w}$, of $\mathbf{w}_t$ we have from the closedness of F that $F(\mathbf{w}) \leq \liminf F(\mathbf{w}_{t_k}) \leq F(\mathbf{w}_0) < \infty$. Since $-\nabla \ell$ is continuous, $-\nabla \ell(C)$ is a compact subset of $\operatorname{int}(\operatorname{dom} f^*)$, the interior of $\operatorname{dom} f^*$. Note that f^* is continuous on the interior of its domain, therefore its subdifferential is locally bounded on $-\nabla \ell(C)$, see e.g. (Borwein and

Vanderwerff, 2010, Proposition 4.1.26). A standard compactness argument then establishes the boundedness of $(\partial f^*)(-\nabla \ell(C))$. Thus $\{\mathbf{d}_t\}_t$ is bounded.

(b): Note first that the boundedness of $\{\mathbf{w}_t\}_t$ follows immediately from the boundedness assumption on the sublevel set, thanks to the monotonic property of $F(\mathbf{w}_t)$. Since $\nabla \ell$ is uniformly continuous, the set $\{-\nabla \ell(\mathbf{w}_t)\}_t$ is again bounded. On the other hand, we know from (Borwein and Vanderwerff, 2010, Theorem 4.4.13, Proposition 4.1.25) that f is supercoercive iff ∂f^* maps bounded sets into bounded sets. Therefore $\{\mathbf{d}_t\}_t$ is again bounded.

(c): This is clear.

Note that the convexity of ℓ is not used in this proof.

A.2 Proof of Theorem 4

Since ℓ is L-smooth, for the sequences $\{\mathbf{w}_t\}$ and $\{\mathbf{d}_t\}$ generated by Algorithm 1, and for all $\eta \in [0,1]$, we have

$$\ell(\mathbf{w}_t + \eta(\mathbf{d}_t - \mathbf{w}_t)) \leq \ell(\mathbf{w}_t) + \eta \left\langle \mathbf{d}_t - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \right\rangle + \tfrac{L\eta^2}{2} \left\| \mathbf{d}_t - \mathbf{w}_t \right\|^2. \tag{63}$$

Since the subroutine is `Relaxed` and the subproblem (13) is solved up to some $\varepsilon_t \geq 0$,

$$\begin{aligned} F(\mathbf{w}_{t+1}) &\leq \ell(\mathbf{w}_t) + \eta_t \left\langle \mathbf{d}_t - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \right\rangle + \tfrac{L\eta_t^2}{2} \left\| \mathbf{d}_t - \mathbf{w}_t \right\|^2 + (1-\eta_t) f(\mathbf{w}_t) + \eta_t f(\mathbf{d}_t) \\ &\leq F(\mathbf{w}_t) - \eta_t \mathsf{G}(\mathbf{w}_t) + \tfrac{L\eta_t^2}{2} \left\| \mathbf{d}_t - \mathbf{w}_t \right\|^2 + \eta_t \varepsilon_t \\ &= F(\mathbf{w}_t) - \eta_t \mathsf{G}(\mathbf{w}_t) + \eta_t^2 (\varepsilon_t/\eta_t + \tfrac{L}{2} \left\| \mathbf{d}_t - \mathbf{w}_t \right\|^2). \end{aligned}$$

Define $\Delta_t := F(\mathbf{w}_t) - F(\mathbf{w})$ and $\mathsf{G}_t := \mathsf{G}(\mathbf{w}_t)$. Thus

$$\Delta_{t+1} \leq \Delta_t - \eta_t \mathsf{G}_t + \eta_t^2 (\varepsilon_t/\eta_t + \tfrac{L}{2} \left\| \mathbf{d}_t - \mathbf{w}_t \right\|^2), \qquad \text{and} \qquad \Delta_t \leq \mathsf{G}_t. \tag{64}$$

Plug the latter into the former and expand:

$$\Delta_{t+1} \leq \pi_t (1-\eta_0) \Delta_0 + \sum_{s=0}^{t} \frac{\pi_t}{\pi_s} \eta_s^2 (\varepsilon_s/\eta_s + \tfrac{L}{2} \left\| \mathbf{d}_s - \mathbf{w}_s \right\|^2). \tag{65}$$

To prove the second claim, we have from (64) $\eta_t \mathsf{G}_t \leq \Delta_t - \Delta_{t+1} + \eta_t^2 (\varepsilon_t/\eta_t + \tfrac{L}{2} \left\| \mathbf{d}_t - \mathbf{w}_t \right\|^2)$. Summing from k to t:

$$\left(\min_{k \leq s \leq t} \mathsf{G}_s \right) \sum_{s=k}^{t} \eta_s \leq \sum_{s=k}^{t} \eta_s \mathsf{G}_s \leq F(\mathbf{w}_k) - F(\mathbf{w}_{t+1}) + \sum_{s=k}^{t} \eta_s^2 (\varepsilon_s/\eta_s + \tfrac{L}{2} \left\| \mathbf{d}_s - \mathbf{w}_s \right\|^2).$$

Rearranging completes the proof.

A.3 Proof of Theorem 5

As $\eta_t = \frac{2}{t+2}$, we have $\eta_0 = 1$ and $\pi_t = \frac{2}{(t+1)(t+2)}$. For the first claim (upper bound on $F(\mathbf{w}_t)$) all we need to verify is that, by induction, $\frac{1}{(t+1)(t+2)} \sum_{s=0}^{t} \frac{s+1}{s+2} \leq \frac{1}{t+4}$. This is elementary.

We prove the second claim by a sequence of calculations similar to that in (Freund and Grigas, 2016). First, using (19) and the bound on $F(\mathbf{w}_t)$ in (20) with $t = 1$ and $\mathbf{w} = \mathbf{w}_2$:

$$\tilde{\mathsf{G}}_1^1 \leq \tfrac{1}{\eta_1} \left(F(\mathbf{w}_1) - F(\mathbf{w}_2) + \tfrac{\eta_1^2}{2} (\delta + L_F) \right) \leq \tfrac{3}{2} \left(\tfrac{1}{2} (\delta + L_F) + \tfrac{2}{9} (\delta + L_F) \right) = \tfrac{13}{12} (\delta + L_F),$$

proving the second claim on $\tilde{\mathsf{G}}_1^t$ for $t = 1, 2$ (note that $\tilde{\mathsf{G}}_1^2 \leq \tilde{\mathsf{G}}_1^1$ by definition). For $t \geq 3$, we consider $k = t/2 - 1$ if t is even and $k = (t+1)/2 - 1$ otherwise. Clearly $k \geq 1$, hence

$$\begin{aligned} \sum_{s=k}^{t} \eta_s &= 2 \sum_{s=k}^{t} \frac{1}{s+2} \geq 2 \int_{k-1}^{t} \frac{1}{s+2} \mathrm{d}s = 2 \ln \frac{t+2}{k+1} \geq 2 \ln 2, \\ \sum_{s=k}^{t} \eta_s^2/2 &= 2 \sum_{s=k}^{t} \frac{1}{(s+2)^2} \leq 2 \int_{k}^{t+1} \frac{1}{(s+2)^2} \mathrm{d}s = 2 \left(\frac{1}{k+2} - \frac{1}{t+3} \right). \end{aligned}$$

Using again (19), and the bound on $F(\mathbf{w}_t)$ in (20) with $t = k$ and $\mathbf{w} = \mathbf{w}_{t+1}$:

$$\tilde{\mathsf{G}}_1^t \leq \tilde{\mathsf{G}}_k^t \leq \frac{\delta + L_F}{2 \ln 2} \left(\frac{2}{k+3} + \frac{2}{k+2} - \frac{2}{t+3} \right) \leq \frac{\delta + L_F}{\ln 2} \left(\frac{2}{t+2} + \frac{1}{t+3} \right) \leq \frac{3(\delta + L_F)}{t \ln 2}.$$

A.4 Proof of Theorem 8

A proof can be based on the simple observation that the objective value $F^\star$ satisfies

$$F^\star := \inf_{\mathbf{w}} \{\ell(\mathbf{w}) + f(\mathbf{w})\} = \inf_{\mathbf{w}, \rho : \kappa(\mathbf{w}) \leq \rho} \ell(\mathbf{w}) + h(\rho). \tag{66}$$

Note that if ρ were known, the theorem could have been proved as before. The key idea behind the step size choice (34) is to ensure that Algorithm 2 performs almost the same as if it knew the *unknown* but fixed constant $\rho = \kappa(\mathbf{w})$ beforehand.

Consider an arbitrary $\mathbf{w}$ and let $\rho = \kappa(\mathbf{w})$. We also use the shorthand $\hat{F}_t := \ell(\mathbf{w}_t) + h(\rho_t) \geq \ell(\mathbf{w}_t) + f(\mathbf{w}_t) = F(\mathbf{w}_t)$. Then, the following chain of inequalities can be verified:

$$\begin{aligned} \hat{F}_{t+1} &:= \ell(\mathbf{w}_{t+1}) + h(\rho_{t+1}) \\ &\leq \hat{F}_t + \langle \theta_t \mathbf{a}_t - \eta_t \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \rangle + \tfrac{L}{2} \left\| \theta_t \mathbf{a}_t - \eta_t \mathbf{w}_t \right\|^2 - \eta_t h(\rho_t) + \eta_t h(\theta_t/\eta_t) \\ &\leq \hat{F}_t + \eta_t \left\langle \tfrac{\rho}{\alpha_t} \mathbf{a}_t - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \right\rangle + \tfrac{L\eta_t^2}{2} \left\| \tfrac{\rho}{\alpha_t} \mathbf{a}_t - \mathbf{w}_t \right\|^2 - \eta_t h(\rho_t) + \eta_t h(\rho/\alpha_t) \\ &\leq \hat{F}_t + \min_{\mathbf{z} : \kappa(\mathbf{z}) \leq \rho} \eta_t \left\langle \mathbf{z} - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \right\rangle + \eta_t \rho \varepsilon_t + \tfrac{L\eta_t^2}{2} \left\| \tfrac{\rho}{\alpha_t} \mathbf{a}_t - \mathbf{w}_t \right\|^2 - \eta_t h(\rho_t) + \eta_t h(\rho/\alpha_t) \\ &= \hat{F}_t + \min_{\mathbf{z} : \kappa(\mathbf{z}) \leq \rho} \eta_t \left\langle \mathbf{z} - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \right\rangle - \eta_t h(\rho_t) + \eta_t h(\rho) \\ &\qquad + \eta_t^2 \Big(\underbrace{\tfrac{L}{2} \left\| \tfrac{\rho}{\alpha_t} \mathbf{a}_t - \mathbf{w}_t \right\|^2 + (\rho \varepsilon_t + h(\rho/\alpha_t) - h(\rho))/\eta_t}_{:= \delta_t} \Big) \\ &= \hat{F}_t + \eta_t^2 \delta_t - \eta_t \Big[\underbrace{\langle \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \rangle + h(\rho_t) - \min_{\mathbf{z} : \kappa(\mathbf{z}) \leq \rho} \langle \mathbf{z}, \nabla \ell(\mathbf{w}_t) \rangle + h(\rho)}_{:= \hat{\mathsf{G}}(\mathbf{w}_t)} \Big], \end{aligned}$$

where the first inequality is because the subroutine is `Relaxed`, the second inequality follows from the minimality of θ_t in (34), and the third inequality is due to the choice of $\mathbf{a}_t$ in Line 3 of Algorithm 2.

Recall that $\rho = \kappa(\mathbf{w})$. Moreover, due to the convexity of ℓ,

$$\begin{aligned}\hat{\mathsf{G}}(\mathbf{w}_t) &= \langle \mathbf{w}_t, \nabla \ell(\mathbf{w}_t)\rangle + h(\rho_t) - \min_{\mathbf{z}:\kappa(\mathbf{z})\le\rho} \big(\langle \mathbf{z}, \nabla \ell(\mathbf{w}_t)\rangle + h(\rho)\big) \\ &= h(\rho_t) - h(\kappa(\mathbf{w})) + \max_{\mathbf{z}:\kappa(\mathbf{z})\le\kappa(\mathbf{w})} \langle \mathbf{w}_t - \mathbf{z}, \nabla \ell(\mathbf{w}_t)\rangle \\ &\ge h(\rho_t) - h(\kappa(\mathbf{w})) + \max_{\mathbf{z}:\kappa(\mathbf{z})\le\kappa(\mathbf{w})} \ell(\mathbf{w}_t) - \ell(\mathbf{z}) \\ &\ge \hat{F}_t - F(\mathbf{w}).\end{aligned}$$

Thus we have retrieved the recursion:

$$\hat{F}_{t+1} - F(\mathbf{w}) \le \hat{F}_t - F(\mathbf{w}) - \eta_t \hat{\mathsf{G}}(\mathbf{w}_t) + \eta_t^2 \delta_t, \qquad \text{and} \qquad \hat{F}_t - F(\mathbf{w}) \le \hat{\mathsf{G}}(\mathbf{w}_t).$$

Summing the indices as in the proof of Theorem 4 and noting that $F(\mathbf{w}_t) \le \hat{F}_t$ for all t completes the proof.

Appendix B. Proofs for Section 4

B.1 Proof of Theorem 15

The first equivalence in Theorem 15 is well-known since Grothendieck's work on tensor products of Banach spaces (where it is usually called the projective tensor norm). The second equality follows directly from the arithmetic-geometric mean inequality.

We first note that the atomic set $\mathcal{A}$ in (48) is compact, so is its convex hull $\operatorname{conv}\mathcal{A}$. It is also easy to see that $\mathcal{A}$ is a connected set (by considering the continuous map $(\mathbf{u},\mathbf{v}) \mapsto \mathbf{u}\mathbf{v}^\top$).

$$\begin{aligned}\text{By (4), } \kappa(W) &= \inf\left\{\rho \ge 0 : W = \rho \sum\nolimits_i \sigma_i \mathbf{a}_i, \sigma_i \ge 0, \sum\nolimits_i \sigma_i = 1, \mathbf{a}_i \in \mathcal{A}\right\} \qquad (67)\\ &= \inf\left\{\rho \ge 0 : W \in \rho\ \operatorname{conv}\mathcal{A}\right\}.\end{aligned}$$

Since $\operatorname{conv}\mathcal{A}$ is compact with $\mathbf{0} \in \operatorname{int}(\operatorname{conv}\mathcal{A})$ we know the infimum above is attained. Thus there exist $\rho \ge 0$ and $C \in \operatorname{conv}\mathcal{A}$ so that $W = \rho C$ and $\kappa(W) = \rho$. Applying Caratheodory's theorem for connected sets (Rockafellar and Wets, 1998, Theorem 2.29) we know $C = \sum_{i=1}^t \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ for some $\sigma_i \ge 0$, $\sum_i \sigma_i = 1$, $\mathbf{u}_i\mathbf{v}_i^\top \in \mathcal{A}$ and $t \le mn$. Let $U = \sqrt{\rho}\,[\sqrt{\sigma_1}\mathbf{u}_1, \ldots, \sqrt{\sigma_t}\mathbf{u}_t]$, and $V = \sqrt{\rho}\,[\sqrt{\sigma_1}\mathbf{v}_1, \ldots, \sqrt{\sigma_t}\mathbf{v}_t]^\top$. We then have $W = UV$ and $\kappa(W) = \rho \ge \frac{1}{2}\sum_{i=1}^t \left(\|U_{:i}\|_c^2 + \|V_{i:}\|_r^2 \right)$, which proves one side of Theorem 15.

On the other hand, consider any U and V that satisfy

$$W = UV = \sum_{j=1}^t \|U_{:j}\|_c \|V_{j:}\|_r \sum_{i=1}^t \frac{\|U_{:i}\|_c \|V_{i:}\|_r}{\sum_{j=1}^t \|U_{:j}\|_c \|V_{j:}\|_r} \frac{U_{:i}}{\|U_{:i}\|_c} \frac{V_{i:}}{\|V_{i:}\|_r},$$

assuming w.l.o.g. that $\|U_{:i}\|_c \|V_{i:}\|_r \ne 0$ for all $1 \le i \le t$. This, together with the definition (67), gives the other half of the equality: $\kappa(W) \le \sum_{i=1}^t \|U_{:i}\|_c \|V_{i:}\|_r$. The proof is completed by making the elementary observation that $\sum_{i=1}^t \|U_{:i}\|_c \|V_{i:}\|_r \le \frac{1}{2}\sum_{i=1}^t \left(\|U_{:i}\|_c^2 + \|V_{i:}\|_r^2 \right)$.

B.2 Proof of Theorem 16

Since $U_{\text{init}} = (\sqrt{1-\eta_t}U_t, \sqrt{\theta_t}\mathbf{u}_t)$ and $V_{\text{init}} = (\sqrt{1-\eta_t}V_t^\top, \sqrt{\theta_t}\mathbf{v}_t)^\top$, it is obvious that $\tilde{W}_{t+1} = U_{\text{init}}V_{\text{init}}$. Note $\|\mathbf{u}_t\|_c \le 1$ and $\|\mathbf{v}_t\|_r \le 1$. By construction

$$\begin{aligned}\ell(W_{t+1}) + \lambda\rho_{t+1} &= \ell(U_{t+1}V_{t+1}) + \frac{\lambda}{2}\sum_{i=1}^{t+1}\left(\|(U_{t+1})_{:i}\|_c^2 + \|(V_{t+1})_{i:}\|_r^2\right) \\ &= \mathcal{F}_{t+1}(U_{t+1}, V_{t+1}) \ \le\ \mathcal{F}_{t+1}(U_{\text{init}}, V_{\text{init}}) \\ &\le \ell((1-\eta_t)U_tV_t + \theta_t\mathbf{u}_t\mathbf{v}_t^\top) + \lambda\theta_t + \lambda\frac{(1-\eta_t)}{2}\sum_{i=1}^{t}\left(\|(U_t)_{:i}\|_c^2 + \|(V_t)_{i:}\|_r^2\right) \\ &= \ell((1-\eta_t)W_t + \theta_t\mathbf{u}_t\mathbf{v}_t^\top) + \lambda(1-\eta_t)\rho_t + \lambda\theta_t, \qquad (68)\end{aligned}$$

where the inequality is by the definition of U_{init} and V_{init}, and the last equality is by the definition of ρ_t in (54). In addition, $\kappa(W_{t+1}) \le \rho_{t+1}$ follows from (54) and Theorem 15.

B.3 Proof of Theorem 21

By (50), the square of the polar is

$$(\kappa^\circ(G))^2 = \max\left\{\mathbf{c}^\top ZZ^\top \mathbf{c} : \|\mathbf{a}\|_2 \le 1, \|\mathbf{b}\|_2 \le 1\right\}, \quad \text{where} \quad \mathbf{c} = \begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}. \qquad (69)$$

Since it maximizes a convex quadratic function over a convex set, there must be a maximizer attained at an extreme point of the feasible region. Therefore the problem is equivalent to

$$\begin{aligned}(\kappa^\circ(G))^2 &= \max\left\{\mathbf{c}^\top ZZ^\top \mathbf{c} : \|\mathbf{a}\|_2 = 1, \|\mathbf{b}\|_2 = 1\right\} \qquad (70)\\ &= \max\left\{\operatorname{tr}(SZZ^\top) : \operatorname{tr}(SI_1) = 1, \operatorname{tr}(SI_2) = 1, S \succeq 0, \operatorname{rank}(S) = 1\right\}, \qquad (71)\end{aligned}$$

where $S = \mathbf{c}\mathbf{c}^\top$. Key to this proof is the fact that the rank one constraint can be dropped without breaking the equivalence. To see why, notice that without this rank constraint the feasible region is the intersection of the positive semidefinite cone with two hyperplanes, hence the rank of all its extreme points must be upper bounded by one (Pataki, 1998). Furthermore the linearity of the objective implies that there must be a maximizing solution attained at an extreme point of the feasible region. Therefore we can continue by

$$\begin{aligned}(\kappa^\circ(G))^2 &= \max\left\{\operatorname{tr}(SZZ^\top) : \operatorname{tr}(SI_1) = 1, \operatorname{tr}(SI_2) = 1, S \succeq 0\right\} \\ &= \max_{S\succeq 0}\min_{\mu_1,\mu_2\in\mathbb{R}} \operatorname{tr}(SGG^\top) - \mu_1(\operatorname{tr}(SI_1) - 1) - \mu_2(\operatorname{tr}(SI_2) - 1) \\ &= \min_{\mu_1,\mu_2\in\mathbb{R}}\max_{S\succeq 0} \operatorname{tr}(SGG^\top) - \mu_1(\operatorname{tr}(SI_1) - 1) - \mu_2(\operatorname{tr}(SI_2) - 1) \\ &= \min\left\{\mu_1 + \mu_2 : \mu_1, \mu_2 \in \mathbb{R}, GG^\top \preceq \mu_1 I_1 + \mu_2 I_2\right\} \qquad (72)\\ &= \min\left\{\mu_1 + \mu_2 : \mu_1 \ge 0, \mu_2 \ge 0, \left\|D_{\mu_2/\mu_1}G\right\|_{\text{sp}}^2 \le \mu_1 + \mu_2\right\} \qquad (73)\\ &= \min\left\{\|D_\mu G\|_{\text{sp}}^2 : \mu \ge 0\right\}, \qquad (74)\end{aligned}$$

where the third equality is due to the Lagrangian duality, the fifth equality follows from the equivalence $\left\|D_{\mu_2/\mu_1}G\right\|_{\text{sp}}^2 \leq \mu_1+\mu_2 \iff D_{\mu_2/\mu_1}GG^\top D_{\mu_2/\mu_1} \preceq (\mu_1+\mu_2)I \iff GG^\top \preceq (\mu_1+\mu_2)D_{\mu_2/\mu_1}^{-1}D_{\mu_2/\mu_1}^{-1} = \mu_1 I_1 + \mu_2 I_2$, and the last equality is obtained through the re-parameterization $\mu = \mu_2/\mu_1, \nu = \mu_1+\mu_2$ and the elimination of ν.

B.4 Solving the multi-view polar

Here we show how to efficiently compute the polar (dual norm) in the multi-view setting.

Let us first backtrack in (73), which produces the optimal μ_1 and μ_2 via

$$\mu_1+\mu_2 = \|D_\mu G\|_{\text{sp}}^2, \qquad \text{and} \qquad \frac{\mu_2}{\mu_1} = \mu. \tag{75}$$

Then the KKT condition for the step from (72) to (73) can be written as

$$M \succeq \mathbf{0}, \qquad M\mathbf{c} = \mathbf{0}, \qquad \|\mathbf{a}\|_2 = \|\mathbf{b}\|_2 = 1, \quad \text{where} \quad M := \mu_1 I_1 + \mu_2 I_2 - GG^\top. \tag{76}$$

By duality, this is a sufficient and necessary condition for $\mathbf{a}$ and $\mathbf{b}$ to be the solution of the polar operator. Given the optimal μ_1 and μ_2, the first condition $M \succeq \mathbf{0}$ must have been satisfied already. Let the null space of M be spanned by an orthonormal basis $\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}$. Then we can parameterize $\mathbf{c}$ as $\mathbf{c} = Q\boldsymbol{\alpha}$ where $Q = [\mathbf{q}_1, \ldots, \mathbf{q}_k] =: \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$. By (76),

$$0 = \|\mathbf{a}\|_2^2 - \|\mathbf{b}\|_2^2 = \boldsymbol{\alpha}' \left(Q_1^\top Q_1 - Q_2^\top Q_2\right) \boldsymbol{\alpha}. \tag{77}$$

Let $P\Sigma P^\top$ be the eigen-decomposition of $Q_1^\top Q_1 - Q_2^\top Q_2$ with Σ being diagonal and P being unitary. Let $\mathbf{v} = P^\top \boldsymbol{\alpha}$. Then $\mathbf{c} = QP\mathbf{v}$ and (77) simply becomes $\mathbf{v}^\top \Sigma \mathbf{v} = 0$. In addition, (76) also implies $2 = \|\mathbf{c}\|_2^2 = \mathbf{v}^\top P^\top Q^\top QP\mathbf{v} = \mathbf{v}^\top \mathbf{v}$. So finally the optimal solution to the polar problem (70) can be completely characterized by

$$\left\{ \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} = QP\mathbf{v} : \mathbf{v}^\top \Sigma \mathbf{v} = 0, \|\mathbf{v}\|_2^2 = 2 \right\}, \tag{78}$$

which is simply a linear system after a straightforward change of variable. The major computational cost for recovery is to find the null space of M, which can be achieved by QR decomposition. The complexity of eigen-decomposition for $Q_1^\top Q_1 - Q_2^\top Q_2$ depends on the dimension of the null space of M, which is usually low in practice.

References

Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Good practice in large-scale learning for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):507–520, 2014.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

Francis Bach. Convex relaxations of structured matrix factorizations. HAL:00861118, 2013.

Francis Bach. Duality between subgradient and conditional gradient methods. *SIAM Journal of Optimization*, 25(1):115–129, 2015.

Francis Bach and Michael Jordan. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley, 2006.

Francis Bach, Julien Mairal, and Jean Ponce. Convex sparse matrix factorizations. arXiv:0812.1869v1, 2008.

Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27(4):450–468, 2012.

Amir Beck and Shimrit Shtern. Linearly convergent away-step conditional gradient for non-strongly convex functions, 2015.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31:167–175, 2003.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Stephen Becker, Jérôme Bobin, and Emmanuel J. Candès. NESTA: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2009.

Jonathan M. Borwein and John D. Vanderwerff. *Convex Functions: Constructions, Characterizations and Counterexamples*. Cambridge University Press, 2010.

David M. Bradley and James Andrew Bagnell. Convex coding. In *Conf. on Uncertainty in Artificial Intelligence*, 2009.

Kristian Bredies and Dirk A. Lorenz. Iterated hard shrinkage for minimization problems with sparsity constraints. *SIAM Journal on Scientific Computing*, 30(2):657–683, 2008.

Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data*. Springer, 2011.

Samuel Burer and Renato D C Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2009.

Michael D. Canon and Clifton D. Cullum. A tight upper bound on the rate of convergence of Frank-Wolfe algorithm. *SIAM Journal on Control*, 6:509–516, 1968.

Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S.Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6): 805–849, 2012.

H. Cheng, Y. Yu, X. Zhang, E. Xing, and D. Schuurmans. Scalable and sound low-rank tensor learning. In *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, 2016.

Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms*, 6(4):1–30, 2010.

Tij De Bie, Nello Cristianini, and Roman Rosipal. Eigenproblems in pattern recognition. In *Handbook of Geometric Computing*, pages 129–170, 2005.

Vladimir Fedorovich Dem'yanov and Alexander M. Rubinov. The minimization of a smooth convex functional on a convex set. *SIAM Journal on Control*, 5:280–294, 1967.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-fei. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.

Gideon Dror, Noam Koenigstein, Yehuda Koren, and Markus Weimer. The Yahoo! music dataset and KDD-Cup'11. *JMLR Workshop and Conference Proceedings: Proceedings of KDD Cup 2011 Competition*, 18:3–18, 2012.

Miroslav Dudik, Zaid Harchaoui, and Jerome Malick. Lifted coordinate descent for learning with trace-norm regularizations. In *Proc. Intl. Conf. on Artificial Intelligence and Statistics*, 2012.

J. Dunn. Rates of convergence for conditional gradient algorithms near singular and nonsingular extremals. *SIAM Journal on Control and Optimization*, 17:187–211, 1979.

J. Dunn and S. Harshbarger. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62:432–444, 1978.

Yonina C. Eldar and Gitta Kutyniok, editors. *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

Robert Freund and Paul Grigas. New analysis and results for the Frank-Wolfe method. *Mathematical Programming*, 155(1):199–230, 2016.

Masao Fukushima and Hisashi Mine. A generalized proximal point algorithm for certain non-convex minimization problems. *International Journal of Systems Science*, 12(8): 989–1000, 1981.

Dan Garber and Elad Hazan. Faster rates for the Frank-Wolfe method over strongly-convex sets. In *Proc. Intl. Conf. on Machine Learning*, 2015.

Dan Garber and Elad Hazan. A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. *SIAM Journal on Optimization*, 26(3):1493–1528, 2016.

Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:643–660, 2001.

J. Guélat and P. Marcotte. Some comments on Wolfe's 'away step'. *Mathematical Programming*, 35:110–119, 1986.

Zaid Harchaoui, Anatoli Juditsky, and Arkadi Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152:75–112, 2015.

Elad Hazan. Sparse approximate solutions to semidefinite programs. In *Latin American Conference on Theoretical Informatics*, 2008.

Charles A. Holloway. An extension of the Frank and Wolfe method of feasible directions. *Mathematical Programming*, 6(1):14–27, 1974.

Cho-Jui Hsieh and Peder A. Olsen. Nuclear norm minimization via active subspace selection. In *Proc. Intl. Conf. on Machine Learning*, 2014.

Laurent Jacob, Guillaume Obozinski, and Jean P. Vert. Group lasso with overlap and graph lasso. In *Proc. Intl. Conf. on Machine Learning*, 2009.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proc. Intl. Conf. on Machine Learning*, 2013.

Martin Jaggi and Marek Sulovsky. A simple algorithm for nuclear norm regularized problems. In *Proc. Intl. Conf. on Machine Learning*, 2010.

Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. Factorized latent spaces with structured sparsity. In *Advances in Neural Information Processing Systems*, pages 982–990, 2010.

Seyoung Kim and Eric P. Xing. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genetics*, 5(8):1–18, 2009.

Jacek Kuczyński and Henryk Woźniakowski. Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122, 1992.

S. Lacoste-Julien and M. Jaggi. On the global linear convergence of Frank-Wolfe optimization variants. In *Advances in Neural Information Processing Systems*, 2015.

Sören Laue. A hybrid algorithm for convex semidefinite optimization. In *Proc. Intl. Conf. on Machine Learning*, 2012.

Evgeny S. Levitin and Boris T. Polyak. Constrained minimization problems. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50, 1966.

Gerard Meyer. Accelerated Frank–Wolfe algorithms. *SIAM Journal on Control*, 12:655–655, 1974.

Charles A. Micchelli, Jean M. Morales, and Massimiliano Pontil. Regularizers for structured sparsity. *Advances in Computational Mathematics*, 38(3):455–489, 2013.

Bamdev Mishra, Gilles Meyer, Francis Bach, and Rodolphe Sepulchre. Low-rank optimization with trace norm penalty. *SIAM Journal on Optimization*, 23(4):2124–2149, 2013.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming, Series B*, 140:125–161, 2013.

Guillaume Obozinski and Francis Bach. Convex relaxation for combinatorial penalties. Technical Report HAL 00694765, 2012.

Gabor Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23(2):339–358, 1998.

Ting Kei Pong, Paul Tseng, Shuiwang Ji, and Jieping Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20 (6):3465–3489, 2010.

Ralph Tyrell Rockafellar and Roger J-B Wets. *Variational Analysis*. Springer, 1998.

Shai Shalev-Shwartz, Nathan Srebro, and Tong Zhang. Trading accuracy for sparsity in optimization problem with sparsity constraint. *SIAM Journal on Optimization*, 20(6): 2807–2832, 2010.

Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, 2005.

Ambuj Tewari, Pradeep Ravikumar, and Inderjit S. Dhillon. Greedy algorithms for structurally constrained high dimensional problems. In *Advances in Neural Information Processing Systems*, 2011.

Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6:615–640, 2010.

Paul Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Mathematical Programming, Series B*, 125:263–295, 2010.

Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2010.

Martha White, Yaoliang Yu, Xinhua Zhang, and Dale Schuurmans. Convex multi-view subspace learning. In *Advances in Neural Information Processing Systems*, 2012.

P. Wolfe. Convergence theory in nonlinear programming. In *Integer and Nonlinear Programming*. North-Holland, 1970.

Yaoliang Yu and Dale Schuurmans. Rank/norm regularization with closed-form solutions: Application to subspace clustering. In *Conf. on Uncertainty in Artificial Intelligence*, 2011.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of The Royal Statistical Society Series B*, 68(1):49–67, 2006.

Xiaotong Yuan and Shuicheng Yan. Forward basis selection for pursuing sparse representations over a dictionary. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):3025–3036, 2013.

Hyokun Yun, Hsiang-Fu Yu, Cho-Jui Hsieh, S V N Vishwanathan, and Inderjit Dhillon. NOMAD: Nonlocking, stochastic multi-machine algorithm for asynchronous and decentralized matrix completion. In *Proc. Intl. Conf. on Very Large Data Bases*, 2014.

Xinhua Zhang, Yaoliang Yu, Martha White, Ruitong Huang, and Dale Schuurmans. Convex sparse coding, subspace learning, and semi-supervised extension. In *Proc. Nat'l. Conf. Artificial Intelligence*, 2011.

Xinhua Zhang, Yaoliang Yu, and Dale Schuurmans. Accelerated training for matrix-norm regularization: A boosting approach. In *Advances in Neural Information Processing Systems*, 2012.

Yong Zhuang, Wei-Sheng Chin, Yu Juan, and Chih-Jen Lin. A fast parallel SGD for matrix factorization in shared memory systems. In *Proc. ACM Recommender System Conference*, 2013.

Journal of Machine Learning Research 18 (2017) 1-31 Submitted 2/17; Revised 11/17; Published 12/17

Following the Leader and Fast Rates in Online Linear Prediction: Curved Constraint Sets and Other Regularities*

Ruitong Huang† ruitong@ualberta.ca
Department of Computing Science
University of Alberta
Edmonton T6G 2E8, Canada

Tor Lattimore‡ tor.lattimore@gmail.com
Independent Researcher
Canberra, Australia

András György a.gyorgy@imperial.ac.uk
Department of Electrical and Electronic Engineering
Imperial College London
South Kensington Campus, London SW7 2BT, UK

Csaba Szepesvári szepesva@cs.ualberta.ca
Department of Computing Science
University of Alberta
Edmonton T6G 2E8, Canada

Editor: Manfred Warmuth

Abstract

Follow the leader (FTL) is a simple online learning algorithm that is known to perform well when the loss functions are convex and positively curved. In this paper we ask whether there are other settings when FTL achieves low regret. In particular, we study the fundamental problem of linear prediction over a convex, compact domain with non-empty interior. Amongst other results, we prove that the curvature of the boundary of the domain can act as if the losses were curved: In this case, we prove that as long as the mean of the loss vectors have positive lengths bounded away from zero, FTL enjoys logarithmic regret, while for polytope domains and stochastic data it enjoys finite expected regret. The former result is also extended to strongly convex domains by establishing an equivalence between the strong convexity of sets and the minimum curvature of their boundary, which may be of independent interest. Building on a previously known meta-algorithm, we also get an algorithm that simultaneously enjoys the worst-case guarantees and the smaller regret of FTL when the data is 'easy'. Finally, we show that such guarantees are achievable directly (e.g., by the follow the regularized leader algorithm or by a shrinkage-based variant of FTL) when the constraint set is an ellipsoid.

Keywords: online linear optimization, follow the leader, logarithmic regret, strongly convex decision set, curvature

1. Introduction

Learning theory traditionally has been studied in a statistical framework, discussed at length, for example, by Shalev-Shwartz and Ben-David (2014). The issue with this approach is that the analysis of the performance of learning methods seems to critically depend on whether the data generating mechanism satisfies some probabilistic assumptions. Realizing that these assumptions are not necessarily critical, much work has been devoted recently to studying learning algorithms in the so-called online learning framework (Cesa-Bianchi and Lugosi, 2006). The online learning framework makes minimal assumptions about the data generating mechanism, while allowing one to replicate results of the statistical framework through online-to-batch conversions (Cesa-Bianchi et al., 2004). By following a minimax approach, however, results proven in the online learning setting, at least initially, led to rather conservative results and algorithm designs, failing to capture how more regular, "easier" data, may give rise to faster learning. This is problematic as it may suggest overly conservative learning strategies, missing opportunities to extract more information when the data is nicer. Also, it is hard to argue that data resulting from passive data collection, such as weather data, would ever be adversarially generated (though it is equally hard to defend that such data satisfies precise stochastic assumptions). Realizing this issue, during recent years much work has been devoted to understanding what regularities and how can lead to faster learning speed. For example, much work has been devoted to showing that faster learning speed (smaller "regret") can be achieved in the online convex optimization setting when the loss functions are "curved", such as when the loss functions are strongly convex or exp-concave, or when the losses show small variations, or the best prediction in hindsight has a small total loss, and that these properties can be exploited in an adaptive manner (e.g., Merhav and Feder 1992, Freund and Schapire 1997, Gaivoronski and Stella 2000, Cesa-Bianchi and Lugosi 2006, Hazan et al. 2007, Bartlett et al. 2007, Kakade and Shalev-Shwartz 2009, Orabona et al. 2012, Rakhlin and Sridharan 2013, van Erven et al. 2015, Foster et al. 2015).

In this paper we contribute to this growing literature by studying online linear prediction and the follow the leader (FTL) algorithm. Online linear prediction is arguably the simplest yet fundamental of all the learning settings, and lies at the heart of online convex optimization, while it also serves as an abstraction of core learning problems such as prediction with expert advice. FTL, the online analogue of empirical risk minimization of statistical learning, is the simplest learning strategy, one can think of. Although the linear setting removes the possibility of exploiting the curvature of losses, there are multiple ways online learning problems can present data that allows for small regret, even for FTL. As is well known, in the worst case, FTL suffers a linear regret (e.g., Example 2.2 of Shalev-Shwartz 2012). However, for "curved" losses (e.g., exp-concave losses), FTL was shown to achieve small (logarithmic) regret (see, e.g., Merhav and Feder 1992; Cesa-Bianchi and Lugosi 2006; Gaivoronski and Stella 2000; Hazan et al. 2007).

We take a thorough look at FTL in the case when the losses are linear, but the problem perhaps exhibits other regularities. The motivation comes from the simple observation that, for prediction over the simplex, when the loss vectors are selected independently of each other from a distribution with a bounded support with a nonzero mean, FTL quickly locks onto selecting the loss-minimizing vertex of the simplex, achieving finite expected regret. In this case, FTL is an excellent algorithm. In fact, FTL is shown to be the minimax optimizer for the binary losses in the stochastic expert setting in the paper of Kotłowski (2016). Thus, we ask the question of whether there are other regularities that allow FTL to achieve nontrivial performance guarantees. Our main result shows that when the decision set (or constraint set) has a sufficiently "curved" boundary (or equivalently, if it is strongly convex) and the linear loss is bounded away from 0, FTL is able to achieve logarithmic regret even in the adversarial setting, thus opening up a new way to prove fast rates not based on the curvature of losses, but on that of the boundary of the constraint set and non-singularity of the linear loss. In a matching lower bound we show that this regret bound is essentially unimprovable. We also show an alternate bound for polytope constraint sets, which allows us to prove that (under certain technical conditions) for stochastic problems the expected regret of FTL will be finite. To finish, we use $(\mathcal{A},\mathcal{B})$-prod of Sani et al. (2014) to design an algorithm that adaptively interpolates between the worst case $O(\sqrt{n \log n})$ regret and the smaller regret bounds, which we prove here for "easy data." We also show that if the constraint set is an ellipsoid, both the follow the regularized leader (FTRL) algorithm and a combination of FTL and shrinkage, which we call follow the shrunken leader

*. An earlier version of this paper was published in NIPS 2016 (Huang et al., 2016).
†. Ruitong Huang is now at Borealis AI, Edmonton, Canada.
‡. Tor Lattimore is now at DeepMind, London, UK.

©2017 Ruitong Huang, Tor Lattimore, András György, and Csaba Szepesvári.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-079.html.

(FTSL), achieve logarithmic regret for easy data. Simulation results on artificial data complement the theoretical findings.

While we believe that we are the first to point out that the curvature of the constraint set $\mathcal{W}$ can help in speeding up learning, this effect is known in convex optimization since at least the work of Levitin and Polyak (1966), who showed that exponential rates are attainable for strongly convex constraint sets if the norm of the gradients of the objective function admit a uniform lower bound. More recently, Garber and Hazan (2015) proved an $O(1/n^2)$ optimization error bound (with problem-dependent constants) for the Frank-Wolfe algorithm for strongly convex and smooth objectives and strongly convex constraint sets. The effect of the shape of the constraint set was also discussed by Abbasi-Yadkori (2009) who demonstrated $O(\sqrt{n})$ regret in the linear bandit setting. Although at a high level these results are similar to ours, our proof technique is rather different.

2. Preliminaries, Online Learning and the Follow the Leader Algorithm

We consider the standard framework of online convex optimization, where a learner and an environment interact in a sequential manner over n rounds: In every round $t = 1, \dots, n$, first the learner predicts $w_t \in \mathcal{W}$. Then the environment picks a loss function $\ell_t \in \mathcal{L}$, and the learner suffers loss $\ell_t(w_t)$ and observes ℓ_t. Here, $\mathcal{W}$ is a compact and convex subset of the d-dimensional Euclidean space $\mathbb{R}^d$ with non-empty interior, and $\mathcal{L}$ is a set of convex functions mapping $\mathcal{W}$ to the reals. The elements of $\mathcal{L}$ are called loss functions. The performance of the learner is measured in terms of its regret,

$$R_n = \sum_{t=1}^{n} \ell_t(w_t) - \min_{w \in \mathcal{W}} \sum_{t=1}^{n} \ell_t(w)\,.$$

The simplest possible case, which will be the focus of this paper, is when the losses are linear, that is, when $\ell_t(w) = \langle f_t, w\rangle$ for some $f_t \in \mathcal{F} \subset \mathbb{R}^d$. In fact, the linear case is not only simple, but is also fundamental since the case of nonlinear loss functions can be reduced to it: Indeed, even if the losses are nonlinear, defining $f_t \in \partial\ell_t(w_t)$ to be a subgradient[1] of ℓ_t at w_t and letting $\tilde{\ell}_t(u) = \langle f_t, u\rangle$, by the definition of subgradients, $\ell_t(w_t) - \ell_t(u) \le \ell_t(w_t) - (\ell_t(w_t) + \langle f_t, u - w_t\rangle) = \tilde{\ell}_t(w_t) - \tilde{\ell}_t(u)$, hence for any $u \in \mathcal{W}$,

$$\sum_t \ell_t(w_t) - \sum_t \ell_t(u) \le \sum_t \tilde{\ell}_t(w_t) - \sum_t \tilde{\ell}_t(u)\,.$$

In particular, if an algorithm keeps the regret small no matter how the linear losses are selected (even when allowing the environment to pick losses based on the choices of the learner), the algorithm can also be used to keep the regret small in the nonlinear case.

Hence, in what follows we will study the linear case $\ell_t(w) = \langle f_t, w\rangle$ and, in particular, we will study the regret of the so-called "Follow The Leader" (FTL) learner, which in round $t \ge 2$ picks

$$w_t = \operatorname*{argmin}_{w \in \mathcal{W}} \sum_{i=1}^{t-1} \ell_i(w)\,.$$

For the first round, $w_1 \in \mathcal{W}$ is picked in an arbitrary manner. When $\mathcal{W}$ is compact, the optimal w of $\min_{w\in\mathcal{W}} \sum_{i=1}^{t-1}\langle w, f_t\rangle$ is attainable, which we will assume henceforth. If multiple minimizers exist, we simply fix one of them as w_t. We will also assume that $\mathcal{F}$ is non-empty, compact and convex.

One problem of the linearization technique is that if some algorithm's performance depends on some additional properties of the linear loss function, linearization may not preserve these and could lead to suboptimal performance. For example, if the loss functions are strongly convex and the optimum in hindsight (in fact, w_{n+1}) is an inner point of $\mathcal{W}$, FTL has no chance to do well, since

1. We let $\partial g(x)$ denote the subdifferential of a convex function $g : \mathrm{dom}(g) \to \mathbb{R}$ at x, that is, $\partial g(x) = \left\{\theta \in \mathbb{R}^d \,|\, g(x') \ge g(x) + \langle \theta, x' - x\rangle \ \forall x' \in \mathrm{dom}(g)\right\}$, where $\mathrm{dom}(g) \subset \mathbb{R}^d$ is the domain of g.

it will always predict points on the boundary. Thus, while our results extend from linear losses to arbitrary convex functions, some of the conditions of our regret bounds may be violated or the constants in the bounds might blow up, possibly leading to trivial or weak regret bounds. Thus, in practice, one should always check if the linearization step makes sense. On the positive side, no problem occurs if the optimum is outside of $\mathcal{W}$.

2.1 Support Functions

Let $\Theta_t = -\frac{1}{t}\sum_{i=1}^{t} f_i$ be the negative average of the first t vectors in $(f_t)_{t=1}^n$, $f_t \in \mathcal{F}$. For convenience, we define $\Theta_0 := 0$. Thus, for $t \ge 2$,

$$w_t = \operatorname*{argmin}_{w\in\mathcal{W}} \sum_{i=1}^{t-1}\langle w, f_i\rangle = \operatorname*{argmin}_{w\in\mathcal{W}}\langle w, -\Theta_{t-1}\rangle = \operatorname*{argmax}_{w\in\mathcal{W}}\langle w, \Theta_{t-1}\rangle\,.$$

Denote by $\Phi(\Theta) = \max_{w\in\mathcal{W}}\langle w, \Theta\rangle$ the so-called *support function* of $\mathcal{W}$. The support function, being the maximum of linear and hence convex functions, is itself convex. Further Φ is positive homogenous: for $a \ge 0$ and $\theta \in \mathbb{R}^d$, $\Phi(a\theta) = a\Phi(\theta)$. It follows then that the epigraph $\mathrm{epi}(\Phi) = \left\{(\theta, z) \,|\, z \ge \Phi(\theta), z \in \mathbb{R}, \theta \in \mathbb{R}^d\right\}$ of Φ is a cone, since for any $(\theta, z) \in \mathrm{epi}(\Phi)$ and $a \ge 0$, $az \ge a\Phi(\theta) = \Phi(a\theta)$, $(a\theta, az) \in \mathrm{epi}(\Phi)$ also holds.

The differentiability of the support function is closely tied to whether in the FTL algorithm the choice of w_t is uniquely determined:

Proposition 1 *Let $\mathcal{W} \ne \emptyset$ be convex and closed. Fix Θ and let $\mathcal{Z} := \{w \in \mathcal{W} \,|\, \langle w, \Theta\rangle = \Phi(\Theta)\}$. Then, $\partial\Phi(\Theta) = \mathcal{Z}$ and, in particular, $\Phi(\Theta)$ is differentiable at Θ if and only if $\max_{w\in\mathcal{W}}\langle w, \Theta\rangle$ has a unique optimizer. In this case, $\nabla\Phi(\Theta) = \operatorname{argmax}_{w\in\mathcal{W}}\langle w, \Theta\rangle$.*

The proposition follows from Danskin's theorem when $\mathcal{W}$ is compact (e.g., Proposition B.25 of Bertsekas 1999), but a simple direct argument, presented in Appendix A.1 for completeness, can also be used to show that it also remains true even when $\mathcal{W}$ is unbounded. By Proposition 1, when Φ is differentiable at Θ_{t-1}, $w_t = \nabla\Phi(\Theta_{t-1})$.

2.2 A Motivating Example

We close this section with an example demonstrating how fast rates can be achieved by the FTL algorithm. Consider the case when the losses are independent and identically distributed (i.i.d.), which means that (f_t) is an i.i.d. sequence with expectation $\mu \in \mathbb{R}^d$. Then $\mathbb{E}[\Theta_t] = -\mu$, and we have $\|\Theta_t + \mu\|_2 = O(1/\sqrt{t})$ with high probability. For $\mathcal{W}$ being the unit ball of $\mathbb{R}^d$ one has $w_t = \Theta_t/\|\Theta_t\|_2$ and therefore a crude bound suggests that $\|w_t - w^*\|_2 = O(1/\sqrt{t})$ where w^* is the optimal decision in hindsight, overall predicting that $\mathbb{E}[R_n] = O(\sqrt{n})$. On the other hand, in the rest of the paper we provide conditions when the expected regret can be much smaller than this. Below we give a simple geometric explanation how it can happen.

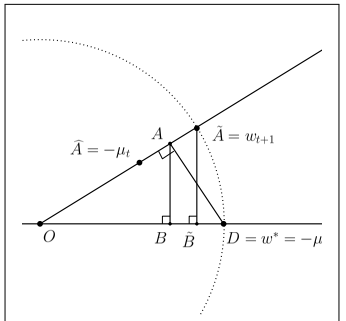


Figure 1: Illustration of how fast rates can be achieved by FTL.

Let $\mathcal{W} = \{w \,|\, \|w\|_2 \le 1\}$ and consider a stochastic setting where the f_t are i.i.d. samples with expectation $\mathbb{E}[f_t] = \mu = (-1, 0, \dots, 0)$ and $\|f_t\|_\infty \le M$ almost surely. It is straightforward to see that $w^* = (1, 0, \dots, 0)$, and thus $\langle w^*, \mu\rangle = -1$. Let $\mu_t = -\Theta_t$ denote our estimate of μ after t time steps; then $\|\mu_t - \mu\| = O(1/\sqrt{t})$ with high

probability. Now consider Fig. 1: The origin is denoted by O, the optimal prediction $w^* = -\mu$ by D, and $-\mu_t$ by $\hat{A}$. Then the prediction of FTL at time t is $\tilde{A}$, the intersection of the line connecting O and $\hat{A}$ with the unit sphere, and its instantaneous excess loss is $\langle \overrightarrow{OA}, \overrightarrow{OD} \rangle - 1 = |\overline{\tilde{B}D}|$ where $\tilde{B}$ is the orthogonal projection of $\tilde{A}$ to $\overline{OD}$. Next we give a simple geometric argument showing that if $|\overline{\hat{A}D}| \le \epsilon$ then $|\overline{\tilde{B}D}| \le \epsilon^2$. Since $|\overline{\hat{A}D}| = \|\mu_t - \mu\| = O(1/\sqrt{t})$ with high probability, this means that the excess error at time t is $O(1/t)$, making the regret $O(\log n)$ in n time steps, much smaller than the previously anticipated $O(\sqrt{n})$ regret. To finish, let A denote the orthogonal projection of D to the line connecting O and $\hat{A}$; then the Pythagorean theorem implies that $|\overline{OA}| \le |\overline{OD}| = |\overline{O\tilde{A}}|$, and so $A \in \overline{O\tilde{A}}$. Therefore, the orthogonal projection of A to $\overline{OD}$, denoted by B, belongs to the segment $\overline{O\tilde{B}}$, and so $|\overline{BD}| \ge |\overline{\tilde{B}D}|$. Since the triangles OAD and ABD are similar, we have $\frac{|\overline{BD}|}{|\overline{AD}|} = \frac{|\overline{AD}|}{|\overline{OD}|}$. Therefore, $|\overline{BD}| \le |\overline{AD}|^2 \le |\overline{\hat{A}D}|^2$ (by the definition of A), implying $|\overline{\tilde{B}D}| \le |\overline{\hat{A}D}|^2$, which we wanted to prove.

3. Non-Stochastic Analysis of FTL

We start by rewriting the regret of FTL in an equivalent form, which shows that we can expect FTL to enjoy a small regret when successive weight vectors move little.

Proposition 2 *The regret R_n of FTL satisfies the following identity:*

$$R_n = \sum_{t=1}^{n} t \, \langle w_{t+1} - w_t, \Theta_t \rangle \, .$$

The result is a direct corollary of Lemma 9 of McMahan (2010), which holds for any sequence of losses (even non-convex). It is also a tightening of the well-known inequality $R_n \le \sum_{t=1}^{n} \ell_t(w_t) - \ell_t(w_{t+1})$, which again holds for arbitrary loss sequences (e.g., Lemma 2.1 of Shalev-Shwartz, 2012). To keep the paper self-contained, we give a direct proof based on the summation by parts formula:

Proof The summation by parts formula states that for any $u_1, v_1, \ldots, u_{n+1}, v_{n+1}$ reals, $\sum_{t=1}^{n} u_t \, (v_{t+1} - v_t) = (u_{t+1} v_{t+1} - u_1 v_1) - \sum_{t=1}^{n} (u_{t+1} - u_t) \, v_{t+1}$. Applying this to the definition of regret with $u_t := w_{t,\cdot}$ and $v_{t+1} := t\Theta_t$, we get

$$\begin{aligned} R_n &= -\sum_{t=1}^{n} \langle w_t, t\Theta_t - (t-1)\Theta_{t-1} \rangle + \langle w_{n+1}, n\Theta_n \rangle \\ &= -\left\{ \cancel{\langle w_{n+1}, n\Theta_n \rangle} - 0 - \sum_{t=1}^{n} \langle w_{t+1} - w_t, t\Theta_t \rangle \right\} + \cancel{\langle w_{n+1}, n\Theta_n \rangle} . \end{aligned}$$

■

Our next proposition gives another identity for the regret. Although this formula is not directly needed for the rest of the paper, it provides interesting insights: as opposed to the previous result, it is independent of w_t, and directly connects the sequence $(\Theta_t)_t$ to the geometric properties of $\mathcal{W}$ through the support function Φ. A similar expression for a general "Follow the Regularized Leader" algorithm was also derived by Abernethy et al. (2014). For this proposition, we will momentarily assume that Φ is differentiable at $(\Theta_t)_{t \ge 1}$.

Proposition 3 *Assume Φ is differentiable at $\Theta_1, \ldots, \Theta_n$. Then*

$$R_n = \sum_{t=1}^{n} t \, D_\Phi(\Theta_t, \Theta_{t-1}) \, , \tag{1}$$

where $D_\Phi(\theta', \theta) = \Phi(\theta') - \Phi(\theta) - \langle \nabla\Phi(\theta), \theta' - \theta \rangle$ is the Bregman divergence of Φ and we use the convention that $\nabla\Phi(0) = w_1$.

Proof Let $v = \operatorname{argmax}_{w \in \mathcal{W}} \langle w, \theta \rangle$, $v' = \operatorname{argmax}_{w \in \mathcal{W}} \langle w, \theta' \rangle$. When Φ is differentiable at θ,

$$D_\Phi(\theta', \theta) = \Phi(\theta') - \Phi(\theta) - \langle \nabla\Phi(\theta), \theta' - \theta \rangle = \langle v', \theta' \rangle - \langle v, \theta \rangle - \langle v, \theta' - \theta \rangle = \langle v' - v, \theta' \rangle \, . \tag{2}$$

Therefore, by Proposition 2, $R_n = \sum_{t=1}^{n} t \langle w_{t+1} - w_t, \Theta_t \rangle = \sum_{t=1}^{n} t \, D_\Phi(\Theta_t, \Theta_{t-1})$. ■

When Φ is non-differentiable at some of the points $\Theta_1, \ldots, \Theta_n$, the equality in the above proposition can be replaced with inequalities. Defining the upper Bregman divergence $\overline{D}_\Phi(\theta', \theta) = \sup_{w \in \partial\Phi(\theta)} \Phi(\theta') - \Phi(\theta) - \langle w, \theta' - \theta \rangle$ and the lower Bregman divergence $\underline{D}_\Phi(\theta', \theta)$ similarly with inf instead of sup, we can easily obtain an analogue of Proposition 3:

$$\sum_{t=1}^{n} t \, \underline{D}_\Phi(\Theta_t, \Theta_{t-1}) \le R_n \le \sum_{t=1}^{n} t \, \overline{D}_\Phi(\Theta_t, \Theta_{t-1}) \, . \tag{3}$$

3.1 Constraint Sets with Positive Curvature

The previous results show in an implicit fashion that the curvature of $\mathcal{W}$ controls the regret. Before presenting our first main results, which make this connection explicit, we define some basic notions from differential geometry related to the curvature, while some extra details are presented in Appendix A.2 (all differential geometry concept and results that we need can be found in Section 2.5 of the book of Schneider, 2014).

3.1.1 Curvature and Strong Convexity

Given a twice continuously differentiable planar curve γ in $\mathbb{R}^2$, there exists a parametrization with respect to the curve length s, such that $\|\gamma'(s)\| = \| \, (x'(s), y'(s)) \, \| = \sqrt{x'(s)^2 + y'(s)^2} = 1$. Under the curve length parametrization, the curvature of γ at $\gamma(s)$ is $\|\gamma''(s)\|$. Define the unit normal vector $\mathbf{n}(s)$ as the unit vector that is perpendicular to $\gamma'(s)$.[2] Note that $\mathbf{n}(s) \cdot \gamma'(s) = 0$. Thus $0 = (\mathbf{n}(s) \cdot \gamma'(s))' = \mathbf{n}'(s) \cdot \gamma'(s) + \mathbf{n}(s) \cdot \gamma''(s)$, and $\|\gamma''(s)\| = \|\mathbf{n}(s) \cdot \gamma''(s)\| = \|\mathbf{n}'(s) \cdot \gamma'(s)\| = \|\mathbf{n}'(s)\|$. Therefore, the curvature of γ at point $\gamma(s)$ is the length of the differential of its unit normal vector.

Denote the boundary of $\mathcal{W}$ by $\mathrm{bd}(\mathcal{W})$ and a tangent plane of $\mathrm{bd}(\mathcal{W})$ at point w by $T_w\mathcal{W}$. We shall assume that $\mathcal{W}$ is twice continuously differentiable, that is, $\mathrm{bd}(\mathcal{W})$ is a twice continuously differentiable submanifold of $\mathbb{R}^d$. Then $T_w\mathcal{W}$ is unique, and there exists a unique unit vector at w that is perpendicular to $T_w\mathcal{W}$ and points outward of $\mathcal{W}$. In fact, one can define a continously differentiable normal unit vector field on $\mathrm{bd}(\mathcal{W})$, $u_\mathcal{W} : \mathrm{bd}(\mathcal{W}) \to \mathbb{S}^{d-1}$, the so-called Gauss map, which maps a boundary point $w \in \mathrm{bd}(\mathcal{W})$ to the unique outer normal vector to $\mathcal{W}$ at w, where $\mathbb{S}^{d-1} = \left\{ x \in \mathbb{R}^d \,|\, \|x\|_2 = 1 \right\}$ denotes the unit sphere in $\mathbb{R}^d$. Since $u_\mathcal{W}(w)$ maps $\mathrm{bd}(\mathcal{W})$ to unit vectors, the differential of the Gauss map, $\nabla u_\mathcal{W}(w)$, defines a linear endomorphism of $T_w\mathcal{W}$. Moreover, $\nabla u_\mathcal{W}(w)$ is a self-adjoint operator, with nonnegative eigenvalues. The differential of the Gauss map, $\nabla u_\mathcal{W}(w)$, describes the curvature of $\mathrm{bd}(\mathcal{W})$ via the second fundamental form. In particular, the *principal curvatures* of $\mathrm{bd}(\mathcal{W})$ at $w \in \mathrm{bd}(\mathcal{W})$ are defined as the eigenvalues of $\nabla u_\mathcal{W}(w)$. Perhaps a more intuitive, yet equivalent definition, is that the principal curvatures are the eigenvalues of the Hessian of $f = f_w$ in the parameterization $t \mapsto w + t - f_w(t) u_\mathcal{W}(w)$ of $\mathrm{bd}(\mathcal{W})$, which is valid in a small open neighborhood of w, where $f_w : T_w\mathcal{W} \to [0, \infty)$ is a suitable convex, nonnegative valued function that also satisfies $f_w(0) = 0$ (see Fig. 2). Thus, the principal curvatures at some point $w \in \mathrm{bd}(\mathcal{W})$ describe the local shape of $\mathrm{bd}(\mathcal{W})$ up to the second order. In this paper, we are interested in the minimum principal curvature at $w \in \mathrm{bd}(\mathcal{W})$, which can be interpreted as the minimum curvature at w over all the planar curves $\gamma \in \mathrm{bd}(\mathcal{W})$ that go through w.

2. There exist two unit vectors that are perpendicular to $\gamma'(s)$ for each point on γ. Pick the ones that are consistently oriented.

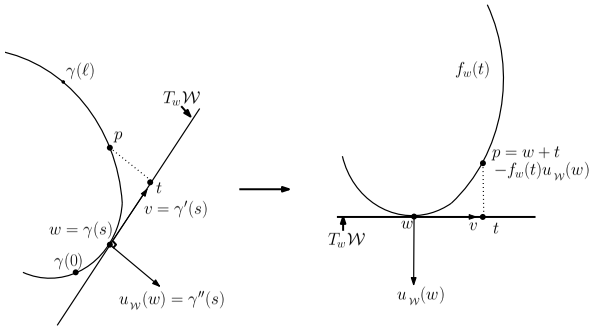


Figure 2: Some differential geometry notations.

A related concept that has been used in convex optimization to show fast rates is that of a strongly convex constraint set (Levitin and Polyak, 1966; Garber and Hazan, 2015): $\mathcal{W}$ is λ-strongly convex with respect to the norm $\|\cdot\|$ if, for any $x, y \in \mathcal{W}$ and $\gamma \in [0,1]$, the $\|\cdot\|$-ball with origin $\gamma x+(1-\gamma)y$ and radius $\gamma(1-\gamma)\lambda \|x-y\|^2/2$ is included in $\mathcal{W}$. That is, for any $z \in \mathbb{R}^d$ with $\|z\| = 1$, $\gamma x+(1-\gamma)y+\gamma(1-\gamma)\frac{\lambda}{2}\|x-y\|^2 z \in \mathcal{W}$. Next we show that a convex body $\mathcal{W}$ with twice continuously differentiable boundary is λ-strongly convex with respect to $\|\cdot\|_2$ if and only if the principal curvatures of the surface $\mathrm{bd}(\mathcal{W})$ are all at least λ.[3] In the rest of the paper, $B_r(x) = \{y \in \mathbb{R}^d \,|\, \|x-y\|_2 \le r\}$ will denote the Euclidean ball of radius r centered at x (in case x is the origin, it will often be omitted).

Proposition 4 *Let $\mathcal{W} \subset \mathbb{R}^d$ be a convex body with with twice continuously differentiable boundary and support function φ, and let λ be an arbitrary positive number. Then the following statements are equivalent:*

(i) The smallest principal curvature of $\mathcal{W}$ is at least λ.

(ii) $\mathcal{W} = \cap_{\theta \in \mathbb{S}^{d-1}} B_{1/\lambda}(w_\theta - \theta/\lambda)$ where $w_\theta \in \partial\varphi(\theta) \subset \mathrm{bd}(\mathcal{W})$.

(iii) $\mathcal{W}$ is λ-strongly convex.

Condition (ii), which is actually the definition of Polovinkin (1996) for strongly convex sets, means that $\mathcal{W}$ can be obtained as the intersection of closed balls of radius $1/\lambda$, such that there is one ball for every boundary point w and tangent hyperplane P where the ball touches P at w. Note that a ball with radius $1/\lambda$ satisfies all conditions: (i) and (ii) by definition, while (iii) holds, e.g., by Example 13 of Journée et al. (2010).

3.1.2 Regret Bounds

As promised, our next result connects the principal curvatures of $\mathrm{bd}(\mathcal{W})$ to the regret of FTL and shows that FTL enjoys logarithmic regret for highly curved surfaces, as long as $\|\Theta_t\|_2$ is bounded away from zero.

Theorem 5 *Assume $d \ge 2$ and let $\mathcal{W} \subset \mathbb{R}^d$ be a convex body with twice continuously differentiable boundary. Let $M = \max_{f \in \mathcal{F}} \|f\|_2$ and assume that Φ is differentiable at $(\Theta_t)_t$. Assume that the principal curvatures of the surface $\mathrm{bd}(\mathcal{W})$ are all at least λ_0 for some constant $\lambda_0 > 0$ (that is, $\mathcal{W}$ is λ_0 strongly convex) and $L_n := \min_{1 \le t \le n} \|\Theta_t\|_2 > 0$. Choose $w_1 \in \mathrm{bd}(\mathcal{W})$. Then*

$$R_n \le \frac{2M^2}{\lambda_0 L_n}(1+\log n)\,.$$

Before presenting the proof of the theorem, we discuss some of its implications and refinements. After the proof we will provide some examples of constraint sets with positive minimum principal curvature.

Remark 6 As we will show later in an essentially matching lower bound, this bound is tight, showing that the forte of FTL is when L_n is bounded away from zero and λ_0 is large. Note that the bound is vacuous as soon as $L_n = O(\log n/n)$ and is worse than the minimax bound of $O(\sqrt{n})$ when $L_n = o(\log n/\sqrt{n})$. One possibility to reduce the bound's sensitivity to L_n is to use the trivial bound $\langle w_{t+1} - w_t, \Theta_t\rangle \le LW = L \sup_{w,w' \in \mathcal{W}} \|w - w'\|_2$ for indices t when $\|\Theta_t\|_2 \le L$ (with an arbitrary $L > 0$). Then, by optimizing the bound over L, one gets a data-dependent bound of the form

$$R_n \le \inf_{L>0}\left(\frac{2M^2}{\lambda_0 L}(1+\log n) + LW\sum_{t=1}^n t\,\mathbb{I}\left(\|\Theta_t\|_2 \le L\right)\right), \tag{4}$$

which is more complex, but is free of L_n and thus reflects the nature of FTL better. Note that in the case of stochastic problems, where $f_1, \dots, f_n$ are independent and identically distributed (i.i.d.) with $\mu := -\mathbb{E}[\Theta_t] \ne 0$, the probability that $\|\Theta_t\|_2 < \|\mu\|_2/2$ is exponentially small in t. Thus, selecting $L = \|\mu\|_2/2$ in the previous bound, the contribution of the expectation of the second term is $O(\|\mu\|_2 W)$, giving an overall bound of the form $O(\frac{M^2}{\lambda_0\|\mu\|_2}\log n + \|\mu\|_2 W)$. On the other hand, if $\|\Theta_t\|_2 = 1$ if t is odd and $\|\Theta_t\| = 0$ otherwise (such an example is trivial to construct), the optimal choice of L in the above bound is $L = \Theta\left(\frac{M}{n}\sqrt{\frac{\log n}{\lambda_0 W}}\right)$ leads to a vacuous $O\left(Mn\sqrt{\frac{W\log n}{\lambda_0}}\right)$ regret bound.

Remark 7 Now consider the case of i.i.d. losses in a bit more detail. Assume that $\mathcal{W} = \mathcal{F} = B_1$, the Euclidean unit ball centered at the origin, and assume $\mathbb{E}[f_t] = \mu \ne 0$. Then it is straightforward to derive a high probability lower bound for $\|\Theta_t\|$: Using that $\mathbb{E}\|\Theta_t\|_2^2 = \|\mu\|_2^2 + \frac{\sigma^2}{t}$ where $\sigma^2 = \mathbb{E}\|f_i\|_2^2 - \|\mu\|_2^2$, we get

$$\mathbb{P}\left[\|\Theta_t\|_2 \le \frac{\|\mu\|_2}{2}\right] = \mathbb{P}\left[\|\Theta_t\|_2^2 - \mathbb{E}\|\Theta_t\|_2^2 \le -\frac{3\|\mu\|_2^2}{4} - \frac{\sigma^2}{t}\right] \le e^{-\frac{t}{18}\left(\frac{3\|\mu\|_2^2}{4} + \frac{\sigma^2}{t}\right)^2} \le e^{-t\frac{\|\mu\|_2^4}{32}}\,,$$

where the first inequality is due to McDiarmid's inequality (Boucheron et al., 2013) after noticing that changing a single f_i to some $f_i' \in \mathcal{F}$ may change the value of $\|\Theta_t\|_2^2$ by at most $6/t$. Combining this with (4) for $L = \|\mu\|_2/2$, we get

$$\mathbb{E}R_n \le \frac{4}{\|\mu\|_2}(1+\log n) + \frac{\|\mu\|_2}{4\sinh\left(\frac{\|\mu\|_2^4}{32}\right)} \le \frac{4}{\|\mu\|_2}(1+\log n) + O(1/\|\mu\|_2^7)\,. \tag{5}$$

Koolen et al. (2016) also proved an $O(\log n)$ bound on the expected regret of the sophisticated algorithm MetaGrad for the above case (Theorem 3 and Lemma 5 of their paper): in particular, they showed that MetaGrad achieves $O(Bd\log n)$ regret where $B = \frac{2\lambda_{\max}}{\|\mu\|}$ with $\lambda_{\max}$ being the maximum eigenvalue of $\mathbb{E}[f_t f_t^\top]$. Since $\lambda_{\max}$ can be as large as 1 (if $\|f_t\|_2 = 1$), (5) can improve (asymptotically) a factor of d over this regret bound. On the other hand, if f_t is uniformly distributed on the half unit sphere (e.g., the first coordinate of f_t is nonnegative with probability 1), Koolen et al. (2016) shows that $B \le \frac{24}{\sqrt{d}}$, which leads to an $O(\sqrt{d}\log n)$ regret, essentially matching (5), as one can show that $\frac{c}{\sqrt{d}} \le \|\mu\|_2 \le \frac{1}{\sqrt{d}}$ for some constant c.

3. Following Schneider (2014), a convex body in $\mathbb{R}^d$ is any compact, convex subset of $\mathbb{R}^d$ with non-empty interior.

Proof of Theorem 5 Fix $\theta_1, \theta_2 \in \mathbb{R}^d$ and let $w^{(1)} = \operatorname{argmax}_{w\in\mathcal{W}}\langle w, \theta_1\rangle$, $w^{(2)} = \operatorname{argmax}_{w\in\mathcal{W}}\langle w, \theta_2\rangle$. Note that if $\theta_1, \theta_2 \neq 0$ then $w^{(1)}, w^{(2)} \in \operatorname{bd}(\mathcal{W})$. Below we will show that

$$\langle w^{(1)} - w^{(2)}, \theta_1\rangle \le \frac{1}{2\lambda_0}\frac{\|\theta_2 - \theta_1\|_2^2}{\|\theta_2\|_2}\,. \tag{6}$$

Proposition 2 shows that bounding the regret is equivalent to bounding $\langle w_{t+1} - w_t, \Theta_t\rangle$. We then apply (6), which shows that the regret can be bounded by controlling the stability of Θ_t. A straightforward calculation shows that Θ_t cannot move much: for any norm $\|\cdot\|$ on $\mathcal{F}$, we have

$$\begin{aligned}
\|\Theta_t - \Theta_{t-1}\| &= \left\|\frac{1}{t-1}\sum_{i=1}^{t-1} f_i - \frac{1}{t}\sum_{i=1}^{t} f_i\right\| = \left\|\sum_{i=1}^{t-1}\left(\frac{1}{t-1} - \frac{1}{t}\right) f_i - \frac{1}{t} f_t\right\| \\
&\le \left\|\sum_{i=1}^{t-1}\left(\frac{1}{t-1} - \frac{1}{t}\right) f_i\right\| + \left\|\frac{1}{t} f_t\right\| = \left\|\sum_{i=1}^{t-1}\frac{1}{t(t-1)} f_i\right\| + \left\|\frac{1}{t} f_t\right\| \\
&= \frac{1}{t}\left\|\frac{1}{t-1}\sum_{i=1}^{t-1} f_i\right\| + \frac{1}{t}\|f_t\| \le \frac{2}{t} M\,.
\end{aligned} \tag{7}$$

where $M = \max_{f\in\mathcal{F}}\|f\|$ is a constant that depends on $\mathcal{F}$ and the norm $\|\cdot\|$.

Combining inequality (6) with Proposition 2 and (7), we get

$$\begin{aligned}
R_n &= \sum_{t=1}^{n} t\langle w_{t+1} - w_t, \Theta_t\rangle \le \sum_{t=1}^{n}\frac{t}{2\lambda_0}\frac{\|\Theta_t - \Theta_{t-1}\|_2^2}{\|\Theta_{t-1}\|_2} \\
&\le \frac{2M^2}{\lambda_0}\sum_{t=1}^{n}\frac{1}{t\|\Theta_{t-1}\|_2} \le \frac{2M^2}{\lambda_0 L_n}\sum_{t=1}^{n}\frac{1}{t} \le \frac{2M^2}{\lambda_0 L_n}(1 + \log n)\,.
\end{aligned}$$

To finish the proof we need to show (6). Below we provide a derivation based on the definition of principal curvature. Using the equivalence between the principal curvature and the modulus of strong convexity (cf., Proposition 4), we also provide an alternative proof in Appendix A.3 based on strong convexity, which leads to the slightly weaker result (24), loosing a constant factor of 4.

The following elementary lemma relates the cosine of the angle between two vectors θ_1 and θ_2 to the squared normalized distance between the two vectors, thereby reducing our problem to bounding the cosine of this angle. For brevity, we denote by $\cos(\theta_1, \theta_2)$ the cosine of the angle between θ_1 and θ_2.

Lemma 8 *For any non-zero vectors $\theta_1, \theta_2 \in \mathbb{R}^d$,*

$$1 - \cos(\theta_1, \theta_2) \le \frac{1}{2}\frac{\|\theta_1 - \theta_2\|_2^2}{\|\theta_1\|_2\|\theta_2\|_2}. \tag{8}$$

Proof Note that $\|\theta_1\|_2\|\theta_2\|_2\cos(\theta_1, \theta_2) = \langle\theta_1, \theta_2\rangle$. Therefore, (8) is equivalent to $2\|\theta_1\|_2\|\theta_2\|_2 - 2\langle\theta_1, \theta_2\rangle \le \|\theta_1 - \theta_2\|_2^2$, which, by algebraic manipulations, is itself equivalent to $0 \le (\|\theta_1\|_2 - \|\theta_2\|_2)^2$. ■

Given this result, it suffices to show that $\cos(\theta_1, \theta_2) \le 1 - \lambda_0\langle w^{(1)} - w^{(2)}, \frac{\theta_1}{\|\theta_1\|_2}\rangle$, which we prove using the tools from differential geometry introduced in Section 3.1.1. Let $\tilde{\theta}_i = \frac{\theta_i}{\|\theta_i\|_2}$ for $i = 1, 2$. The angle between θ_1 and θ_2 is the same as the angle between the normalized vectors $\tilde{\theta}_1$ and $\tilde{\theta}_2$. To calculate the cosine of the angle between $\tilde{\theta}_1$ and $\tilde{\theta}_2$, let P be a plane spanned by $\tilde{\theta}_1$ and $w^{(1)} - w^{(2)}$ and passing through $w^{(1)}$ (P is uniquely determined if $\tilde{\theta}_1$ is not parallel to $w^{(1)} - w^{(2)}$; if there are multiple planes, just pick any of them). Further, let $\hat{\theta}_2 \in \mathbb{S}^{d-1}$ be the unit vector along the projection of $\tilde{\theta}_2$ onto the plane P, as indicated in Fig. 3. Clearly, $\cos(\tilde{\theta}_1, \tilde{\theta}_2) \le \cos(\tilde{\theta}_1, \hat{\theta}_2)$.

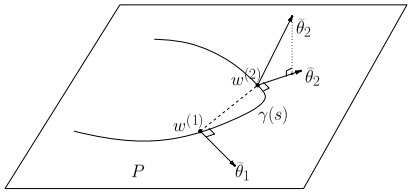


Figure 3: Illustration of the construction used in the proof of (6).

Consider a curve $\gamma(s)$ on $\operatorname{bd}(\mathcal{W})$ connecting $w^{(1)}$ and $w^{(2)}$ that is defined by the intersection of $\operatorname{bd}(\mathcal{W})$ and P and is parametrized by its curve length s so that $\gamma(0) = w^{(1)}$ and $\gamma(l) = w^{(2)}$, where l is the length of the curve γ between $w^{(1)}$ and $w^{(2)}$. Note that since γ is parametrized by its length, $\|\gamma'(s)\|_2 = 1$ for all $s \in [0, l]$. Let $u_{\mathcal{W}}(w)$ denote the outer normal vector to $\mathcal{W}$ at w as before, and let $u_\gamma : [0, l] \to \mathbb{S}^{d-1}$ denote the Gauss map of the planar curve γ, that is, $u_\gamma(s) = \hat{\theta}$ where $\hat{\theta}$ is the unit vector parallel to the projection of $u_{\mathcal{W}}(\gamma(s))$ on the plane P. Now, for any $\theta \in \mathbb{S}^{d-1}$, $w_\theta = \operatorname{argmax}_{w\in\mathcal{W}}\langle w, \theta\rangle$ is a point where a hyperplane with normal vector θ touches $\mathcal{W}$, thus, $u_{\mathcal{W}}(w_\theta) = \theta$. Therefore, $u_\gamma(0) = \tilde{\theta}_1$ and $u_\gamma(l) = \hat{\theta}_2$. In fact γ exists in two versions since $\mathcal{W}$ is a compact convex body, hence the intersection of P and $\operatorname{bd}(\mathcal{W})$ is a closed curve. Of these two versions we choose the one that satisfies that $\langle\gamma'(s), \tilde{\theta}_1\rangle \le 0$ for $s \in [0, l]$.[4] Given the above, we have

$$\cos(\tilde{\theta}_1, \hat{\theta}_2) = \langle\hat{\theta}_2, \tilde{\theta}_1\rangle = 1 + \langle\hat{\theta}_2 - \tilde{\theta}_1, \tilde{\theta}_1\rangle = 1 + \Big\langle\int_0^l u'_\gamma(s)\,\mathrm{d}s, \tilde{\theta}_1\Big\rangle = 1 + \int_0^l \langle u'_\gamma(s), \tilde{\theta}_1\rangle\,\mathrm{d}s. \tag{9}$$

Note that γ is a planar curve on $\operatorname{bd}(\mathcal{W})$, thus its curvature $\lambda(s)$ satisfies $\lambda(s) \ge \lambda_0$ for any $s \in [0, l]$. Also, for all $s \in [0, l]$, $\gamma'(s)$ is a unit vector parallel to P (since γ is parametrized by its curve length). Moreover, $u'_\gamma(s)$ is parallel to $\gamma'(s)$ since $u_\gamma(s)$ is the Gauss map, and $\lambda(s) = \|u'_\gamma(s)\|_2$. Therefore,

$$\langle u'_\gamma(s), \tilde{\theta}_1\rangle = \|u'_\gamma(s)\|_2\langle\gamma'(s), \tilde{\theta}_1\rangle \le \lambda_0\langle\gamma'(s), \tilde{\theta}_1\rangle,$$

where the last inequality holds because $\langle\gamma'(s), \tilde{\theta}_1\rangle \le 0$. Plugging this into (9), we get the desired

$$\cos(\tilde{\theta}_1, \hat{\theta}_2) \le 1 + \lambda_0\int_0^l \langle\gamma'(s), \tilde{\theta}_1\rangle\,\mathrm{d}s = 1 + \lambda_0\Big\langle\int_0^l \gamma'(s)\,\mathrm{d}s, \tilde{\theta}_1\Big\rangle = 1 - \lambda_0\langle w^{(1)} - w^{(2)}, \tilde{\theta}_1\rangle\,.$$

Reordering and combining with (8) we obtain

$$\langle w^{(1)} - w^{(2)}, \tilde{\theta}_1\rangle \le \frac{1}{\lambda_0}\Big(1 - \cos(\tilde{\theta}_1, \hat{\theta}_2)\Big) \le \frac{1}{\lambda_0}\left(1 - \cos(\theta_1, \theta_2)\right) \le \frac{1}{2\lambda_0}\frac{\|\theta_1 - \theta_2\|_2^2}{\|\theta_1\|_2\|\theta_2\|_2}\,.$$

Multiplying both sides by $\|\theta_1\|_2$ gives (6), thus, finishing the proof. ■

Next we present the smallest principal curvature of some common convex bodies (the proofs are relegated to the appendix), often used as constraint sets in machine learning.

Example 1 *(i) The smallest principal curvature λ_0 of the Euclidean ball $\mathcal{W} = \{w \,|\, \|w\|_2 \le r\}$ of radius r satisfies $\lambda_0 = \frac{1}{r}$.*

4. γ' and u'_γ denote the derivatives of γ and u_γ, respectively, which exist since $\operatorname{bd}(\mathcal{W})$ is twice continuously differentiable. When $s = 0$ or $s = l$, it suffices to take the corresponding one-sided derivatives or, equivalently, extend the definitions of γ and u_γ to an interval $[-\epsilon, l + \epsilon]$ for some $\epsilon > 0$.

(ii) Let Q be a positive definite matrix. If $\mathcal{W} = \left\{ w \,|\, w^\top Q w \le 1 \right\}$ then $\lambda_0 = \lambda_{\min}/\sqrt{\lambda_{\max}}$, where $\lambda_{\min}$ and $\lambda_{\max}$ are the minimal, respectively, maximal eigenvalues of Q.

(iii) Let $p > 1$ and $\mathcal{W} = \{w \,|\, \|w\|_p \le 1\}$. If $p > 2$, then $\lambda_0 = 0$. Otherwise, if $1 < p \le 2$, then

$$\lambda_0 = \min_{w \in \mathrm{bd}(\mathcal{W})} \min_{v \in \mathbb{S}^{d-1} : \langle w^{\odot(p-1)}, v \rangle = 0} (p-1) \frac{v^\top \operatorname{diag}\left(|w_1|^{p-2}, \cdots, |w_d|^{p-2} \right) v}{\|w^{\odot(p-1)}\|_2} \ge (p-1) d^{\frac{1}{2} - \frac{1}{p}} \,,$$

where $w^{\odot(p-1)} = (|w_1|^{p-1}, \ldots, |w_d|^{p-1})$ and $\operatorname{diag}(a_1, \ldots, a_k)$ denotes a $k \times k$ diagonal matrix with diagonal entries $a_1, \ldots, a_k$.

(iv) In general, let $\phi : \mathbb{R}^d \to \mathbb{R}$ be a twice continuously differentiable convex function. Then, for $\mathcal{W} = \{w \,|\, \phi(w) \le 1\}$, $\lambda_0 = \min_{w \in \mathrm{bd}(\mathcal{W})} \min_{v \in \mathbb{S}^{d-1} : \langle \phi'(w), v \rangle = 0} \frac{v^\top \nabla^2 \phi(w) v}{\|\phi'(w)\|_2}$.

Some of the results above have been derived in the literature based on seemingly different but equivalent assumptions explored in Proposition 4: (i) is a standard result in books on differential geometry; Polovinkin (1996) derived (ii) based on the strong convexity definition (ii) in Proposition 4, while (iii) was proved by Garber and Hazan (2015) based on the strong convexity definition (iii) in Proposition 4. Other examples of strongly sets convex sets, that is, sets with positive minimal principal curvature, can be found in the paper of Garber and Hazan (2015).

Our last result in this section is a lower bound for the linear game, showing that FTL achieves the optimal rate under the condition that $\min_t \|\Theta_t\|_2 \ge L > 0$.

Theorem 9 *Let $\lambda, L \in (0,1)$. Assume that $\{(1,-L),(-1,-L)\} \subset \mathcal{F}$ and let*

$$\mathcal{W} = \left\{ (x,y) \in \mathbb{R}^2 : x^2 + \frac{y^2}{\lambda^2} \le 1 \right\}$$

be an ellipsoid with principal curvature λ. Then, for any learning strategy, there exists a sequence of losses in $\mathcal{F}$ such that $\|\Theta_t\|_2 \ge L$ for all t and

$$R_n \ge \frac{1}{84\sqrt{2}} \frac{1}{\lambda L} \log n - \frac{1}{\lambda L} \left(\frac{2}{1 - e^{-\lambda^2 L^2}} + \frac{\pi^2}{108} \right) \,. \tag{10}$$

The theorem states that the regret of any learning strategy can be made at least as large as $\Omega(\log n/(L\lambda))$. Note that by Example 1, the minimal principal curvature of $\mathcal{W}$ in the above theorem is λ. In fact, it is not too hard to extend the above argument for any set $\mathcal{W}$ such that there is $w \in \mathrm{bd}(\mathcal{W})$ where the curvature is λ, and the curvature is a continuous function in a neighborhood of w over the boundary $\mathrm{bd}(\mathcal{W})$. The constants in the bound then depend on how fast the curvature changes within this neighborhood. In the case above, for small λL, the n-independent term in (10) is of order $1/(\lambda L)^3$.

Proof We define a random loss sequence, and we will show that no algorithm on this sequence can achieve an $o(\log n/(\lambda_0 L))$ regret. Let P be a random variable with $\mathrm{Beta}(K,K)$ distribution for some $K > 0$, and, given P, assume that $X_t, t \ge 1$ are i.i.d. Bernoulli random variables with parameter P. Let $f_t = X_t(1,-L) + (1-X_t)(-1,-L) = (2X_t - 1, -L)$. Thus, the second coordinate of f_t is always $-L$, and so $\|\Theta_t\|_2 = \left\| \frac{1}{t} \sum_{i=1}^t f_i \right\|_2 \ge L$. Furthermore, the conditional expectation of the loss vector is $f^p \triangleq \mathbb{E}[\, f_t |\, P = p] = (2p-1, -L)$.

Note that X_t is a function of f_t for all t; thus the conditional expectation of P, given $f_1, \ldots, f_{t-1}$, can be determined by the well-known formula $\hat{P}_{t-1} = \mathbb{E}[\, P | f_1 \ldots f_{t-1}] = \frac{K + \sum_{i=1}^{t-1} X_i}{2K + t - 1}$. Given p, denote the optimizer of f^p by w^p, that is, $w^p = \operatorname{argmin}_{w \in \mathcal{W}} \langle w, f^p \rangle$. Then the Bayesian optimal

choice in round t is

$$\begin{aligned}
\operatorname*{argmin}_{w \in \mathcal{W}} \mathbb{E}\left[\left[\langle w, f^P \rangle \right| f_1 \ldots f_{t-1} \right] &= \operatorname*{argmin}_{w \in \mathcal{W}} \left\langle w, \mathbb{E}\left[f^P \right| f_1 \ldots f_{t-1} \right] \right\rangle \\
&= \operatorname*{argmin}_{w \in \mathcal{W}} \left\langle w, f^{\hat{P}_{t-1}} \right\rangle \\
&= w^{\hat{P}_{t-1}} \,,
\end{aligned} \tag{11}$$

where the first equality follows by linearity of the inner product, the second since f^p is a linear function of p and the third by the definition of w^p.

Thus, denoting by W_t the prediction of an arbitrary algorithm in round t, the expected regret can be bounded from below as

$$\begin{aligned}
\mathbb{E}[R_n] &= \mathbb{E}\left[\max_{w \in \mathcal{W}} \sum_{t=1}^n \langle W_t - w, f_t \rangle \right] = \mathbb{E}\left[\mathbb{E}\left[\max_{w \in \mathcal{W}} \sum_{t=1}^n \langle W_t - w, f_t \rangle \middle| P \right] \right] \\
&\ge \mathbb{E}\left[\mathbb{E}\left[\sum_{t=1}^n \langle W_t - w^P, f_t \rangle \middle| P \right] \right] = \mathbb{E}\left[\sum_{t=1}^n \mathbb{E}\left[\langle W_t - w^P, f_t \rangle \right| P, f_1, \ldots, f_{t-1} \right] \right] \\
&= \mathbb{E}\left[\sum_{t=1}^n \mathbb{E}\left[\langle W_t - w^P, f^P \rangle \right| f_1, \ldots, f_{t-1} \right] \right] \qquad (12) \\
&\ge \mathbb{E}\left[\sum_{t=1}^n \min_{w \in \mathcal{W}} \mathbb{E}\left[\langle w - w^P, f^P \rangle \right| f_1, \ldots, f_{t-1} \right] \right] \\
&= \mathbb{E}\left[\sum_{t=1}^n \mathbb{E}\left[\left\langle w^{\hat{P}_{t-1}} - w^P, f^P \right\rangle \right| f_1, \ldots, f_{t-1} \right] \right] \qquad (13) \\
&= \sum_{t=1}^n \mathbb{E}\left[\left\langle w^{\hat{P}_{t-1}} - w^P, f^P \right\rangle \right] \,,
\end{aligned}$$

where (12) holds because of the independence of the f_s given P and since W_t is chosen based on $f_1, \ldots, f_{t-1}$ (but not on P), and (13) holds by (11).

By Lemma 17, given in Appendix A.4, we have

$$\begin{aligned}
\sum_{t=1}^n \mathbb{E}\left[\left\langle w^{\hat{P}_{t-1}} - w^P, f^P \right\rangle \right] &\ge \frac{\lambda L}{2} \sum_{t=1}^n \mathbb{E}\left[\frac{\left(\frac{2\hat{P}_{t-1} - 2P}{\lambda L} \right)^2}{\sqrt{1 + \left(\frac{1-2P}{\lambda L} \right)^2} \left(1 + \left(\frac{1 - 2\hat{P}_{t-1}}{\lambda L} \right)^2 \right)} \right] \qquad (14) \\
&= \frac{2}{\lambda L} \sum_{t=1}^n \mathbb{E}\left[\frac{1}{\sqrt{1 + \left(\frac{1-2P}{\lambda L} \right)^2}} \mathbb{E}\left[\frac{(\hat{P}_{t-1} - P)^2}{1 + \left(\frac{1 - 2\hat{P}_{t-1}}{\lambda L} \right)^2} \middle| P \right] \right] \\
&\ge \frac{2}{\lambda L} \sum_{t=1}^n \mathbb{E}\left[\frac{1}{\sqrt{1 + \left(\frac{1-2P}{\lambda L} \right)^2}} \mathbb{E}\left[\frac{(\hat{P}_{t-1} - P)^2}{1 + 2\left(\frac{1-2P}{\lambda L} \right)^2 + 2\left(\frac{2P - 2\hat{P}_{t-1}}{\lambda L} \right)^2} \middle| P \right] \right] , \qquad (15)
\end{aligned}$$

where in the last step we used $(a+b)^2 \le a^2 + b^2$. Let $\mathcal{G}_t$ be the event that $|\hat{P}_t - P| \le \frac{K|1-2P|}{2K+t} + \frac{t\lambda L}{2K+t}$; note that $\mathcal{G}_t$ holds with high probability by Lemma 18 in Appendix A.4. Then, lower bounding the

first term by 0, (15) can be lower bounded by

$$\begin{aligned}
&\frac{2}{\lambda L}\sum_{t=1}^{n-1}\mathbb{E}\left[\frac{1}{\sqrt{1+\left(\frac{1-2P}{\lambda L}\right)^2}}\mathbb{E}\left[\frac{(\hat{P}_t-P)^2}{1+2\left(\frac{1-2P}{\lambda L}\right)^2+2\left(\frac{2P-2\hat{P}_t}{\lambda L}\right)^2}\mathbb{I}(\mathcal{G}_t)\middle| P\right]\right]\\
&\geq \frac{2}{\lambda L}\sum_{t=1}^{n-1}\mathbb{E}\left[\frac{1}{\sqrt{1+\left(\frac{1-2P}{\lambda L}\right)^2}}\frac{\mathbb{E}\left[(\hat{P}_t-P)^2\mathbb{I}(\mathcal{G}_t)\middle| P\right]}{\left(1+2\left(\frac{1-2P}{\lambda L}\right)^2+2\left(\frac{2K}{2K+t}\frac{|1-2P|}{\lambda L}+\frac{2t}{2K+t}\right)^2\right)}\right]\\
&\geq \frac{2}{\lambda L}\sum_{t=1}^{n-1}\mathbb{E}\left[\frac{1}{\sqrt{1+\left(\frac{1-2P}{\lambda L}\right)^2}}\frac{\mathbb{E}\left[(\hat{P}_t-P)^2\mathbb{I}(\mathcal{G}_t)\middle| P\right]}{\left(9+4\left(\frac{1-2P}{\lambda L}\right)^2+8\frac{|1-2P|}{\lambda L}\right)}\right].
\end{aligned}$$

Combining the above, and using $(\hat{P}_t-P)^2\leq 1$ together with the upper bound on the probability of the event $\mathcal{G}_t^c$, the complement of $\mathcal{G}_t$, given in Lemma 18, we get

$$\begin{aligned}
\mathbb{E}[R_n] &\geq \frac{2}{\lambda L}\sum_{t=1}^{n-1}\mathbb{E}\left[\frac{1}{\sqrt{1+\left(\frac{1-2P}{\lambda L}\right)^2}}\frac{\mathbb{E}\left[(\hat{P}_t-P)^2\middle| P\right]-\mathbb{P}[\mathcal{G}_t^c]}{\left(9+4\left(\frac{1-2P}{\lambda L}\right)^2+8\frac{|1-2P|}{\lambda L}\right)}\right]\\
&\geq \frac{2}{\lambda L}\sum_{t=1}^{n-1}\left(\mathbb{E}\left[\frac{1}{\sqrt{1+\left(\frac{1-2P}{\lambda L}\right)^2}}\frac{\mathbb{E}\left[(\hat{P}_t-P)^2\middle| P\right]}{\left(9+4\left(\frac{1-2P}{\lambda L}\right)^2+8\frac{|1-2P|}{\lambda L}\right)}\right]-e^{-(t-1)\lambda^2L^2}\right)\\
&\geq \frac{2}{\lambda L}\left(\sum_{t=1}^{n-1}\mathbb{E}\left[\frac{1}{\sqrt{1+\left(\frac{1-2P}{\lambda L}\right)^2}}\frac{\mathbb{E}\left[(\hat{P}_t-P)^2\middle| P\right]}{\left(9+4\left(\frac{1-2P}{\lambda L}\right)^2+8\frac{|1-2P|}{\lambda L}\right)}\right]-\frac{1}{1-e^{-\lambda^2L^2}}\right). \qquad (16)
\end{aligned}$$

Now, by Lemma 19, given in Appendix A.4, we have

$$\mathbb{E}\left[(\hat{P}_t-P)^2\middle| P\right]=\frac{K^2(1-2P)^2}{(2K+t)^2}+\frac{tP(1-P)}{(2K+t)^2}\geq P(1-P)\left(\frac{1}{t}-\frac{2}{t(2K+t)}\right).$$

Combining this with (16) and introducing the constant

$$C=\mathbb{E}\left[\frac{1}{\sqrt{1+\left(\frac{1-2P}{\lambda L}\right)^2}}\frac{P(1-P)}{\left(9+4\left(\frac{1-2P}{\lambda L}\right)^2+8\frac{|1-2P|}{\lambda L}\right)}\right]$$

we obtain, for any $K>0$,

$$\begin{aligned}
\mathbb{E}[R_n] &\geq \frac{2}{\lambda L}\left[-\frac{1}{1-e^{-\lambda^2L^2}}+\sum_{t=1}^{n-1}C\left(\frac{1}{t}-\frac{2}{t(2K+t)}\right)\right] \qquad (17)\\
&\geq \frac{2C}{\lambda L}\log n-\frac{1}{\lambda L}\left(\frac{2}{1-e^{-\lambda^2L^2}}+\frac{C\pi^2}{3}\right). \qquad (18)
\end{aligned}$$

where we used $\sum_{t=1}^{n-1}\geq\int_1^n 1/t=\log n$ and $\sum_{t=1}^{n-1}1/(t(2K+t))\leq\sum_{t=1}^{\infty}1/t^2=\pi^2/6$. It remains to calculate a constant lower bound for C that is independent of λ and L. Denote $\frac{|1-2P|}{\lambda L}$ by Y; then $0\leq P(1-P)=\frac{1-Y^2\lambda^2L^2}{4}\leq 1/4$. Define $\widehat{\mathcal{G}}$ to be the event when $|Y|\leq 1$. Since P has Beta(K,K) distribution, $\mathbb{E}[P]=\frac{1}{2}$ and $\mathrm{Var}(P)=\frac{1}{8K}$. Therefore, by Chebyshev's inequality,

$$\mathbb{P}\left[\widehat{\mathcal{G}}^c\right]=\mathbb{P}\left[\left|P-\frac{1}{2}\right|>\frac{\lambda L}{2}\right]\leq\frac{1}{2K\lambda^2L^2}.$$

Therefore,

$$\begin{aligned}
C&=\mathbb{E}\left[\frac{1}{\sqrt{1+Y^2}}\frac{1-Y^2\lambda^2L^2}{4(9+4Y^2+8Y)}\right]\geq\mathbb{E}\left[\frac{1}{\sqrt{1+Y^2}}\frac{1-Y^2\lambda^2L^2}{4(9+4Y^2+8Y)}\mathbb{I}(\widehat{\mathcal{G}})\right]\\
&\geq\frac{1}{84\sqrt{2}}\mathbb{E}\left[(1-Y^2\lambda^2L^2)\mathbb{I}(\widehat{\mathcal{G}})\right]\geq\frac{1}{84\sqrt{2}}\left(\mathbb{E}\left[1-Y^2\lambda^2L^2\right]-\mathbb{P}\left[\widehat{\mathcal{G}}^c\right]\right)\\
&\geq\frac{1}{84\sqrt{2}}\left(1-\mathbb{E}\left[(1-2P)^2\right]-\frac{1}{2K\lambda^2L^2}\right)=\frac{1}{84\sqrt{2}}\left(1-\frac{1}{2K}-\frac{1}{2K\lambda^2L^2}\right)\\
&\geq\frac{1}{84\sqrt{2}}\cdot\frac{1}{2}
\end{aligned}$$

for any $K\geq 1+\frac{1}{\lambda^2L^2}$. Hence,

$$\mathbb{E}[R_n]\geq\frac{1}{84\sqrt{2}}\frac{1}{\lambda L}\log n-\frac{1}{\lambda L}\left(\frac{2}{1-e^{-\lambda^2L^2}}+\frac{\pi^2}{108}\right),$$

where we used the trivial upper bound $C\leq 1/36$ (obtained by maximizing the argument of the expectation in P in the definition of C by selecting $P=1/2$). The result is completed by noting that the worst-case regret is at least as big as the expected regret, thus, for every n, there exist a P and a sequence of loss vectors $f_1,\ldots,f_n$ such that the regret R_n satisfies (10). ■

3.2 Other Regularities

So far we have looked at the case when FTL achieves a low regret due to the curvature of $\mathrm{bd}(\mathcal{W})$. The next result characterizes the regret of FTL when $\mathcal{W}$ is a polytope, which has a flat, non-smooth boundary and thus Theorem 5 is not applicable. For this statement recall that given some norm $\|\cdot\|$, its dual norm is defined by $\|w\|_*=\sup_{\|v\|\leq 1}\langle v,w\rangle$.

Theorem 10 *Assume that $\mathcal{W}$ is a polytope and that Φ is differentiable at Θ_i, $i=1,\ldots,n$. Let $w_t=\mathrm{argmax}_{w\in\mathcal{W}}\langle w,\Theta_{t-1}\rangle$, $W=\sup_{w_1,w_2\in\mathcal{W}}\|w_1-w_2\|_*$ and $F=\sup_{f_1,f_2\in\mathcal{F}}\|f_1-f_2\|$. Then the regret of FTL is*

$$R_n\leq W\sum_{t=1}^{n}t\,\mathbb{I}(w_{t+1}\neq w_t)\|\Theta_t-\Theta_{t-1}\|\leq FW\sum_{t=1}^{n}\mathbb{I}(w_{t+1}\neq w_t).$$

Note that when $\mathcal{W}$ is a polytope, w_t is expected to "snap" to some vertex of $\mathcal{W}$. Hence, we expect the regret bound to be non-vacuous, if, e.g., Θ_t "stabilizes" around some value. Some examples after the proof will illustrate this.

Proof Let $v=\mathrm{argmax}_{w\in\mathcal{W}}\langle w,\theta\rangle$, $v'=\mathrm{argmax}_{w\in\mathcal{W}}\langle w,\theta'\rangle$. Similarly to the proof of Theorem 5,

$$\begin{aligned}
\langle v'-v,\theta'\rangle&=\langle v',\theta'\rangle-\langle v',\theta\rangle+\langle v',\theta\rangle-\langle v,\theta\rangle+\langle v,\theta\rangle-\langle v,\theta'\rangle\\
&\leq\langle v',\theta'\rangle-\langle v',\theta\rangle+\langle v,\theta\rangle-\langle v,\theta'\rangle=\langle v'-v,\theta'-\theta\rangle\leq W\,\mathbb{I}(v'\neq v)\|\theta'-\theta\|,
\end{aligned}$$

where the first inequality holds because $\langle v',\theta\rangle\leq\langle v,\theta\rangle$. Therefore, by (7),

$$R_n=\sum_{t=1}^{n}t\,\langle w_{t+1}-w_t,\Theta_t\rangle\leq W\sum_{t=1}^{n}t\,\mathbb{I}(w_{t+1}\neq w_t)\|\Theta_t-\Theta_{t-1}\|\leq FW\sum_{t=1}^{n}\mathbb{I}(w_{t+1}\neq w_t).$$

■

As noted before, since $\mathcal{W}$ is a polytope, w_t is (generally) attained at the vertices. In this case, the

epigraph of Φ is a polyhedral cone. Then, the event when $w_{t+1} \neq w_t$, that is, when the "leader" switches corresponds to when Θ_t and Θ_{t-1} belong to different linear regions corresponding to different linear pieces of the graph of Φ.

We now spell out a corollary for the stochastic setting. In particular, in this case FTL will often enjoy a constant regret:

Corollary 11 (Stochastic setting) *Assume that $\mathcal{W}$ is a polytope and that $(f_t)_{1\le t\le n}$ is an i.i.d. sequence of random variables such that $\mathbb{E}[f_i] = \mu$ and $\|f_i\|_\infty \le M$. Let $W = \sup_{w_1,w_2\in\mathcal{W}} \|w_1 - w_2\|_1$. Further assume that there exists a constant $r > 0$ such that Φ is differentiable for any ν such that $\|\nu - \mu\|_\infty \le r$. Then,*

$$\mathbb{E}[R_n] \le 2MW\left(3 + \frac{2M^2}{r^2}\log\left(\frac{2M^2 d}{r^2}\right)\right).$$

The existence of an r such that Φ is differentiable for any ν such that $\|\nu - \mu\|_\infty \le r$ is equivalent to that Φ is differentiable at μ. By Proposition 1, this condition requires that at μ, $\max_{w\in\mathcal{W}}\langle w, \theta\rangle$ has a unique optimizer (note that the volume of the set of vectors θ with multiple optimizers is zero). On the other hand, r should be selected to be the radius of the largest ball such that the optimal decisions for the expected losses μ and ν (i.e., the maximizers defining $\Phi(-\mu)$ and $\Phi(-\nu)$) belong to the same face of $\mathcal{W}$.

Proof Let $V = \{\nu \,|\, \|\nu - \mu\|_\infty \le r\}$. Note that the epigraph of the function Φ is a polyhedral cone. Since Φ is differentiable in the interior of V, $\{(\theta, \Phi(\theta)) \,|\, \theta \in V\}$ is a subset of a linear subspace. Therefore, for $-\Theta_t, -\Theta_{t-1} \in V$, $w_{t+1} = w_t$. Hence, by Theorem 10,

$$\mathbb{E}[R_n] \le 2MW\sum_{t=1}^{n}\mathbb{P}[-\Theta_t, -\Theta_{t-1} \notin V] \le 4MW\left(1 + \sum_{t=1}^{n}\mathbb{P}[-\Theta_t \notin V]\right). \tag{19}$$

On the other hand, note that $\|f_i\|_\infty \le M$. Then

$$\mathbb{P}[-\Theta_t \notin V] = \mathbb{P}\left[\left\|\frac{1}{t}\sum_{i=1}^{t} f_i - \mu\right\|_\infty \ge r\right] \le \sum_{j=1}^{d}\mathbb{P}\left[\left|\frac{1}{t}\sum_{i=1}^{t} f_{i,j} - \mu_j\right| \ge r\right] \le 2de^{-\frac{tr^2}{2M^2}},$$

where the last inequality is due to Hoeffding's inequality. Now, using that for any $\alpha > 0$ and $\tau > 0$, $\sum_{t=\tau+1}^{n}\exp(-\alpha t) \le \int_\tau^n \exp(-\alpha t)dt \le \frac{1}{\alpha}\exp(-\alpha\tau)$, from (19) we obtain

$$\mathbb{E}[R_n] \le 2MW\left(1 + \tau + \frac{2d}{\alpha}e^{-\alpha\tau}\right).$$

Setting $\alpha = \frac{r^2}{2M^2}$ and $\tau = \frac{1}{\alpha}\log(d/\alpha)$ in the above bound finishes the proof. ■

4. Adaptive Algorithms

While FTL can exploit the curvature of the surface of the constraint set to achieve $O(\log n)$ regret, it requires the curvature condition and $\min_t \|\Theta_t\|_2 \ge L$ being bounded away from zero, or it may suffer linear regret. On the other hand, many algorithms such as the follow the regularized leader (FTRL) are known to achieve a regret guarantee of $O(\sqrt{n})$ even for the worst-case data in the linear setting (see, e.g., Shalev-Shwartz, 2012). This raises the question of whether one can have an algorithm that can achieve constant or $O(\log n)$ regret in the respective settings of Corollary 11 or Theorem 5, while it still maintains $O(\sqrt{n})$ regret for worst-case data. One way to design an adaptive algorithm is to use the $(\mathcal{A}, \mathcal{B})$-prod algorithm of Sani et al. (2014), trivially leading to the following result:

Algorithm 1 Follow The Shrunken Leader (FTSL)

Predict $w_1 = 0$;
for $t = 2, ..., n-1$ **do**
  FTL: Compute $\tilde{w}_t = \operatorname{argmin}_{w\in\mathcal{W}}\langle w, F_{t-1}\rangle$.
  Shrinkage: Predict $w_t = \frac{\|F_{t-1}\|_2}{\sqrt{\|F_{t-1}\|_2^2 + t + 2}}\tilde{w}_t$.
end for
FTL: Compute $\tilde{w}_n = \operatorname{argmin}_{w\in\mathcal{W}}\langle w, F_{n-1}\rangle$.
Shrinkage: Predict $w_n = \frac{\|F_{n-1}\|_2}{\sqrt{\|F_{n-1}\|_2^2 + n}}\tilde{w}_n$.

Proposition 12 *Consider $(\mathcal{A}, \mathcal{B})$-prod of Sani et al. (2014), where algorithm $\mathcal{A}$ is chosen to be FTRL with an appropriate regularization term, while $\mathcal{B}$ is chosen to be FTL. Then the regret of the resulting hybrid algorithm $\mathcal{H}$ enjoys the following guarantees:*

- *If FTL achieves constant regret as in the setting of Corollary 11, then the regret of $\mathcal{H}$ is also constant.*
- *If FTL achieves a regret of $O(\log n)$ as in the setting of Theorem 5, then the regret of $\mathcal{H}$ is also $O(\log n)$.*
- *Otherwise, the regret of $\mathcal{H}$ is at most $O(\sqrt{n\log n})$.*

In the next section we show that if the constraint set is an ellipsoid, it is possible to design adaptive algorithms directly.

4.1 Adaptive Algorithms for Ellipsoid Constraint Sets

In this section we provide some interesting results about adaptive algorithms for the case when $\mathcal{W}$ is an ellipsoid in $\mathbb{R}^d$. First, we show that a variant of FTL using shrinkage as regularization has $O(\log n)$ regret when $\|\Theta_t\|_2 \ge L > 0$ for all t, but it also has $O(\sqrt{n})$ worst case guarantee. Furthermore, we show that the standard FTRL algorithm is adaptive if the constraint set is an ellipsoid and the loss vectors are stochastic. Throughout the section we will use the notation $F_t = -t\,\Theta_t = \sum_{i=1}^{t} f_i$.

4.1.1 Follow the Shrunken Leader

In this section we are going to analyze a combination of the FTL algorithm and the idea of shrinkage often used for regularization purposes in statistics. We assume that $\mathcal{W}$ is the d-dimensional unit ball and, without loss of generality, we further assume that $\|f\|_2 \le 1$ for all $f \in \mathcal{F}$. The Follow The Shrunken Leader (FTSL) algorithm is given in Algorithm 1. The main idea of the algorithm is to predict a shrunken version of the FTL prediction, in this way keeping it away from the boundary of $\mathcal{W}$. The next theorem shows that the right amount of shrinkage leads to a robust, adaptive algorithm.

Theorem 13 *Assume that $\mathcal{W} = \left\{x \in \mathbb{R}^d \,|\, \|x\|_2 \le 1\right\}$ and $\|f\|_2 \le 1$ for all $f \in \mathcal{F}$. Then the regret of FTSL is $O(\sqrt{n})$. If, in addition, there exists an $L > 0$ such that $\|\Theta_t\|_2 \ge L$ for $1 \le t \le n$, then the regret of is $O(\log n/L)$.*

Proof By the definition of F_t and $\mathcal{W}$, $\tilde{w}_t = -F_{t-1}/\|F_{t-1}\|_2$ for $t \ge 2$. Let $\sigma_n = \frac{\|F_{n-1}\|_2}{\sqrt{\|F_{n-1}\|_2^2 + n}}$. Our proof follows the idea of Abernethy et al. (2008). We compute the upper bound on the value of the game for each round backwards for $t = n, n-1, \ldots, 1$, by solving the optimal strategies for f_t. The

value of the game using FTSL is defined as

$$\begin{aligned} V_n &= \max_{f_1,\dots,f_n} \sum_{t=1}^{n} \langle w_t, f_t\rangle - \min_{w\in\mathcal{W}} \langle w, F_n\rangle \\ &= \max_{f_1,\dots,f_{n-1}} \sum_{t=1}^{n-1} \langle w_t, f_t\rangle + \underbrace{\max_{f_n} \|F_{n-1}+f_n\|_2 + \langle f_n, w_n\rangle}_{=:U_n} \end{aligned}$$

We first prove that U_n, the second term above, is bounded from above by $\sqrt{\|F_{n-1}\|_2^2+n}$. To see this, let $f_n = a_n\tilde{F}_{n-1} + b_n\Omega_{n-1}$ where $\tilde{F}_{n-1}$ is the unit vector parallel to F_{n-1} and Ω_{n-1} is a unit vector orthogonal to F_{n-1}. Furthermore, since $\|f_n\|_2 \le 1$, we have $a_n^2 + b_n^2 \le 1$. Thus,

$$\begin{aligned} U_n &= \max_{f_n} \sqrt{\|F_{n-1}\|_2^2 + 2a_n\|F_{n-1}\|_2 + a_n^2 + b_n^2} - a_n\sigma_n \\ &\le \max_{a} \sqrt{\|F_{n-1}\|_2^2 + 2a\|F_{n-1}\|_2 + n} - a\sigma_n \\ &= \sqrt{\|F_{n-1}\|_2^2 + n}, \end{aligned}$$

where the last equality follows since the maximum is attained at $a = 0$. A similar statement holds for the other time indices: for any $t \ge 1$,

$$\max_{f_t} \sqrt{\|F_{t-1}+f_t\|_2^2 + t + 1} + \langle f_t, w_t\rangle \le \sqrt{\|F_{t-1}\|_2^2 + t} + \frac{1}{\sqrt{t}}\,. \tag{20}$$

Before proving this inequality, we show how it implies the first statement of the theorem:

$$\begin{aligned} V_n &\le \max_{f_1,\dots,f_{n-1}} \sum_{t=1}^{n-1} \langle w_t, f_t\rangle + \sqrt{\|F_{n-1}\|_2^2 + n} \\ &\le \max_{f_1,\dots,f_{n-2}} \sum_{t=1}^{n-2} \langle w_t, f_t\rangle + \sqrt{\|F_{n-2}\|_2^2 + n - 1} + \frac{1}{\sqrt{n-1}} \\ &\le \dots \\ &\le 1 + \sum_{t=1}^{n-1} \frac{1}{\sqrt{t}} \le 2(1+\sqrt{n-1}). \end{aligned}$$

Moreover, if $\|\Theta_t\|_2 \ge L > 0$ for $1 \le t \le n$, a stronger version of (20) also holds:

$$\max_{f_t} \sqrt{\|F_{t-1}+f_t\|_2^2 + t + 1} + \langle f_t, w_t\rangle \le \sqrt{\|F_{t-1}\|_2^2 + t} + \frac{1}{(t-1)L}. \tag{21}$$

This implies the second statement of the theorem, since

$$\begin{aligned} V_n &\le \max_{f_1,\dots,f_{n-1}} \sum_{t=1}^{n-1} \langle w_t, f_t\rangle + \sqrt{\|F_{n-1}\|_2^2 + n} \\ &\le \max_{f_1,\dots,f_{n-2}} \sum_{t=1}^{n-2} \langle w_t, f_t\rangle + \sqrt{\|F_{n-2}\|_2^2 + n - 1} + \frac{1}{(n-1)L} \\ &\le \dots \\ &\le 1 + \sum_{t=1}^{n-1} \frac{1}{tL} \le 1 + \frac{1}{L} + \frac{\log(n-1)}{L}\,. \end{aligned}$$

To finish the proof, it remains to show (20) and (21). Let $f_t = a_t\tilde{F}_{t-1} + b_t\Omega_{t-1}$ where $\tilde{F}_{t-1}$ is the unit vector parallel to F_{t-1} and Ω_{t-1} is a unit vector orthogonal to F_{t-1}. Since $\|f_t\|_2 \le 1$, observe that $a_t^2 + b_t^2 = \|f_t\|_2 \le 1$. Furthermore, let $\sigma_t = \frac{\|F_{t-1}\|_2}{\sqrt{\|F_{t-1}\|_2^2+t+2}}$. Then, for any $t \ge 1$,

$$\begin{aligned} \Delta_t &= \max_{f_t} \sqrt{\|F_{t-1}\|_2^2 + 2a_t\|F_{t-1}\|_2 + a_t^2 + b_t^2 + t + 1} - a_t\sigma_t - \sqrt{\|F_{t-1}\|_2^2 + t} \\ &\le \max_{a_t} \sqrt{\|F_{t-1}\|_2^2 + 2a_t\|F_{t-1}\|_2 + t + 2} - a_t\sigma_t - \sqrt{\|F_{t-1}\|_2^2 + t} \\ &= \sqrt{\|F_{t-1}\|_2^2 + t + 2} - \sqrt{\|F_{t-1}\|_2^2 + t} \\ &= \frac{2}{\sqrt{\|F_{t-1}\|_2^2 + t + 2} + \sqrt{\|F_{t-1}\|_2^2 + t}} \\ &\le \frac{1}{\sqrt{t}}. \end{aligned} \tag{22}$$

This proves (20). Moreover, if $\|F_{t-1}\|_2 = \|(t-1)\Theta\|_2 \ge (t-1)L > 0$, by (22) we obtain

$$\Delta_t \le \frac{2}{\sqrt{\|F_{t-1}\|_2^2 + t + 2} + \sqrt{\|F_{t-1}\|_2^2 + t}} \le \frac{1}{\|F_{t-1}\|_2} \le \frac{1}{(t-1)L},$$

proving (21). ■

Remark 14 The above result can easily be extended to the case when $\mathcal{W}$ is an ellipsoid, that is, $\mathcal{W} = \{w \,|\, w^\top Q w \le 1\}$ for some positive-definite matrix Q. Transforming the predictions as $\hat{w}_t = Q^{1/2}w_t$ and the losses $\hat{f}_t = Q^{-1/2}f_t$, we see that, for all t, the new prediction $\hat{w}_t$ belongs to the unit ball (i.e., $\hat{w}_t \in B_1$) and $\langle w_t, f_t\rangle = \langle \hat{w}_t, \hat{f}_t\rangle$. Thus, the value of the game can be bounded as

$$\begin{aligned} V_n &= \max_{f_1,\dots,f_n\,:\,\|f_i\|_2\le 1} \sum_{t=1}^{n} \langle w_t, f_t\rangle - \min_{w\in\mathcal{W}} \langle w, F_n\rangle \\ &= \max_{f_1,\dots,f_n\,:\,\|f_i\|_2\le 1} \sum_{t=1}^{n} \langle \hat{w}_t, Q^{-1/2}f_t\rangle - \min_{\hat{w}\in B_1} \langle \hat{w}, Q^{-1/2}F_n\rangle \\ &= \max_{\hat{f}_1,\dots,\hat{f}_n\,:\,\|Q^{1/2}\hat{f}_i\|_2\le 1} \sum_{t=1}^{n} \langle \hat{w}_t, \hat{f}_t\rangle - \min_{\hat{w}\in B_1} \langle \hat{w}, \widehat{F}_n\rangle \\ &\le \max_{\hat{f}_1,\dots,\hat{f}_n\,:\,\|\hat{f}_i\|_2\le \frac{1}{\sqrt{\lambda_{\min}}}} \sum_{t=1}^{n} \langle \hat{w}_t, \hat{f}_t\rangle - \min_{\hat{w}\in B_1} \langle \hat{w}, \widehat{F}_n\rangle \\ &= \frac{1}{\sqrt{\lambda_{\min}}} \max_{\tilde{f}_1,\dots,\tilde{f}_n\,:\,\|\tilde{f}_i\|_2\le 1} \sum_{t=1}^{n} \langle \hat{w}_t, \tilde{f}_t\rangle - \min_{\hat{w}\in B_1} \langle \hat{w}, \tilde{F}_n\rangle\,, \end{aligned}$$

where $\lambda_{\min}$ is the minimal eigenvalue of the matrix Q, $\widehat{F}_n = \sum_{t=1}^n \hat{f}_t$, $\tilde{f}_t = \sqrt{\lambda_{\min}}\,\hat{f}_t$, and $\tilde{F}_n = \sqrt{\lambda_{\min}}\,\widehat{F}_n = \sum_{t=1}^n \tilde{f}_t$. Thus, playing over $\mathcal{W}$ with loss vectors from the unit ball is equivalent to playing over the unit ball B_1 against losses over $\mathcal{W}$ (note that $\|Q^{1/2}\hat{f}_i\|_2 \le 1$ is equivalent to $\hat{f}_i \in \mathcal{W}$), which can be reduced to playing against losses with maximum Euclidean norm 1. Thus, an algorithm for the ellipsoid constraint set $\mathcal{W}$ is to run FTSL over the unit ball B_1 with the transformed losses $\tilde{f}_t$, and predict $w_t = Q^{-1/2}\hat{w}_t$ where $\hat{w}_t$ is the prediction of FTSL. Assuming that $\|\Theta_t\|_2 \ge L$ for all t in the original problem, in the transformed problem we have $\|\sqrt{\lambda_{\min}}\,Q^{-1/2}\Theta_t\|_2 \ge L\sqrt{\lambda_{\min}/\lambda_{\max}}$ where $\lambda_{\max}$ is the largest eigenvalue of Q. Hence, the regret of the algorithm (in both the original

and the transformed problems) is at most $O\big(\frac{\sqrt{\lambda_{\max}}}{L\lambda_{\min}}\log n\big)$. Note that this is exactly the same rate as we can obtain from Theorem 5 and Example 1 (ii) for the (non-adaptive) FTL algorithm for the ellipsoid constraint set $\mathcal{W}$ (a closer inspection of the constants shows that the leading constant for FTSL is actually a factor of 2 better).

4.1.2 FTRL for Stochastic Losses

This section shows that when $\mathcal{W}$ is the unit ball B_1, FTRL with regularizer $R(w)=\frac{1}{2}\|w\|^2$ is an adaptive algorithm achieving logarithmic regret for stochastic losses. To fix the notation, in round t, FTLR predicts

$$w_t = \operatorname*{argmin}_{w\in\mathcal{W}} \eta_t\langle F_{t-1}, w\rangle + R(w),$$

if $t>1$ and $w_1=0$. It has been well known that FTRL with $\eta_t = 1/\sqrt{t-1}$ is guaranteed to achieve $O(\sqrt{n})$ regret in the adversarial setting (see, e.g., Shalev-Shwartz, 2012). It remains to prove that FTRL indeed achieves a fast rate in the stochastic setting.

Theorem 15 *Assume that the sequence of loss vectors, $f_1,\dots,f_n\in\mathbb{R}^d$ satisfies $\|f_t\|_2\le 1$ almost surely and $\mathbb{E}[f_t]=\mu$ for all t with some $\|\mu\|_2>0$. Then FTRL with $\eta_t=1/\sqrt{t-1}$ suffers $O(\log n)$ regret .*

Proof Using $R(w)=\frac{1}{2}\|w\|^2$ as its regularization, in round $t>1$ FTRL predicts

$$w_t = \operatorname*{argmin}_{w\in\mathcal{W}} \eta_t\langle F_{t-1}, w\rangle + R(w) = \begin{cases} \frac{1}{\sqrt{t-1}}F_{t-1} & \text{if } \|F_{t-1}\|\le\sqrt{t-1}\,; \\ \frac{F_{t-1}}{\|F_{t-1}\|} & \text{otherwise.}\end{cases} \tag{23}$$

For any $1\le t\le n$, denote the event $\|F_t\|\ge\sqrt{t}$ by $\mathcal{E}_t$. Note that if $\|F_{t-1}\|\ge\sqrt{t-1}$, FTRL predicts exactly the same w_t as FTL. Denote the accumulated loss of FTL in n rounds by $\mathcal{L}_n^{FTL}$. Thus, the regret of FTRL is

$$\begin{aligned}\mathbb{E}[R_n] &= \mathbb{E}\left[\sum_{t=1}^n\langle f_t,w_t\rangle - \min_{w\in\mathcal{W}}\langle f_t,w\rangle\right] \\ &= \mathbb{E}\left[\sum_{t=1}^n\langle f_t,w_t\rangle - \mathcal{L}_n^{FTL}\right] + \mathbb{E}\left[\mathcal{L}_n^{FTL} - \min_{w\in\mathcal{W}}\langle f_t,w\rangle\right] \\ &\le 2\sum_{t=1}^n \mathbb{P}[\mathcal{E}_t^c] + O(\log n),\end{aligned}$$

where to obtain the last inequality we applied (23) for the first term, while the second term is $O(\log n)$ by Remark 7. It remains to bound the first term, $2\sum_{t=1}^n\mathbb{P}[\mathcal{E}_t^c]$ in the above. For any $t>\frac{4}{\|\mu\|_2^2}$,

$$\begin{aligned}\mathbb{P}\left[\|F_t\|_2\le\sqrt{t}\right] &\le \mathbb{P}\left[\|F_t\|_2<\frac{t}{2}\|\mu\|_2\right] \le \sum_{i=1}^d\mathbb{P}\left[|F_{t,i}|<\frac{t}{2}|\mu_i|\right] \\ &\le \sum_{i=1}^d\mathbb{P}\left[|F_{t,i}-t\mu_i|>\frac{t}{2}|\mu_i|\right] \le 2\sum_{i=1}^d e^{-\frac{\mu_i^2}{4}t}\end{aligned}$$

Thus,

$$\begin{aligned}\sum_{t=1}^n\mathbb{P}[\mathcal{E}_t^c] &= \sum_{t=1}^{4/\|\mu\|_2^2}\mathbb{P}[\mathcal{E}_t^c] + \sum_{t=4/\|\mu\|_2^2}^n\mathbb{P}[\mathcal{E}_t^c] \\ &\le \frac{4}{\|\mu\|_2^2} + 2\sum_{i=1}^d\sum_{t=0}^n e^{-\frac{\mu_i^2}{4}t} \\ &\le \frac{4}{\|\mu\|_2^2} + 2\sum_{i=1}^d\frac{1}{1-e^{-\frac{\mu_i^2}{4}}} \\ &\le \frac{4}{\|\mu\|_2^2} + 2\sum_{i=1}^d\frac{\mu_i^2}{4} = \frac{4}{\|\mu\|_2^2} + \frac{\|\mu\|_2^2}{2}\,.\end{aligned}$$

where in the last inequality we used $1/(1-e^{-a})\le a$. Therefore, if $\|\mu\|>0$, the regret of FTRL satisfies

$$\mathbb{E}[R_n]\le\frac{8}{\|\mu\|_2^2}+\|\mu\|_2^2+O(\log n)=O(\log n).$$

■

Remark 16 Similarly to FTSL, the above result can be extended to ellipsoid constraint sets with an adequate choice of the regularizer $R(w)$. Assume that $\mathcal{W}=\{w\,|\,w^\top Qw\le 1\}$ for some positive definite matrix Q, and let $R(w)=\frac{1}{2}w^\top Qw$. Then

$$w_t = \operatorname*{argmin}_{w\in\mathcal{W}}\eta_t\langle F_{t-1},w\rangle + R(w) = Q^{-1/2}\operatorname*{argmin}_{\tilde{w}\in B_1}\eta_t\langle Q^{-1/2}F_{t-1},\tilde{w}\rangle + \frac{1}{2}\|\tilde{w}\|_2^2,$$

and

$$R_n = \sum_{t=1}^n\langle f_t,w_t\rangle - \min_{w\in\mathcal{W}}\langle f_t,w\rangle = \sum_{t=1}^n\langle Q^{-1/2}f_t,\tilde{w}_t\rangle - \min_{w\in B_1}\langle Q^{-1/2}f_t,\tilde{w}\rangle.$$

Thus, the problem is equivalent to the case of a unit ball constraint set with the loss vector $Q^{-1/2}f_t$ for time t, and FTRL with the selected regularizer achieves $O(\sqrt{n})$ worst case regret and $O(\log n)$ regret in the case of an i.i.d. loss sequence. Whether FTRL with a constraint-set-independent regularizer $R(w)=\frac{1}{2}\|w\|_2^2$ achieves similar adaptivity, remains an open question.

5. Simulations

We performed three simulations to illustrate the differences between FTL, FTRL with the regularizer $R(w)=\frac{1}{2}\|w\|_2^2$ when $w_t=\operatorname{argmin}_{w\in\mathcal{W}}\sum_{i=1}^{t-1}\langle f_{i-1},w\rangle+R(w)$, and the adaptive algorithm $(\mathcal{A},\mathcal{B})$-prod (AB) using FTL and FTRL as its candidates, which we shall call AB(FTL,FTRL).

For the experiments the constraint set $\mathcal{W}$ was chosen to be a slightly elongated ellipsoid in the 4-dimensional Euclidean space, with volume matching that of the 4-dimensional unit ball. The actual ellipsoid is given by $\mathcal{W}=\{w\in\mathbb{R}^4\,|\,w^\top Qw\le 1\}$ where Q is randomly generated as

$$Q=\begin{pmatrix} 4.3367 & 3.6346 & -2.2250 & 3.5628 \\ 3.6346 & 3.9966 & -2.3613 & 3.2817 \\ -2.2250 & -2.3613 & 2.0589 & -2.1295 \\ 3.5628 & 3.2817 & -2.1295 & 3.4206 \end{pmatrix}.$$

We experimented with three types of data to illustrate the behavior of the different algorithms: stochastic, "half-adversarial", and "worst-case" data (worst-case for FTL), as will be explained below. The first two data sets are random, so the experiments were repeated 100 times, and we report the average regret with its standard deviation; the worst case data is deterministic, so there no repetition was needed. For each experiment, we set $n = 2500$. The regularization coefficient for the FTRL, and the learning rate for AB were chosen based on their theoretical bounds minimizing the worst-case regret.

5.1 Stochastic Data

In this setting we used the following model to generate f_t: Let $(\hat{f}_t)_t$ be an i.i.d. sequence drawn from the 4-dimensional standard normal distribution, and let $\tilde{f}_t = \hat{f}_t / \left\|\hat{f}_t\right\|_2$. Then, f_t is defined as $f_t = \tilde{f}_t + Le_1$ where $e_1 = (1, 0, \dots, 0)^\top$. Therefore, $\mathbb{E}\left[\left\|\frac{1}{t}\sum_{s=1}^{t} f_s\right\|_2\right] \to L$ as $t \to \infty$. In the experiments we picked $L \in \{0, 0.1\}$.

The results are shown in Fig. 4. On the left-hand side we plotted the regret against the logarithm of the number of rounds, while on the right-hand side we plotted the regret against the square root of the number of rounds, together with the standard deviation of the results over the 100 independent runs. As can be seen from the figures, when $L = 0.1$, the growth-rate of the regret of FTL is indeed logarithmic, while when $L = 0$, the growth-rate is $\Theta(\sqrt{n})$. In particular, when $L = 0.1$, FTL enjoys a major advantage compared to FTRL, while for $L = 0$, FTL and FTRL perform essentially the same (in this special case, the regret of FTL will indeed be $O(\sqrt{n})$ as w_t will stay bounded, but $\|\Theta_t\| = O(1/\sqrt{t})$). As expected, AB(FTL,FTRL), gets the better of the two regrets with little to no extra penalty.

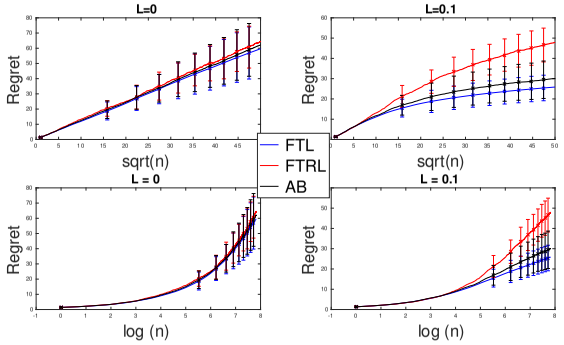


Figure 4: Regret of FTL, FTRL and AB(FTL,FTRL) against time for stochastic data.

5.2 "Half-Adversarial" Data

The half-adversarial data used in this experiment is the optimal solution for the adversary in the *linear game* when $\mathcal{W}$ is the unit ball (Abernethy et al., 2008). This data is generated as follows: The sequence $\hat{f}_t$ for $t = 1, \dots, n$ is generated randomly in the $(d-1)$-dimensional subspace $S = \text{span}\{e_2, \dots, e_d\}$ (here e_i is the ith unit vector in $\mathbb{R}^d$) as follows: $\hat{f}_1$ is drawn from the uniform distribution on the unit sphere of S (actually $\mathbb{S}^{d-1}$. For $t = 2, \dots, n$, $\hat{f}_t$ is drawn from the uniform distribution on the

unit sphere of the intersection of S and the hyperplane perpendicular to $\sum_{i=1}^{t-1} \hat{f}_i$ and going through the origin. Then, $f_t = Le_1 + \sqrt{1 - L^2}\hat{f}_t$ for some $L \geq 0$.

The results are reported in Fig. 5. When $L = 0$, the regret of both FTL and FTRL grows as $O(\sqrt{n})$. When $L = 0.1$, FTL achieves $O(\log n)$ regret, while the regret of FTRL appears to be $O(\sqrt{n})$. AB(FTL,FTRL) closely matches the regret of FTL.

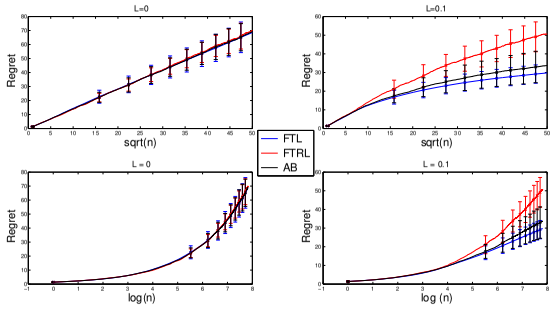


Figure 5: Experimental results for "half-adversarial" data.

5.3 Worst-Case Data

We also tested the algorithms on data where FTL is known to suffer linear regret, mainly to see how well AB(FTL,FTRL) is able to deal with this setting. In this case, we set $f_{t,i} = 0$ for all t and $i \geq 2$, while for the first coordinate, $f_{1,1} = 0.9$, and $f_{t,1} = 2(t \mod 2) - 1$ for $t \geq 2$.

The results are reported in Fig. 6. It can be seen that the regret of FTL is linear (as one can easily verify theoretically), and AB(FTL,FTRL) succeeds to adapt to FTRL, and they both achieve a much smaller $O(\sqrt{n})$ regret.

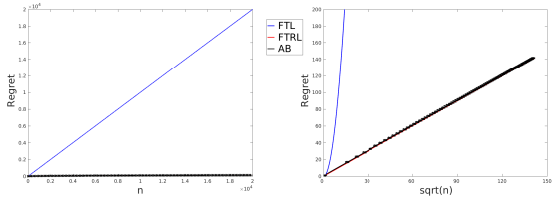


Figure 6: Experimental results for worst-case data.

5.4 The Unit Ball

We close this section by comparing the performance of our adaptive algorithms on the unit ball, namely, FTL, FTSL, FTLR, and AB(FTL,FTRL). All these algorithms are parametrized as above. The problem setup is similar to the stochastic data setting and the worst-case data setting. Again, we consider a 4-dimensional setting, that is, $\mathcal{W}$ is the unit ball in $\mathbb{R}^4$ centered at the origin. The

worst-case data is generated exactly as above, while the generation process of the stochastic data is slightly modified to increase the difference between FTLR and FTL: we sample the i.i.d. vectors $\hat{f}_t$ from a zero-mean normal distribution with independent components whose variance is $1/16$, and let $\tilde{f}_t = \hat{f}_t$ if $\|\hat{f}_t\|_2 \le 1$ and $\tilde{f}_t = \hat{f}_t / \left\|\hat{f}_t\right\|_2$ when $\left\|\hat{f}_t\right\|_2 > 1$ (i.e., we only normalize if $\hat{f}_t$ falls outside of the unit ball). The reason of this modification is to encourage the occurrence of the event $\|F_{t-1}\|_2 < \sqrt{t-1}$. Recall that when $\|F_{t-1}\|_2 \ge \sqrt{t-1}$, the prediction of FTRL matches that of FTL, so we are trying to create some data where their behavior is actually different. As a result, we will be able to observe that the predictions of FTL and FTRL are different in the early rounds. Finally, as before, we let $f_t = \tilde{f}_t + Le_1$, and set the time horizon to $n = 20,000$.

The results of the simulation of the stochastic data setting are shown in Figure 7. In the case of $L = 0.1$, FTRL suffers more regret at the beginning for some rounds, but then succeeds to match the performance of FTL. The results of the simulation of the worst-case data setting are shown in Figure 8, where FTSL has similar performance as FTRL.

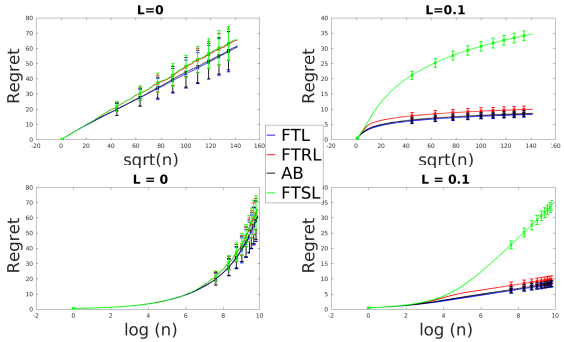


Figure 7: Experimental results for stochastic data when $\mathcal{W}$ is the unit ball.

6. Conclusion

FTL is a simple method that is known to perform well in many settings, while existing worst-case results fail to explain its good performance. While taking a thorough look at why and when FTL can be expected to achieve small regret, we discovered that the curvature of the boundary of the constraint and having average loss vectors bounded away from zero help keep the regret of FTL small. These conditions are significantly different from previous conditions on the curvature of the loss functions which have been considered extensively in the literature. It would be interesting to further investigate this phenomenon for other algorithms or in other learning settings.

Acknowledgments

The authors would like to thank the anonymous reviewers for their comments that helped improving the presentation of the paper, in particular, for the alternative proof of (6) based on strong convexity. This work was supported in part by the Alberta Innovates Technology Futures through the Alberta Ingenuity Centre for Machine Learning and by NSERC. During part of this work, T. Lattimore was

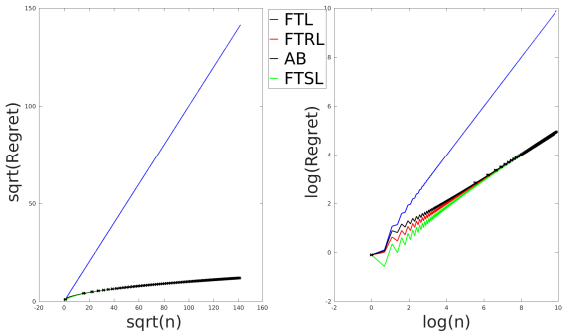


Figure 8: Experimental results for worst-case data when $\mathcal{W}$ is the unit ball.

with the Department of Computing Science, University of Alberta and with the School of Informatics and Computing, Indiana University Bloomington.

Appendix A. Technical Results

A.1 Proof of Proposition 1

Under the extra condition that $\mathcal{W}$ is compact the result follows from Danskin's theorem (e.g., Proposition B.25 of Bertsekas 1999). However, compactness is not required. For completeness, we provide a short, direct proof. We need to show that $\mathcal{Z} = \partial\varphi(\Theta)$ where recall that

$$\partial\varphi(\Theta) = \left\{u \in \mathbb{R}^d \,|\, \varphi(\Theta) + \langle u, \cdot - \Theta\rangle \le \varphi(\cdot)\right\} = \left\{u \in \mathbb{R}^d \,|\, \varphi(\Theta) \le \langle u, \Theta\rangle + \varphi(\cdot) - \langle u, \cdot\rangle\right\} .$$

Since $\mathcal{Z} \subset \mathcal{W}$, if $w \in \mathcal{Z}$, $\varphi(\Theta') \ge \langle w, \Theta'\rangle$ for any Θ' by the definition of φ. Hence, $\varphi(\Theta) = \langle w, \Theta\rangle \le \langle w, \Theta\rangle + \varphi(\Theta') - \langle w, \Theta'\rangle$ for any Θ', implying that $w \in \partial\varphi(\Theta)$.

On the other hand, assume $w \in \partial\varphi(\Theta)$. Then $\varphi(\Theta) \le \langle w, \Theta\rangle$ since $\varphi(0) = \langle w, 0\rangle = 0$. Since $\mathcal{W}$ is closed, $\mathcal{Z}$ is also closed. Therefore, if $w \notin \mathcal{Z}$, the strict separation theorem (applied to $\{w\}$, a convex compact set, and $\mathcal{Z}$, a convex closed set) implies that there exists $\rho \in \mathbb{R}^d$ such that $\langle z, \rho\rangle < \langle w, \rho\rangle$ for all $z \in \mathcal{Z}$. Let $\Theta' = \Theta + \rho$. Then, $\varphi(\Theta') = \max_{u\in\mathcal{W}} \langle u, \Theta\rangle + \langle u, \rho\rangle < \varphi(\Theta) + \langle w, \Theta' - \Theta\rangle \le \langle w, \Theta'\rangle \le \varphi(\Theta')$, a contradiction. Hence, $w \in \mathcal{Z}$.

A.2 Preliminaries in Differential Geometry

In this section we present some extra details on the tools we use from differential geometry. We focus on providing the intuitive picture and foundational results used in the paper, omitting formal definitions that do not directly contribute to this goal. The interested reader can find a detailed formal treatment, for example, in Section 2.5 of the book of Schneider (2014).

A.2.1 Planar Curves

We only consider twice continuously differentiable curves in this paper, defined as injective twice continuously differentiable functions $\gamma : [a, b] \to X$ from an interval $[a, b] \subset \mathbb{R}$ to a differentiable manifold $X \subset \mathbb{R}^d$ such that $\gamma'(u) \ne 0$. Given a differentiable curve γ, we define its length between $\gamma(u)$ and $\gamma(v)$ as $\ell([u, v]) = \int_u^v \|\gamma'(s)\| ds$ for $u, v \in [a, b]$; throughout this section $\|\cdot\|$ denotes the

Euclidean norm, and—with a slight abuse of notation—we will also use ℓ to denote the length of a curve or an interval. Given a twice continuously differentiable bijective mapping $r : [a,b] \to J$ (where $J \subset \mathbb{R}$ is an interval), γ can be reparametrized as a twice continuously differentiable function from J to X, by $\tilde{\gamma}(\ell) = \gamma(r^{-1}(\ell))$. In particular, when the mapping is $r(u) = \ell([a,u])$, that is, the curve is reparametrized by the curve length, we have $r^{-1}(\ell([a,u])) = u$ and $r'(u) = \|\gamma'(u)\| > 0$. Moreover, since $\ell = r(r^{-1}(\ell))$, we also have $1 = \frac{d\, r(r^{-1}(\ell))}{d\,\ell} = r'(u)\frac{d\, r^{-1}(\ell)}{d\,\ell}$ where $r(u) = \ell$. Thus, $\frac{d\, r^{-1}(\ell)}{d\,\ell} = \frac{1}{r'(u)} = \frac{1}{\|\gamma'(u)\|}$, and so

$$\|\tilde{\gamma}'(\ell)\| = \left\| \frac{d\,\gamma(r^{-1}(\ell))}{d\,\ell} \right\| = \left| \frac{d\, r^{-1}(\ell)}{d\,\ell} \right| \|\gamma'(u)\| = 1.$$

Thus, if the curve is parametrized by the its length, its gradient is always a unit vector. For the rest of this section, we always assume this parametrization.

A planar curve is a curve in a 2-dimensional plane. Given a point $\gamma(u)$ on the curve, one can compute its tangent vector in the plane by $\gamma'(u)$. Note that since γ is parametrized by its curve length, $\gamma'(u)$ is a unit vector. To measure how *curved* a planar curve is at point $\gamma(u)$, we define its curvature as

$$\kappa(\gamma(u)) = \left\| \frac{d\,\gamma'(u)}{d\,u} \right\| = \|\gamma''(u)\|_2.$$

Furthermore, since $\gamma'(u)$ is a unit vector,

$$0 = \frac{d\,1}{d\,u} = \frac{d\,\|\gamma'(u)\|^2}{d\,u} = \frac{d\,\langle \gamma'(u),\, \gamma'(u)\rangle}{d\,u} = 2\langle \gamma'(u),\, \gamma''(u)\rangle,$$

thus $\gamma'(u)$ is perpendicular to $\gamma''(u)$.

A.2.2 Manifolds, Tangent Plane, and Principal Curvature

A manifold M of dimension d is a Hausdorff topological space that is locally homeomorphic to $\mathbb{R}^d$. Given a convex body (a convex body is a compact, convex subset of $\mathbb{R}^d$ with non-empty interior) $\mathcal{W} \subset \mathbb{R}^d$, its boundary is a manifold $M = \mathrm{bd}(\mathcal{W})$ of dimension $d-1$. Assume that M is twice continuously differentiable, and let ζ denote the standard embedding map from M to $\mathcal{W}$.[5] Now let $\gamma_1, \gamma_2 : [-1,1] \to M$ be two curves (not necessarily parametrized by curve length), such that $\gamma_1(0) = \gamma_2(0) = w$. The two curves are equivalent at the point w if and only if their derivatives are equal at the point u, that is, $(\zeta \circ \gamma_1)'(0) = (\zeta \circ \gamma_2)'(0)$ and the tangent vector embedded in $\mathbb{R}^d$ associated with this equivalence class is $(\zeta \circ \gamma_1)'(0)$. The set of all the tangent vectors form the tangent space, denoted by $T_w M$ or $T_w \mathcal{W}$. One can verify that $T_w M$ is a $d-1$ dimensional hyperplane in $\mathbb{R}^d$. Note that this definition of tangent space is consistent with the 'natural' tangent plane in the Euclidean space $\mathbb{R}^3$.[6]

Since $T_w M$ is a $d-1$ dimensional hyperplane in $\mathbb{R}^d$, there exists a unique vector that is perpendicular to $T_w M$, is of length 1 and points outward of $\mathcal{W}$ (note that in this sense $M = \mathrm{bd}(\mathcal{W})$ is oriented): this vector is called the Gauss vector at point w for $\mathcal{W}$. The mapping $u_{\mathcal{W}} : \mathrm{bd}(\mathcal{W}) \to \mathbb{S}^{d-1}$ that maps every $w \in \mathrm{bd}(\mathcal{W})$ to the corresponding Gauss vector is called the Gauss map. Since M is twice continuously differentiable, the Gauss map $u_{\mathcal{W}}$ is continuously differentiable. One can actually show that $\nabla u_{\mathcal{W}}(w)$, the so-called Weingarten map, is a self-adjoint operator with nonnegative eigenvalues, which can be represented as a $(d-1)\times(d-1)$ positive semidefinite matrix. The eigenvalues of this matrix (or the self-adjoint operator) are called the principal curvatures of M at

5. A detailed discussion of manifolds including local charts and atlases can be found in Section 2.5 of the book of Schneider (2014), together with a formal definition of differentiability. To build the intuition required to follow the arguments in the paper it is sufficient to think of a manifold as the boundary of a convex body.
6. More generally, one can define the tangent space without embedding it to $\mathbb{R}^d$, but, for simplicity, we only consider here the definitions through the embedding ζ, which is always possible and allows to perform calculations in $\mathbb{R}^d$.

point w. Intuitively, how fast the Gauss map $u_{\mathcal{W}}$ changes characterizes the curvature of the manifold M. An interesting property of the operator $\nabla u_{\mathcal{W}}(w)$ is that it maps $T_w M$ to itself: for any unit vector $v \in T_w M$, $0 = \lim_{\epsilon\to 0} \frac{\partial 1}{\partial \epsilon} = \lim_{\epsilon\to 0} \frac{\partial \|u_{\mathcal{W}}(w+\epsilon v)\|_2^2}{\partial \epsilon} = 2\lim_{\epsilon\to 0}\langle \nabla u_{\mathcal{W}}(w+\epsilon v)v,\, u_{\mathcal{W}}(w+\epsilon v)\rangle = 2\langle \nabla u_{\mathcal{W}}(w)v,\, u_{\mathcal{W}}(w)\rangle$, thus $\nabla u_{\mathcal{W}}(w)v$ is perpendicular to $u_{\mathcal{W}}(w)$, thus belongs to $T_w M$.

A.3 Technical Proofs Related to Strongly Convex Sets and Principal Curvatures

Proof of Proposition 4 We show that (i) implies (ii), (ii) implies (iii), and (iii) implies (i). We start with showing that (i) implies (ii). First note that all principal curvatures of the d-dimensional ball $B = B_{1/\lambda}$ with radius $1/\lambda$ (centered at the origin) are λ. Therefore, (i) and Theorem 3.2.9 of Schneider (2014) implies that there is a convex body $\mathcal{M}$ such that $\mathcal{W} + \mathcal{M} = B$, where for two sets, $S_1, S_2 \subset \mathbb{R}^d$, $S_1 + S_2$ is defined as $\{s_1 + s_2 \,|\, s_1 \in S_1, s_2 \in S_2\}$. For any $\theta \in \mathbb{S}^{d-1}$, let $m_\theta \in \mathrm{argmax}_{m\in\mathcal{M}}\langle m, \theta\rangle$. Then clearly $w_\theta + m_\theta$ maximizes $\langle b, \theta\rangle$ for $b \in \mathcal{W} + \mathcal{M}$. Therefore, $\mathcal{W} + m_\theta$ is a subset of B and touches it at $w_\theta + m_\theta$, or equivalently $\mathcal{W} \subset B - m_\theta$ and they touch each other, and a tangent hyperplane with normal vector θ, in w_θ. This proves that (i) implies (ii).

Next we prove that (ii) implies (iii). Assuming (ii) holds, let $w \in \mathcal{W}$ be any point in the interior of $\mathcal{W}$, and let $p \in \mathrm{bd}(\mathcal{W})$ be the closest boundary point to w, and recall that $T_p\mathcal{W}$ is the tangent space of $\mathcal{W}$ at p. By construction, $B_{\|w-p\|_2}(w)$ touches the boundary of $\mathcal{W}$ at p (in the sense that they do not intersect, but they can have multiple common points), and so $w - p$ is orthogonal to $T_p\mathcal{W}$. Therefore, $B_{\|w-p\|_2}(w)$ also touches the boundary of the ball $B = B_{1/\lambda}(p + \frac{w-p}{\lambda\|w-p\|_2})$, which contains $\mathcal{W}$ by assumption (ii). Now consider any two points $x, y \in \mathcal{W}$ and $\gamma \in [0,1]$ such that $w = \gamma x + (1-\gamma)y$. Then the ball with radius $\lambda\gamma(1-\gamma)\|x-y\|_2^2/2$ centered at w is contained in B, since B is λ-strongly convex. But then its radius is at most $\|p - w\|_2$, and so it is also contained in $\mathcal{W}$. This shows that $\mathcal{W}$ is λ-strongly convex, thus (iii) holds.

To finish the proof of the proposition, assume (iii). To prove that (i) holds, we have to show that for any point w on $\mathrm{bd}(\mathcal{W})$ and for any unit vector $v \in T_w\mathcal{W}$, the curvature of the boundary along v is at least λ. Using the same notations as in Fig. 2 in Section 3.1, let P be the hyperplane spanned by v and the outer normal vector $u_{\mathcal{W}}(w)$ of $\mathcal{W}$ at point w, and consider the planar curve γ defined by $\mathrm{bd}(\mathcal{W}) \cap P$. Using v as the axis of a local 2-dimensional coordinate system, a point $\gamma(s)$ on the curve γ in the neighborhood of w can be expressed as $\gamma(s) = w + sv - f(s)u_{\mathcal{W}}(w)$, where w serves as the origin in the local coordinate system, f is the restriction of the function $f_w(sv)$ (see Section 3.1) to P (to simplify the notation, we denote it by $f(s)$, omitting v and w), and the curve γ is the epigraph of the function f, as in Fig. 9.

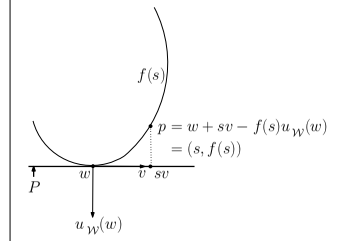


Figure 9: The local coordinate system at w.

Note that $f'(0) = 0$, and by Proposition 2.1 of Pressley (2010), the curvature of γ at p can be obtained as

$$\left.\frac{f''(s)}{\sqrt{1+f'(s)^2}^3}\right|_{s=0} = f''(0)\ .$$

Now since $w(s), w(-s) \in \mathcal{W}$ for a sufficiently small s, the strong convexity of $\mathcal{W}$ applied to $w(s)$ and $w(-s)$ with $\gamma = 1/2$ implies that $q = \frac{w(s)+w(-s)}{2} + \frac{\lambda}{8}\|w(s) - w(-s)\|_2^2 u \in \mathcal{W}$. Substituting the definition of $w(s)$ and $w(-s)$, we get

$$q = p - u\left[\frac{f(s)+f(-s)}{2} - \frac{\lambda}{8}\Big(4s^2 + (f(s)-f(-s))^2\Big)\right].$$

Therefore, $q \in \mathcal{W}$ implies $f(s) + f(-s) \geq \lambda s^2$, and so

$$f''(0) = \lim_{s \to 0} \frac{\frac{f(s)-f(0)}{s} - \frac{f(0)-f(-s)}{s}}{s} = \frac{f(s)+f(-s)}{s^2} \geq \lambda.$$

Thus (i) holds, finishing the proof of the proposition. ∎

Proof of a weakened variant of (6) based on strong convexity
Given θ_1 and θ_2, for any $0 < \gamma < 1$, define $\theta_\gamma = \gamma\theta_1 + (1-\gamma)\theta_2$ and

$$w_\gamma = \gamma w^{(1)} + (1-\gamma)w^{(2)} + \frac{\lambda_0}{2}\gamma(1-\gamma)\|w^{(1)} - w^{(2)}\|_2^2 \frac{\theta_\gamma}{\|\theta_\gamma\|_2} .$$

By the strong convexity of $\mathcal{W}$, $w_\gamma \in \mathcal{W}$, and so by the definition and convexity of the support function Φ, we have

$$\langle w_\gamma, \theta_\gamma \rangle \leq \Phi(\theta_\gamma) \leq \gamma\Phi(\theta_1) + (1-\gamma)\Phi(\theta_2) = \gamma\langle w^{(1)}, \theta_1\rangle + (1-\gamma)\langle w^{(2)}, \theta_2\rangle.$$

Plugging in the definitions of w_γ and θ_γ, and applying the Cauchy-Schwarz inequality, we obtain

$$\frac{\lambda_0}{2}\gamma(1-\gamma)\|w^{(1)} - w^{(2)}\|_2^2\|\theta_\gamma\|_2 \leq \gamma(1-\gamma)\langle \theta_1 - \theta_2, w^{(1)} - w^{(2)}\rangle \leq \gamma(1-\gamma)\|\theta_1 - \theta_2\|_2\|w^{(1)} - w^{(2)}\|_2 .$$

Rearranging and letting $\gamma \to 0$ implies

$$\|w^{(1)} - w^{(2)}\|_2 \leq \frac{2}{\lambda_0}\frac{\|\theta_1 - \theta_2\|_2}{\|\theta_2\|_2} .$$

Finally, the definition of $w^{(2)}$ and the Cauchy-Schwarz inequality yields

$$\begin{aligned}\langle w^{(1)} - w^{(2)}, \theta_1\rangle &\leq \langle w^{(1)} - w^{(2)}, \theta_1\rangle + \underbrace{\langle w^{(2)} - w^{(1)}, \theta_2\rangle}_{\geq 0} = \langle w^{(1)} - w^{(2)}, \theta_1 - \theta_2\rangle \\ &\leq \|w^{(1)} - w^{(2)}\|_2\|\theta_1 - \theta_2\|_2 \leq \frac{2}{\lambda_0}\frac{\|\theta_1 - \theta_2\|_2^2}{\|\theta_2\|_2}, \end{aligned} \tag{24}$$

finishing the proof. ∎

Proof of Example 1 We start with proving the last statement, part (iv), which implies the rest. Fix $w \in \mathrm{bd}(\mathcal{W})$. Note that $\phi'(w)$ is a normal vector at w for $\mathrm{bd}(\mathcal{W})$, thus $T_w\mathcal{W} = \{v : \langle v, \phi'(w)\rangle\}$. Then the Gauss map $u_\mathcal{W}$ of $\mathcal{W}$ satisfies $u_\mathcal{W}(w) = \frac{\phi'(w)}{\|\phi'(w)\|_2}$ for $w \in \mathrm{bd}(\mathcal{W})$. According to Schneider (2014, page 105), the principal curvatures of $\mathcal{W}$ at w are the eigenvalues of the Weingarten map $W_w(v)$, which is a linear map from $T_w\mathcal{W}$ to itself defined through the derivative of $u_\mathcal{W}$: $W_w(v) = \langle \frac{du_\mathcal{W}}{dw}, v\rangle$. In our case,

$$W_w(v) = \left\langle \frac{du_\mathcal{W}}{dw}, v \right\rangle = \frac{\nabla^2\phi(w)v}{\|\phi'(w)\|_2} - \frac{\phi'(w)\nabla^2\phi(w)\phi'(w)^\top v}{\|\phi'(w)\|_2^3} = \frac{\nabla^2\phi(w)v}{\|\phi'(w)\|_2} ,$$

where in the last step we used that $\phi'(w)$ is orthogonal to the tangent space $T_w\mathcal{W}$ (since it is parallel to the normal vector $u_\mathcal{W}(w)$), and $v \in T_w\mathcal{W}$. Therefore, the smallest principal curvature at w is the smallest eigenvalue $\min_{v \in \mathbb{S}^{d-1}:\langle \phi'(w), v\rangle = 0} \frac{v^\top\nabla^2\phi(w)v}{\|\phi'(w)\|_2}$. Taking minimum over all $w \in \mathrm{bd}(\mathcal{W})$ finishes the proof.

Now part (ii) follows for $\phi(w) = w^\top Q w$, as we need to minimize $v^\top Q v/\|w^\top Q\|_2$. It is easy to see that the denominator is maximized when $w \in \mathrm{bd}(\mathcal{W})$ is an eigenvector of Q corresponding to

$\lambda_{\max}$ (with length $1/\sqrt{\lambda_{\max}}$), and the numerator is minimized (for arbitrary $v \in \mathbb{S}^{d-1}$) when v is an eigenvector of Q corresponding to $\lambda_{\min}$. Since the two eigenvectors, w and v are orthogonal (or can be chosen to be orthogonal if they are not unique), v is orthogonal to $\phi'(w) = Qw = \lambda_{\max}w$, and hence it is a valid minimizer. This completes the proof of part (ii), and part (i) follows as a special case.

Part (iii) follows similarly: Due to symmetry, it is enough to consider w in the nonnegative quadrant (i.e., $w_i \geq 0$ for all i). Calculating the first and second derivatives of $\phi(w) = \|w\|_p = \left(\sum_{i=1}^d w_i^p\right)^{1/p}$, for any $w \in \mathrm{bd}(\mathcal{W})$ (i.e., $\|w\|_p = 1$), we obtain

$$\phi'(w) = \left(\sum_{i=1}^d w_i^p\right)^{1/p-1} w^{\odot(p-1)} = w^{\odot(p-1)} \quad \text{and} \quad \nabla^2\phi(w) = (p-1)\,\mathrm{diag}\left(w_1^{p-2}, \cdots, w_d^{p-2}\right).$$

When $p > 2$, picking $w = (1, 0, 0, \cdots, 0)$, one can easily verify that $\lambda_0 = 0$. For $1 < p \leq 2$, $|w_i|^{p-2} \geq 1$ since $|w_i| \leq 1$ by the assumption that $\|w\|_p = 1$. Thus, the minimum eigenvalue of $\mathrm{diag}\left(w_1^{p-2}, \cdots, w_d^{p-2}\right)$ is at least 1, and so $\lambda_0 \geq (p-1)/\|w^{\odot(p-1)}\|_2$ since $v^\top\,\mathrm{diag}\left(w_1^{p-2}, \cdots, w_d^{p-2}\right)v \geq 1$. Defining q via $1/p + 1/q = 1$, for any $w \in \mathrm{bd}(\mathcal{W})$ (i.e., with $\|w_p\| = 1$), Hölder's inequality implies

$$\|w^{\odot(p-1)}\|_2 \leq d^{\frac{1}{2}-\frac{1}{q}}\|w^{\odot(p-1)}\|_q = d^{\frac{1}{2}-\frac{1}{q}}\left(\sum_{i=1}^d w_i^{(p-1)q}\right)^{\frac{1}{q}} = d^{\frac{1}{2}-\frac{1}{q}}\left(\sum_{i=1}^d w_i^p\right)^{\frac{1}{q}} = d^{\frac{1}{2}-\frac{1}{q}} = d^{\frac{1}{p}-\frac{1}{2}} .$$

Thus, $\lambda_0 \geq (p-1)d^{\frac{1}{2}-\frac{1}{p}}$, as desired. ∎

A.4 Technical Lemmas for the Lower Bound, Theorem 9

Lemma 17 *Under the assumptions of Theorem 9, for any $0 < P_1, P_2 < 1$,*

$$\left\langle w^{P_2} - w^{P_1}, f^{P_1}\right\rangle \geq \frac{\lambda L}{2}\frac{\left(\frac{2P_2-2P_1}{\lambda L}\right)^2}{\sqrt{1 + \left(\frac{1-2P_1}{\lambda L}\right)^2\left(1 + \left(\frac{1-2P_2}{\lambda L}\right)^2\right)}} .$$

Proof It is easy to see that for any p, w^p is on the boundary of $\mathcal{W}$, that is, $w^p = \mathrm{argmin}_{w\in\mathcal{W}}\langle w, f^p\rangle = (\cos(\varphi^p), \lambda\sin(\varphi^p))$ for some φ^p. Then $\langle w^p, f^p\rangle = (2p-1)\cos(\varphi^p) - \lambda L\sin(\varphi^p)$, and so taking the derivative it is easy to verify that $\tan(\varphi^p) = \frac{\lambda L}{1-2p}$ and $\sin(\varphi^p) = \frac{\lambda L}{\sqrt{(\lambda L)^2 + (1-2p)^2}} > 0$. Thus, $1 - 2P_1 = \frac{\lambda L\cos(\varphi^{P_1})}{\sin(\varphi^{P_1})}$. To simplify notation, let $\varphi_1 = \varphi^{P_1}$ and $\varphi_2 = \varphi^{P_2}$. Then,

$$\begin{aligned}\left\langle w^{P_2} - w^{P_1}, f^{P_1}\right\rangle &= \left\langle \begin{pmatrix} \cos\varphi_2 - \cos\varphi_1 \\ \lambda(\sin\varphi_2 - \sin\varphi_1)\end{pmatrix}, \begin{pmatrix} \frac{-\lambda L\cos\varphi_1}{\sin\varphi_1} \\ -L\end{pmatrix}\right\rangle \\ &= -\lambda L\left((\cos(\varphi_2) - \cos(\varphi_1))\frac{\cos(\varphi_1)}{\sin(\varphi_1)} + (\sin(\varphi_2) - \sin(\varphi_1))\right) \\ &= \frac{-\lambda L}{\sin(\varphi_1)}\left(\cos(\varphi_2)\cos(\varphi_1) - \cos^2(\varphi_1) + \sin(\varphi_1)\sin(\varphi_2) - \sin^2(\varphi_1)\right) \\ &= \frac{\lambda L}{\sin(\varphi_1)}\left(1 - \cos(\varphi_2)\cos(\varphi_1) - \sin(\varphi_1)\sin(\varphi_2)\right) \\ &= \frac{\lambda L}{\sin(\varphi_1)}\left(1 - \cos(\varphi_1 - \varphi_2)\right) \\ &= \frac{\lambda L}{\sin(\varphi_1)}\left(\frac{1}{2}\left(\cos(\varphi_1 - \varphi_2) - 1\right)^2 + \frac{1}{2}\sin^2(\varphi_1 - \varphi_2)\right)\end{aligned}$$

$$\geq \frac{\lambda L}{2\sin(\varphi_1)}\sin^2(\varphi_1-\varphi_2)$$
$$= \frac{\lambda L}{2}\sin(\varphi_1)\sin^2\varphi_2\left(\cot(\varphi_1)-\cot(\varphi_2)\right)^2 .$$

The proof is finished by substituting $\cot(\varphi_i) = \frac{1-2P_i}{\lambda L}$, $\sin(\varphi_1) = \frac{1}{\sqrt{1+\left(\frac{1-2P_1}{\lambda L}\right)^2}}$ and $\sin^2(\varphi_2) = \frac{1}{1+\left(\frac{1-2P_2}{\lambda L}\right)^2}$. ■

Lemma 18 (Concentration of $\hat{P}_t$) *For any $u > 0$,*

$$\mathbb{P}\left[|\hat{P}_t - P| > \frac{K}{2K+t}|1-2P| + \frac{t}{2K+t}u \,\middle|\, P\right] \leq 2\exp(-tu^2) .$$

Proof Recall that $\hat{P}_t = \frac{K+\sum_{i=1}^t X_i}{2K+t}$. Thus,

$$\begin{aligned}\mathbb{P}\left[|\hat{P}_t - P| > u \,\middle|\, P\right] &= \mathbb{P}\left[\left|\frac{K+\sum_{i=1}^t X_i}{2K+t} - P\right| > \frac{K}{2K+t}|1-2P| + \frac{t}{2K+t}u \,\middle|\, P\right] \\ &= \mathbb{P}\left[\left|\sum_{i=1}^t X_i - Pt + K(1-2P)\right| > K|1-2P| + tu \,\middle|\, P\right] \\ &\leq \mathbb{P}\left[\left|\sum_{i=1}^t X_i - Pt\right| > tu \,\middle|\, P\right], \end{aligned} \tag{25}$$

where the last inequality is due to $\mathbb{P}\left[|A+b| > c\right] \leq \mathbb{P}\left[|A| > c - |b|\right]$. Note that conditioned on P, $X_1, \ldots, X_t$ are independent Bernoulli random variables with expectation P, thus (25) holds by Hoeffding's inequality (see, e.g., (Cesa-Bianchi and Lugosi, 2006, Corollary A.1)). ■

Lemma 19

$$\mathbb{E}\left[(P-\hat{P}_t)^2 \,\middle|\, P\right] = \frac{K^2(1-2P)^2}{(2K+t)^2} + \frac{tP(1-P)}{(2K+t)^2} .$$

Proof Recall that $\hat{P}_t = \frac{K+\sum_{i=1}^t X_i}{2K+t}$.Thus,

$$\begin{aligned}\mathbb{E}\left[(P-\hat{P}_t)^2 \,\middle|\, P\right] &= \mathbb{E}\left[\left(\frac{K(1-2P)}{2K+t} + \frac{\sum_{i=1}^t X_i - Pt}{2K+t}\right)^2 \,\middle|\, P\right] \\ &= \frac{K^2(1-2P)^2}{(2K+t)^2} + \frac{1}{(2K+t)^2}\mathbb{E}\left[\left(\sum_{i=1}^t X_i - tP\right)^2 \,\middle|\, P\right] \\ &= \frac{K^2(1-2P)^2}{(2K+t)^2} + \frac{tP(1-P)}{(2K+t)^2} ,\end{aligned}$$

where the second equality is due to $\mathbb{E}\left[\sum_{i=1}^t X_i - Pt \,\middle|\, P\right] = 0$, and the last equality is due to that conditioned on P, $\sum_{i=1}^t X_i$ has a Binomial distribution with parameters t and P. ■

References

Y. Abbasi-Yadkori. Forced-exploration based algorithms for playing in bandits with large action sets. Master's thesis, University of Alberta, 2009.

J. Abernethy, P.L. Bartlett, A. Rakhlin, and A. Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Conference on Learning Theory (COLT)*, pages 415–424, 2008.

J. Abernethy, C. Lee, A. Sinha, and A. Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory (COLT)*, pages 807–823, 2014.

P.L. Bartlett, E. Hazan, and A. Rakhlin. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems (NIPS)*, pages 65–72, 2007.

D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. Information Theory*, 50(9):2050–2057, 2004.

D.J. Foster, A. Rakhlin, and K. Sridharan. Adaptive online learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3357–3365, 2015.

Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.

A.A. Gaivoronski and F. Stella. Stochastic nonstationary optimization for finding universal portfolios. *Annals of Operations Research*, 100(1–4):165–188, 2000.

D. Garber and E. Hazan. Faster rates for the Frank-Wolfe method over strongly-convex sets. In *International Conference on Machine Learning (ICML)*, pages 541–549, 2015.

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

R. Huang, T. Lattimore, A. György, and Cs. Szepesvári. Following the leader and fast rates in linear prediction: curved constraint sets and other regularities. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4970–4978. 2016.

M. Journée, Y. Nesterov, P. Richtárik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *Journal of Machine Learning Research*, 11:517–553, 2010.

S. M. Kakade and S. Shalev-Shwartz. Mind the duality gap: Logarithmic regret algorithms for online optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1457–1464, 2009.

Wouter M. Koolen, Peter Grünwald, and Tim van Erven. Combining adversarial guarantees and stochastic fast rates in online learning. *CoRR*, abs/1605.06439, 2016. URL `http://arxiv.org/abs/1605.06439`.

W. Kotłowski. Minimax strategy for prediction with expert advice under stochastic assumptions. *Algorithmic Learning Theory (ALT)*, 2016.

E.S. Levitin and B.T. Polyak. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50, 1966.

H.B. McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and implicit updates. arXiv, 2010. URL `http://arxiv.org/abs/1009.3240`.

N. Merhav and M. Feder. Universal sequential learning and decision from individual data sequences. In *ACM Workshop on Computational Learning Theory (COLT)*, pages 413—427, 1992.

F. Orabona, N. Cesa-Bianchi, and C. Gentile. Beyond logarithmic bounds in online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 823–831, 2012.

E. S. Polovinkin. Strongly convex analysis. *Sbornik: Mathematics*, 187(2):259, 1996.

A. N. Pressley. *Elementary differential geometry*. Springer Science & Business Media, 2010.

A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In *Conference on Learning Theory (COLT)*, pages 993–1019, 2013.

A. Sani, G. Neu, and A. Lazaric. Exploiting easy data in online optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 810–818, 2014.

R. Schneider. *Convex Bodies: The Brunn–Minkowski Theory*. Encyclopedia of Mathematics and its Applications. Cambridge Univ. Press, 2nd edition, 2014.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2012.

S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

T. van Erven, P. Grünwald, N. Mehta, M. Reid, and R. Williamson. Fast rates in statistical and online learning. *Journal of Machine Learning Research (JMLR)*, 16:1793–1861, 2015. Special issue in Memory of Alexey Chervonenkis.

Journal of Machine Learning Research 18 (2017) 1-48 Submitted 8/16; Revised 10/17; Published 11/17

Regularization and the small-ball method II: complexity dependent error rates

Guillaume Lecué GUILLAUME.LECUE@ENSAE.FR
ENSAE
5 avenue Henry Le Chatelier
91120 Palaiseau, France

Shahar Mendelson SHAHAR@TX.TECHNION.AC.IL
Department of Mathematics
Technion, Haifa, Israel

Editor: Zhihua Zhang

Abstract

We study estimation properties of regularized procedures of the form

$$\hat{f} \in \operatorname*{argmin}_{f \in F} \Big(\frac{1}{N} \sum_{i=1}^{N} \big(Y_i - f(X_i)\big)^2 + \lambda \Psi(f) \Big)$$

for a convex class of functions F, regularization function $\Psi(\cdot)$ and some well chosen regularization parameter λ, where the given data is an independent sample $(X_i, Y_i)_{i=1}^N$.

We obtain bounds on the L_2 estimation error rate that depend on the complexity of the true model $F^* := \{f \in F : \Psi(f) \leq \Psi(f^*)\}$, where $f^* \in \operatorname{argmin}_{f \in F} \mathbb{E}(Y - f(X))^2$ and the (X_i, Y_i)'s are independent and distributed as (X, Y). Our estimate holds under weak stochastic assumptions – one of which being a small-ball condition satisfied by F – and for rather flexible choices of regularization functions $\Psi(\cdot)$. Moreover, the result holds in the learning theory framework: we do not assume any a-priori connection between the output Y and the input X.

As a proof of concept, we apply our general estimation bound to various choices of Ψ, for example, the ℓ_p and S_p-norms (for $p \geq 1$), weak-ℓ_p, atomic norms, max-norm and SLOPE. In many cases, the estimation rate almost coincides with the minimax rate in the class F^*.

Keywords: Empirical processes theory, high-dimensional Statistics, regularization, learning theory, minimax rates.

1. Introduction

In the standard learning framework, one would like to approximate / predict an unknown random variable Y using functions from a given class F, and to do so using only random data. To be more accurate, let $(\mathcal{X}, \mu)$ be a probability space and consider a class of functions F on $(\mathcal{X}, \mu)$. Let X be distributed according to μ and set $X_1, \ldots, X_N \in \mathcal{X}$ to be N independent copies of X.

Given an unknown random variable Y, let $\mathcal{D} = (X_i, Y_i)_{i=1}^N$ be a sample selected according to the joint distribution of (X, Y). One would like to use the data $\mathcal{D}$ and construct a (random) function $\hat{f}(\cdot) = \hat{f}(\mathcal{D}, \cdot) \in F$, with $\hat{f}(X)$ serving as a good guess of Y.

©2017 Guillaume Lecué and Shahar Mendelson.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/16-422.html.

While there are various interpretations of the meaning of 'a good guess', the notion we will focus on here is as follows.

In a typical problem, very little is assumed on the target Y or on the measure μ; on the other hand, the class F is known and a typical assumption is that F **is convex and closed** in $L_2(\mu)$. Therefore, the functional $f \to \mathbb{E}(f(X) - Y)^2$ has a unique minimizer in F,

$$f^* = \operatorname*{argmin}_{f \in F} \mathbb{E}(Y - f(X))^2. \tag{1.1}$$

The notion of 'a good guess' is that $\hat{f}$ is close to f^* in $L_2(\mu)$, and one would like to obtain a high probability bound on the $L_2(\mu)$ distance of the form

$$\left\| \hat{f} - f^* \right\|_{L_2}^2 = \mathbb{E}\left[\big(f^*(X) - \hat{f}(X)\big)^2 | \mathcal{D} \right] \leq \alpha_N^2. \tag{1.2}$$

In this case, α_N^2 is called a *rate of convergence*, the *error rate* or the $L_2(\mu)$-*estimation rate* of the problem.

Clearly, one has to pay a price for allowing a rather general target Y. Also, to have any hope that $f^*(X)$ is reasonably close to Y, one has to consider large classes, leading to an error α_N^2 that is often too large to be of any use.

A possible way of bypassing the fact that F may be very large, is the classical approach to *regularization*, where a certain property one believes f^* to possess is emphasized by penalizing functions that do not have that property. The penalty is endowed via a *regularization function* $\Psi(\cdot)$, defined on an appropriate subspace $E \subset L_2(\mu)$ that contains F, and for which $\Psi(f^*)$ is believed to be small (though one does not know that for certain). As a consequence, regularization procedures are designed to fit the data and to have a small Ψ value at the same time. One way of achieving that is to search for functions in F that realize a good tradeoff between fitting that data, which is measured via an empirical loss function $P_N \ell_f$, and the size of the regularization term $\lambda \Psi(f)$.

Definition 1.1 *The Regularized Empirical Risk Minimization procedure (RERM) is defined by*

$$\hat{f} \in \operatorname*{argmin}_{f \in F} \big(P_N \ell_f + \lambda \Psi(f) \big), \tag{1.3}$$

where here and throughout the article, $P_N h$ denotes the empirical mean of h, ℓ_f is the loss function associated with f and λ is the so-called regularization parameter.

We only consider the square loss $\ell_f(x, y) = (y - f(x))^2$, and thus,

$$P_N \ell_f = \frac{1}{N} \sum_{i=1}^{N} (Y_i - f(X_i))^2.$$

A well known example to this, the "classical approach" to regularization, is the cubic smoothing spline that can be obtained with the choice

$$\Psi(f) = \int \big(f''(t)\big)^2 \, dt.$$

Another well-studied example is of the form

$$\Psi(f) = \int_{\mathbb{R}^d} \frac{\left|\bar{f}(t)\right|^2}{\bar{G}(t)} dt$$

where the integration is with respect to the Lebesgue measure, $\bar{f}$ is the Fourier transform of f and $\bar{G}$ is some positive function tending to zero when $|t|$ goes to infinity (cf. Girosi et al. (1995)). In fact, this type of regularization methods dates back to Tikhonov (Tikhonov (1943)) and is sometimes called Tikhonov regularization; it is also known as L_2-regularization or Ridge regularization (Golub et al. (1979)).

These methods and others like them have been used to "smooth" estimators that have poor generalization capability because of their tendency to over-fit the data, and for the corresponding regularization functions, having a small Ψ value is a guarantee of smoothness. We refer to Hastie et al. (2009) for other examples of regularization functions that have been used to "smooth" estimators.

We said "classical approach to regularization" because in the more modern approach the aim is somewhat different. One uses a penalty that seemingly has little to do with the property one wishes to emphasize (usually, some notion of *sparsity*). Yet somehow, almost magically, the penalty enhances a hidden property and the resulting error rate does not depend on $\Psi(f^*)$ but on that hidden property of f^*. We call such error rates **sparsity-dependent error rates**.

The first part of this article (Lecué and Mendelson (to appear)) has dealt with the modern approach to regularization. Here we would like to complete the picture by exploring bounds that depend on $\Psi(f^*)$ rather than on some hidden sparsity structure of f^*. Such error rates will be called **complexity-dependent error rates**, since the aim is to obtain rates of convergence that depend on the complexity of the unknown true model $\{f \in F : \Psi(f) \leq \Psi(f^*)\}$. Of course, the two approaches may sometimes be combined advantageously (see some examples below).

In this context, we will consider regularization functions that satisfy the following properties, which are more general than the ones considered in Lecué and Mendelson (to appear).

Assumption 1.1 *A function $\Psi : E \to \mathbb{R}_+$ is a regularization function if*

- *It is nonnegative, even, convex and $\Psi(0) = 0$.*
- *There is a constant $\eta \geq 1$, for which, for every $f, h \in E$,*

$$\Psi(f + h) \leq \eta(\Psi(f) + \Psi(h)).$$

- *For every $0 \leq \alpha \leq 1$ and $h \in E$, $\Psi(\alpha h) \leq \alpha \Psi(h)$.*

Remark 1.2 *Classical Model Selection regularization functions, such as the cardinality of the support of a vector or the rank of a matrix, are usually not convex and do not satisfy Assumption 1.1. Such examples are therefore not considered in what follows.*

1.1 Classical vs. modern

As mentioned above, the direction we take here is closely related to the classical approach to regularization and is rather different from the modern approach. To explain the differences we shall use the celebrated LASSO estimator (cf. Tibshirani (1996); Donoho and Johnstone (1994)) as an example.

Let F be a class of linear functionals on $\mathbb{R}^d$ of the form $\langle t, \cdot \rangle$. Set $t^* \in \operatorname{argmin}_{t \in \mathbb{R}^d} \mathbb{E}(Y - \langle X, t \rangle)^2$, and consider the RERM (1.3) with the ℓ_1^d-norm, $\|t\|_1 = \sum_{i=1}^d |t_i|$, serving as a regularization function. Let

$$\hat{t} \in \operatorname*{argmin}_{t \in \mathbb{R}^d} \left(\frac{1}{N} \sum_{i=1}^N \left(Y_i - \langle X_i, t \rangle \right)^2 + \lambda \|t\|_1 \right),$$

and the resulting minimizer is the LASSO estimator.

Estimation, de-noising, prediction and support recovery results have been obtained for the LASSO in the last decades (see, for example, Tibshirani (1996), Bickel et al. (2009), and the books Giraud (2015); Bühlmann and van de Geer (2011) and Koltchinskii (2011) for additional references).

The LASSO has been used in high-dimensional problems, in which the aim was to enhance a low-dimensional structure. The hope was that if the signal t^* were sparse (that is, supported on relatively few coordinates), the regularization procedure $\hat{t}$ would estimate t^* with an error rate depending on the cardinality of the support of t^*, denoted by $\|t^*\|_0 = |\{j \in \{1, \ldots, d\} : t_j^* \neq 0\}|$.

However, if t^* happens to be 'well-spread' rather than sparse, though with a reasonable ℓ_1^d norm, the sparsity-dependent error rate is useless, while a complexity-dependent error rate, which yields bounds in terms of $\|t^*\|_1$, is sharper. The obvious example is $t_1^* = (1/d, ..., 1/d)$ and $t_2^* = (1, 0, ..., 0)$: although $\|t_1^*\|_1 = \|t_2^*\|_1 = 1$, the cardinalities of their supports are very different, and sparsity-dependent error rates when $t^* = t_1^*$ are likely to be bad.

Examples of that nature are the reason why error rates combining both sparsity and complexity have been obtained for the LASSO. A typical example is Corollary 9.1 in Koltchinskii (2011). To formulate it, Let $W_1, \cdots, W_N$ be N independent, centered subgaussian variables with variance σ^2 and set $x_1, \ldots, x_N$ to be N deterministic vectors in $\mathbb{R}^d$. Assume that the design matrix, $\Gamma = N^{-1/2} \sum_{i=1}^N \langle x_i, \cdot \rangle e_i$, whose rows are $x_i/\sqrt{N}$, satisfies some Restricted Isometry Property (cf. Candès et al. (2006)). If $Y_i = \langle x_i, t^* \rangle + W_i$, $i = 1, \ldots, N$, then for a well chosen regularization parameter λ, one has, with high probability,

$$\mathbb{E}\langle X, \hat{t} - t^* \rangle^2 \leq C \min \left\{ \frac{\sigma^2 \|t^*\|_0 \log d}{n}, \sigma \|t^*\|_1 \sqrt{\frac{\log d}{n}} \right\} \tag{1.4}$$

for a suitable absolute constant C.

The error rate from (1.4) consists of two components: the sparsity-dependent error term $\sigma^2(\|t^*\|_0 \log d)/n$, and the complexity-dependent error term $\sigma \|t^*\|_1 \sqrt{(\log d)/n}$, and in what follows we will present a few other examples that combine the two rates – because the procedure one uses to obtain both types of rate is the same.

The aim of this article is to address the "complexity-based" aspect of the problem: to study regularization problems in which one believes that the $\Psi(f^*)$ is relatively small, and obtain an error rate that depends on $\Psi(f^*)$ rather than on some sparsity property of f^*.

1.2 Attaining Minimax rates

A natural benchmark for measuring the success of a regularization method is the minimax error rate, assuming that $\Psi(f^*)$ is known: if one is given additional information on $\Psi(f^*)$, e.g., that $f^* \in \{f : \Psi(f) \leq R\}$, one may consider the estimation problem in $\{f : \Psi(f) \leq R\}$ using the given random data. Such a problem has an optimal error rate (called the minimax rate): it is the best rate any learning procedure may achieve in the class $\{f : \Psi(f) \leq R\}$ given the random data $(X_i, Y_i)_{i=1}^N$. This minimax rate will serve as our benchmark, and will be compared with the error rates that we obtain.

Of course, one *is not* given additional information on $\Psi(f^*)$ and it is reasonable to expect that the error rate of the regularization procedure will be significantly slower than this benchmark. The question we shall study here focuses on that gap. In fact, we will show that the price one has to pay for not knowing $\Psi(f^*)$ is surprisingly small, under rather weak assumptions.

From a technical perspective, all regularization-based procedures share one crucial aspect: the calibration of the regularization parameter λ. That choice is very important as λ is an essential component in ensuring that the error rate of the estimator $\hat{f}$ is well-behaved. Thus, to study the gap between the regularization error rate and the minimax rate, one has to identify the right choice of λ.

Question 1.3 *What is the 'correct choice' of the regularization parameter λ, and given that choice, what is the rate of convergence of RERM? Specifically, how far is the resulting rate from the one that could have been achieved had $\Psi(f^*)$ been given in advance?*

An answer to Question 1.3 requires one to identify λ; to find a high probability upper bound on $\|\hat{f} - f^*\|_{L_2(\mu)}^2$ where $\hat{f}$ is defined in Definition 1.1 for that choice of λ; and then to compare the error rate to the minimax rate of the estimation problem in the class $\{f \in F : \Psi(f) \leq \Psi(f^*)\}$ (the so-called true model).

The strategy we use below follows a similar path to Lecué and Mendelson (to appear) and is based on the small ball method, introduced in Mendelson (2015, 2014); Koltchinskii and Mendelson (2015); Mendelson (To appear).

1.3 The small-ball method

Given a closed and convex class F and an unknown target Y, recall that $f^* \in F$ is a minimizer in F of the functional $f \to \mathbb{E}(f(X) - Y)^2$.

The excess loss functional associated with $f \in L_2(\mu)$ is

$$\begin{aligned} f \to \mathcal{L}_f(X,Y) =& \ell_f(X,Y) - \ell_{f^*}(X,Y) = (f(X) - Y)^2 - (f^*(X) - Y)^2 \\ =& (f - f^*)^2(X) + 2(f^*(X) - Y)(f - f^*)(X). \end{aligned} \tag{1.5}$$

Moreover, since F is closed and convex, then by the characterization of the nearest point map in a Hilbert space,

$$\mathbb{E}(f^*(X) - Y)(f - f^*)(X) \geq 0 \quad \text{for every } f \in F;$$

thus

$$\frac{1}{N}\sum_{i=1}^N (f^*(X_i) - Y_i)(f - f^*)(X_i) \geq \frac{1}{N}\sum_{i=1}^N (f^*(X_i) - Y_i)(f - f^*)(X_i) - \mathbb{E}(f^*(X) - Y)(f - f^*)(X). \tag{1.6}$$

Let E be a subspace that contains F and set $\Psi(\cdot)$ to be a regularization function on E (i.e., a functional that satisfies Assumption 1.1). Set $\rho \geq 0$ and put

$$K_\rho(f^*) = \{h \in E : \Psi(h - f^*) \leq \rho\},$$

which, by the convexity of Ψ, is a convex set.

Definition 1.4 *For every $\lambda > 0$ and any $f \in L_2(\mu)$, define the regularized excess loss by*

$$\mathcal{L}_f^\lambda = (\ell_f + \lambda\Psi(f)) - (\ell_{f^*} + \lambda\Psi(f^*)) = \mathcal{L}_f + \lambda\left(\Psi(f) - \Psi(f^*)\right).$$

Note that for every sample $(X_i, Y_i)_{i=1}^N$, a minimizer $\hat{f}$ of the empirical regularized loss functional (1.3) also minimizes in F the empirical regularized excess loss $f \to P_N\mathcal{L}_f^\lambda$. Hence, since $\mathcal{L}_{f^*}^\lambda = 0$, it follows that for every $(X_i, Y_i)_{i=1}^N$, the empirical regularized excess loss in $\hat{f}$ is non-positive:

$$P_N\mathcal{L}_{\hat{f}}^\lambda \leq 0. \tag{1.7}$$

This observation is at the heart of our analysis, as it allows one to exclude functions f in F that satisfy $P_N\mathcal{L}_f^\lambda > 0$ as potential minimizers of the empirical regularized loss function. Our strategy is therefore to show that if $f \in F$ and $\|f - f^*\|_{L_2(\mu)}$ is not 'too small', then necessarily $P_N\mathcal{L}_f^\lambda > 0$ (for the right choice of λ); hence, functions cannot be minimizers of the empirical regularized (excess) loss function.

To simplify notation, set $\xi = Y - f^*(X)$,

$$\mathcal{M}_{f-f^*}(X,Y) = \xi(f - f^*)(X) - \mathbb{E}\xi(f - f^*)(X) \quad \text{and} \quad \mathcal{Q}_{f-f^*}(X) = (f - f^*)^2(X);$$

therefore, combining (1.5) and (1.6),

$$P_N\mathcal{L}_f \geq P_N\mathcal{Q}_{f-f^*} - 2\left|P_N\mathcal{M}_{f-f^*}\right|. \tag{1.8}$$

The main step in the small-ball method is to find a lower bound on the quadratic process $f \to P_N\mathcal{Q}_{f-f^*}$ and an upper bound on $f \to |P_N\mathcal{M}_{f-f^*}|$. The two estimates should hold with high probability on certain subsets of F. Then, they have to be compared with the behaviour of the regularization term $\lambda(\Psi(f) - \Psi(f^*))$ on those sets to ensure that $P_N\mathcal{L}_f^\lambda > 0$.

A uniform lower bound on the quadratic component $P_N\mathcal{Q}_{f-f^*}$ can be obtained under the following small-ball condition:

Assumption 1.2 *Assume that there are constants $\kappa > 0$ and $0 < \varepsilon \leq 1$, for which, for every $f, h \in F$,*

$$Pr\left(|f(X) - h(X)| \geq \kappa \|f - h\|_{L_2(\mu)}\right) \geq \varepsilon.$$

There are numerous examples in which Assumption 1.2 may be verified for κ and ε that are absolute constants and we refer the reader to Mendelson (2014, 2015); Lecué and Mendelson (2017); Mendelson (To appear); Koltchinskii and Mendelson (2015); Rudelson and Vershynin (2015) for some of them.

To put Assumption 1.2 in some perspective, recall that the class $F = \{f_t = \langle \cdot, t\rangle : t \in \mathbb{R}^d\}$ is *identifiable* if for every $t_1, t_2 \in \mathbb{R}^d$, $Pr(f_{t_1} \neq f_{t_2}) > 0$, (where the probability is taken with respect to the underlying measure μ). By linearity, this condition is equivalent to assuming that for every $t \in \mathbb{R}^d$, $Pr(|\langle X, t\rangle| > 0) > 0$. Thus, the small-ball condition (which in this case implies that for every $t \in \mathbb{R}^d$, $Pr(|\langle X, t\rangle| \geq \kappa \|\langle X, t\rangle\|_{L_2}) \geq \varepsilon$) is simply a uniform estimate on the degree of identifiability of class F and is therefore a rather weak assumption from a statistical point of view.

Now, let us introduce two complexity parameters that play a central role in our analysis. Let D be the unit ball in $L_2(\mu)$ and for $r > 0$ set

$$rD_{f^*} = \{f \in L_2(\mu) : \|f - f^*\|_{L_2(\mu)} \leq r\} = f^* + rD.$$

Definition 1.5 *Given a class F of functions and $\tau > 0$, let*

$$r_Q(F, \tau) = r_Q(F, f^*, \tau) = \inf\left\{r > 0 : \mathbb{E} \sup_{f \in F \cap rD_{f^*}} \left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i (f - f^*)(X_i)\right| \leq \tau r\right\},$$

where $(\varepsilon_i)_{i=1}^N$ are independent, symmetric, $\{-1, 1\}$-valued random variables that are also independent of $(X_i, Y_i)_{i=1}^N$.

Set

$$\phi_N(F, f^*, s) = \sup_{f \in F \cap sD_{f^*}} \left|\frac{1}{\sqrt{N}}\sum_{i=1}^N \varepsilon_i \xi_i (f - f^*)(X_i)\right| \tag{1.9}$$

and put

$$r_M(F, \tau, \delta) = r_M(F, f^*, \tau, \delta) = \inf\left\{s > 0 : Pr\left(\phi_N(F, f^*, s) \leq \tau s^2 \sqrt{N}\right) \geq 1 - \delta\right\}.$$

One may show the following (see Theorem 3.1 in Mendelson (2015)):

Theorem 1.6 *Let F be a closed, convex class of functions that satisfies Assumption 1.2 with constants κ and ε, and set $\theta = \kappa^2\varepsilon/16$. For every $\delta \in (0, 1)$, with probability at least $1 - \delta - 2\exp(-N\varepsilon^2/2)$ one has both:*

- *for every $f \in F$,*

$$|P_N \mathcal{M}_{f - f^*}| \leq \frac{\theta}{4} \max\left\{\|f - f^*\|^2_{L_2(\mu)}, r_M^2\left(F, \theta/5, \delta/4\right)\right\},$$

- *for every $f \in F$ with $\|f - f^*\|_{L_2(\mu)} \geq r_Q\left(F, \kappa\varepsilon/32\right)$,*

$$P_N \mathcal{Q}_{f - f^*} \geq \theta \|f - f^*\|^2_{L_2(\mu)}.$$

In particular, with probability at least $1 - \delta - 2\exp(-N\varepsilon^2/2)$, $P_N \mathcal{L}_f \geq \frac{\theta}{2}\|f - f^*\|^2_{L_2(\mu)}$ for every $f \in F$ that satisfies

$$\|f - f^*\|_{L_2(\mu)} \geq \max\left\{r_M\left(F, \theta/5, \delta/4\right), r_Q\left(F, \kappa\varepsilon/32\right)\right\}.$$

Remark 1.7 *An immediate outcome of Theorem 1.6 is that with high probability, a minimizer in F of the empirical excess-loss functional $P_N \mathcal{L}_f$ must satisfy*

$$\|\tilde{f} - f^*\|_{L_2(\mu)} \leq \max\left\{r_M\left(F, \theta/5, \delta/4\right), r_Q\left(F, \kappa\varepsilon/32\right)\right\}. \tag{1.10}$$

In fact, results from Lecué and Mendelson show that (1.10) is optimal in the minimax sense under additional mild technical assumptions on F when the data is assumed to satisfy the Gaussian regression model, that is, when the targets are of the form $Y = f_0(X) + W$ for $f_0 \in F$ and W that is a centered Gaussian random variable, independent of X. *Empirical risk minimization* performed in the set

$$F^* = \{f \in F : \Psi(f) \leq \Psi(f^*)\}$$

yields

$$\|\tilde{f} - f^*\|_{L_2} \leq \max\left\{r_M\left(F^*, \theta/5, \delta/4\right), r_Q\left(F^*, \kappa\varepsilon/32\right)\right\}, \tag{1.11}$$

and the r.h.s. of (1.11) is the minimax rate of the estimation problem in F^* (up to some mild technical assumptions). This will serve as a benchmark for the performance of the regularization procedure (1.3).

1.4 The Main result

Let $F \cap K_\rho(f^*) = \{f \in F : \Psi(f - f^*) \leq \rho\}$ and observe that these are convex subsets of F. To simplify notation, set

$$r_M(\rho) = r_M\left(F \cap K_\rho(f^*), \frac{\kappa^2\varepsilon}{80}, \frac{\delta}{4}\right) \text{ and } r_Q(\rho) = r_Q\left(F \cap K_\rho(f^*), \frac{\kappa\varepsilon}{32}\right). \tag{1.12}$$

Let $r(\cdot)$ be a function that satisfies that for every $\rho \geq 0$, one has

$$r(\rho) \geq \max\{r_M(\rho), r_Q(\rho)\}. \tag{1.13}$$

It should be noted that $r(\rho)$ may depend on f^*, and that it also depends on other parameters – like δ, κ and ε. We will not specify the dependence on those parameters, but rather, only on the radius ρ.

When sharp estimates on $r_M(\rho)$ and $r_Q(\rho)$ can be obtained, one may set $r(\rho)$ to be of the order of $\max\{r_M(\rho), r_Q(\rho)\}$ for the class $F \cap K_\rho(f^*)$. This choice leads to the rate appearing in Theorem 1.6 and in (1.10). As we mentioned previously, the rate $\max\{r_M(\rho), r_Q(\rho)\}$ is optimal in the minimax sense under additional mild technical assumptions on $F \cap K_\rho(f^*)$.

More information on the $r(\rho)$ may be found in Section 1 of Lecué and Mendelson or in Mendelson (2015).

AS it happens, the geometry of the sets $F \cap K_\rho(f^*)$ (see Figure 1) determines both the error rate and the regularization parameter λ, while $r(\rho)$ measures the sets' 'sizes'.

The choice of λ is made as follows:

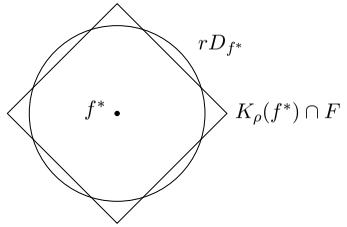


Figure 1: Localization of the set $F \cap K_\rho(f^*)$, i.e. its intersection with $L_2(\mu)$-balls of various radii r for the right choice of radius ρ, plays a central role in the analysis of the quadratic and multiplier processes.

Let

$$\mathcal{O}(\rho) = \sup\left(|P_N \mathcal{M}_{f-f^*}| : f \in F \cap K_\rho(f^*) \cap r(\rho) D_{f^*}\right)$$

and for $\tau > 0$ and $0 < \delta < 1$, set

$$\gamma_{\mathcal{O}}(\rho, \tau, \delta) = \inf\left\{x > 0 : Pr\left(\mathcal{O}(\rho) \leq \tau x\right) \geq 1 - \delta\right\}$$

and

$$\gamma_{\mathcal{O}}(\rho) = \gamma_{\mathcal{O}}(\rho, 3/(80\eta^3), \delta).$$

In other words, $\gamma_{\mathcal{O}}(\rho)$ is proportional to the smallest possible upper estimate on $\mathcal{O}(\rho)$ that still holds with probability at least $1 - \delta$.

Definition 1.8 *For any $\tau > 0$ and $0 < \delta < 1$, set*

$$\lambda_0(\delta, \tau) = \sup_{\rho > 0, f^* \in F} \frac{\gamma_{\mathcal{O}}(\rho, \tau, \delta)}{\rho}.$$

To compare $\lambda_0(\delta, \tau)$ with r_Q and r_M, first note that $r_M(\rho)$ and $\mathcal{O}(\rho)$ both depend on properties of the multiplier processes indexed by localizations of $F \cap K_\rho(f^*)$, and recall that symmetrized and centered processes are essentially equivalent. Second, if $r(\rho) = r_M(\rho)$ then $\gamma_{\mathcal{O}}(\rho) \sim r_M^2(\rho)$; moreover, $\gamma_{\mathcal{O}}(\rho)$ is trivially bounded by $\sim r^2(\rho)$ for the right choice of τ and δ. However, if $r_M(\rho) \leq r_Q(\rho)$, that is, when $r(\rho) = r_Q(\rho)$ – which is the case when ρ is very large – one may find that $\gamma_{\mathcal{O}}(\rho)$ is actually significantly smaller than $\sim r^2(\rho)$. This observation is of crucial importance because of the choice of the regularization parameter: for the right choice of τ, $\gamma_{\mathcal{O}}(\rho) \leq r^2(\rho)$ and

$$\lambda_0(\delta, \tau) \leq \sup_{\rho > 0, f^* \in F} r^2(\rho)/\rho;$$

thus, one may be tempted to select the latter as a regularization term. However, there are natural examples in which $\sup_{\rho>0} r^2(\rho)/\rho = \infty$, rendering that choice impossible, whereas $\sup_{\rho>0, f^* \in F} \gamma_{\mathcal{O}}(\rho)/\rho$ turns out to be finite. Of course, there are still cases in which $\sup_{\rho>0, f^* \in F} r^2(\rho)/\rho$ is finite, and $\lambda_0(\delta, \tau)$ is of the same order as $\sup_{\rho>0, f^* \in F} r_M^2(\rho)/\rho$, though that is not the generic situation.

We now come to the main result of the article.

Theorem 1.9 *Let F be a closed, convex class of functions that satisfies Assumption 1.2 with constants κ and ε. Set $\Psi(\cdot)$ to be a regularization function that satisfies Assumption 1.1 with constant η. Furthermore, assume that $\lim_{\rho \to 0} r(\rho) = 0$ and put $\lambda > \lambda_0(\delta, 3/(80\eta^3))$.*

If $\hat{f}$ is the RERM with a regularization parameter λ as in (1.3), then with probability at least $1 - 2\delta - 2\exp(-N\varepsilon^2/2)$,

$$\|\hat{f} - f^*\|_{L_2(\mu)}^2 \leq \max\left\{r^2\left(10\eta\Psi(f^*)\right), \left(\frac{32}{\kappa^2\varepsilon}\right)\lambda\Psi(f^*)\right\}. \tag{1.14}$$

Observe that λ_0 depends only on the oscillations of the multiplier process. Hence, if the problem is noise-free then $\lambda_0 = 0$, showing that any regularization parameter $\lambda > 0$ would do. Moreover, in that case $r_M(\rho) = 0$ and so one can choose $r(\rho) \geq r_Q(\rho)$ obtaining an error rate that depends only on $r_Q^2(10\eta\Psi(f^*))$.

As noted previously, if one considers empirical risk minimization performed in $F^* = \{f \in F : \Psi(f) \leq \Psi(f^*)\}$, the resulting error rate is $\|\tilde{f} - f^*\|_{L_2(\mu)}^2 \leq c_0 r^2(c\Psi(f^*))$ for a suitable absolute constant c_0 and a constant c that depends on κ, ε and δ; moreover, under some minor additional assumptions, that rate is optimal in the minimax sense (cf. Lecué and Mendelson) when one takes $r(\rho) \sim \max\left\{r_M(\rho), r_Q(\rho)\right\}$. Hence, up to constants involved, the first term in Theorem 1.9 is essentially the minimax rate that one can obtain if $\Psi(f^*)$ were known.

If one chooses $\lambda \sim \lambda_0(\delta, \tau)$ for $\tau = 3/(80\eta^3)$ then the second term in (1.14) is of the order of

$$\lambda\Psi(f^*) = \left(\sup_{\rho, f^*} \frac{\gamma_{\mathcal{O}}(\rho, \tau, \delta)}{\rho}\right) \cdot \Psi(f^*).$$

Note that for ρ that is of the order of $\Psi(f^*)$, one has

$$\frac{\gamma_{\mathcal{O}}(\rho, \tau, \delta)}{\rho} \cdot \Psi(f^*) \leq c_1 \gamma_{\mathcal{O}}(\rho, \tau, \delta) \leq c_2 r^2(c_3\Psi(f^*)),$$

which coincides with the first term, up to the constants involved. Thus, the price that one has to pay for not knowing $\Psi(f^*)$ is manifested in the need to take the supremum over all possible choices of ρ in the second term, rather than considering only the level $\rho \sim \Psi(f^*)$.

Thankfully, there are many natural cases in which that price is rather small, allowing for satisfactory outcomes of Theorem 1.9 that are close to the minimax rate.

2. Proof of Theorem 1.9

Before we present the proof, let us introduce some notation. Throughout, absolute constants or constants that depend on other parameters are denoted by c, C, c_1, c_2, etc., (and,

of course, we will specify when a constant is absolute and when it depends on other parameters). The values of these constants may change from line to line. The notation $x \sim y$ (resp. $x \lesssim y$) means that there exist absolute constants $0 < c < C$ for which $cy \leq x \leq Cy$ (resp. $x \leq Cy$). If $b > 0$ is a parameter, then $x \lesssim_b y$ means that $x \leq C(b)y$ for some constant $C(b)$ that depends only on b.

The normed space ℓ_p^d is $\mathbb{R}^d$ endowed with the norm $\|x\|_p = \left(\sum_j |x_j|^p\right)^{1/p}$; the corresponding unit ball is denoted by B_p^d and the unit Euclidean sphere in $\mathbb{R}^d$ is S^{d-1}.

Finally, from here on we write Pr and $\|\ \|_{L_2}$ without specifying the underlying measure.

The proof of Theorem 1.9 follows an almost identical path as the proof of Theorem 3.2 from Lecué and Mendelson (to appear). The differences between the two arguments are minor and their source is the fact that unlike Lecué and Mendelson (to appear), here we do not assume that Ψ is a norm. In Remark 2.2 we will outline how a version of Theorem 1.9 may be derived directly from Theorem 3.2 in Lecué and Mendelson (to appear) when Ψ is a norm.

Theorem 1.9 is an immediate outcome of the following lemma:

Lemma 2.1 *Let $\lambda_0 = \lambda_0(\delta, 3/(80\eta^3))$ and set $\lambda > \lambda_0$. If $\lim_{\rho\to 0} r(\rho) = 0$, $\rho \geq 5\eta\Psi(f^*)$ and $\rho > 0$, then with probability at least $1 - 2\delta - 2\exp(-N\varepsilon^2/2)$,*

$$\|\hat{f} - f^*\|_{L_2}^2 \leq \max\left\{r^2(\rho), (32/(\kappa^2\varepsilon))\lambda\Psi(f^*)\right\}.$$

To see how Lemma 2.1 can be used to conclude the proof of Theorem 1.9, observe that if $\Psi(f^*) > 0$, one may simply select $\rho = 5\eta\Psi(f^*)$ in the lemma. If, on the other hand, $\Psi(f^*) = 0$, let $(\gamma_n)_{n=1}^\infty$ be a positive sequence decreasing to 0 and set $\mathcal{A}_n = \{\|\hat{f} - f^*\|_{L_2} \leq \gamma_n\}$, which is a decreasing sequence of events. If $Pr(\mathcal{A}_n) \geq 1 - \nu$ for some $0 \leq \nu \leq 1$ and every n then $Pr(\{\hat{f} = f^*\}) \geq 1 - \nu$. Since $\lim_{\rho\to 0} r(\rho) = 0$, one may apply Lemma 2.1 to each member of a nonnegative sequence ρ_n that decreases to zero and for which $\gamma_n = r(\rho_n)$ decreases to zero. By Lemma 2.1, $Pr(\mathcal{A}_n) \geq 1 - 2\delta - 2\exp(-N\varepsilon^2/2)$ for every n and the proof of Theorem 1.9 follows.

Proof of Lemma 2.1. Fix f^* and set $\rho > 0$ that satisfies $\rho \geq 5\eta\Psi(f^*)$. Let

$$F_1 = \{f \in F : \Psi(f - f^*) \leq \rho\} = F \cap K_\rho(f^*),$$

and

$$F_2 = \{f \in F : \Psi(f - f^*) = \rho\}.$$

Clearly, F_1 is a convex set that contains f^*, and by the continuity of the real-valued function $t \to \Psi(f^* + t(f - f^*))$, every ray $[f^*, f)$ that originates in f^* and passes through some $f \in F\backslash F_1$ intersects F_2.

Let $\theta = \kappa^2\varepsilon/16$ and set

$$r_Q(\rho) = r_Q(F_1, \kappa\varepsilon/32) \quad \text{and} \quad r_M(\rho) = r_M(F_1, \theta/5, \delta/4).$$

There is an event $\mathcal{A}_0$ of probability at least $1-\delta-2\exp(-N\varepsilon^2/2)$, and for every $(X_i, Y_i)_{i=1}^N \in \mathcal{A}_0$ the following holds:

- If $f \in F_1$ and $\|f - f^*\|_{L_2} \geq r_Q(\rho)$ then

$$\frac{1}{N}\sum_{i=1}^N (f - f^*)^2(X_i) \geq \theta\|f - f^*\|_{L_2}^2.$$

- If $f \in F_1$ then

$$\left|\frac{1}{N}\sum_{i=1}^N \xi_i(f - f^*)(X_i) - \mathbb{E}\xi(f - f^*)(X)\right| \leq \frac{\theta}{4}\max\{\|f - f^*\|_{L_2}^2, r_M^2(\rho)\}.$$

In particular, if $f \in F_1$ and $\|f - f^*\|_{L_2} \geq r(\rho) \geq \max\{r_M(\rho), r_Q(\rho)\}$ then

$$P_N\mathcal{L}_f \geq \frac{\theta}{2}\|f - f^*\|_{L_2}^2.$$

By the choice of λ, there is an event $\mathcal{A}_1$ of probability at least $1 - \delta$ on which if $f \in F_1$ and $\|f - f^*\|_{L_2} \leq r(\rho)$, then

$$\left|\frac{2}{N}\sum_{i=1}^N \xi_i(f - f^*)(X_i) - \mathbb{E}\xi(f - f^*)(X)\right| < \frac{3}{80\eta^3}\lambda\rho < \frac{3}{5\eta}\lambda\rho. \tag{2.1}$$

Set $\mathcal{A} = \mathcal{A}_0 \cap \mathcal{A}_1$ and let $(X_i, Y_i)_{i=1}^N \in \mathcal{A}$. The proof now follows in three steps:

(1) Show that the functional $f \to P_N\mathcal{L}_f^\lambda$ is bounded from below – away from zero – in F_2.

(2) An outcome of (1) is that if $f \in F\backslash F_1$, $P_N\mathcal{L}_f^\lambda > 0$; hence, $\hat{f} \notin F\backslash F_1$.

(3) Finally, pin-point $\hat{f}$ within $F_1 = \{f \in F : \Psi(f - f^*) \leq \rho\}$.

Step 1. Fix $f \in F_2$ and note that by the 'triangle inequality' satisfied by Ψ,

$$\Psi(f) \geq \eta^{-1}\Psi(f - f^*) - \Psi(f^*).$$

Recall that $\eta^{-1}\Psi(f - f^*) \geq \eta^{-1}\rho \geq 5\Psi(f^*)$ and thus, $\Psi(f) - \Psi(f^*) \geq (3/5)\eta^{-1}\rho$. Hence, if $\|f - f^*\|_{L_2} \geq r(\rho)$ then

$$P_N\mathcal{L}_f^\lambda \geq (\theta/2)\|f - f^*\|_{L_2}^2 + \lambda\rho \cdot \frac{3}{5\eta} > 0.$$

On the other hand, by the choice of λ, if $\|f - f^*\|_{L_2} \leq r(\rho)$ then

$$\begin{aligned} P_N\mathcal{L}_f^\lambda &\geq -\left|\frac{1}{N}\sum_{i=1}^N \xi_i(f - f^*)(X_i) - \mathbb{E}\xi(f - f^*)(X)\right| + \lambda\left(\Psi(f) - \Psi(f^*)\right) \\ &\geq -\left|\frac{1}{N}\sum_{i=1}^N \xi_i(f - f^*)(X_i) - \mathbb{E}\xi(f - f^*)(X)\right| + \lambda\rho \cdot \frac{3}{5\eta} > 0. \end{aligned}$$

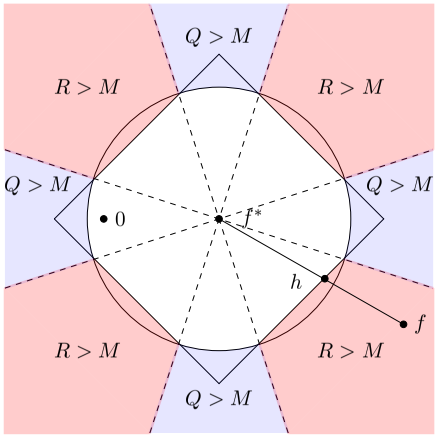


Figure 2: $P_N\mathcal{L}_f^\lambda > 0$ for two different reasons: either $Q > M$ – the quadratic component dominates the multiplier component, or $R > M$ – the regularization component dominates the multiplier component. Unlike Theorem 3.2 in Lecué and Mendelson (to appear), here we choose $\rho \sim \Psi(f^*)$ to ensure that $0 \in F \cap K_\rho(f^*)$.

It should be noted that the same proof shows that on the event $\mathcal{A}$, for every $f \in F_2$,

$$P_N\mathcal{L}_f + \frac{\lambda}{2\eta^2}(\Psi(f) - \Psi(f^*)) > 0, \tag{2.2}$$

a fact that will be used below. Indeed, $(\lambda/2\eta^2)\cdot(\Psi(f) - \Psi(f^*)) \geq (\lambda/2\eta^2)\cdot(3\rho/5\eta)$ and by (2.1), if $\|f - f^*\|_{L_2} \leq r(\rho)$ then $P_N\mathcal{L}_f \geq -(3/80)\cdot(\lambda\rho/\eta^3)$.

Step 2. Let $f \in F\backslash F_1$ and note that by the convexity of F and the continuity of Ψ on rays, there is some $h \in F_2$ and $R > 1$ for which $f = f^* + R(h - f^*)$. Thus,

$$P_N\mathcal{L}_f^\lambda = \frac{R^2}{N}\sum_{i=1}^N (h-f^*)^2(X_i) + \frac{2R}{N}\sum_{i=1}^N \xi_i(h-f^*)(X_i) + \lambda\left(\Psi(f) - \Psi(f^*)\right).$$

Observe that

$$\Psi(f) - \Psi(f^*) \geq \frac{R}{2\eta^2}\left(\Psi(h) - \Psi(f^*)\right); \tag{2.3}$$

indeed,

$$\Psi(f^* + R(h-f^*)) \geq \eta^{-1}\Psi(R(h-f^*)) - \Psi(f^*) \geq R\eta^{-1}\Psi(h-f^*) - \Psi(f^*),$$

and thus it suffices to show that

$$\frac{R}{\eta}\Psi(h-f^*) \geq \frac{R}{2\eta^2}\Psi(h) + 2\Psi(f^*).$$

But since $\Psi(h-f^*) \geq 5\eta\Psi(f^*)$, $\eta \geq 1$ and $R \geq 1$, one has

$$\begin{aligned}\frac{R}{\eta}\Psi(h-f^*) &\geq \frac{R}{2\eta}\Psi(h-f^*) + \frac{R}{2}\Psi(f^*) + 2R\Psi(f^*)\\ &\geq \frac{R}{2\eta}\left(\Psi(h-f^*) + \Psi(f^*)\right) + 2\Psi(f^*) \geq \frac{R}{2\eta^2}\Psi(h) + 2\Psi(f^*),\end{aligned}$$

and (2.3) follows.

Finally, applying (2.2) to $h \in F_2$,

$$\begin{aligned}P_N\mathcal{L}_f^\lambda &\geq \frac{R^2}{N}\sum_{i=1}^N (h-f^*)^2(X_i) + \frac{2R}{N}\sum_{i=1}^N \xi_i(h-f^*)(X_i) + \lambda\frac{R}{2\eta^2}\left(\Psi(h) - \Psi(f^*)\right)\\ &\geq R\left(P_N\mathcal{L}_h + \frac{\lambda}{2\eta^2}(\Psi(h) - \Psi(f^*))\right) > 0,\end{aligned}$$

and $\hat{f} \notin F\backslash F_1$.

Step 3. Turning to $F_1 = \{f \in F : \Psi(f-f^*) \leq \rho\} = F \cap K_\rho(f^*)$, recall that if $f \in F_1$ and $\|f - f^*\|_{L_2} \geq r(\rho)$, then $P_N\mathcal{L}_f \geq (\theta/2)\|f-f^*\|_{L_2}^2$; hence, if f is a potential minimizer and $\|f - f^*\|_{L_2} \geq r(\rho)$ then

$$0 \geq P_N\mathcal{L}_f^\lambda \geq (\theta/2)\|f-f^*\|_{L_2}^2 + \lambda\left(\Psi(f) - \Psi(f^*)\right) \geq (\theta/2)\|f-f^*\|_{L_2}^2 - \lambda\Psi(f^*),$$

and

$$\|\hat{f} - f^*\|_{L_2}^2 \leq \frac{2\lambda}{\theta}\Psi(f^*),$$

as claimed. ∎

Remark 2.2 *If Ψ happens to be a norm (which is an assumption slightly stronger than Assumption 1.1), one may apply Theorem 3.2 from Lecué and Mendelson (to appear) directly. Indeed, and using the notation from Lecué and Mendelson (to appear) if $\rho \gtrsim \Psi(f^*)$ then the set $K = \{f : \Psi(f-f^*) \leq \rho/20\}$ contains a Ψ-ball around 0, and $\Gamma_{f^*}(\rho)$ – the collection of norming functionals (i.e., the sub-gradient of Ψ) of any $h \in K$ – is the entire unit ball in the dual space to (E,Ψ). Recall that*

$$\Delta(\rho) = \inf_h \sup_{z^* \in \Gamma_{f^*}(\rho)} z^*(h - f^*),$$

where the infimum is taken in the set

$$\{h \in F : \ \Psi(h-f^*) = \rho \text{ and } \|h - f^*\|_{L_2} \leq r(\rho)\}.$$

Since $\Gamma_{f^*}(\rho)$ is the entire dual unit ball, it follows that $\Delta(\rho) = \rho$, and Theorem 3.2 in Lecué and Mendelson (to appear) may be applied. The desired version of Theorem 1.9 now follows from Remark 3.3 in Lecué and Mendelson (to appear).

3. Towards the examples - preliminary estimates

It is rather obvious that any implementation of Theorem 1.9 requires specific estimates on r_M, r_Q and λ_0. This section is devoted to some preliminary facts that will play an instrumental part in establishing such estimates.

Our main interest is the study of upper bounds on the three processes used to define the parameter r_M, r_Q and $\gamma_{\mathcal{O}}$, and which have the following forms:

$$(*)=\sup_{f\in F}\left|\sum_{i=1}^N \varepsilon_i\xi_i f(X_i)\right|,\quad (**)=\sup_{f\in F}\left|\sum_{i=1}^N (\xi_i f(X_i)-\mathbb{E}\xi f(X))\right| \text{ and } \mathbb{E}\sup_{f\in F}\left|\sum_{i=1}^N \varepsilon_i f(X_i)\right|,$$

where $X_1,...,X_N$ are independent and distributed according to the underlying measure μ, $\xi_1,...,\xi_N$ are independent copies of $\xi\in L_q$ for some $q>2$ (though $(\xi_i)_{i=1}^N$ need not be independent of $(X_i)_{i=1}^N$), and $(\varepsilon_i)_{i=1}^N$ are independent, symmetric $\{-1,1\}$-valued random variables that are independent of $(X_i)_{i=1}^N$ and $(\xi_i)_{i=1}^N$.

Standard symmetrization methods (see, e.g., Giné and Zinn (1984); Ledoux and Talagrand (1991); van der Vaart and Wellner (1996)) show that $(*)$ and $(**)$ are equivalent in expectation and in deviation (up to a slight restriction on the deviation parameter). We will present one example in which this symmetrization argument is carried out in full (Theorem 4.2), but in the other examples we will only consider the symmetrized case.

3.1 Estimates for subgaussian classes

The first result is from Mendelson (2016), under the assumption that F is an L-subgaussian class of functions.

Definition 3.1 *A class of functions $F\subset L_2(\mu)$ is L-subgaussian if for every $f,h\in F\cup\{0\}$ and every $u\geq 1$,*

$$Pr(|f(X)-h(X)|\geq Lu\|f-h\|_{L_2(\mu)})\leq 2\exp(-u^2/2),$$

where X is distributed according to μ.

Let $F\subset L_2(\mu)$ and set $\{G_f: f\in F\}$ to be the centered, canonical Gaussian process indexed by F (i.e., the covariance operator of the process is $\mathbb{E}G_fG_g=\mathbb{E}f(X)g(X)$ for every $f,g\in F$). Put

$$\ell^*(F)=\mathbb{E}\sup_{f\in F}G_f,\quad\text{and}\quad d_2(F)=\sup_{f\in F}\|f\|_{L_2(\mu)}. \tag{3.1}$$

Theorem 3.2 (Corollary 1.10 in Mendelson (2016)) *Let X be distributed according to μ, set $\xi\in L_q$ for some $q>2$ and assume that $F\subset L_2(\mu)$ is an L-subgaussian class. There are constants c,c_0,c_1,c_2 and c_3 that depend only on q, for which, for any $w,u>c$, with probability at least*

$$1-\frac{c_0\log^q N}{w^qN^{q/2-1}}-2\exp\left(-c_1u^2(\ell^*(F)/d_2(F))^2\right),$$

$$\sup_{f\in F}\left|\frac{1}{N}\sum_{i=1}^N \varepsilon_i\xi_i f(X_i)\right|\leq c_2Lwu\|\xi\|_{L_q}\frac{\ell^*(F)}{\sqrt{N}}$$

and

$$\sup_{f\in F}\left|\frac{1}{N}\sum_{i=1}^N \xi_i f(X_i)-\mathbb{E}\xi f(X)\right|\leq c_2Lwu\|\xi\|_{L_q}\frac{\ell^*(F)}{\sqrt{N}}.$$

Corollary 3.3 *Using the notation and assumptions of Theorem 3.2, let $\xi=Y-f^*(X)$ and assume that $\xi\in L_q$ for some $q>2$. Fix $\tau>0$ and $0<\delta<1$, and set $A>0$ for which*

$$c_2Lwu\|\xi\|_{L_q}\ell^*\left(F\cap AD_{f^*}\right)\leq\tau A^2\sqrt{N}. \tag{3.2}$$

If

$$\delta\geq\frac{c_0\log^q N}{w^qN^{q/2-1}}+2\exp\left(-c_1a_0u^2\right) \tag{3.3}$$

then $r_M(F,\tau,\delta)\leq A$.

Proof. Clearly, it follows from Theorem 3.2 that if

$$\delta\geq\frac{c_0\log^q N}{w^qN^{q/2-1}}+2\exp\left(-c_1u^2\left(\frac{\ell^*(F\cap AD_{f^*})}{d_2(F\cap AD_{f^*})}\right)^2\right)$$

then $r_M(F,\tau,\delta)\leq A$. The claim follows because if $F\cap AD_{f^*}$ is nonempty,

$$\frac{\ell^*(F\cap AD_{f^*})}{d_2(F\cap AD_{f^*})}\geq a_0$$

for a suitable absolute constant a_0. ∎

Remark 3.4 *The estimate in Corollary 3.3 can be rather loose. The reason for the suboptimal estimate is that usually, the Gaussian mean-width $\ell^*(F\cap AD_{f^*})$ is much larger than $d_2(F\cap AD_{f^*})$. For example, let $F=\{\langle t,\cdot\rangle: t\in S^{d-1}\}$ be the class of linear functionals on $\mathbb{R}^d$ indexed by the Euclidean unit ball. Assume that X is an isotropic vector – that is, its covariance structure coincides with the standard Euclidean structure on $\mathbb{R}^d$; that $f^*=0$; and that $A\leq 1$. Then $F\cap AD_{f^*}=\{\langle t,\cdot\rangle: \|t\|_2\leq A\}$, $d_2(F\cap AD_{f^*})=A$ and $\ell^*(F\cap AD_{f^*})=A\sqrt{d}$, implying that*

$$\frac{\ell^*(F\cap AD_{f^*})}{d_2(F\cap AD_{f^*})}\geq\sqrt{d} \tag{3.4}$$

which is significantly larger than an absolute constant.

Having said that, the question of an accurate probability estimate is not the main issue of this article and we will not explore that point further.

Next, we provide an estimate on $\gamma_{\mathcal{O}}(\rho,\tau,\delta)$ that follows from Theorem 3.2 when F is L-subgaussian and $\xi\in L_q$ for some $q>2$. The proof is identical to the one of Corollary 3.3 and is omitted.

Corollary 3.5 *Let F be a closed, convex L-subgaussian class of functions and let $\xi = Y - f^*(X) \in L_q$ for some $q > 2$. Set $w, u > c$, $\tau > 0$, $0 < \delta < 1$ and $\rho > 0$. If $A > 0$ satisfies*

$$c_2 Lwu \, \|\xi\|_{L_q} \, \ell^*\left(F \cap K_\rho(f^*) \cap r(\rho) D_{f^*}\right) \leq \tau A \sqrt{N}.$$

and

$$\delta \geq \frac{c_0 \log^q N}{w^q N^{q/2-1}} + 2\exp\left(-c_1 a_0 u^2\right)$$

then $\gamma_{\mathcal{O}}(\rho, \tau, \delta) \leq A$.

Finally, when F is an L-subgaussian class it follows from a standard chaining argument (cf. Talagrand (2005) or Mendelson (2016)) that

$$\mathbb{E} \sup_{f \in F} \left| \frac{1}{N} \sum_{i=1}^N \varepsilon_i f(X_i) \right| \leq \frac{c_0 L \ell^*(F)}{\sqrt{N}}. \tag{3.5}$$

This observation will be used in what follows to upper bound r_Q.

3.2 Estimates under a limited moment condition

In this section we shall consider the case of a class that need not be subgaussian, but rather, the growth of moments of class members is well-behaved up to some point. More accurately, we will assume that there is some p_0 for which, for every $f, h \in F \cup \{0\}$ and $2 \leq p \leq p_0$,

$$\|f - h\|_{L_p} \leq L\sqrt{p} \|f - h\|_{L_2}. \tag{3.6}$$

In contrast, a subgaussian condition is equivalent to having $\|f - h\|_{L_p} \leq L\sqrt{p}\|f - h\|_{L_2}$ for every $p \geq 2$.

The motivation behind this type of limited moment assumption is the LASSO estimator. Recent results on properties of the *basis pursuit algorithm* in $\mathbb{R}^d$ Lecué and Mendelson (2017); Dirksen et al. (2014) indicate that the limited moment assumption (3.6) for $p_0 \sim \log d$ should suffice for an optimal estimate on the performance of the LASSO – as if the class were subgaussian.

When analyzing the LASSO via the computation of the fixed points r_M and r_Q, one encounters the following scenario. Let $X = (x_j)_{j=1}^d$ be a random vector in $\mathbb{R}^d$ and set $X_1, \ldots, X_N$ to be independent copies of X. Let $X_i(j)$ be the j-th coordinate of X_i and thus $(X_i(j))_{i=1}^N$ is a random vector with independent coordinates, distributed as x_j.

Consider the random variables appearing in the definition of r_M and r_Q in the LASSO case:

$$\max_{1 \leq j \leq d} \left| \sum_{i=1}^N \varepsilon_i X_i(j) \right|, \tag{3.7}$$

and

$$\max_{1 \leq j \leq d} \left| \sum_{i=1}^N \varepsilon_i \xi_i X_i(j) \right| \tag{3.8}$$

The aim of this section is to derive upper bounds on (3.7) in expectation and (3.8) in deviation when each x_j satisfies that

$$\|x_j\|_{L_p} \leq L\sqrt{p}\|x_j\|_{L_2}$$

for $p \lesssim \log d$. Note that an upper bound on the centered empirical process involved in the definition of $\gamma_{\mathcal{O}}(\rho)$ will follow from a symmetrization argument and a bound on (3.8).

The obvious difference between (3.7) and (3.8) are the multipliers $(\xi_i)_{i=1}^N$: although the x_j's have $\sim \log d$ moments, ξ may be heavy-tailed, in the sense that it only belongs to L_q for some fixed $q > 2$; this difference makes the analysis of (3.8) more difficult.

Upper bounds on (3.7) and (3.8) are obtained under the following assumption.

Assumption 3.1 *Let $N \leq d$, $t \geq 4$ and set $p_0 = t \log d$. Assume that $p_0 \lesssim N$ (and note that $p_0 \geq \log N$) and that for every $1 \leq j \leq d$ and $p \leq p_0$, $\|x_j\|_{L_p} \leq L\sqrt{p}\|x_j\|_{L_2}$. Consider $\xi \in L_q$ for some $q > 2$; let $r = \min\{1/2 + q/4, 2\}$; set r' to be the conjugate index of r; and assume that $4r' \max\{2, 1 + a_0/a_1\} \leq t \log N$ (where a_0 and a_1 are two absolute constants to be specified later – in Lemma 7.3 and Lemma 7.4).*

Under this assumption we will prove the following:

Theorem 3.6 *Let the random vector X and $\xi = Y - f^*(X)$ satisfy Assumption 3.1. Then,*

$$\mathbb{E} \max_{1 \leq i \leq d} \left| \frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i X_i(j) \right| \leq c_0 \sqrt{\log d} \cdot L \max_{1 \leq j \leq d} \|x_j\|_{L_2}. \tag{3.9}$$

Also, for every $u > 2, v > 0, w \geq 2$ and for $p = p_0/2$ and $m = p/\log(eN/p)$, one has that with probability at least

$$1 - \frac{\exp(-p/2)}{u^{2p}} - \frac{4\exp(-p/2)}{u^{c_1 m}} - \frac{c_2 \log^q N}{w^q N^{q/2-1}} - 2\exp(-v^2 t \log d), \tag{3.10}$$

$$\max_{1 \leq j \leq d} \left| \sum_{i=1}^N \varepsilon_i \xi_i X_i(j) \right| \leq c_3(q)(uw + u^2 v) L \|\xi\|_{L_q} \sqrt{N} \sqrt{t \log d} \max_{1 \leq j \leq d} \|x_j\|_{L_2}. \tag{3.11}$$

The proofs of both estimates in Theorem 3.6 follow from a more general result, established in Mendelson (2016), on the supremum of a centered multiplier process under a limited moment assumption like (3.6). Although the estimate in Mendelson (2016) is stated for the centered empirical process (cf. Section 4 there) its proof is actually based on an estimate on the symmetrized process. The proof of Theorem 3.6 will be presented in a final section of this article.

4. The LASSO under a limited moment assumption

In this section, we obtain complexity-dependent error rates for the LASSO. Our aim is to show that the LASSO (almost) achieves the minimax rates of convergence in the true model. Observe that in this case, the true model is the smallest ℓ_1^d-ball centered at 0 that contains t^*. Thus, the price one has to pay for not knowing $\|t^*\|_1$ is rather minimal.

The rate we shall be comparing the LASSO's performance to is the minimax rate of the following problem. Let $X \sim \mathcal{N}(0, I_{d\times d})$ and set $\xi \sim \mathcal{N}(0,\sigma^2)$ to be independent of X. Let $\rho > 0$, consider an unknown $t_0 \in \rho B_1^d$ and put $Y = \langle X, t_0\rangle + \xi$.

Let c_0, c_1 and c_2 be well-chosen absolute constants and consider the cases $\log d \le N \le c_0 d$ or $c_1 d \le N$. Following Lecué and Mendelson, if

$$s_M^2(\rho) = c_2 \begin{cases} \frac{\sigma^2 d}{N} & \text{if } \rho^2 N \ge \sigma^2 d^2 \\ \rho\sigma\sqrt{\frac{1}{N}\log\left(\frac{e\sigma d}{\rho\sqrt{N}}\right)} & \text{if } \sigma^2 \log d \le \rho^2 N \le \sigma^2 d^2 \\ \rho\sigma\sqrt{\frac{\log d}{N}} & \text{if } \rho^2 N \le \sigma^2 \log d \end{cases} \quad \text{and} \quad s_Q^2(\rho) \begin{cases} = 0 & \text{if } N \ge c_0 d \\ \lesssim \rho^2/d & \text{if } c_0 d \le N \le c_1 d \\ \sim \frac{\rho^2}{N}\log\left(\frac{d}{N}\right) & \text{if } N \le c_1 d, \end{cases}$$

then the minimax rate of convergence in the class ρB_1^d is

$$\max\left\{s_M^2(\rho), s_Q^2(\rho)\right\} \tag{4.1}$$

when $\rho \ge \sigma\sqrt{(\log d)/N}$ and ρ^2 when $\rho \le \sigma\sqrt{(\log d)/N}$. Note that when $c_0 d \le N \le c_1 d$ (i.e. $N \sim d$), $s_Q^2(\rho)$ decays rapidly from $\frac{\rho^2}{N}\log(d/N)$ to 0 and there are no precise estimates on the minimax rate in that range.

It turns out that for this problem – the so-called Gaussian linear model – the minimax rate in ρB_1^d is achieved by the Empirical Risk Minimization procedure (see, e.g., Lecué and Mendelson); however, an underlying assumption is that ρ is given. Thanks to regularization, and specifically, thanks to the LASSO, one does not need to know the value of $\|t_0\|_1$ in advance to achieve the minimax rate, at least up to a logarithmic term. In fact, as will be explained below, the optimal rate can be achieved using regularization in a much more general framework than just the Gaussian linear model.

In what follows we focus on the performance of the LASSO in the high-dimensional case, that is, when $N \le c_1 d$. One may do the same when $N \ge c_0 d$ and we leave that to the reader.

Let X be a random vector in $\mathbb{R}^d$ and consider the class of linear functionals $F = \{f_t = \langle\cdot, t\rangle : t \in \mathbb{R}^d\}$. In particular, if $Y \in L_2$ is an arbitrary target random variable then $f^* = f_{t^*} = \langle\cdot, t^*\rangle$ satisfies

$$t^* \in \operatorname*{argmin}_{t\in\mathbb{R}^d} \mathbb{E}\big(Y - \langle X, t\rangle\big)^2. \tag{4.2}$$

As noted in the introduction, the regularization function associated with the LASSO is the ℓ_1^d-norm: for every $t = (t_j)_{j=1}^d \in \mathbb{R}^d$,

$$\Psi(f_t) = \|t\|_1 = \sum_{j=1}^d |t_j|.$$

Clearly, as a norm, the ℓ_1^d-regularization function satisfies Assumption 1.1 for $\eta = 1$.

The LASSO with regularization parameter λ produces

$$\hat{t} \in \operatorname*{argmin}_{t\in\mathbb{R}^d}\Big(\frac{1}{N}\sum_{i=1}^N \big(Y_i - \langle X_i, t\rangle\big)^2 + \lambda\|t\|_1\Big), \tag{4.3}$$

and one would like to control $\|f_{\hat{t}} - f_{t^*}\|_{L_2}^2 = \mathbb{E}\langle X, \hat{t} - t^*\rangle^2$, where the expectation is taken with respect to X conditionally on the data.

It should be noted that despite the LASSO's popularity, there are relatively few results in the random design scenario we are interested in (see, e.g., Bartlett et al. (2012), Massart and Meynet (2011) and chapter 8.2 in Koltchinskii (2011)). The overwhelming majority of existing results have been obtained for the linear model with subgaussian noise and a fixed design (i.e., each data point is of the form $Y_i = \langle t^*, z_i\rangle + \xi_i$), and the deterministic design matrix, whose rows are the vectors z_i, which satisfies some form of the *restricted isometry property* – for example, the *Restricted Eigenvalue Condition* (REC) from Bickel et al. (2009) or the *Compatibility Condition* (CC) from van de Geer (2007)).

To define the restricted eigenvalue condition, let us introduce the following notation: for $x \in \mathbb{R}^d$ and a set $S_0 \subset \{1,\dots,d\}$ of cardinality $|S_0| \le s$, let S_1 be the set of indices of the m largest coordinates of $(|x_i|)_{i=1}^d$ that are outside S_0. Let $x_{S_{01}}$ be the restriction of x to the set $S_{01} = S_0 \cup S_1$.

Definition 4.1 (Bickel et al. (2009)) *Let Γ be an $N \times d$ matrix. For $c_0 \ge 1$ and an integer $1 \le s \le m \le d$ for which $m + s \le d$, the* ***restricted eigenvalue constant*** *is*

$$\kappa(s, m, c_0) = \min\left\{\frac{\|\Gamma x\|_2}{\|x_{S_{01}}\|_2} : S_0 \subset \{1,\dots,d\}, |S_0| \le s, \left\|x_{S_0^c}\right\|_1 \le c_0\|x_{S_0}\|_1\right\}.$$

The matrix Γ satisfies the ***Restricted Eigenvalue Condition (REC) of order s with a constant c*** *if $\kappa(s, s, 3) \ge c$.*

One can show (see, Bickel et al. (2009), Bühlmann and van de Geer (2011)) that if Γ satisfies REC and $\lambda \gtrsim \sigma\sqrt{(\log d)/N}$, then with high probability (with respect to the noise), simultaneously for every $1 \le p \le 2$,

$$\left\|\hat{t} - t^*\right\|_p^p \lesssim_p \|t^*\|_0\left(\frac{\sigma}{\kappa(s,s,3)}\sqrt{\frac{\log d}{N}}\right)^p \tag{4.4}$$

where $\|t^*\|_0$ is the cardinality of the support of t^*.

The main result in this section is an estimate on $\|\hat{f} - f^*\|_{L_2}^2$ that depends on $\|t^*\|_1$ rather than on the cardinality of the support of t^* (we refer to Lecué and Mendelson (to appear) for sparsity-dependent rates of convergence for the LASSO in the same framework as we consider here). Such a result follows from Theorem 1.9, and to that end, one has to construct a function $r(\cdot)$ as in (1.13) and to compute $\lambda_0(\delta,\gamma)$ as in Definition 1.8. We will do so in the following case: Set $a_2 \ge 4$, $2 \le p_0 = a_2 \log d \lesssim N$, $q > 2$, $r = \min\{1/2 + q/4, 2\}$ and r' that is the conjugate index of r. Assume that $4r'\max\{2, 1 + a_0/a_1\} \le a_2 \log N$ (which is equivalent to assuming that $q > 2 + c_1/\log N$ for some constant $c_1 = c_1(a_0, a_1, a_2)$). Let $X = (x_j)_{j=1}^d$ be a random vector and note that the coordinates $x_1, ..., x_d$ need not be independent.

Assumption 4.1 *Using the above notation, assume that there are constants κ_0, κ and ε for which the following holds:*

- *For every* $1 \le j \le d$ *and every* $2 \le p \le p_0$, $\|x_j\|_{L_p} \le \kappa_0 \sqrt{p}\, \|x_j\|_{L_2}$.
- X *satisfies a small-ball condition with constants* κ *and* ε*; that is, for every* $t \in \mathbb{R}^d$,

$$Pr\left(|\langle X, t\rangle| \ge \kappa \left\|\langle X, t\rangle\right\|_{L_2} \right) \ge \varepsilon. \tag{4.5}$$

- $\xi = Y - f^*(X) \in L_q$.

To put this assumption in some perspective, note that an obvious underlying condition in any estimation problem with respect to the squared loss is that $\mathbb{E}(f(X)-Y)^2$ is defined for any $f \in F$, and in particular, that $\xi = Y - f^*(X) \in L_2$. Thus, assuming that $\xi \in L_q$ for some $q > 2 + c_1/\log N$ is not very restrictive. Also, as noted previously, the small-ball assumption is rather minimal.

The most restrictive component of Assumption 4.1 is the moment assumption on the coordinates of X, namely that their moments exhibit a subgaussian behavior, up to, roughly, $p \sim \log d$.

While this assumption can be weakened to other types of moment growth condition (e.g., $\|x_j\|_{L_p} \le \kappa_0 p^\alpha \|x_j\|_{L_2}$ for some $\alpha \ge 1/2$ and up to $p \sim \log d$), the resulting analysis is more involved (see Lecué and Mendelson (2017)), and will not be explored here.

Finally, Lecué and Mendelson (2017) shows that even if one assumes a subgaussian behavior of the coordinates x_i, but only up to $p \sim (\log d)/(\log\log d)$, Basis Pursuit may fail to recover even a 1-sparse vector, implying that the choice of p_0 in Assumption 4.1 can not be relaxed significantly.

Given any $\rho \ge 0$, set $M = \max_{1\le j\le d} \|x_j\|_{L_2}$, let $\sigma_q = \|\xi\|_q$ and put

$$\Lambda(\rho) = \frac{\kappa_0 \rho M}{\kappa^2 \varepsilon} \sqrt{\frac{\log d}{N}}.$$

Moreover, for $R(t) = \mathbb{E}(Y - \langle X, t\rangle)^2$, one has

$$R(t) - R(t^*) = \mathbb{E}\langle X, t - t^*\rangle^2,$$

because $\langle X, t^*\rangle$ is the best approximation of Y in a closed subspace of L_2. Thus, the estimation bounds also lead to excess risk bounds.

Theorem 4.2 *There are absolute constants* $c_0, ..., c_6$ *for which the following holds. Assume that* X *and* $\xi = Y - f^*(X)$ *satisfy Assumption 4.1 and that* $N \le d$. *Let* $u > 2, v > 0$ *and* $w \ge 2$, *and set* $p = (a_2/2)\log d$ *and* $m = p/\log(eN/p)$. *Put*

$$\delta = \frac{\exp(-p/2)}{u^{2p}} - \frac{4\exp(-p/2)}{u^{c_0 m}} - \frac{c_1 \log^q N}{w^q N^{q/2-1}} - 2\exp(-v^2 t \log d) \tag{4.6}$$

and set

$$r^2(\rho) = c_2 \begin{cases} (uw + u^2 w)\sigma_q \Lambda(\rho) & \text{if } N \ge (\kappa\varepsilon/32)^2 d \\ \max\left\{ (uw + u^2 v)\sigma_q \Lambda(\rho), \kappa^2 \Lambda^2(\rho) \right\} & \text{otherwise.} \end{cases}$$

If $\hat{t}$ *is produced by the LASSO for a regularization parameter*

$$\lambda > c_4(uw + u^2 v)\kappa_0 \|\xi\|_{L_q} \eta^3 M \sqrt{\frac{\log d}{N}},$$

then with probability at least $1 - 5\delta - 2\exp(-\varepsilon^2 N/2)$,

$$R(\hat{t}) - R(t^*) = \left\|\langle X, \hat{t} - t^*\rangle\right\|_{L_2}^2 \le c_5 \max\left\{ r^2(c_6 \|t^*\|_1), \frac{\lambda}{\kappa^2 \varepsilon} \|t^*\|_1 \right\}.$$

Observe that like known estimates on the LASSO, and despite imposing considerably weaker assumptions on X and Y, the regularization parameter in Theorem 4.2 is of the order of $\|\xi\|_{L_q}\sqrt{(\log d)/N}$. And, when $\|\xi\|_{L_q}$ is equivalent to σ – the L_2 norm of ξ – then for $N \lesssim d$, the rate of convergence is

$$c(M) \max\left\{ \sigma \|t^*\|_1 \sqrt{\frac{\log d}{N}}, \|t^*\|_1^2 \frac{\log d}{N} \right\}$$

for a constant $c(M)$ that depends only on M.

Hence, up to a logarithmic factor, the LASSO attains the minimax rate in $\|t^*\|_1 B_1^d$ when $\log d \le N \lesssim d$ and when $\|t^*\|_1 \ge \sigma\sqrt{\log d/N}$; moreover, it does so without knowing in advance the identity of the true model $\|t^*\|_1 B_1^d$.

Note that one may want to combine the sparsity-dependent error rate from Theorem 1.3 in Lecué and Mendelson (to appear) and the complexity-dependent error rate from Theorem 4.2. To simplify the formulation we assume that X is subgaussian, in which case the probability estimate in Theorem 4.2 can be improved and the third condition in Assumption 4.1 (i.e. that $q > 2 + c_1/\log N$) can be relaxed to only $q > 2$ (see more details in the next section, and, in particular, Theorem 5.4). Combining the two approaches, one has that when X is isotropic and L-subgaussian, and when $\xi \in L_q$ for some $q > 2$ then for any $u, w > c$ with probability larger than $1 - \delta$ for

$$\delta = 2\exp(-c_2 N/L^8) - \frac{c_0 \log^q N}{w^q N^{q/2-1}} - c_0 \exp(-c_1 u^2/L^2),$$

the LASSO estimator $\hat{t}$ with the universal regularization parameter $\|\xi\|_{L_q}\sqrt{(\log d)/N}$ satisfies that

$$\left\|\hat{t} - t^*\right\|_2^2 \lesssim_{L,q} \min\left\{ \frac{\|t^*\|_0 \sigma^2 \log d}{N}, \max\left\{ \sigma \|t^*\|_1 \sqrt{\frac{\log d}{N}}, \|t^*\|_1^2 \frac{\log d}{N} \right\} \right\} \tag{4.7}$$

when $N \gtrsim \|t^*\|_0 \log(d/\|t^*\|_0)$.

Observe that (4.7) seemingly exhibits a different rate than Corollary 9.1 in Koltchinskii (2011) (see also (1.4)), the difference being the extra (and necessary) $\|t^*\|_1^2 \frac{\log d}{N}$ term in (4.7). This extra term appears only in the random design scenario, and the rates of convergence of the LASSO appear to deteriorate when

$$\sigma\sqrt{\frac{N}{\log d}} \le \|t^*\|_1 \le \sigma\sqrt{\|t^*\|_0}. \tag{4.8}$$

However, the sparsity-dependent error rate, and therefore Equation (4.7), holds only when $N \gtrsim \|t^*\|_0 \log(d/\|t^*\|_0)$. And, when $N \gtrsim \|t^*\|_0 \log d$ (which is only slightly larger than $\|t^*\|_0 \log(d/\|t^*\|_0)$), the error rates in the two scenarii (random and deterministic design) are the same and are given by

$$\min\left\{\frac{\sigma^2 \|t^*\|_0 \log d}{N}, \sigma \|t^*\|_1 \sqrt{\frac{\log d}{N}}\right\}.$$

Proof of Theorem 4.2. As noted previously, since $\|\cdot\|_1$ is a norm, $\Psi(t) = \|t\|_1$ satisfies Assumption 1.1 for $\eta = 1$ and Theorem 1.9 may be applied. To that end, one has to control $r(\rho) \equiv \max\{r_M(\rho), r_Q(\rho)\}$ and $\lambda_0(\delta,\tau)$. In what follows we will invoke the results of Section 3 and estimate these parameters.

Set $F(f^*,\rho) = F \cap K_\rho(f^*) - f^*$ and recall that $r_Q(\rho) = r_Q(F \cap K_\rho(f^*), \kappa\varepsilon/32)$ is determined by the behavior of

$$(\star) = \mathbb{E} \sup_{f \in F(f^*,\rho)\cap rD} \left|\frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i f(X_i)\right|; \tag{4.9}$$

as a consequence, it suffices to upper bound $(\star)$. Let $\mathcal{E} = \{t \in \mathbb{R}^d : \mathbb{E}\langle X,t\rangle^2 \le 1\}$, put $\mathcal{E}^\circ$ to be the polar of $\mathcal{E}$ (that is, $\mathcal{E}^\circ = \{u : \sup_{t\in\mathcal{E}} |\langle u,t\rangle| \le 1\}$), and set $\|t\|_{\mathcal{E}} = \sup_{x\in\mathcal{E}}\langle x,t\rangle$. Thus,

$$(\star) = \mathbb{E} \sup_{t\in\rho B_1^d \cap r\mathcal{E}} \left|\frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i\langle X_i,t\rangle\right| \le \min\left\{\mathbb{E}\sup_{t\in\rho B_1^d}\left|\frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i\langle X_i,t\rangle\right|, \mathbb{E}\sup_{t\in r\mathcal{E}}\left|\frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i\langle X_i,t\rangle\right|\right\}$$
$$= \min\left\{\rho\mathbb{E}\left\|\frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i X_i\right\|_{\ell_\infty^d}, r\mathbb{E}\left\|\frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i X_i\right\|_{\mathcal{E}^\circ}\right\}.$$

It is standard to verify (see, for instance, the proof of Lemma 2.2 in Lecué and Mendelson (2016)) that

$$\mathbb{E}\left\|\frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i X_i\right\|_{\mathcal{E}^\circ} \le \sqrt{d}.$$

Moreover, by (3.9),

$$\mathbb{E}\left\|\frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i X_i\right\|_{\ell_\infty^d} = \mathbb{E}\max_{1\le j\le d}\left|\frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i X_i(j)\right| \le c_0\kappa_0\sqrt{\log d}\max_{1\le j\le d}\|x_j\|_{L_2}.$$

Therefore,

$$(\star) \le \min\left\{c_0\rho\kappa_0\sqrt{\log d}\max_{1\le j\le d}\|x_j\|_{L_2}, r\sqrt{d}\right\}, \tag{4.10}$$

and setting $\gamma = \kappa\varepsilon/32$, one has

$$r_Q(\rho) \le \begin{cases} 0 & \text{if } N \ge \gamma^2 d \\ \frac{c_0\rho\kappa_0}{\gamma}\sqrt{\frac{\log d}{N}}M & \text{otherwise.}\end{cases}$$

Next, let us establish a high probability upper bound on $r_M(\rho) = r_M(F\cap K_\rho(f^*), \kappa^2\varepsilon/80, \delta/4)$. Note that

$$\phi_N(F\cap K_\rho(f^*), f^*, s) = \sup_{t\in\rho B_1^d\cap s\mathcal{E}}\left|\frac{1}{\sqrt{N}}\sum_{i=1}^N \varepsilon_i\xi_i\langle X_i,t\rangle\right| \le \rho\max_{1\le j\le d}\left|\frac{1}{\sqrt{N}}\sum_{i=1}^N \varepsilon_i\xi_i X_i(j)\right|.$$

Applying the second part of Theorem 3.6 for $u > 2, v > 0$, $w \ge 2$, $p = (a_2/2)\log d, m = p/\log(eN/p)$ and

$$\delta = \frac{\exp(-p/2)}{u^{2p}} - \frac{4\exp(-p/2)}{u^{c_0 m}} - \frac{c_1\log^q N}{w^q N^{q/2-1}} - 2\exp(-v^2 t\log d), \tag{4.11}$$

it follows that with probability at least $1-\delta$,

$$\phi_N(F\cap K_\rho(f^*), f^*, s) \le c_2\kappa_0(uw+u^2v)\|\xi\|_{L_q}\rho M\sqrt{\log d};$$

thus,

$$r_M^2(\rho) \le \frac{c_3\kappa_0(uw+u^2v)}{\kappa^2\varepsilon}\|\xi\|_{L_q}\rho M\sqrt{\frac{\log d}{N}}.$$

Finally, let us identify an upper bound on $\lambda_0(\delta,\tau)$ for $\tau = 3/(80\eta^3)$. Let $\{e_1,\dots,e_d\}$ be the canonical basis of $\mathbb{R}^d$. Since $K_\rho(f^*) = \{t : \|t-t^*\|_1 \le \rho\}$, we have

$$(\star_1) = \sup_{f\in F\cap K_\rho(f^*)\cap r(\rho)D_{f^*}}\left(\frac{1}{N}\sum_{i=1}^N \xi_i(f-f^*)(X_i) - \mathbb{E}\xi(f-f^*)(X)\right)$$
$$\le \rho\max_{t-t^*\in\{\pm e_1,\dots,\pm e_d\}}\left(\frac{1}{N}\sum_{i=1}^N \xi_i\langle X_i,t-t^*\rangle - \mathbb{E}\xi\langle X,t-t^*\rangle\right)$$
$$= \rho\max_{t\in\{\pm e_1,\dots,\pm e_d\}}\left(\frac{1}{N}\sum_{i=1}^N \xi_i\langle X_i,t\rangle - \mathbb{E}\xi\langle X,t\rangle\right).$$

Recall that $X = (x_j)_{j=1}^d$. By a standard symmetrization argument (see, for example, Lemma 2.3.7 in van der Vaart and Wellner (1996)), if $z \ge 4\max_{1\le j\le d}\sqrt{\mathrm{Var}(\xi x_j)/N}$ then

$$Pr\Big(\max_{t\in\{\pm e_1,\dots,\pm e_d\}}\frac{1}{N}\sum_{i=1}^N \xi_i\langle X_i,t\rangle - \mathbb{E}\xi\langle X,t\rangle \ge z\Big) \le 4Pr\Big(\max_{t\in\{\pm e_1,\dots,\pm e_d\}}\frac{1}{N}\sum_{i=1}^N \varepsilon_i\xi_i\langle X_i,t\rangle \ge \frac{z}{4}\Big).$$

Note that $\sqrt{\mathrm{Var}(\xi x_j)} \le \sqrt{\mathbb{E}\xi^2 x_j^2} \le \|\xi\|_{L_q}\|x_j\|_{L_{2q'}}$ where q' is the conjugate index of $q/2$. Therefore, $\sqrt{\mathrm{Var}(\xi x_j)} \le \kappa_0\sqrt{2q'}\|\xi\|_{L_q}M$ as long as $2q' \le a_2\log d$, i.e., when $q \ge 2+$

$2/(a_2 \log d - 1)$ – which is the case under Assumption 4.1. Therefore, applying the second part of Theorem 3.6 for δ as in (4.11), it follows that with probability at least $1-\delta$,

$$(\star_1) \leq \rho c_0 \kappa_0 (uw + u^2 v) \|\xi\|_{L_q} M \sqrt{\frac{\log d}{N}},$$

and, for $\tau = 3/(80\eta^3)$ one may select

$$\lambda_0(\delta, \tau) = c_4 \kappa_0 (uw + u^2 v) \|\xi\|_{L_q} \eta^3 w M \sqrt{\frac{\log d}{N}}.$$

■

5. Regularization methods for subgaussian classes

In this section, we assume that X is a random vector that takes its values in a Hilbert space $\mathcal{H}$. The main examples we consider are when $\mathcal{H}$ is the d-dimensional Euclidean space and when it is the space of $m \times T$ matrices endowed with the Frobenius norm.

The inner product in $\mathcal{H}$ is denoted by $\langle \cdot, \cdot \rangle$, and the norm and unit ball endowed by the inner product are denoted by $\|\cdot\|_{\mathcal{H}}$ and $B_{\mathcal{H}} = \{t \in \mathcal{H} : \|t\|_{\mathcal{H}} \leq 1\}$ respectively.

There is another natural Hilbertian structure on $\mathcal{H}$, endowed by $\Sigma = \mathbb{E} X X^{\top}$, the covariance operator associated with the random vector X. The corresponding unit ball $\mathcal{E} = \{t \in \mathcal{H} : \mathbb{E}\langle X, t\rangle^2 \leq 1\}$, is an ellipsoid in $\mathcal{H}$.

Let $T \subset \mathcal{H}$ be a closed and convex set and put

$$t^* \in \operatorname*{argmin}_{t \in T} \mathbb{E}(Y - \langle X, t\rangle)^2.$$

Let $\Psi(\cdot)$ be a regularization function on $\mathcal{H}$ that satisfies Assumption 1.1 and set

$$\hat{t} \in \operatorname*{argmin}_{t \in T} \left(\frac{1}{N} \sum_{i=1}^N \left(Y_i - \langle X_i, t\rangle\right)^2 + \lambda \Psi(t) \right) \tag{5.1}$$

for a well-chosen regularization parameter λ.

Unlike the results of the previous section, in what follows we assume that $F = \{\langle t, \cdot \rangle : t \in T\}$ is an L-subgaussian class (see Definition 3.1). Moreover, F satisfies a small-ball property with constants that depend only on L. Indeed, observe that for every $t \in T$

$$\|\langle X, t\rangle\|_{L_4} \lesssim L \|\langle X, t\rangle\|_{L_2},$$

and applying the Paley-Zygmund inequality (see, e.g., Corollary 3.3.2 in de la Peña and Giné (1999)),

$$Pr\left(|\langle X, t\rangle| \geq \kappa \|\langle X, t\rangle\|_{L_2} \right) \geq \varepsilon \text{ for } \kappa = 1/2 \text{ and } \varepsilon = c/L^4. \tag{5.2}$$

From here on we say that the random vector X taking its values in $\mathcal{H}$ is L-subgaussian if the class consisting of all the linear functionals on $\mathcal{H}$, i.e., $\{\langle t, \cdot \rangle : t \in \mathcal{H}\}$, is L-subgaussian. Also, throughout this section, we assume that $\xi = Y - f^*(X) \in L_q$ for some $q > 2$, and denote $\sigma_q = \|\xi\|_{L_q}$ and

$$T \cap K_\rho(t^*) = \{t \in T : \Psi(t - t^*) \leq \rho\}.$$

5.1 'Heavy tailed' noise

Thanks to the subgaussian assumption, both $r(\rho)$ and $\lambda_0 = \lambda_0(\delta, 3/(80\eta^3))$ may be determined using the Gaussian mean-widths of the sets $T \cap K_\rho(t^*)$ for all $\rho > 0$. Recall that for $T_0 \subset \mathcal{H}$ the Gaussian mean-width of T_0 is $\ell^*(T_0) = \mathbb{E}\sup_{t \in T_0} G_t$, where $(G_t)_{t \in T_0}$ is the centered canonical Gaussian process indexed by T_0 with covariance structure given by $\mathbb{E} G_{t_1} G_{t_2} = \mathbb{E}\langle X, t_1\rangle \langle X, t_2\rangle$ for every $t_1, t_2 \in T$.

Definition 5.1 *Let $r\mathcal{E}_{t^*} = \left\{t \in \mathcal{H} : \|\langle t - t^*, \cdot\rangle\|_{L_2(\mu)} \leq r\right\} = t^* + r\mathcal{E}$, and for $\alpha, \beta > 0$ set*

$$\tilde{r}_Q(\rho, \alpha) = \inf\left\{ r > 0 : \ell^*\left(T \cap K_\rho(t^*) \cap r\mathcal{E}_{t^*}\right) \leq \alpha r \sqrt{N} \right\}$$

and

$$\tilde{r}_M(\rho, \beta) = \inf\left\{ r > 0 : \ell^*\left(T \cap K_\rho(t^*) \cap r\mathcal{E}_{t^*}\right) \leq \beta r^2 \sqrt{N} \right\}.$$

Let c_0 be an absolute constant to be specified later. Fix $u, w > c$, and ε and κ as in (5.2). Consider

$$\alpha = \frac{\kappa \varepsilon}{c_0 L}, \quad \beta = \frac{\kappa^2 c_1 \varepsilon}{Lwu \|\xi\|_{L_q}}, \quad \text{and} \quad \gamma = c_0 \eta^3 L w u \|\xi\|_{L_q}, \tag{5.3}$$

put

$$r(\rho) \geq \max\left\{\tilde{r}_Q(\rho, \alpha), \tilde{r}_M(\rho, \beta)\right\} \tag{5.4}$$

and set

$$\lambda_0(\gamma) = \gamma \sup_{\rho > 0, t^* \in T} \frac{\ell^*\left(T \cap K_\rho(t^*) \cap r(\rho)\mathcal{E}_{t^*}\right)}{\rho \sqrt{N}}. \tag{5.5}$$

The first result we present is rather general and holds for any closed and convex subset $T \subset \mathcal{H}$ and any regularization function satisfying Assumption 1.1. It allows one to take into account an additional constraint on the "signal" $t^* \in T$.

Theorem 5.2 *There are absolute constants c, c_1 and c_2 for which the following holds. Let Ψ be a regularization function satisfying Assumption 1.1. Assume that X is L-subgaussian for some $L > 0$ and that $\xi = Y - \langle X, t^*\rangle$ is in L_q for some $q > 2$.*

If $\hat{t}$ is given by (5.1) for a regularization parameter $\lambda > \lambda_0(\gamma)$ as in (5.5), then with probability larger than

$$1 - 2\exp(-N\varepsilon^2/8) - \frac{c_0 \log^q N}{w^q N^{q/2-1}} - c_0 \exp(-c_1 u^2/L^2), \tag{5.6}$$

$$\|\langle X, \hat{t} - t^*\rangle\|_{L_2}^2 \leq \max\left\{ r(10\eta\Psi(t^*))^2, (32/\kappa^2\varepsilon)\lambda\Psi(t^*) \right\}$$

for $r(\cdot)$ given by (5.4).

Proof. The proof follows from Theorem 1.9 by estimating $r(\rho)$ and λ_0 using the 'local' Gaussian mean-widths of the sets $T \cap K_\rho(t^*)$.

Since X is L-subgaussian, the process $\{\langle X,t\rangle : t\in\mathcal{H}\}$ is L-subgaussian. Setting $F=\{\langle t,\cdot\rangle : t\in\mathcal{H}\}$ and $f^*=\langle t^*,\cdot\rangle$, a standard chaining argument shows that

$$\mathbb{E}\sup_{f\in F\cap K_\rho(f^*)\cap rD_{f^*}}\Big|\frac{1}{N}\sum_{i=1}^N\varepsilon_i(f-f^*)(X_i)\Big|\le c_0L\frac{\ell^*(T\cap K_\rho(t^*)\cap r\mathcal{E}_{t^*})}{\sqrt{N}}.$$

Thus,

$$r_Q\Big(F\cap K_\rho(f^*),\frac{\kappa\varepsilon}{32}\Big)\le\tilde{r}_Q(\rho,\alpha). \tag{5.7}$$

As for the fixed point associated with the multiplier process, it follows from Corollary 3.3 that

$$r_M\Big(F\cap K_\rho(f^*),\frac{\kappa^2\varepsilon}{160},\frac{\delta}{4}\Big)\le\tilde{r}_M(\rho,\beta) \tag{5.8}$$

for β as defined in (5.3), and as long as

$$\frac{\delta}{4}\ge\frac{c_0\log^q N}{w^qN^{q/2-1}}+2\exp(-c_1u^2/L^2).$$

Finally by Corollary 3.5, $\lambda_0(\delta,\gamma)\le\lambda_0(\gamma)$. The claim now follows from Theorem 1.9. ∎

If one is to apply Theorem 5.2, an essential component is an upper bound on $\ell^*(T\cap K_\rho(t^*)\cap r\mathcal{E}_{t^*})$, leading to estimates on r and λ. To simplify the analysis we shall use an additional assumption on Ψ:

Assumption 5.1 *Assume that for every $x,y\in\mathcal{H}$ and $\lambda\ge0$,*

$$\Psi(x)=\Psi(-x),\qquad\Psi(x+y)\le\eta\big(\Psi(x)+\Psi(y)\big)\ \text{and}\ \Psi(\lambda x)\le\lambda\Psi(x). \tag{5.9}$$

Also, recall that $\mathcal{E}=\{t\in\mathcal{H}:\mathbb{E}\langle X,t\rangle^2\le1\}$, $\sigma_q=\|\xi\|_q$ and set $K=\{t\in\mathcal{H}:\Psi(t)\le1\}$.

Theorem 5.3 *Assume that Ψ satisfies Assumption 5.1 and that the assumptions of Theorem 5.2 hold. Let $\Lambda(\rho)\ge\rho\ell^*(K)/\sqrt{N}$ for every $\rho>0$, $w,u>c$ and consider the RERM*

$$\hat{t}\in\operatorname*{argmin}_{t\in\mathcal{H}}\Big(\frac{1}{N}\sum_{i=1}^N(Y_i-\langle X_i,t\rangle)^2+c_0\eta^3Lwu\sigma_q\Lambda(\Psi(t))\Big).$$

Then, with probability larger than the one in (5.6),

$$R(\hat{t})-R(t^*)=\big\|\langle X,\hat{t}-t^*\rangle\big\|_{L_2}^2\le c_0r^2(10\eta\Psi(t^*))$$

where, for α and β defined in (5.3) and, for any $\rho\ge0$,

$$r^2(\rho)=\begin{cases}\frac{\Lambda(\rho)}{\beta} & \text{if } N\ge\big(\ell^*(\mathcal{E})/\alpha\big)^2\\ \max\Big\{\frac{\Lambda(\rho)}{\beta},\frac{\Lambda^2(\rho)}{\alpha^2}\Big\} & \text{otherwise.}\end{cases} \tag{5.10}$$

Proof. The result follows immediately from Theorem 5.2. Indeed, for every $\rho>0$, $r>0$ and $t^*\in T=\mathcal{H}$,

$$\ell^*(T\cap K_\rho(t^*)\cap r\mathcal{E}_{t^*})=\ell^*(K_\rho(0)\cap r\mathcal{E})\le\ell^*(\rho K\cap r\mathcal{E})\le\min\{\rho\ell^*(K),r\ell^*(\mathcal{E})\},$$

because $K_\rho(0)\subset\rho K=\{\rho t:t\in K\}$.

∎

Note that in a d-dimensional space, the trivial bound $\ell^*(\mathcal{E})\le\sqrt{d}$ holds (see, e.g., Lemma 2.2 in Lecué and Mendelson (2016)). Therefore, one only needs to control $\ell^*(K)$. In the next section, we provide several examples of applications of Theorem 5.3 that follow from estimates on $\ell^*(K)$. We will simplify the analysis further by assuming that there is some compatibility between the norm $\|\cdot\|_{\mathcal{H}}$ and the one endowed by the covariance structure of X:

Assumption 5.2 *Assume that X is isotropic; that is, for every $t\in\mathcal{H}$, $\big(\mathbb{E}\langle X,t\rangle^2\big)^{1/2}=\|t\|_{\mathcal{H}}$.*

Observe that Assumption 5.2 implies that $\ell^*(K)=\mathbb{E}\sup_{t\in K}G_t$, where $(G_t)_{t\in K}$ is the canonical Gaussian process indexed by K with the covariance $\mathbb{E}G_{t_1}G_{t_2}=\langle t_1,t_2\rangle$ for every $t_1,t_2\in K$. Indeed, this follows because the inner-product in $\mathcal{H}$ coincides with the one endowed by $L_2(\mu)$.

5.2 Regularization methods in $\mathbb{R}^d$

Consider a regularization function $\Psi(\cdot)$ satisfying Assumption 5.1. Assume that X is L-subgaussian and isotropic in $\mathbb{R}^d$ with respect to the standard Euclidean inner-product, and that $\xi\in L_q$ for some $q>2$. Let $u,w>c$. For any $\rho\ge0$ set $\Lambda(\rho)\ge\rho\ell^*(K)/\sqrt{N}$ and put

$$r^2(\rho)\sim_{L,q}\begin{cases}wu\sigma_q\Lambda(\rho) & \text{when } N\gtrsim_L d\\ \max\big\{wu\sigma_q\Lambda(\rho),\Lambda^2(\rho)\big\} & \text{otherwise.}\end{cases}$$

It follows from Theorem 5.3 that if

$$\hat{t}\in\operatorname*{argmin}_{t\in\mathbb{R}^d}\Big(\frac{1}{N}\sum_{i=1}^N(Y_i-\langle X_i,t\rangle)^2+c_0\eta^3Lwu\sigma_q\Lambda(\Psi(t))\Big) \tag{5.11}$$

then with probability larger than the one in (5.6)

$$\big\|\langle\hat{t}-t^*,\cdot\rangle\big\|_{L_2(\mu)}^2\lesssim r^2(10\eta\Psi(t^*)).$$

As a consequence, one can derive an estimation result for (5.11) whenever $\ell^*(K)$ may be controlled from above. In the following section, we shall apply this observation to some classical problems and compare the error rates obtained by the RERM (5.11) to the minimax rate in the true model $\{t\in T:\Psi(t)\le\Psi(t^*)\}$.

Example: ℓ_p-regularization for $1\le p\le\infty$. Consider the regularization function $\Psi(t)=\|t\|_p$ for some $p\ge1$. Assumption 5.1 holds with $\eta=1$ because $\|\cdot\|_p$ is a norm. In order to apply the general result for the RERM in (5.11), one has to estimate the Gaussian mean-width of the unit ball associated with the regularization function $\Psi(\cdot)=\|\cdot\|_p$.

In the range $1\le p\le1+(\log d)^{-1}$, we obtain the same performance as the LASSO, because $B_1^d\subset B_p^d\subset cB_1^d$ for a suitable absolute constant c, implying that $\ell^*(B_p^d)\sim\ell^*(B_1^d)\sim\sqrt{\log(ed)}$.

When $1+(\log(ed))^{-1}\le p$, set r to be the conjugate index for p and one may easily verify that $\ell^*(B_p^d)\sim\sqrt{r}d^{1/r}$.

Applying Theorem 5.3, one has the following:

Theorem 5.4 *Under the assumptions of Theorem 5.3 and using its notation,*

- *If* $1 \le p \le 1 + 1/(\log d)$ *and*

$$\hat{t} \in \operatorname*{argmin}_{t \in \mathbb{R}^d} \Big(\frac{1}{N} \sum_{i=1}^N (Y_i - \langle X_i, t \rangle)^2 + c_2 \eta_p^3 L w u \sigma_q \|t\|_p \sqrt{\frac{\log d}{N}} \Big)$$

then with probability larger than the one in (5.6),

$$\big\|\hat{t} - t^*\big\|_2^2 \lesssim_{p,L,q} \begin{cases} w u \sigma_q \|t^*\|_p \sqrt{\frac{\log d}{N}} & \text{if } N \gtrsim_L d, \\ \max\Big\{ w u \sigma_q \|t^*\|_p \sqrt{\frac{\log d}{N}}, \|t^*\|_p^2 \frac{\log d}{N} \Big\} & \text{otherwise.} \end{cases}$$

- *If* $p \ge 1 + 1/(\log d)$ *and*

$$\hat{t} \in \operatorname*{argmin}_{t \in \mathbb{R}^d} \Big(\frac{1}{N} \sum_{i=1}^N (Y_i - \langle X_i, t \rangle)^2 + c_2 \sigma_q L w u \|t\|_p \frac{\sqrt{p/(p-1)} d^{(p-1)/p}}{\sqrt{N}} \Big),$$

then with probability larger than the one in (5.6)

$$\big\|\hat{t} - t^*\big\|_2^2 \lesssim_{L,q} \begin{cases} w u \sigma_q \|t^*\|_p \frac{d^{(p-1)/p}}{p\sqrt{N}} & \text{if } N \gtrsim_L d, \\ \max\Big\{ w u \sigma_q \|t^*\|_p \frac{d^{(p-1)/p}}{\sqrt{N}}, \|t^*\|_p^2 \frac{d^{2(p-1)/p}}{N} \Big\} & \text{otherwise.} \end{cases}$$

Remark. [**the case** $0 < p < 1$] Despite being a non-convex function, ℓ_p-regularization for $0 < p < 1$ has attracted much attention in the context of Signal Processing and High-Dimensional Statistics. Among the problems studied using ℓ_p regularization were the linear regression model with a deterministic design (cf. Raskutti et al. (2011); Rigollet and Tsybakov (2011); Verzelen (2012)); the sequence space model Donoho and Johnstone (1994); Abramovich et al. (2006); and the random design linear regression model Wang et al. (2014).

From our point of view, there is no particular restriction on p as long as the regularization function satisfies Assumption 5.1. We can therefore consider the regularization function $\Psi(t) = \|t\|_p$ for any $0 < p < 1$. In that range of p, Assumption 5.1 holds for $\eta = \eta_p = 2^{1/p}$ (see, for example, page 2 in Edmunds and Triebel (1996)) and the Gaussian mean width of the "unit ball" associated with $\Psi(\cdot) = \|\cdot\|_p$ for $0 < p < 1$ can also be computed.

To that end, let $\{e_1, \dots, e_d\}$ be the canonical basis of $\mathbb{R}^d$. Since $\{\pm e_1, \dots, \pm e_d\} \subset B_p^d \subset B_1^d$ for $p < 1$, it is evident that $\ell^*(B_p^d) \sim \sqrt{\log d}$. Thus, the error rates of the LASSO from Theorem 4.2 dominate all the ℓ_p-regularization rates when $0 < p \le 1$.

However, the resulting rate is not the minimax rate in the true model, as can be seen from Raskutti et al. (2011). Indeed, fix $0 < p \le 1$. Consider an unknown $t^* \in \rho B_p^d$ and the corresponding Gaussian linear model $Y_i = \langle x_i, t^* \rangle + W_i, i = 1, \dots, N$, where the matrix whose rows are $(x_i)_{i=1}^N$ satisfies some RIP property and $W_1, \cdots, W_N$ are independent, centered Gaussian variables with variance σ^2. For specific asymptotic rates in N and d (see

Raskutti et al. (2011) for a precise formulation), the authors show that minimax rate of the problem is given by

$$\sigma^2 \rho \Big(\frac{\log d}{N} \Big)^{1 - \frac{p}{2}},$$

and similar results have been obtained in Wang et al. (2014). Thus, our estimate recovers the minimax rate in the true model only when $p = 1$. When $0 < p < 1$, it is possible that the choice of the $\Psi(t) = \|t\|_p$ is suboptimal, and instead one should use $\Psi(t) = \|t\|_p^p$ as was suggested in Rohde and Tsybakov (2011) for the problem of S_p-regularization for $0 < p \le 1$.

Example: weak-ℓ_p-regularization for $0 < p \le 1$. Weak-ℓ_p norms have been used to model sparsity in High-Dimensional Statistics (see, for instance, Abramovich et al. (2006); Wang et al. (2014)). To define those norms, let $t_1^* \ge t_2^* \ge \ldots \ge t_d^*$ be the non-increasing rearrangement of $(|t_i|)_{i=1}^d$. Set $\|t\|_{p\infty} = \max_{1 \le j \le d} j^{1/p} t_j^*$ and put $B_{p\infty}^d = \{t \in \mathbb{R}^d : t_j^* \le j^{-1/p} \text{ for every } 1 \le j \le d\}$.

One can use the following well-known fact (see, e.g., Theorem B in Gordon et al. (2007)) to control the Gaussian mean-width of the unit ball associated with $\|\cdot\|_{p\infty}$.

Proposition 5.5 *For* $0 < p \le 1$,

$$\ell^*(B_{p\infty}) \lesssim \begin{cases} \frac{\sqrt{\log d}}{p-1} & \text{if } 0 < p < 1 \\ \big(\log d\big)^{3/2} & \text{if } p = 1. \end{cases}$$

Applying Theorem 5.3, one has the following:

Theorem 5.6 *Under the assumptions of Theorem 5.3 and using its notation,*

- *If* $p < 1$ *and*

$$\hat{t} \in \operatorname*{argmin}_{t \in \mathbb{R}^d} \Big(\frac{1}{N} \sum_{i=1}^N (Y_i - \langle X_i, t \rangle)^2 + c_2 \eta_p^3 \sigma_q L w u \|t\|_{p\infty} \sqrt{\frac{\log d}{N}} \Big),$$

then with probability larger than the one in (5.6)

$$\big\|\hat{t} - t^*\big\|_2^2 \lesssim_{p,L,q} \begin{cases} \sigma_q w u \|t^*\|_{p\infty} \sqrt{\frac{\log d}{N}} & \text{if } N \gtrsim_L d, \\ \max\Big\{ \sigma_q w u \|t^*\|_p \sqrt{\frac{\log d}{N}}, \|t^*\|_{p\infty}^2 \frac{\log d}{N} \Big\} & \text{otherwise.} \end{cases}$$

- *If* $p = 1$ *and*

$$\hat{t} \in \operatorname*{argmin}_{t \in \mathbb{R}^d} \Big(\frac{1}{N} \sum_{i=1}^N (Y_i - \langle X_i, t \rangle)^2 + c_2 \eta_1^3 \sigma_q \|t\|_{1\infty} \sqrt{\frac{\log^3 d}{N}} \Big),$$

then with probability larger than the one in (5.6)

$$\big\|\hat{t} - t^*\big\|_2^2 \lesssim_{L,\delta,q} \begin{cases} \sigma_q w u \|t^*\|_{1\infty} \sqrt{\frac{\log^3 d}{N}} & \text{if } N \gtrsim_L d, \\ \max\Big\{ \sigma_q w u \|t^*\|_{1\infty} \sqrt{\frac{(\log d)^3}{N}}, \|t^*\|_{1\infty}^2 \frac{\log^3 d}{N} \Big\} & \text{otherwise.} \end{cases}$$

Example: the Micchelli, Morales and Pontil's regularization functions.
Let Θ be a nonempty convex cone in $[0,\infty)^d$, and for every $t \in \mathbb{R}^d$ set

$$\Omega(t|\Theta) = \inf_{\theta\in\Theta} \frac{1}{2}\sum_{j=1}^d \Big(\frac{t_j^2}{\theta_j} + \theta_j\Big). \tag{5.12}$$

It was shown in Maurer et al. (2010) that $\Omega(\cdot|\Theta)$ is a norm on $\mathbb{R}^d$.

This family of norms captures several classical regularization functions, by an appropriate choice of the cone Θ. For instance, the ℓ_1^d-norm is obtained by selecting $\Theta = [0,\infty)^d$. Also, the *group LASSO* introduced in Yuan and Lin (2006) is generated by a cone: indeed, if $(G_1,\cdots,G_T)$ is a partition of $\{1,\dots,d\}$ and

$$\Theta = \{\theta \in [0,\infty)^d \text{ that is constant within each group } G_\ell\}, \tag{5.13}$$

then

$$\Omega(t|\Theta) = \sum_{\ell=1}^T \sqrt{|G_\ell|}\,\|t_{|G_\ell}\|_2\,,$$

where $|G_\ell|$ is the cardinality of the set of coordinates G_ℓ and $t_{|G_\ell}$ is the restriction of t to G_ℓ.

Error bounds for procedures that use $\Psi(t) = \Omega(t|\Theta)$ as regularization functions have been established in Maurer and Pontil (2012), under the assumption that the loss functions is bounded and Lipschitz (see Theorem 1 there). Naturally, the squared loss is not covered by such a result because it is not bounded in $\mathbb{R}^d$, nor is it Lipschitz. Our aim is to provide a version of the result from Maurer and Pontil (2012) for the quadratic loss and when X is subgaussian and the noise belongs to L_q for some $q > 2$. The first step in that direction is the following estimate on the Gaussian mean width of the corresponding unit ball.

Proposition 5.7 *Let Θ be a nonempty convex cone in $[0,\infty)^d$ and set $\mathcal{B} = \{t : \Omega(t|\Theta) \le 1\}$. Let S_1^{d-1} be the unit sphere of ℓ_1^d and put $\mathcal{E}x$ to be the set of extreme points of $\Theta \cap S_1^{d-1}$. If $M = \max_{a\in\mathcal{E}x}\|a\|_\infty^{1/2}$, then, for an absolute constant c,*

$$\ell^*(\mathcal{B}) \le 1 + cM\sqrt{2\log\big(|\mathcal{E}x|\big)}. \tag{5.14}$$

The proof of Proposition 5.7 may be derived in various ways (see a similar result in Maurer and Pontil (2012)), though we will use a chaining argument which actually leads to a stronger estimate than (5.14).

Definition 5.8 *Let $T \subset \mathbb{R}^d$ and set $\|\cdot\|$ to be a norm on $\mathbb{R}^d$. For every $\alpha > 1$ let*

$$\gamma_\alpha(T,\|\cdot\|) = \inf_{(T_s)}\sup_{t\in T}\sum_{s=0}^\infty 2^{s/\alpha}\,\|\pi_{s+1}t - \pi_s t\|\,;$$

the infimum is taken with respect to all sequences (T_s) of subsets of T for which $|T_0| = 1$ and for $s \ge 1$, $|T_s| \le 2^{2^s}$, and $\pi_s t$ denotes the nearest point to t in T_s with respect to $\|\cdot\|$.

Clearly, if T is finite then $\gamma_\alpha(T,\|\ \|) \lesssim \sup_{t\in T}\|t\|\cdot\log^{1/\alpha}|T|$.

Proof of Proposition 5.7. It is straightforward to verify (see, e.g., Maurer et al. (2010)) that the dual norm to $\Omega(\cdot|\Theta)$ is

$$\Omega^*(t|\Theta) = \max_{a\in\mathcal{E}x}\Big(\sum_{j=1}^d a_j t_j^2\Big)^{1/2}. \tag{5.15}$$

Let $g_1,\dots,g_d$ be independent, standard Gaussian random variables, Applying a Bernstein type inequality for a sum of independent ψ_1 random variables (see Corollary 2.10 in Talagrand (1994)), it follows that for every $a_1,\dots,a_N$, every $u > 0$ and any $s \in \mathbb{N}$,

$$Pr\left[\Big|\sum_{j=1}^d a_j(g_j^2-1)\Big| \ge u2^{s/2}\|a\|_2 + u^2 2^s\|a\|_\infty\right] \le 2\exp(c_1 2^s u^2).$$

Hence, using a standard chaining argument,

$$\mathbb{E}\sup_{a\in\mathcal{E}x}\sum_{j=1}^d a_j g_j^2 \le 1 + c_2\left(\gamma_2(\mathcal{E}x,\|\ \|_2) + \gamma_1(\mathcal{E}x,\|\ \|_\infty)\right).$$

Now one may apply the trivial estimates on γ_1 and γ_2. Firstly, $\gamma_1(\mathcal{E}x,\|\cdot\|_\infty) \lesssim M^2\log(|\mathcal{E}x|)$, and secondly, noting that $|\sum_{j=1}^d a_j| \le \|a\|_1 = 1$ and thus $\|a\|_2 \le \|a\|_\infty^{1/2}$, one has $\gamma_2(\mathcal{E}x,\|\cdot\|_2) \lesssim M\sqrt{\log(|\mathcal{E}x|)}$. Therefore, by Jensen's inequality,

$$\mathbb{E}\sup_{a\in\mathcal{E}x}\Big(\sum_{j=1}^d a_j g_j^2\Big)^{1/2} \le 1 + cM\sqrt{\log(|\mathcal{E}x|)}.$$

■

Theorem 5.9 *Using the notation above and of Theorem 5.3, let*

$$\Lambda(t) = \Omega(t|\Theta)M\sqrt{\frac{\log(|\mathcal{E}x|)}{N}}.$$

If

$$\hat{t} \in \operatorname*{argmin}_{t\in\mathbb{R}^d}\Big(\frac{1}{N}\sum_{i=1}^N (Y_i - \langle X_i,t\rangle)^2 + c_2\sigma_q L w u\Lambda(t)\Big)$$

then with probability larger than the one in (5.6)

$$\|\langle\hat{t}-t^*,\cdot\rangle\|_{L_2(\mu)}^2 \lesssim_{L,q} \begin{cases} \sigma_q w u\Lambda(t^*) & \text{if } N \gtrsim_L d,\\ \max\left\{\sigma_q L w u\Lambda(t^*),\Lambda^2(t^*)\right\} & \text{otherwise.}\end{cases}$$

When $\Theta = [0,\infty)^d$ then $M\sqrt{\log(\mathcal{E}x)} \lesssim \sqrt{\log d}$. Hence, Theorem 5.9 yields the same error rate as the one obtained for the LASSO in Theorem 4.2 and in Theorem 5.4, though under a stronger assumption on X. This is not surprising because when $\Theta = [0,\infty)^d$, $\Omega(t|\Theta) = \|t\|_1$ and the resulting RERM is just the LASSO.

In the case of the group LASSO, for Θ as in (5.13), one has $M\sqrt{\log(\mathcal{E}x)} \lesssim \sqrt{\log|T|}$, and $\Lambda(t^*) \sim \Omega(t^*|\Theta)M\sqrt{(\log|T|)/N}$.

Example: The SLOPE regularization

In Su and Candès (2016); Bogdan et al. (2015), the authors introduced the regularization function:

$$\Psi(t) = \|t\|_{SLOPE} = \sum_{j=1}^d \lambda_j t_j^*$$

where $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$ and $t_1^* \geq \cdots \geq t_d^* \geq 0$ is the non-increasing rearrangement of $(|t_i|)_{i=1}^d$.

In Su and Candès (2016) the given data is generated by the Gaussian linear model $Y_i = \langle X_i, t^*\rangle + W_i, i = 1,\ldots,N$ for a Gaussian design $X_i \sim \mathcal{N}(0, N^{-1}I_{d\times d})$ (note that the covariance matrix is normalized by $1/N$) and a centered Gaussian noise W_i with variance σ^2 that is independent of the design X_i. Setting $\Phi^{-1}(\alpha)$ to be the α-th quantile of a standard normal distribution and $q \in (0,1)$, the weights were chosen to be

$$\lambda_i = \Phi^{-1}(1 - iq/(2d)), \tag{5.16}$$

and, for this choice of weights, SLOPE was defined by

$$\hat{t} \in \operatorname*{argmin}_{t\in\mathbb{R}^d}\Big(\frac{1}{2N}\sum_{i=1}^N (Y_i - \langle X_i, t\rangle)^2 + \sigma\frac{\|t\|_{SLOPE}}{\sqrt{N}}\Big).$$

The result in Su and Candès (2016) is of an asymptotic nature:

Theorem 5.10 (Theorem 1.2 Su and Candès (2016)) *Let $0 < \varepsilon < 1$ and set $1 \leq k \leq d$ that satisfy $k/d = o(1)$ and $(k\log d)/N = o(1)$ when $N \to \infty$. Then,*

$$\lim_{N\to\infty}\sup_{\|t^*\|_0\leq k} Pr\Big(\frac{N\left\|\hat{t} - t^*\right\|_2^2}{2\sigma^2 k\log(d/k)} > 1 + 3\varepsilon\Big) = 0,$$

where the supremum is taken with respect to all vectors that are supported on at most k coordinates.

It was shown in Su and Candès (2016) that $2\sigma^2 k\log(d/k)/N$ is the (asymptotic) minimax rate for t^* that is k-sparse.

The article Su and Candès (2016) (see Section 6 there) raises the question of extending Theorem 5.10 beyond the Gaussian case, especially when the coordinates of X may be correlated. We study this question in the context of sparse recovery and for an arbitrary choice of weights in Lecué and Mendelson (to appear), leading to error bounds that depend on $\|t^*\|_0$. Here, we obtain a complexity-dependent error rate that depends on $\|t^*\|_{SLOPE}$.

Proposition 5.11 *Set $\mathcal{B} = \{t \in \mathbb{R}^d : \|t\|_{SLOPE} \leq 1\}$. There exists an absolute constant C, for which, if $M = \max_{1\leq j\leq d}\lambda_j^{-1}\sqrt{\log(ed/j)}$, then $\ell^*(\mathcal{B}) \leq CM$.*

Proof. The proof is an outcome of a standard binomial estimate. Let $G = (g_i)_{i=1}^d$ be a standard Gaussian vector and observe that

$$\ell^*(\mathcal{B}) = \mathbb{E}\sup_{t\in\mathcal{B}}\langle G, t\rangle \leq \mathbb{E}\sup_{t\in\mathcal{B}}\sum_{j=1}^d g_j^* t_j^* \leq \mathbb{E}\sup_{t\in\mathcal{B}}\sum_{j=1}^d \frac{g_j^*}{\lambda_j}\lambda_j t_j^* \leq \mathbb{E}\max_{1\leq j\leq d}\frac{g_j^*}{\lambda_j}.$$

For $u \geq 1$,

$$Pr\Big(\max_{1\leq j\leq d}\frac{g_j^*}{\lambda_j} \geq u\Big) \leq \sum_{j=1}^d Pr\left(g_j^* \geq u\lambda_j\right) \leq \sum_{j=1}^d \binom{d}{j} Pr^j\left(|g| \geq u\lambda_j\right)$$
$$\leq 2\sum_{j=1}^d \exp\Big(j\log\Big(\frac{ed}{j}\Big) - c_1 j u^2\lambda_j^2\Big) \leq 2\exp(-c_2u^2),$$

where the last inequality follows if one sets $u^2 \geq \max_j \lambda_j^{-2}\log(ed/j)$. The proof is concluded by integrating the tails. ∎

Theorem 5.3 leads to estimation properties of SLOPE.

Theorem 5.12 *Using the notation of Theorem 5.3, if $\Psi(t) = \|t\|_{SLOPE}$, $\max_j \lambda_j^{-1}\sqrt{\log(ed/j)} \leq C$ and*

$$\hat{t} \in \operatorname*{argmin}_{t\in\mathbb{R}^d}\Big(\frac{1}{N}\sum_{i=1}^N (Y_i - \langle X_i, t\rangle)^2 + c_2\sigma_q Lwu\frac{\|t\|_{SLOPE}}{\sqrt{N}}\Big),$$

then with probability larger than the one in (5.6),

$$\left\|\hat{t} - t^*\right\|_2^2 \lesssim_{L,q,C} \begin{cases} \frac{\sigma_q wu\|t^*\|_{SLOPE}}{\sqrt{N}} & \text{if } N \gtrsim_L d, \\ \max\Big\{\frac{\sigma_q Lwu\|t^*\|_{SLOPE}}{\sqrt{N}}, \frac{\|t^*\|_{SLOPE}^2}{N}\Big\} & \text{otherwise.}\end{cases}$$

As in the case of the LASSO, one may combine the sparsity-dependent error rate for SLOPE from Lecué and Mendelson (to appear) and the complexity-dependent error rate from Theorem 5.12. To that end, assume that X is isotropic, L-subgaussian and that the noise ξ is in L_q for some $q > 2$. Then, with probability larger than the one in (5.6)

$$\left\|\hat{t} - t^*\right\|_2^2 \lesssim_{L,q,C} \min\left\{\frac{\sigma_q\|t^*\|_0}{N}\log\Big(\frac{ed}{\|t^*\|_0}\Big), \max\left\{\frac{\sigma_q wu\|t^*\|_{SLOPE}}{\sqrt{N}}, \frac{\|t^*\|_{SLOPE}^2}{N}\right\}\right\},$$

for $N \gtrsim \|t^*\|_0\log(ed/\|t^*\|_0)$.

5.3 Regularization methods in $\mathbb{R}^{m\times T}$

In this section, we assume that X takes values in the set of $m\times T$ matrices, endowed with the inner product $\langle A,B\rangle=\sum_{u,v}A_{uv}B_{uv}$. We consider $A^*\in\operatorname{argmin}_{A\in\mathbb{R}^{m\times T}}\mathbb{E}\big(Y-\langle X,A\rangle\big)^2$ and thus $\langle X,A^*\rangle$ is the best (linear) approximation of Y in the L_2 sense.

Let $\Lambda(\rho)\geq\rho\ell^*(K)/\sqrt{N}$ for all $\rho>0$, $u,w>C$ and set

$$\hat{A}\in\operatorname*{argmin}_{A\in\mathbb{R}^{m\times T}}\Big(\frac{1}{N}\sum_{i=1}^N(Y_i-\langle X_i,A\rangle)^2+c_2\eta^3\sigma_q Lwu\Lambda(\Psi(A))\Big). \tag{5.17}$$

By Theorem 5.3, with probability larger than the one in (5.6)

$$\Big\|\hat{A}-A\Big\|_2^2=\Big\|\langle X,\hat{A}-A^*\rangle\Big\|_{L_2}^2\lesssim r(10\eta\Psi(A^*))^2$$

where for $\rho\geq 0$,

$$r(\rho)^2\sim_{L,q}\begin{cases}\sigma_q wu\Lambda(\rho) & \text{when } N\gtrsim_L mT\\ \max\Big\{\sigma_q wu\Lambda(\rho),\Lambda^2(\rho)\Big\} & \text{otherwise.}\end{cases}$$

Let us turn to estimates on $\ell^*(K)$ for the unit balls of the regularization functions used in the *matrix completion* and *collaborative filtering* problems.

Example: S_p-regularization for $p\geq 1$.

For any $A\in\mathbb{R}^{m\times T}$, let $\sigma_1(A)\geq\sigma_2(A)\geq\cdots\geq\sigma_{m\wedge T}(A)$ be the ordered singular values of A and set $m\wedge T=\min\{m,T\}$. Recall that the p-Schatten norm $\|\cdot\|_{S_p}$ of A is defined by

$$\|A\|_{S_p}=\Big(\sum_{j=1}^{m\wedge T}\sigma_j(A)^p\Big)^{1/p}.$$

Schatten norms have been used extensively in *matrix completion* and in *collaborative filtering*. Exact reconstruction properties of various procedures have been established via the minimization of the S_1-norm, constrained to matching the data (see, e.g., Candès and Plan (2011); Candès and Tao (2010); Candès and Recht (2009); Gross (2011); Chandrasekaran et al. (2012)). S_1 regularization has also been used in the noisy setup for independent subgaussian noise and, in most cases, for subgaussian or deterministic designs, in Koltchinskii et al. (2011); Rohde and Tsybakov (2011); Koltchinskii (2011); Negahban and Wainwright (2012); Gaïffas and Lecué (2011); Klopp (2014).

A result that is closely related to ours is Theorem 9.2 from Koltchinskii (2011), in which X is isotropic and L-subgaussian; ξ is a symmetric random variable that is independent of X and for which $\|\xi\|_{\psi_\alpha}<\infty$ for some $\alpha\geq 1$ (cf. Rao and Ren (1991) for more details on the ψ_α-norms); and the target is $Y=\langle X,A^*\rangle+\xi$.

Let $N\gtrsim m\cdot\operatorname{rank}(A^*)$ and set

$$\lambda\gtrsim\max\left\{\|\xi\|_2\sqrt{\frac{m(t+\log m)}{N}},\|\xi\|_{\psi_\alpha}\log^{1/\alpha}\Big(\frac{\|\xi\|_{\psi_\alpha}}{\|\xi\|_{L_2}}\Big)\frac{\sqrt{m}(t+\log N)(t+\log m)}{N}\right\}.$$

The S_1-regularization procedure with regularization parameter λ satisfies that for every $t>0$, with probability larger than $1-3\exp(-t)-\exp(-c_0N)$,

$$\Big\|\hat{A}-A^*\Big\|_{S_2}^2\lesssim\min\left\{\lambda\|A^*\|_{S_1},\lambda^2\operatorname{rank}(A^*)\right\}. \tag{5.18}$$

In comparison, an estimation result for S_p-norm regularization (for any $p\geq 1$) follows from Theorem 5.3, and does not require any assumptions on the "noise" $\xi=Y-\langle A^*,X\rangle$, other than $\xi\in L_q$ for some $q>2$. in particular, ξ need not belong to ψ_α, nor does it have to be independent of X. The result uses the following estimate on the Gaussian mean-width of the unit ball of S_p-norms (see, for instance, Proposition 1.4.4 in Chafaï et al. (2012)):

Proposition 5.13 *Let $p\geq 1$ and set B_p^{mT} to be the unit ball of $\|\cdot\|_{S_p}$. Then*

$$\ell^*(B_p^{mT})\sim\min\{m,T\}^{1-1/p}\sqrt{m+T}.$$

Combining the previous result with Theorem 5.3, one obtains the following:

Theorem 5.14 *Assume that the assumptions of Theorem 5.3 hold. Let $\Lambda_p(\rho)=\rho\min\{m,T\}^{1-1/p}\sqrt{\frac{m+T}{N}}$ and*

$$\hat{A}\in\operatorname*{argmin}_{A\in\mathbb{R}^{m\times T}}\Big(\frac{1}{N}\sum_{i=1}^N(Y_i-\langle X_i,A\rangle)^2+c_2\sigma_q Lwu\Lambda_p(\|A\|_{S_p})\Big).$$

Then with probability at least $1-\delta$,

$$\Big\|\hat{A}-A^*\Big\|_{S_2}^2\lesssim_{p,L,q}\begin{cases}\sigma_q wu\Lambda_p(\|A^*\|_{S_p}) & \text{if } N\gtrsim_L mT,\\ \max\Big\{\sigma_q wu\Lambda_p(\|A^*\|_{S_p}),\Lambda_p^2(\|A^*\|_{S_p})\Big\} & \text{otherwise.}\end{cases}$$

Remark. As in the vector case mentioned earlier, Theorem 5.3 also applies for S_p-regularization for $0<p<1$. In that case, Assumption 1.1 is satisfied for $\eta=2^{1/p}$ and the Gaussian mean width of the S_p-unit ball satisfies $\ell^*(B_p^{mT})\lesssim\sqrt{m+T}$. It therefore follows from Theorem 5.3 that under the same assumptions as in Theorem 5.3 and for $\Lambda(\rho)=\rho\sqrt{(m+T)/N}$ for all $\rho>0$, the RERM

$$\hat{A}\in\operatorname*{argmin}_{A\in\mathbb{R}^{m\times T}}\Big(\frac{1}{N}\sum_{i=1}^N(Y_i-\langle X_i,A\rangle)^2+c_2\sigma_q Lwu\Lambda(\|A\|_{S_p})\Big),$$

satisfies, with probability larger than the one in (5.6),

$$\Big\|\hat{A}-A^*\Big\|_{S_2}^2\lesssim_{p,L,q}\begin{cases}\sigma_q wu\Lambda(\|A^*\|_{S_p}) & \text{if } N\gtrsim_L mT,\\ \max\Big\{\sigma_q wu\Lambda(\|A^*\|_{S_p}),\Lambda^2(\|A^*\|_{S_p})\Big\} & \text{otherwise.}\end{cases}$$

Observe that just as in the vector case, when $0<p<1$ this rate is not the minimax rate in the true model $\|A^*\|_{S_p}B_p^{mT}$. Indeed, Rohde and Tsybakov (2011) provides the minimax rate, and, in fact, also shows that the minimax rate may be attained using $\Psi(A)=\|A\|_{S_p}^p$

as a regularization function. To be more accurate, Rohde and Tsybakov (2011) considers the following problem: let $x_1, \dots, x_N$ be N deterministic matrices in $\mathbb{R}^{m\times T}$ satisfying some RIP property and set W_i to be N independent, standard Gaussian variables with variance σ^2. Set $Y_i = \langle x_i, A^*\rangle + W_i, i = 1, \dots, N$, leading to the so-called matrix regression model with Gaussian noise and a deterministic design. It is shown in Rohde and Tsybakov (2011) that when ρB_p^{mT} for some $0 < p \le 1$, the minimax rate of the problem in ρB_p^{mT} is

$$\sigma^2\rho^p\Big(\frac{m+T}{N}\Big)^{1-\frac{p}{2}}$$

in some specific range of N, σ and ρ. Our result recovers this rate only for $p = 1$.

Example: Max-norm regularization. The max-norm of a matrix is defined by

$$\|A\|_{max} = \inf_{A=UV^\top} \|U\|_{2\to\infty}\|V\|_{2\to\infty},$$

with the infimum taken with respect to all pairs of matrices U, V for which $A = UV^\top$.

Constrained empirical risk minimization procedures that are based on the max-norm have been used in Cai and Zhou (2016); Nathan and Adi (2005) for bounded and Lipschitz loss functions, and in Lecué and Mendelson for the squared loss and for a subgaussian and isotropic design vector X and a subgaussian noise ξ independent of X. Consider the matrix regression model $Y_i = \langle X_i, A^*\rangle + W_i, i = 1, \dots, N$ where $X_1, \dots, X_N$ are independent isotropic and subgaussian matrices, $W_1, \dots, W_N$ are independent centered gaussian variables with variance σ^2 that are independent of the X_i's and A^* belongs to the max-norm ball of radius ρ. One may show that the minimax rate in this case is

$$\max\left\{\sigma\rho\sqrt{\frac{(mT)(m+T)}{N}}, \frac{\rho^2(mt)(m+T)}{N}\right\}, \tag{5.19}$$

at least in some specific regime of ρ, σ and N (cf. Lecué and Mendelson).

To apply Theorem 5.3, let us estimate the Gaussian mean-width of the unit ball of the max-norm ball, that is, of $\mathcal{B} = \{A \in \mathbb{R}^{m\times T} : \|A\|_{max} \le 1\}$.

Lemma 5.15 *There exists an absolute constant c for which, for every m and T,*

$$\ell^*(\mathcal{B}) \lesssim \sqrt{(mT)(m+T)}.$$

Proof. An application of Grothendieck's inequality (see, e.g., Nathan and Adi (2005)) shows that

$$\mathrm{conv}(\mathcal{X}_\pm) \subset \mathcal{B} \subset K_G\mathrm{conv}(\mathcal{X}_\pm)$$

where K_G is the Grothendieck constant and $\mathcal{X}_\pm = \{uv^\top : u \in \{\pm1\}^m, v \in \{\pm1\}^T\}$. If $\mathfrak{G} = (g_{ij})_{1\le u\le m:1\le v\le T}$ is a standard $m \times T$ Gaussian matrix, it follows from the Gaussian maximal inequality (see, e.g., Chapter 3 in Ledoux and Talagrand (1991)) that

$$\begin{aligned}\ell^*(\mathcal{B}) &= \mathbb{E}\sup_{A\in\mathcal{B}}|\langle\mathfrak{G}, A\rangle| \le K_G\mathbb{E}\sup_{A\in\mathrm{conv}(\mathcal{X}_\pm)}|\langle\mathfrak{G}, A\rangle| \\ &= K_G\mathbb{E}\sup_{A\in\mathcal{X}_\pm}|\langle\mathfrak{G}, A\rangle| \lesssim \max_{A\in\mathcal{X}_\pm}\|A\|_{HS}\sqrt{\log|\mathcal{X}_\pm|} \lesssim \sqrt{(mT)(m+T)}.\end{aligned}$$

■

Theorem 5.16 *Using the assumptions and notation of Theorem 5.3, and setting $\Lambda(\rho) = \rho\sqrt{(mT)(m+T)/N}$, if*

$$\hat{A} \in \operatorname*{argmin}_{A\in\mathbb{R}^{m\times T}}\Big(\frac{1}{N}\sum_{i=1}^N(Y_i - \langle X_i, A\rangle)^2 + c_2\sigma_q L w u\Lambda(\|A\|_{max})\Big),$$

then with probability larger than the one in (5.6)

$$\|\hat{A} - A^*\|_{S_2}^2 \lesssim_{L,q} \begin{cases} \sigma_q w u\Lambda(\|A^*\|_{max}) & \text{if } N \gtrsim_L mT, \\ \max\Big\{\sigma_q w u\Lambda(\|A^*\|_{max}), \Lambda^2(\|A^*\|_{max})\Big\} & \text{otherwise.}\end{cases}$$

As a consequence, we recover the minimax rate of convergence in the matrix regression model with subgaussian design and gaussian noise in the class $\|A^*\|_{max}\mathcal{B}$ thanks to max-norm regularization and without knowing $\|A^*\|_{max}$ in advance.

Example: Atomic-norm regularization.

The atomic-norm has been used in Chandrasekaran et al. (2012) in the context of exact and robust recovery using few Gaussian linear measurements of a signal or of a matrix.

Given $\mathcal{A} \subset \mathbb{R}^{m\times T}$, the elements in $\mathcal{A}$ are called *atoms*. Set $\mathrm{conv}(\mathcal{A})$ to be the convex hull of $\mathcal{A}$ and put

$$\|A\|_{\mathcal{A}} = \inf\{t > 0 : A \in t\cdot\mathrm{conv}(\mathcal{A})\}. \tag{5.20}$$

Even though $\|\cdot\|_{\mathcal{A}}$ need not be a norm (because $\mathrm{conv}(\mathcal{A})$ need not be centrally-symmetric), it is positive homogeneous and satisfies a triangle inequality: for every $A, B \in \mathbb{R}^{m\times T}$ and $\lambda \ge 0$:

$$\|A + B\|_{\mathcal{A}} \le \|A\|_{\mathcal{A}} + \|B\|_{\mathcal{A}} \text{ and } \|\lambda A\|_{\mathcal{A}} = \lambda\|A\|_{\mathcal{A}}.$$

And, if we assume that $\mathcal{A}$ is centrally-symmetric, then $\|\ \|_{\mathcal{A}}$ is a norm, (5.9) is satisfied and Theorem 5.3 applies.

Set $\mathcal{B}$ to be the unit ball with respect to $\|\ \|_{\mathcal{A}}$ and note that $\ell^*(\mathcal{B}) = \ell^*(\mathcal{A})$. For example, assume that $m = T$ and put $\mathcal{A}$ to be the set of all orthogonal matrices. Since the unit ball of the spectral norm is the convex hull of the set of orthogonal matrices, one has $\|\cdot\|_{\mathcal{A}} = \|\cdot\|_{S_2}$ and

$$\ell^*(\mathcal{B}) = \mathbb{E}\|\mathfrak{G}\|_{S_2} \le \sqrt{m}\mathbb{E}\|\mathfrak{G}\|_{S_\infty} \lesssim m.$$

Theorem 5.17 *Using the assumptions and notation of Theorem 5.3, let $\mathcal{A} \subset \mathbb{R}^{m\times T}$ be a centrally-symmetric set of atoms and set $\Lambda(\rho) \ge \rho\ell^*(\mathcal{A})/\sqrt{N}$. If*

$$\hat{A} \in \operatorname*{argmin}_{A\in\mathbb{R}^{m\times T}}\Big(\frac{1}{N}\sum_{i=1}^N(Y_i - \langle X_i, A\rangle)^2 + c_2\sigma_q L w u\Lambda(\|A\|_{\mathcal{A}})\Big)$$

then with probability larger than the one in (5.6)

$$\Big\|\hat{A} - A^*\Big\|_{S_2}^2 \lesssim_{L,q} \begin{cases} \sigma_q w u\Lambda(\|A^*\|_{\mathcal{A}}) & \text{if } N \gtrsim_L mT, \\ \max\Big\{\sigma_q w u\Lambda(\|A^*\|_{\mathcal{A}}), \Lambda^2(\|A^*\|_{\mathcal{A}})\Big\} & \text{otherwise.}\end{cases}$$

6. Conclusions

We have presented a general result (Theorem 1.9) describing statistical properties of a constrained regularized procedure in the learning theoretical framework. This result highlights the role played by the quadratic and multiplier processes in calibrating the regularization parameter λ as well as their effect on the estimation error rate. It appears that:

1. The rates of convergence depend on $\Psi(f^*)$ and we recover the minimax rate in the true model $\{f \in F : \Psi(f) \leq \Psi(f^*)\}$ – up to a logarithmic factor – in many well-studied examples .

2. No statistical model is needed to study RERM; all the analysis has been carried out in the general learning theory setup, and thus without assuming any statistical model. Theorem 1.9 and all its corollaries and applications are true regardless of any relation between the target Y and the input X. For instance, when predicting Y using linear functionals there is no need to assume that $Y = \langle t_0, X\rangle + W$ for an independent noise W; our results hold even if Y were, for instance, a noisy version of a quadratic function of a linear functional of X (e.g. phase retrieval).

   Our analysis shows that despite considering this general framework, the error rate and the regularization parameter used to construct RERM almost match the ones that would have been obtained with more information – namely, a given statistical model. In the examples we considered, statistical models are superfluous to the analysis of RERM and as a consequence, they may actually hide what really determines the error rate and the right choice of a regularization parameter:

   - calibration of the regularization parameter depends only on the multiplier process – which measures the empirical correlation between the noise $Y - f^*(X)$ and the class F. When this correlation is small or even null (in the free-noise case) the regularization parameter will also be small.
   - the key parameters are the structure of the "unit ball" of the regularization function (measured here using the Gaussian mean width) and the noise level, which is measured through the L_q norm of $Y - f^*(X)$.

7. Proof of Theorem 3.6

Following Mendelson (2016), the proof of Theorem 3.6 is based on properties of the following norm:

Definition 7.1 *For a random variable Z and $p \geq 1$, set*

$$\|Z\|_{(p)} = \sup_{1\leq q\leq p} \frac{\|Z\|_{L_q}}{\sqrt{q}}.$$

The $\|\cdot\|_{(p)}$ norm is a 'local' version of the ψ_2 norm. While

$$\|Z\|_{\psi_2} \sim \sup_{q\geq 1} \frac{\|Z\|_{L_q}}{\sqrt{q}},$$

$\|Z\|_{(p)}$ captures the subgaussian behavior of Z up to the p-th moment.

Under Assumption 3.1, a high probability bound on (3.7) can be derived using the next result.

Proposition 7.2 (Lemma 2.8 in Lecué and Mendelson (2017)) *There exists an absolute constant c_0 for which the following holds. Let Z be a mean-zero real-valued random variable and let $Z_1, ..., Z_N$ be independent copies of Z. Let $p_1 \geq 1$ and assume that $\|Z\|_{(p_1)} \leq L$, then*

$$\left\| \frac{1}{\sqrt{N}} \sum_{i=1}^N Z_i \right\|_{(p_1)} \leq c_0 L.$$

Setting $U_j = N^{-1/2} \sum_{i=1}^N \varepsilon_i X_i(j)$ and $p_1 = \log d$ (recalling that $t \geq 1$ and $d \geq N$ in Assumption 3.1), it follows from Proposition 7.2 that

$$\|U_j\|_{L_{p_1}} \leq c_0 L \sqrt{p_1} \|x_j\|_{L_2}.$$

Therefore,

$$Pr\left(\max_{1\leq j\leq d} |U_j| \geq u \right) \leq \sum_{j=1}^d Pr\left(|U_j| \geq u\right) \leq \sum_{j=1}^d \left(\frac{\|U_j\|_{L_{p_1}}}{u} \right)^{p_1}$$
$$\leq d \left(\frac{c_0 L \sqrt{p_1} \max_{1\leq j\leq d} \|x_j\|_{L_2}}{u} \right)^{p_1} = d \left(\frac{c_0 L \sqrt{\log d} \max_{1\leq j\leq d} \|x_j\|_{L_2}}{u} \right)^{\log d}.$$

Let $w \geq e$ and set $u = c_0 L w \sqrt{\log d} \max_{1\leq j\leq d} \|x_j\|_{L_2}$; therefore,

$$Pr\left(\max_{1\leq j\leq d} |U_j| \geq c_1 L w \sqrt{\log d} \max_{1\leq j\leq d} \|x_j\|_{L_2} \right) \leq \left(\frac{e}{w} \right)^{\log d}, \tag{7.1}$$

which is a high probability estimate on (3.7) under a limited moment assumption. Integrating the tail,

$$\mathbb{E} \max_{1\leq j\leq d} \left| \frac{1}{\sqrt{N}} \sum_{i=1}^N \varepsilon_i X_i(j) \right| \lesssim L \sqrt{\log d} \max_{1\leq j\leq d} \|x_j\|_{L_2}$$

proving (3.9).

Next, we obtain high probability bounds on (3.8),—estimates that require some preparation.

Let $j \in \{1, \dots, d\}$ and set $Z_i = X_i(j)$. Consider the Bernoulli sums

$$Q_j = \sum_{i=1}^N \varepsilon_i \xi_i X_i(j) = \sum_{i=1}^N \varepsilon_i \xi_i Z_i.$$

Denote by $(a_i^*)_{i=1}^N$ the non-increasing rearrangement of $(|a_i|)_{i=1}^N$. A straightforward application of Höffding's inequality shows that conditioned on $(\xi_i)_{i=1}^N$ and $(Z_i)_{i=1}^N$, for any $v > 0$,

with probability at least $1-2\exp(-v^2/2)$ relative to $(\varepsilon_i)_{i=1}^N$,

$$|Q_j| \leq \sum_{i\leq m}\xi_i^* Z_i^* + v\left(\sum_{i\geq m}(\xi_i^* Z_i^*)^2\right)^{1/2} \tag{7.2}$$
$$\leq \left(\sum_{i\leq m}(\xi_i^*)^2\right)^{1/2}\left(\sum_{i\leq m}(Z_i^*)^2\right)^{1/2} + v\left(\sum_{i\geq m}(\xi_i^*)^{2r}\right)^{1/2r}\left(\sum_{i\geq m}(Z_i^*)^{2r'}\right)^{1/2r'},$$

where r and r' are conjugate indices.

As a consequence, high probability bounds on the rearrangements (ξ_i^*) and (Z_i^*) can be used to obtain high probability bounds on $|Q_j|$ (and therefore, on $\max_{1\leq j\leq d}|Q_j|$ as well, using the union bound).

The next two observations, whose proofs may be found in Mendelson (2016), give information on the structure of a typical $(Z_i)_{i=1}^N$ when Z has at least $\sim \log d$ subgaussian moments. It turns out that one may decompose $(Z_i)_{i=1}^N$ to a sum of two vectors, supported on disjoint sets: one consists of the largest m coordinates of $(|Z_i|)_{i=1}^N$, and its ℓ_2^N norm is determined by relatively high moments of Z; the other consists of the $N-m$ smaller coordinates of $(|Z_i|)_{i=1}^N$, and if $Z\in L_{q_1}$, its ℓ_r^N norm is well-behaved for $r<q_1$.

The 'level' m depends on the desired probability estimate and on the moments of Z: if one wishes to obtain a probability estimate of $1-2\exp(-p)$ for $p\geq \log N$ (as we will), then Z should have roughly p moments and one should select $m\sim p/\log(eN/p)$.

First, let us consider the smaller coordinates:

Lemma 7.3 (Lemma 3.2 in Mendelson (2016)) *There exist absolute constants a_0 and c_1 for which the following holds. Let $1\leq r_1<q_1$, set $Z\in L_{q_1}$ and put $Z_1,\ldots,Z_N$ to be independent copies of Z. Fix $1\leq p\leq N$, let $u>2$ and set*

$$m=\left\lceil \frac{a_0 p}{((q_1/r_1)-1)\log(4+eN/p)}\right\rceil.$$

If $m>1$, then, with probability at least $1-2u^{-mq_1}\exp(-p)$,

$$\left(\sum_{i=m}^N (Z_i^*)^{r_1}\right)^{1/r_1} \leq c_1\left(\frac{q_1}{q_1-r_1}\right)^{1/r_1} uN^{1/r_1}\|Z\|_{L_{q_1}}$$

and, if $m=1$ and $0<\beta<q_1/r_1-1$ then with probability at least $1-c_2u^{-q_1}N^{-\beta}$,

$$\left(\sum_{i=1}^N |Z_i|^{r_1}\right)^{1/r_1} \leq c_1\Big(\frac{q_1}{q_1-(\beta+1)r_1}\Big)u\,\|Z\|_{L_{q_1}}N^{1/r_1}.$$

Next, we consider the larger coordinates:

Lemma 7.4 (Lemma 3.4 in Mendelson (2016)) *There exists absolute constants a_1 and c_0 for which the following holds. Let $Z_1,\ldots,Z_N$ be independent copies of a random variable Z, set $p\geq \log N$ and put $1\leq m\leq N/2e$ that satisfy $m\leq a_1 p/\log\big(eN/p\big)$. Then, for every $u>1$, with probability at least $1-u^{-2p}\exp(-p)$, one has*

$$\left(\sum_{i=1}^m (Z_i^*)^2\right)^{1/2} \leq c_0 u\sqrt{p}\|Z\|_{(2p)}.$$

In particular, under Assumption 3.1, we apply Lemma 7.3 and Lemma 7.4 to p, q_1 and r_1 defined by

$$2p=t\log d, r_1=2r' \text{ and } q_1=r_1\max\Big\{2,1+\frac{a_0}{a_1}\Big\},$$

where a_0 and a_1 are the absolute constants from Lemma 7.3 and 7.4. We also set

$$m=\left\lceil \frac{a_0 p}{\log(4+eN/p)}\right\rceil \tag{7.3}$$

and observe that if

$$m_0=\left\lceil \frac{a_0 p}{((q_1/r_1)-1)\log(4+eN/p)}\right\rceil \text{ then } m_0\leq m\leq \frac{a_1 p}{\log(eN/p)}$$

and $m_0 q_1\sim m$. Moreover, if κ_0 is a large enough absolute constant and $t\geq\kappa_0$, then $m_0>1$. Recalling that $p\geq 2\log d$ and setting $Z_i=X_i(j)$ for $i=1,\ldots,N$, it follows that for any $u>2$, with probability at least $1-u^{-2p}\exp(-p/2)-2u^{-c_0 m}\exp(-p/2)$, for every $1\leq j\leq d$

$$\left(\sum_{i=1}^m (Z_i^*)^2\right)^{1/2} \lesssim u\sqrt{p}\,\|Z\|_{(2p)} \lesssim uL\sqrt{t\log d}\,\|x_j\|_{L_2} \tag{7.4}$$

and

$$\left(\sum_{i=m}^N (Z_i^*)^{2r'}\right)^{1/2r'} \lesssim u\,\|Z\|_{L_{q_1}}N^{1/2r'} \lesssim uL\sqrt{r'}\,\|x_j\|_{L_2}N^{1/2r'}. \tag{7.5}$$

Let $\xi_1,\ldots,\xi_N$ be independent copies of ξ and recall that $(\xi_i^*)_{i=1}^N$ is the monotone non-increasing rearrangement of $(|\xi_i|)_{i=1}^N$. We apply Lemma 7.3 for $q_1=q$, $r_1=2r$ and set

$$m_1=\left\lceil \frac{a_0 p}{((q_1/r_1)-1)\log(4+eN/p)}\right\rceil.$$

Thus, $m_1>1$ when $t\geq\kappa_0$ for a large enough constant κ_0, and if m is as in (7.3) one has $m\geq m_1$ and $m_1 q_1\sim m$. Hence, for $p=(t/2)\log d$, with probability larger than $1-2u^{-c_0 m}\exp(-p)$,

$$\left(\sum_{i=m}^N (\xi_i^*)^{2r}\right)^{1/2r} \leq c(q)u\,\|\xi\|_{L_q}N^{1/2r}. \tag{7.6}$$

This provides a high probability bound on the smaller coordinates of $(|\xi_i|)_{i=1}^N$. Next, let us turn to estimating the larger ones.

Lemma 7.5 (Lemma 4.3 in Mendelson (2016)) *Let $q > 2$ and assume that $\xi \in L_q$. If $\xi_1, \ldots, \xi_N$ are independent copies of ξ then for every $w > 1$ with probability larger than $1 - c_0 w^{-q} N^{-((q/2)-1)} \log^q N$,*

$$\left(\sum_{i=1}^m \xi_i^2\right)^{1/2} \leq \left(\sum_{i=1}^N \xi_i^2\right)^{1/2} \leq c_1 w \|\xi\|_{L_q} \sqrt{N}.$$

Setting $Z_i = X_i(j)$ for $i = 1, \ldots, N$ and applying (7.4), (7.5), (7.6) and Lemma 7.5, we obtain, with probability larger than

$$1 - \frac{\exp(-p/2)}{u^{2p}} - \frac{4\exp(-p/2)}{u^{c_0 m}} - \frac{c_0 \log^q N}{w^q N^{q/2-1}}$$

that for every $j = 1, \ldots, d$

$$\left(\sum_{i\leq m} (\xi_i^*)^2\right)^{1/2} \left(\sum_{i\leq m} (Z_i^*)^2\right)^{1/2} \lesssim uwL\sqrt{t\log d}\|x_j\|_{L_2}\|\xi\|_{L_q}\sqrt{N}$$

and

$$\left(\sum_{i\geq m} (\xi_i^*)^{2r}\right)^{1/2r} \left(\sum_{i\geq m} (Z_i^*)^{2r'}\right)^{1/2r'} \leq c(q)u^2 L\|\xi\|_{L_q}\sqrt{N}\|x_j\|_{L_2}.$$

Then, by plugging those inequalities in (7.2), it is evident that under Assumption 3.1, for $u > 2, v > 0$, $w \geq 2$, $2p = t\log d$ and $m \sim p/\log(eN/p)$, with probability at least

$$1 - \frac{\exp(-p/2)}{u^{2p}} - \frac{4\exp(-p/2)}{u^{c_0 m}} - \frac{c_1 \log^q N}{w^q N^{q/2-1}} - 2\exp(-v^2 t\log d),$$

$$\max_{1\leq j\leq d}\left|\sum_{i=1}^N \varepsilon_i\xi_i X_i(j)\right| = \max_{1\leq j\leq d}|Q_j| \lesssim_q (uw + u^2 v)L\|\xi\|_{L_q}\sqrt{N}\sqrt{t\log d}\max_{1\leq j\leq d}\|x_j\|_{L_2}.$$

Acknowledgments

We would like to acknowledge support for this project from Investissements d'Avenir (ANR-11-IDEX-0003/Labex Ecodec/ANR-11-LABX-0047) and the Israel Science Foundation, grant 707/14

References

Felix Abramovich, Yoav Benjamini, David L. Donoho, and Iain M. Johnstone. Adapting to unknown sparsity by controlling the false discovery rate. *Ann. Statist.*, 34(2):584–653, 2006. ISSN 0090-5364. doi: 10.1214/009053606000000074. URL http://dx.doi.org/10.1214/009053606000000074.

Peter L. Bartlett, Shahar Mendelson, and Joseph Neeman. ℓ_1-regularized linear regression: persistence and oracle inequalities. *Probab. Theory Related Fields*, 154(1-2):193–224, 2012. ISSN 0178-8051. doi: 10.1007/s00440-011-0367-2. URL http://dx.doi.org/10.1007/s00440-011-0367-2.

Peter J. Bickel, Ya'acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and Dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 2009. ISSN 0090-5364. doi: 10.1214/08-AOS620. URL http://dx.doi.org/10.1214/08-AOS620.

Małgorzata Bogdan, Ewout van den Berg, Chiara Sabatti, Weijie Su, and Emmanuel J. Candès. SLOPE—adaptive variable selection via convex optimization. *Ann. Appl. Stat.*, 9(3):1103–1140, 2015. ISSN 1932-6157. doi: 10.1214/15-AOAS842. URL http://dx.doi.org/10.1214/15-AOAS842.

Peter Bühlmann and Sara van de Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Springer, Heidelberg, 2011. ISBN 978-3-642-20191-2. doi: 10.1007/978-3-642-20192-9. URL http://dx.doi.org/10.1007/978-3-642-20192-9. Methods, theory and applications.

T. Tony Cai and Wen-Xin Zhou. Matrix completion via max-norm constrained optimization. *Electron. J. Stat.*, 10(1):1493–1525, 2016. ISSN 1935-7524. doi: 10.1214/16-EJS1147. URL http://dx.doi.org/10.1214/16-EJS1147.

Emmanuel J. Candès and Yaniv Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Trans. Inform. Theory*, 57(4):2342–2359, 2011. ISSN 0018-9448. doi: 10.1109/TIT.2011.2111771. URL http://dx.doi.org/10.1109/TIT.2011.2111771.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009. ISSN 1615-3375. doi: 10.1007/s10208-009-9045-5. URL http://dx.doi.org/10.1007/s10208-009-9045-5.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2044061. URL http://dx.doi.org/10.1109/TIT.2010.2044061.

Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006. ISSN 0018-9448. doi: 10.1109/TIT.2005.862083. URL http://dx.doi.org/10.1109/TIT.2005.862083.

Djalil Chafaï, Olivier Guédon, Guillaume Lecué, and Alain Pajor. *Interactions between compressed sensing random matrices and high dimensional geometry*, volume 37 of *Panoramas et Synthèses [Panoramas and Syntheses]*. Société Mathématique de France, Paris, 2012. ISBN 978-2-85629-370-6.

Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Found. Comput. Math.*, 12(6):805–849, 2012.

ISSN 1615-3375. doi: 10.1007/s10208-012-9135-7. URL http://dx.doi.org/10.1007/s10208-012-9135-7.

Víctor H. de la Peña and Evarist Giné. *Decoupling*. Probability and its Applications (New York). Springer-Verlag, New York, 1999. ISBN 0-387-98616-2. doi: 10.1007/978-1-4612-0537-1. URL http://dx.doi.org/10.1007/978-1-4612-0537-1. From dependence to independence, Randomly stopped processes. U-statistics and processes. Martingales and beyond.

Sjoerd Dirksen, Guillaume Lecué, and Holger Rauhut. On the gap between rip-properties and sparse recovery conditions. Technical report, 2014. To appear in IEEE Transactions on Information Theory.

David L. Donoho and Iain M. Johnstone. Minimax risk over l_p-balls for l_q-error. *Probab. Theory Related Fields*, 99(2):277–303, 1994. ISSN 0178-8051. doi: 10.1007/BF01199026. URL http://dx.doi.org/10.1007/BF01199026.

D. E. Edmunds and H. Triebel. *Function spaces, entropy numbers, differential operators*, volume 120 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 1996. ISBN 0-521-56036-5. doi: 10.1017/CBO9780511662201. URL http://dx.doi.org/10.1017/CBO9780511662201.

Stéphane Gaïffas and Guillaume Lecué. Sharp oracle inequalities for high-dimensional matrix prediction. *IEEE Trans. Inform. Theory*, 57(10):6942–6957, 2011. ISSN 0018-9448. doi: 10.1109/TIT.2011.2136318. URL http://dx.doi.org/10.1109/TIT.2011.2136318.

Evarist Giné and Joel Zinn. Some limit theorems for empirical processes. *Ann. Probab.*, 12 (4):929–998, 1984. ISSN 0091-1798. With discussion.

Christophe Giraud. *Introduction to high-dimensional statistics*, volume 139 of *Monographs on Statistics and Applied Probability*. CRC Press, Boca Raton, FL, 2015. ISBN 978-1-4822-3794-8.

F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural network architectures. *Neural Computation*, 7:219–269, 1995.

Gene H. Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979. ISSN 0040-1706. doi: 10.2307/1268518. URL http://dx.doi.org/10.2307/1268518.

Yehoram Gordon, Alexandre E. Litvak, Shahar Mendelson, and Alain Pajor. Gaussian averages of interpolated bodies and applications to approximate reconstruction. *J. Approx. Theory*, 149(1):59–73, 2007. ISSN 0021-9045. doi: 10.1016/j.jat.2007.04.007. URL http://dx.doi.org/10.1016/j.jat.2007.04.007.

David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Trans. Inform. Theory*, 57(3):1548–1566, 2011. ISSN 0018-9448. doi: 10.1109/TIT.2011.2104999. URL http://dx.doi.org/10.1109/TIT.2011.2104999.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer, New York, second edition, 2009. ISBN 978-0-387-84857-0. doi: 10.1007/978-0-387-84858-7. URL http://dx.doi.org/10.1007/978-0-387-84858-7. Data mining, inference, and prediction.

Olga Klopp. Noisy low-rank matrix completion with general sampling distribution. *Bernoulli*, 20(1):282–303, 2014. ISSN 1350-7265. doi: 10.3150/12-BEJ486. URL http://dx.doi.org/10.3150/12-BEJ486.

Vladimir Koltchinskii. *Oracle inequalities in empirical risk minimization and sparse recovery problems*, volume 2033 of *Lecture Notes in Mathematics*. Springer, Heidelberg, 2011. ISBN 978-3-642-22146-0. doi: 10.1007/978-3-642-22147-7. URL http://dx.doi.org/10.1007/978-3-642-22147-7. Lectures from the 38th Probability Summer School held in Saint-Flour, 2008, École d'Été de Probabilités de Saint-Flour. [Saint-Flour Probability Summer School].

Vladimir Koltchinskii and Shahar Mendelson. Bounding the smallest singular value of a random matrix without concentration. *Int. Math. Res. Not. IMRN*, (23):12991–13008, 2015. ISSN 1073-7928. doi: 10.1093/imrn/rnv096. URL http://dx.doi.org/10.1093/imrn/rnv096.

Vladimir Koltchinskii, Karim Lounici, and Alexandre B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *Ann. Statist.*, 39(5): 2302–2329, 2011. ISSN 0090-5364. doi: 10.1214/11-AOS894. URL http://dx.doi.org/10.1214/11-AOS894.

Guillaume Lecué and Shahar Mendelson. Learning subgaussian classes: Upper and minimax bounds. In *Topics in Learning Theory*. Société Mathématique de France, (S. Boucheron and N. Vayatis Eds.). To appear.

Guillaume Lecué and Shahar Mendelson. Performance of empirical risk minimization in linear aggregation. *Bernoulli*, 22(3):1520–1534, 2016. ISSN 1350-7265. doi: 10.3150/15-BEJ701. URL http://dx.doi.org/10.3150/15-BEJ701.

Guillaume Lecué and Shahar Mendelson. Sparse recovery under weak moment assumptions. *J. Eur. Math. Soc. (JEMS)*, 19(3):881–904, 2017. ISSN 1435-9855. URL https://doi.org/10.4171/JEMS/682.

Guillaume Lecué and Shahar Mendelson. Regularization and the small-ball method I: sparse recovery. *Ann. Statist.*, to appear.

Michel Ledoux and Michel Talagrand. *Probability in Banach spaces*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*. Springer-Verlag, Berlin, 1991. ISBN 3-540-52013-9. Isoperimetry and processes.

Pascal Massart and Caroline Meynet. The Lasso as an ℓ_1-ball model selection procedure. *Electron. J. Stat.*, 5:669–687, 2011. ISSN 1935-7524. doi: 10.1214/11-EJS623. URL http://dx.doi.org/10.1214/11-EJS623.

Andreas Maurer and Massimiliano Pontil. Structured sparsity and generalization. *J. Mach. Learn. Res.*, 13:671–690, 2012. ISSN 1532-4435.

Andreas Maurer, Charles Micchelli, and Massimiliano Pontil. A family of penalty functions for structured sparsity. *NIPS*, 2010.

Shahar Mendelson. A remark on the diameter of random sections of convex bodies. In *Geometric aspects of functional analysis*, volume 2116 of *Lecture Notes in Math.*, pages 395–404. Springer, Cham, 2014. doi: 10.1007/978-3-319-09477-9_25. URL http://dx.doi.org/10.1007/978-3-319-09477-9_25.

Shahar Mendelson. Learning without concentration. *J. ACM*, 62(3):Art. 21, 25, 2015. ISSN 0004-5411. doi: 10.1145/2699439. URL http://dx.doi.org/10.1145/2699439.

Shahar Mendelson. Upper bounds on product and multiplier empirical processes. *Stochastic Process. Appl.*, 126(12):3652–3680, 2016.

Shahar Mendelson. Learning without concentration for a general loss function. *Probab. Theory Related Fields*, To appear.

Srebro Nathan and Shraibman Adi. Rank, trace-norm and max-norm. *18th Annual Conference on Learning Theory (COLT)*, 2005.

Sahand Negahban and Martin J. Wainwright. Restricted strong convexity and weighted matrix completion: optimal bounds with noise. *J. Mach. Learn. Res.*, 13:1665–1697, 2012. ISSN 1532-4435.

M. M. Rao and Z. D. Ren. *Theory of Orlicz spaces*, volume 146 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, 1991. ISBN 0-8247-8478-2.

Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q-balls. *IEEE Trans. Inform. Theory*, 57(10): 6976–6994, 2011. ISSN 0018-9448. doi: 10.1109/TIT.2011.2165799. URL http://dx.doi.org/10.1109/TIT.2011.2165799.

Philippe Rigollet and Alexandre Tsybakov. Exponential screening and optimal rates of sparse estimation. *Ann. Statist.*, 39(2):731–771, 2011. ISSN 0090-5364. doi: 10.1214/10-AOS854. URL http://dx.doi.org/10.1214/10-AOS854.

Angelika Rohde and Alexandre B. Tsybakov. Estimation of high-dimensional low-rank matrices. *Ann. Statist.*, 39(2):887–930, 2011. ISSN 0090-5364. doi: 10.1214/10-AOS860. URL http://dx.doi.org/10.1214/10-AOS860.

Mark Rudelson and Roman Vershynin. Small ball probabilities for linear images of high-dimensional distributions. *Int. Math. Res. Not. IMRN*, (19):9594–9617, 2015. ISSN 1073-7928. doi: 10.1093/imrn/rnu243. URL http://dx.doi.org/10.1093/imrn/rnu243.

Weijie Su and Emmanuel J. Candès. SLOPE is adaptive to unknown sparsity and asymptotically minimax. *Ann. Statist.*, 44(3):1038–1068, 2016. ISSN 0090-5364. doi: 10.1214/15-AOS1397. URL http://dx.doi.org/10.1214/15-AOS1397.

Michel Talagrand. The supremum of some canonical processes. *Amer. J. Math.*, 116(2): 283–325, 1994. ISSN 0002-9327. doi: 10.2307/2374931. URL http://dx.doi.org/10.2307/2374931.

Michel Talagrand. *The generic chaining*. Springer Monographs in Mathematics. Springer-Verlag, Berlin, 2005. ISBN 3-540-24518-9. Upper and lower bounds of stochastic processes.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996. ISSN 0035-9246. URL http://links.jstor.org/sici?sici=0035-9246(1996)58:1<267:RSASVT>2.0.CO;2-G&origin=MSN.

A. N. Tikhonov. On the stability of inverse problems. *C. R. (Doklady) Acad. Sci. URSS (N.S.)*, 39:176–179, 1943.

Sara A. van de Geer. The deterministic lasso. 2007. In JSM proceedings. American Statistical Association.

Aad W. van der Vaart and Jon A. Wellner. *Weak convergence and empirical processes*. Springer Series in Statistics. Springer-Verlag, New York, 1996. ISBN 0-387-94640-3. With applications to statistics.

Nicolas Verzelen. Minimax risks for sparse regressions: ultra-high dimensional phenomenons. *Electron. J. Stat.*, 6:38–90, 2012. ISSN 1935-7524. doi: 10.1214/12-EJS666. URL http://dx.doi.org/10.1214/12-EJS666.

Zhan Wang, Sandra Paterlini, Fuchang Gao, and Yuhong Yang. Adaptive minimax regression estimation over sparse ℓ_q-hulls. *J. Mach. Learn. Res.*, 15:1675–1711, 2014. ISSN 1532-4435.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 68(1):49–67, 2006. ISSN 1369-7412. doi: 10.1111/j.1467-9868.2005.00532.x. URL http://dx.doi.org/10.1111/j.1467-9868.2005.00532.x.

Journal of Machine Learning Research 18 (2017) 1-25 Submitted 2/15; Revised 9/17; Published 12/17

Matrix Completion with Noisy Entries and Outliers

Raymond K. W. Wong RAYWONG@STAT.TAMU.EDU
Department of Statistics
Texas A&M University
College Station, TX 77843 USA

Thomas C. M. Lee TCMLEE@UCDAVIS.EDU
Department of Statistics
University of California
Davis, CA 95616, USA

Editor: Xiaotong Shen

Abstract

This paper considers the problem of matrix completion when the observed entries are noisy and contain outliers. It begins with introducing a new optimization criterion for which the recovered matrix is defined as its solution. This criterion uses the celebrated Huber function from the robust statistics literature to downweigh the effects of outliers. A practical algorithm is developed to solve the optimization involved. This algorithm is fast, straightforward to implement, and monotonic convergent. Furthermore, the proposed methodology is theoretically shown to be stable in a well defined sense. Its promising empirical performance is demonstrated via a sequence of simulation experiments, including image inpainting.

Keywords: ES-Algorithm, Huber function, robust methods, Soft-Impute, stable recovery

1. Introduction

The goal of matrix completion is to impute those missing entries of a large matrix based on the knowledge of its relatively few observed entries. It has many practical applications, ranging from collaborative filtering (Rennie and Srebro, 2005) to computer visions (Weinberger and Saul, 2006) to positioning (Montanari and Oh, 2010). In addition, its application to recommender systems is perhaps the most well known example, widely made popularized by the so-called Netflix prize problem (Bennett and Lanning, 2007). In this problem a large matrix of movie ratings is partially observed. Each row of this matrix consists of ratings from a particular customer while each column records the ratings on a particular movie. In the Netflix data set, there are around 5×10^5 customers and 2×10^4 movies, with less than 1% of the ratings are observed. Without any prior knowledge, a reasonable full recovery of the matrix is virtually impossible. To overcome this issue, it is common to assume that the matrix is of low rank, reflecting the belief that the users' ratings are based on a relatively small number of factors. This low rank assumption is very sensible in many applications, although the resulting optimizations are combinatorially hard (Srebro and Jaakkola, 2003). To this end, various convex relaxations and related optimization algorithms have been proposed to provide computationally feasible solutions; see, for example, Candès and Recht (2009); Candès and Plan (2010); Keshavan et al. (2010b,a); Mazumder et al. (2010); Marjanovic and Solo (2012) and Hastie et al. (2014).

In addition to computational advances, the theoretical properties of matrix completion using nuclear norm minimization have also been well studied. For example, when the observed entries are noiseless, Candès and Recht (2009) show that perfect recovery of a low rank matrix is possible; see also Keshavan et al. (2010b), Gross (2011) and Recht (2011). This result of Candès and Recht (2009) has been extended to noisy measurements by Candès and Plan (2010): with high probability, the recovery is subject to an error bound proportional to the noise level. Techniques that achieve this desirable property are often referred as *stable*. See also Keshavan et al. (2010a) and Koltchinskii et al. (2011) for other theoretical developments of matrix completion from noisy measurements.

The original formulation of matrix completion assumes those observed entries are noiseless, and is later extended to the more realistic situation where the entries are observed with noise. This paper further extend the formulation to simultaneously allow for both noisy entries and outliers. To the authors knowledge, such an extension has not been considered before, although similar work exists. In Candès et al. (2011) a method called principal component pursuit (PCP) is developed to recover a matrix observed with mostly noiseless entries and otherwise a small amount of outliers. This is done by modeling the observed matrix as a sum of a low rank matrix and a sparse matrix. Zhou et al. (2010) extend this PCP method to noisy entries but assumes the matrix is fully observed, thus it does not fall into the class of matrix completion problems. Lastly Chen et al. (2011) extend PCP to safeguard against special outlying structures, namely outlying columns. However, it works only on outliers and otherwise noiseless entries. Due to the similarity between the matrix completion and principal component analysis, it is worthmentioning that there are some related work (Karhunen, 2011; Luttinen et al., 2012) on robust principal component analysis with missing values.

The primary contribution of this paper is the development of a new robust matrix completion method that can be applied to recover a matrix with missing, noisy and/or outlying entries. This method is shown to be stable in the sense of Candès and Plan (2010), as discussed above. As opposed to the above referenced PCP approach that decomposes the matrix into a sum of a low rank and a sparse matrix, the new approach is motivated by the statistical literature of robust estimation which modifies the least squares criterion to downweigh the effects of outliers. Particularly, we make use of the Huber function for this modification. We provide a theoretical result that establishes an intrinsic link between the two different approaches. To cope with the nonlinearity introduced by the Huber function, we propose a fast, simple, and easy-to-implement algorithm to perform the resulting nonlinear optimization problem. This algorithm is motivated by the ES-Algorithm for robust nonparametric smoothing (Oh et al., 2007). As to be shown below, it can transform a rich class of (non-robust) matrix completion algorithms into algorithms for robust matrix completion.

The rest of this paper is organized as follows. Section 2 provides further background of matrix completion and proposes a new optimization criterion for robust matrix recovery. Fast algorithms are developed in Section 3 for practically computing the robust matrix estimate. Theoretical and empirical properties of the proposed methodology are studied

©2017 Raymond K. W. Wong and Thomas C. M. Lee.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/15-076.html.

in Section 4 and Section 5 respectively. Concluding remarks are given in Section 6, while technical details are relegated to the appendix.

2. Matrix Completion with Noisy Observations and Outliers

Suppose X is an $n_1 \times n_2$ matrix which is observed for only a subset of entries $\Omega_{\text{obs}} \subseteq [n_1] \times [n_2]$, where $[n]$ denotes $\{1, \ldots, n\}$. Let $\Omega_{\text{obs}}^{\perp}$ be the complement of Ω_{obs}. Define the projection operator $\mathcal{P}_{\Omega_{\text{obs}}}$ as $\mathcal{P}_{\Omega_{\text{obs}}} B = C$, where $C_{ij} = B_{ij}$ if $(i,j) \in \Omega_{\text{obs}}$ and $C_{ij} = 0$ if $(i,j) \notin \Omega_{\text{obs}}$, for any $n_1 \times n_2$ matrix $B = (B_{ij})_{i \in [n_1], j \in [n_2]}$. The following is a standard formulation for matrix completion using a low rank assumption:

$$\begin{aligned} \underset{Y}{\text{minimize}} \quad & \text{rank}(Y) \\ \text{subject to} \quad & \frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}} X - \mathcal{P}_{\Omega_{\text{obs}}} Y\|_F^2 \leq e, \end{aligned}$$

where $e > 0$ and $\|\cdot\|_F$ is the Frobenius norm. Carrying out this rank minimization enables a good recovery of any low rank matrix with missing entries. Note that for the reason of accommodating noisy measurements, the constraint above allows for a slight discrepancy between the recovered and the observed matrices.

However, this minimization is combinatorially hard (e.g., Srebro and Jaakkola, 2003). To achieve fast computation, the following convex relaxation is often used:

$$\begin{aligned} \underset{Y}{\text{minimize}} \quad & \|Y\|_* \\ \text{subject to} \quad & \frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}} X - \mathcal{P}_{\Omega_{\text{obs}}} Y\|_F^2 \leq e, \end{aligned}$$

where $\|Y\|_*$ represents the nuclear norm of Y (i.e., the sum of singular values of Y). The Lagrangian form of this optimization is

$$\underset{Y}{\text{minimize}} \quad f(Y|X) \equiv \frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}} X - \mathcal{P}_{\Omega_{\text{obs}}} Y\|_F^2 + \gamma \|Y\|_*, \tag{1}$$

where $\gamma > 0$ has a one-to-one correspondence to e. The squared loss in the first term is used to measure the fitness of the recovered matrix to the observed matrix. It is widely known that such a squared loss is very sensitive to outliers and often leads to unsatisfactory recovery results if such outliers exist. Motivated by the literature of robust statistics (e.g., Huber and Ronchetti, 2011), we propose replacing this squared loss by the Huber loss function

$$\rho_c(x) = \begin{cases} x^2, & |x| \leq c \\ c(2|x| - c), & |x| > c \end{cases},$$

with tuning parameter c. When comparing with the squared loss, the Huber loss downweighs the effects of extreme measurements. Our proposed solution for robust matrix completion is given by the following minimization:

$$\underset{Y}{\text{minimize}} \quad g(Y) \equiv \frac{1}{2} \sum_{(i,j) \in \Omega_{\text{obs}}} \rho_c(X_{ij} - Y_{ij}) + \gamma \|Y\|_*. \tag{2}$$

Note that the convexity of ρ_c guarantees the convexity of the objective criterion (2).

For many robust statistical estimation problems the tuning parameter c is pre-set as $c = 1.345\hat{\sigma}$ to achieve a 95% statistical efficiency, where $\hat{\sigma}$ is an estimate of the standard deviation of the noise. For the current problem, however, the choice of c is suggested by Theorem 4 below: $c = \gamma/\sqrt{n_{(1)}p}$, where $n_{(1)} = \max\{n_1, n_2\}$ and p is the percentage of missing entries. This choice of c was used throughout all our numerical work.

3. Fast Algorithms for Minimization of (2)

Since the gradient of the Huber function is non-linear, (2) is a harder optimization problem when comparing to many typical matrix completion formulations such as (1). As an example, consider (1) when X is fully observed; i.e., $\Omega_{\text{obs}} = [n_1] \times [n_2]$. Through sub-gradient analysis (e.g., Cai et al., 2010; Ma et al., 2011), one can derive a closed-form solution to (1), denoted as $S_\gamma(X)$, where S_γ is the soft-thresholding operator defined in Mazumder et al. (2010), also given in (6) below. However, even if X was fully observed, (2) does not have a closed-form solution. The goal of this section is to develop fast methods for minimizing (2).

3.1 A General Algorithm

In Oh et al. (2007) a method based on the so-called theoretical construct *pseudo data* is proposed for robust wavelet regression. The idea is to transform a Huber-type minimization problem into a sequence of fast and well understood squared loss minimization problems. This subsection modifies this idea and proposes an algorithm to minimizing (2).

As similar to Oh et al. (2007), we define a *pseudo data matrix* as

$$Z = \mathcal{P}_{\Omega_{\text{obs}}} \tilde{Y} + \frac{1}{2}\psi_c(E), \tag{3}$$

where $\tilde{Y}$ is the current estimate of the target matrix, $E = \mathcal{P}_{\Omega_{\text{obs}}} X - \mathcal{P}_{\Omega_{\text{obs}}} \tilde{Y}$ is the "residual matrix", and $\psi_c = \rho_c'$ is the derivative of ρ_c. With a slight notation abuse, when ψ_c is applied to a matrix, it means ψ_c is evaluated in an element-wise fashion. Straightforward algebra shows that the sub-gradient of $f(Y|Z)$ (with respect to Y) evaluated at $\tilde{Y}$,

$$-(\mathcal{P}_{\Omega_{\text{obs}}} Z - \mathcal{P}_{\Omega_{\text{obs}}} \tilde{Y}) + \gamma \partial \|\tilde{Y}\|_*, \tag{4}$$

is equivalent to the sub-gradient of $g(Y)$ (with respect to Y) evaluated at $\tilde{Y}$,

$$-\frac{1}{2}\psi_c(\mathcal{P}_{\Omega_{\text{obs}}} X - \mathcal{P}_{\Omega_{\text{obs}}} \tilde{Y}) + \gamma \partial \|\tilde{Y}\|_*. \tag{5}$$

The proposed algorithm iteratively updates $\tilde{Y} = \arg\min_Y f(Y|Z)$ and Z using (3). Upon convergence (implied by Proposition 1 below), the sub-gradient (4) contains 0 at the converged $\tilde{Y}$ and thus the sub-gradient (5) also contains 0 at this converged $\tilde{Y}$. Therefore this $\tilde{Y}$ is the solution to (2). Details of this algorithm based on pseudo data matrix are given in Algorithm 1.

Algorithm 1 has several attractive properties. First, it can be paired with any existing (non-robust) matrix completion algorithm (or software), as can be easily seen in Step 2(c). This is a huge advantage, as a rich body of existing (non-robust) methods can be made

Algorithm 1 The General Robust Algorithm

1: Perform (non-robust) matrix completion on X and assign $Y^{\text{old}} \leftarrow \arg\min_Y f(Y|X)$. This Y^{old} is the initial estimate (starting point of the algorithm).
2: Repeat:
   (a) Compute $E \leftarrow \mathcal{P}_{\Omega_{\text{obs}}} X - \mathcal{P}_{\Omega_{\text{obs}}} Y^{\text{old}}$.
   (b) Compute $Z \leftarrow \mathcal{P}_{\Omega_{\text{obs}}} Y^{\text{old}} + \frac{1}{2}\psi_c(E)$.
   (c) Perform (non-robust) matrix completion on Z and assign $Y^{\text{new}} \leftarrow \arg\min_Y f(Y|Z)$.
   (d) If
$$\frac{\|Y^{\text{new}} - Y^{\text{old}}\|_F^2}{\|Y^{\text{old}}\|_F^2} < \varepsilon,$$
   exit.
   (e) Assign $Y^{\text{old}} \leftarrow Y^{\text{new}}$.
3: Output Y^{new}.

robust against outliers. Second, once such an (non-robust) algorithm is available, the rest of the implementation is straightforward and simple, and no expensive matrix operations are required. Lastly, it has strong theoretical backup, as to be reported in Section 4.

3.2 Further Integration with Existing Matrix Completion Algorithms

Many existing matrix completion algorithms are iterative. A direct application of Algorithm 1 would lead to an algorithm that is iterations-within-iterations. Although our extensive numerical experience suggests that these direct implementations would typically converge within a few iterations to give a reasonably fast execution time, it would still be advantageous to speed up the overall procedure. Here we show that it is possible to further improve the speed of the overall robust algorithm by embedding the pseudo data matrix idea directly into a non-robust algorithm.

We shall illustrate this with the Soft-Impute algorithm proposed by Mazumder et al. (2010). To proceed we first recall the definition of their thresholding operator S_γ: for any matrix Z of rank r,

$$S_\gamma(Z) = UD_\gamma V^\intercal, \tag{6}$$

where $Z = UDV^\intercal$ is the singular value decomposition of Z, $D = \text{diag}[d_1, \ldots, d_r]$ and $D_\gamma = \text{diag}[(d_1 - \gamma)_+, \ldots, (d_r - \gamma)_+]$. Now the main idea is to suitably replace an iterative matrix estimate with the pseudo data matrix estimate given by (3). With Soft-Impute, the resulting robust algorithm is given in Algorithm 2. We shall call this algorithm Robust-Impute. As to be shown by the numerical studies below, Robust-Impute is very fast and produces very promising empirical results. Our algorithm also has the sparse-plus-low-rank structure in the singular value thresholding step (Step 2a(iii)). This linear algebra structure has positive impact on the computational complexity. See Section 5 of Mazumder et al. (2010) for details. Moreover, the monotonicity and convergence of our algorithm is guaranteed by Proposition 1 and Theorem 2.

Algorithm 2 Robust-Impute

1: Initialize $Y^{\text{old}} = S_{\gamma_1}(\mathcal{P}_{\Omega_{\text{obs}}} X)$ and $Z = X$.
2: Do for $\gamma_1 > \gamma_2 > \cdots > \gamma_K$:
   (a) Repeat:
      (i) Compute $E \leftarrow \mathcal{P}_{\Omega_{\text{obs}}} X - \mathcal{P}_{\Omega_{\text{obs}}} Y^{\text{old}}$.
      (ii) Compute $Z \leftarrow \mathcal{P}_{\Omega_{\text{obs}}} Y^{\text{old}} + \frac{1}{2}\psi_c(E)$
      (iii) Compute $Y^{\text{new}} \leftarrow S_{\gamma_k}(\mathcal{P}_{\Omega_{\text{obs}}} Z + \mathcal{P}_{\Omega_{\text{obs}}^\perp} Y^{\text{old}})$.
      (iv) If
$$\frac{\|Y^{\text{new}} - Y^{\text{old}}\|_F^2}{\|Y^{\text{old}}\|_F^2} < \varepsilon,$$
      exit.
      (v) Assign $Y^{\text{old}} \leftarrow Y^{\text{new}}$.
   (b) Assign $\hat{Y}_{\gamma_k} \leftarrow Y^{\text{new}}$.
3: Output the sequence of solutions $\hat{Y}_{\gamma_1}, \ldots, \hat{Y}_{\gamma_K}$.

4. Theoretical Properties

This section presents some theoretical backups for the proposed methodology.

4.1 Monotonicity and global convergence

We first present the following proposition concerning the monotonicity of the algorithms. The proof can be found in Appendix A.1. We also provide an alternative proof suggested by a referee, based on the idea of alternating minimization, in Appendix A.1

Proposition 1 (Monotonicity) *Let* $Y^{(k)}$ *and*

$$Z^{(k)} = \mathcal{P}_{\Omega_{\text{obs}}} Y^{(k-1)} + \psi_c(\mathcal{P}_{\Omega_{\text{obs}}} X - \mathcal{P}_{\Omega_{\text{obs}}} Y^{(k-1)})/2$$

be, respectively, the estimate and the pseudo data matrix in the k*-th iteration. If* $Y^{(k+1)}$ *is the next estimate such that* $f(Y^{(k+1)}|Z^{(k+1)}) \le f(Y^{(k)}|Z^{(k+1)})$*, then* $g(Y^{(k+1)}) \le g(Y^{(k)})$.

For the general version (Algorithm 1), it is obvious that the condition $f(Y^{(k+1)}|Z^{(k+1)}) \le f(Y^{(k)}|Z^{(k+1)})$ is satisfied as the result of the minimization $Y^{\text{old}} \leftarrow \arg\min_Y f(Y|Z)$. For the specialized version Robust-Impute (Algorithm 2), this condition is implied by Lemma 2 of Mazumder et al. (2010). Therefore both versions are monotonic.

As pointed out by a referee, the proposed algorithms can also be viewed as an instance of the majorization-minimization (MM) algorithm (Lange et al., 2000; Hunter and Lange, 2004). It can be shown that, for $(i, j) \in \Omega_{\text{obs}}$,

$$\begin{aligned}\rho_c(X_{ij} - Y_{ij}) &\le \rho_c(X_{ij} - Y_{ij}^{\text{old}}) - (Y_{ij} - Y_{ij}^{\text{old}})\psi_c(X_{ij} - Y_{ij}^{\text{old}}) + 2 \cdot \frac{1}{2}(Y_{ij} - Y_{ij}^{\text{old}})^2 \\ &= \left[Y_{ij} - Y_{ij}^{\text{old}} - \frac{1}{2}\psi_c(X_{ij} - Y_{ij}^{\text{old}})\right]^2 + \textit{constant} \\ &= (Y_{ij} - Z_{ij})^2 + \textit{constant}.\end{aligned}$$

Therefore, subject to an additive constant that does not depend on Y, $h(Y|Y^{\text{old}}) = f(Y|Z) = (1/2)\sum_{(i,j)\in\Omega_{\text{obs}}}(Z_{ij} - Y_{ij})^2 + \gamma\|Y\|_*$ is a majorization of the objective function g. With this majorization, Algorithm 1 can be viewed as an MM algorithm. Additionally, one can majorize the unobserved entries by $(Y_{ij} - Z_{ij})^2 = (Y_{ij} - Y_{ij}^{\text{old}})^2 \geq 0$ and, together with the above majorization of the observed entries, Algorithm 2 can also be shown as an MM algorithm. Therefore the monotonicity of the proposed algorithms can also be obtained by the general theory of MM algorithm (e.g., Lange, 2010). Moreover, the explicit connection to the MM algorithm allows possible extensions of the current algorithm to other robust loss functions such as Tukey's biweight loss. However, due to non-differentiability of the objective function, the typical convergence analysis of MM algorithm (e.g., Lange, 2010, Ch. 15) does not apply to our case.

We summarize the global convergence rates of both Algorithm 1 and Algorithm 2 in the following theorem.

Theorem 2 *Let $Y^{(k)}$ and $Y^{(0)}$ be, respectively, the estimate in the k-th iteration and the starting point of Algorithm 1 or Algorithm 2 Then for any $k \geq 1$,*

$$\textit{Algorithm 1:} \qquad g(Y^{(k)}) - g(Y^*) \leq \frac{\|\mathcal{P}_{\Omega_{\text{obs}}}Y^{(0)} - \mathcal{P}_{\Omega_{\text{obs}}}Y^*\|_F^2}{2k}, \qquad \forall Y^* \in \mathcal{Y},$$

$$\textit{Algorithm 2:} \qquad g(Y^{(k)}) - g(Y^*) \leq \frac{\|Y^{(0)} - Y^*\|_F^2}{2k}, \qquad \forall Y^* \in \mathcal{Y},$$

where $\mathcal{Y}$ be the set of all global minimizers of g (i.e. $\mathcal{Y} = \arg\min_{Y\in\mathbb{R}^{n_1\times n_2}} g(Y)$).

The global convergence analysis of Algorithm 1 can be carried out similarly as in Beck and Teboulle (2009) for proximal gradient method, despite that Algorithm 1 is not a proximal gradient method. For completeness, we give the proof of Theorem 2 for Algorithm 1 in Appendix A.2.

As for Robust-Impute (Algorithm 2), we can rewrite it as an instance of the proximal gradient method applied to $g(Y) = g_1(Y_1)+g_2(Y_2)$, where $g_1(Y) = (1/2)\sum_{(i,j)\in\Omega_{\text{obs}}}\rho_c(X_{ij}-Y_{ij})$ and $g_2(Y) = \gamma\|Y\|_*$. In our case, the proximal gradient method with step size L iterates over $Y^{(k+1)} = \xi_L(Y^{(k)})$ with

$$\xi_L(\tilde{Y}) = \arg\min_{Y}\left\{g_2(Y) + \frac{L}{2}\left\|Y - \left(\tilde{Y} - \frac{1}{L}\nabla g_1(\tilde{Y})\right)\right\|_F^2\right\},$$

where L is a constant greater than or equal to the Lipstchiz constant of g_1. Note that g_1 has a Lipschitz contant 1. If we take $L = 1$, we have the following simplification.

$$\begin{aligned} g_2(Y) + \frac{L}{2}\left\|Y - \left(\tilde{Y} - \frac{1}{L}\nabla g_1(\tilde{Y})\right)\right\|_F^2 &= g_2(Y) + \frac{1}{2}\left\|Y - \left\{\tilde{Y} + \frac{1}{2}\psi_c(\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y})\right\}\right\|_F^2 \\ &= g_2(Y) + \frac{1}{2}\left\|Y - \left\{\mathcal{P}_{\Omega_{\text{obs}}^\perp}\tilde{Y} + \mathcal{P}_{\Omega_{\text{obs}}}Z\right\}\right\|_F^2. \end{aligned}$$

The minimization of ξ_1 is equivalent to Step 2a(iii) of Algorithm 2. Therefore, the proximal gradient method is the same as Robust-Impute. This connection allows us to apply the convergence results of proximal gradient method to Robust-Impute directly. Theorem 2 for Algorithm 2 follows from Theorem 3.1 of Beck and Teboulle (2009). Lastly, the Nesterov's method (Nesterov, 2007) can be applied directly to accelerate Algorithm 2. The resulted accelerated version is expected to be faster in terms of convergence. However, the acceleration in the Nesterov's method ruins the computationally beneficial sparse-plus-low-rank structure (Mazumder et al., 2010) in the singular vaue thresholding step (Step 2a(iii)). Hence, for large matrices, the non-accelerated version is still preferred in terms of overall computations. The detailed discussion can be found in Section 5 of Mazumder et al. (2010).

4.2 Stable Recovery

Recall the stable property of Candès and Plan (2010) implies that, with high probability, the recovered matrix is subject to an error bound proportional to the noise level. This subsection shows that the robust matrix completion defined by (2) is also stable.

Although the formulation of (2) has its root from classical robust statistics, it is also related to the more recent principal component pursuit (PCP) proposed by Candès et al. (2011). PCP assumes that the entries of the observed matrix are noiseless, and that this matrix can be decomposed as the sum of a low rank matrix and a sparse matrix, where the sparse matrix is treated as the gross error. In Candès et al. (2011) it is shown that using PCP perfect recovery is possible with or without missing entries in the observed matrix. Another notable work by Chandrasekaran et al. (2011) provide completely deterministic conditions for the PCP to succeed under no missing data. See Section 1.5 of Candès et al. (2011) for a detailed comparison between these two pieces of work. For the case of noisy measurements without missing entries, Zhou et al. (2010) extend PCP to stable PCP (SPCP), which is shown to be stable. However, to the best of our knowledge, there is no existing theoretical results for the case of noisy (and/or outlying) measurements with missing entries.

Inspired by She and Owen (2011), we first establish an useful link between robust matrix completion (2) and PCP in the following proposition. The proof can be found in Appendix A.3.

Proposition 3 (Equivalence) *The minimization (2) is equivalent to*

$$\underset{L,S}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}(L+S)\|_F^2 + \gamma\|L\|_* + c\|S\|_1. \tag{7}$$

That is, the minimizing Y of (2) and the minimizing L of (7) coincide.

Minimization (7) has a high degree of similarity to both PCP and SPCP. It is equivalent to

$$\begin{aligned} \underset{L,S}{\text{minimize}} \quad & \|L\|_* + \lambda\|S\|_1 \\ \text{subject to} \quad & \|\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}(L+S)\|_F^2 \leq \delta^2, \end{aligned} \tag{8}$$

where $\lambda = c/\gamma$ and $\delta > 0$ has a one-to-one correspondence to γ. When comparing with PCP, (7) permits the observed matrix to be different from the recovered matrix $(L + S)$ to allow for noisy measurements. When comparing with SPCP, (7) permits missing entries, which is necessary for matrix completion problems.

Proposition 3 has two immediate implications. First, the proposed Algorithm 1 provides a general methodology to turn a large and well-developed class of matrix completion algorithms into algorithms for solving SPCP with missing entries. Second, many useful results from PCP can be borrowed to study the theoretical properties of robust matrix completion (2). In particular, we show that (2) leads to stable recovery. With Proposition 3, it suffices to show that (7) achieves stable recovery of (L_0, S_0') from the data $\mathcal{P}_{\Omega_{\text{obs}}}(X)$ generated by $\mathcal{P}_{\Omega_{\text{obs}}}(L_0 + S_0)$ obeying $\|\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}(L_0 + S_0)\|_F \leq \delta$ and $S_0' = \mathcal{P}_{\Omega_{\text{obs}}}S_0$. Note that $L_0 = X_0$.

We need some notations to proceed. For simplicity, we assume $n = n_1 = n_2$ but our results can be easily extended to rectangular matrices ($n_1 \neq n_2$). The Euclidean inner product $\langle Q, R\rangle$ is defined as trace$(Q^\intercal R)$. Let p_0 be the proportion of observed entries. Write $\Gamma \subset \Omega_{\text{obs}}$ as the set of locations where the measurements are noisy (but not outliters), and $\Omega = \Omega_{\text{obs}}\backslash\Gamma$ as the support of $S_0' = \mathcal{P}_{\Omega_{\text{obs}}}S_0$; i.e., locations of outliers. Denote their complements as, respectively, $\Gamma^\perp$ and $\Omega^\perp$. We define $\mathcal{P}_\Gamma$, $\mathcal{P}_\Omega$, $\mathcal{P}_{\Gamma^\perp}$ and $\mathcal{P}_{\Omega^\perp}$ similarly to the definition of $\mathcal{P}_{\Omega_{\text{obs}}}$. Let r be the rank of L_0 and $UDV^\intercal$ be the corresponding singular value decomposition of L_0, where $U, V \in \mathbb{R}^{n\times r}$ and $D \in \mathbb{R}^{r\times r}$. Similar to Candès et al. (2011), we consider the linear space of matrices

$$T := \{UQ^\intercal + RV^\intercal : Q, R \in \mathbb{R}^{n\times r}\}.$$

Write $\mathcal{P}_T$ and $\mathcal{P}_{T^\perp}$ as the projection operator to T and $T^\perp$ respectively. As in Zhou et al. (2010), we define a set of notations for any pair of matrices $M = (L, S)$. Here, let $\|M\|_F := \sqrt{\|L\|_F^2 + \|S\|_F^2}$ and $\|M\|_\diamond := \|L\|_* + \lambda\|S\|_1$. We also define the projection operators $\mathcal{P}_T \times \mathcal{P}_{\Gamma^\perp} : (L, S) \mapsto (\mathcal{P}_T L, \mathcal{P}_{\Gamma^\perp} S)$ and $\mathcal{P}_{T^\perp} \times \mathcal{P}_\Gamma : (L, S) \mapsto (\mathcal{P}_{T^\perp} L, \mathcal{P}_\Gamma S)$. In our theoretical development, we consider the following special subspaces

$$\Psi := \{(L, S) : L, S \in \mathbb{R}^{n\times n}, \mathcal{P}_{\Omega_{\text{obs}}}L = \mathcal{P}_{\Omega_{\text{obs}}}S, \mathcal{P}_{\Omega_{\text{obs}}^\perp}L = \mathcal{P}_{\Omega_{\text{obs}}^\perp}S = 0\},$$
$$\Psi^\perp := \{(L, S) : L, S \in \mathbb{R}^{n\times n}, \mathcal{P}_{\Omega_{\text{obs}}}L + \mathcal{P}_{\Omega_{\text{obs}}}S = 0\}.$$

And we write the corresponding projection operators as $\mathcal{P}_\Psi$ and $\mathcal{P}_{\Psi^\perp}$ respectively. Let $M_0 = (L_0, S_0')$. Lastly, for any linear operator $\mathcal{A}$, the operator norm, denoted by $\|\mathcal{A}\|$, is $\sup_{\{\|Q\|_F=1\}} \|\mathcal{A}Q\|_F$. In below, we write that an event occurs with high probability if it holds with probability at least $1 - \mathcal{O}(n^{-10})$.

To avoid certain pathological cases (see, e.g., Candès and Recht, 2009), an incoherence condition on U and V is usually assumed. To be specific, this condition with the parameter μ is:

$$\max_i \|U^\intercal e_i\|^2 \leq \frac{\mu r}{n_1}, \quad \max_i \|V^\intercal e_i\|^2 \leq \frac{\mu r}{n_2}, \quad \text{and} \quad \|UV^\intercal\|_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}}, \tag{9}$$

where $\|Q\|$ is the operator norm or 2-norm of matrix Q (i.e., the largest singular value of Q) and $\|Q\|_\infty = \max_{i,j} |Q_{i,j}|$. This condition guarantees that, for small μ, the singular vectors are reasonably spread out.

Theorem 4 (Stable Recovery) *Suppose that L_0 obeys (9) and Ω_{obs} is uniformly distributed among all sets of cardinality $m = p_0 n^2$ with $p_0 > 0$ being the proportion of observed entries. Further suppose that each observed entry is grossly corrupted to be an outlier with probability τ independently of the others. Suppose L_0 and S_0 satisfy $r \leq \rho_r n \mu^{-1} (\log n)^{-2}$ and $\tau \leq \tau_s$ with ρ_r, τ_s being positive numerical constants. Choose $\lambda = 1/\sqrt{np_0}$. Then, with high probability (over the choices of Ω and Ω_{obs}), for any X obeying $\|\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}(L_0 + S_0)\|_F \leq \delta$, the solution $(\hat{L}, \hat{S})$ to (8) satisfies*

$$\|\hat{L} - L_0\|_F \leq \left\{2 + 8\sqrt{n}\left(1 + \sqrt{\frac{8}{p_0}}\right)\right\}\delta \quad \textit{and} \quad \|\hat{S} - S_0'\|_F \leq \left\{2 + 8\sqrt{n}\left(1 + \sqrt{\frac{8}{p_0}}\right)\right\}\sqrt{np_0}\delta,$$

where $S_0' = \mathcal{P}_{\Omega_{\text{obs}}}(S_0)$.

The proof of this theorem can be found in Appendix A.4.

5. Empirical Performances

Two sets of numerical experiments and a real data application were conducted to evaluate the practical performances of the proposed methodology. In particular the performance of the proposed procedure Robust-Impute is compared to the performance of Soft-Impute developed by Mazumder et al. (2010). The reasons Soft-Impute is selected for comparison are that it is one of the most popular matrix completion methods due to its simplicity and scalability, and that it is shown by Mazumder et al. (2010) that it generally produces superior results to other common matrix completion methods such as MMMF of Rennie and Srebro (2005), SVT of Cai et al. (2010) and OptSpace of Keshavan et al. (2010b)

5.1 Experiment 1: Gaussian Entries

This experiment covers those settings used in Mazumder et al. (2010, Section 9) and additional settings with different proportions of missing entries and outliers. For each simulated data set, the target matrix was generated as $X_0 = UV^\intercal$, where U and V are random matrices of size $100 \times r$ with independent standard normal Gaussian entries. Then each entry of X_0 is contaminated by additional independent Gaussian noise with standard deviation σ, which is set to a value such that the signal-to-noise ratio (SNR) is 1. Here SNR is defined as

$$\text{SNR} = s = \sqrt{\frac{\text{Var}(X_0)}{\sigma^2}},$$

where $\text{Var}(X_0)$ is the variance over all the entries of X_0 conditional on U and V. Next, for each entry, with probability p yet another independent Gaussian noise with $\sigma/4$ is added; these entries are treated as outliers. We call this contaminated version of X_0 as X. Lastly, Ω_{obs} is uniformly random over the indices of the matrix with missing proportion as q. In this study, we used two values for r (5, 10), three values for p (0, 0.05, 0.1) and three values for q (0.25, 0.5, 0.75). Thus in total we have 18 simulation settings. For each setting 200 simulated data sets were generated, and both the non-robust method Soft-Impute and the proposed Robust-Impute were applied to recover X_0. We also provide two oracle fittings as references. They are produced by applying Soft-Impute to the simulated data set with outlying observed entries removed (i.e., treated as missing entries), and with outlying observed entries replaced by non-outlying contaminated entries (i.e., contaminated by independent Gaussian noise with standard deivation σ) respectively.

The first oracle fitting is referred to as oracle1 while the second one is called oracle2 in the following.

For the two simulation settings with $r = 10$ and $q = 0.5$, and one with $p = 0$ while the other with $p = 0.1$, Figure 1 summarizes the average number of singular value decompositions (SVDs) used and the average test error. Here test error is defined as

$$\text{Test error} = \frac{\|\mathcal{P}_{\Omega_{\text{obs}}^{\perp}}(X_0 - \hat{X})\|_F^2}{\|\mathcal{P}_{\Omega_{\text{obs}}^{\perp}} X_0\|_F^2},$$

where $\mathcal{P}_{\Gamma}$ is the projection operator to the set of locations of the observed noisy entries (but not outliers) Γ, and $\hat{X}$ is an estimate of X_0. From Figure 1 (Top), one can see that the performance of Robust-Impute is slightly inferior to Soft-Impute in the case of no outliers ($p = 0$), while Robust-Impute gave significantly better results when outliers were present ($p = 0.1$). The inferior performance of Robust-Impute under the absence of outliers is not surprising, as it is widely known in the statistical literature that a small fraction of statistical efficiency would be lost when a robust method is applied to a data set without outliers. However, it is also known that the gain could be substantial if outliers did present.

As for computational requirements, one can see from Figure 1 (Bottom) that Robust-Impute only used slightly more SVDs on average. For ranks greater than 5, the number of SVDs used by Robust-Impute only differs from Soft-Impute on average by less than 1. This suggests that Robust-Impute is slightly more computationally demanding than Soft-Impute.

Similar experimental results were obtained for the remaining 16 simulation settings. For brevity, the corresponding results are omitted here but can be found in the supplementary document.

From this experiment some empirical conclusions can be drawn. When there is no outlier, Soft-Impute gives slightly better results, while with outliers, results from Robust-Impute are substantially better. Since that in practice one often does not know if outliers are present or not, and that Robust-Impute is not much more computationally demanding than Soft-Impute, it seems that Robust-Impute is the choice of method if one wants to be more conservative.

5.2 Experiment 2: Image Inpainting

In this experiment the target matrix is the so-called Lena image that has been used by many authors in the image processing literature. It consists of 256×256 pixels and is shown in Figure 2 (Left). The simulated data sets were generated via contaminating this Lena image by adding Gaussian noises and/or outliers in the following manner. First independent Gaussian noise was added to each pixel, where the standard deviation of the noise was set such that the SNR is 3. Next, 10% of the pixels were selected as outliers, and to them additional independent Gaussian noises with SNR 3/4 were added. In terms of selecting missing pixels, two mechanisms were considered. In the first one 40% of the pixels were randomly chosen as missing pixels, while in the second mechanism only 10% were missing but they were clustered together to form patches. Two typical simulated data sets are shown in Figure 2 (Middle). Note that Theorem 4 does not cover the second missing mechanism.

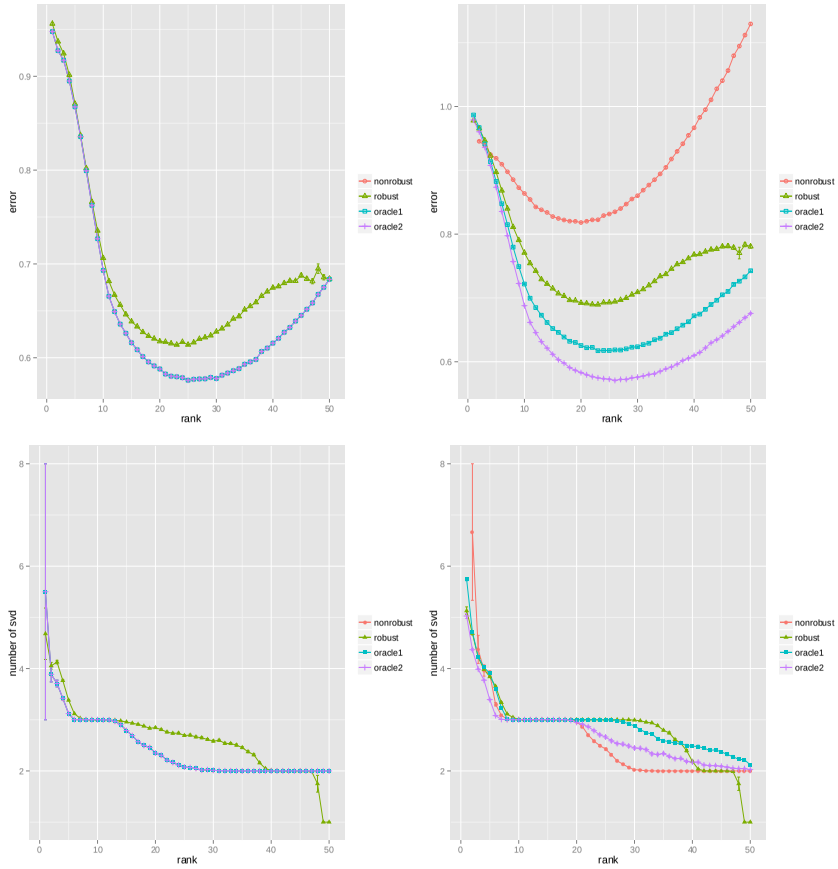


Figure 1: Top: The average test errors with their standard error bands (plus or minus one standard error). Bottom: The average number of singular value decompositions used with standard error bands (plus or minus one standard error). Left: results for the simulation setting: $r = 10$, $p = 0$ and $q = 0.5$. Right: results for the simulation setting: $r = 10$, $p = 0.1$ and $q = 0.5$.

		training error				testing error			
	rank	50	75	100	125	50	75	100	125
independent missing	Soft-Impute	0.0499	0.0351	0.0221	0.0113	0.0578	0.0565	0.0581	0.0620
	Robust-Impute	0.0486	0.0371	0.0282	0.0252	0.0546	0.0540	0.0557	0.0571
clustered missing	Soft-Impute	0.0487	0.0386	0.0296	0.0214	0.0756	0.0751	0.0760	0.0781
	Robust-Impute	0.0468	0.0390	0.0321	0.0268	0.0716	0.0714	0.0723	0.0742

Table 1: The average training and testing errors for the Lena experiment.

For each missing mechanism, 200 data sets were generated and both Soft-Impute and Robust-Impute were applied to reconstruct Lena.

The average training and testing errors[1] of the recovered images of matrix ranks 50, 75, 100 and 125 are reported in Table 1. For both missing mechanisms, Soft-Impute tends to have lower training errors, but larger testing errors when compared to Robust-Impute. In other words, Soft-Impute tends to over-fit the data, and Robust-Impute seems to provide better results. Lastly, for visual evaluation, the recovered image of rank 100 using Robust-Impute is displayed in Figure 1 (Right). From this one can see that the proposed Robust-Impute provided good recoveries under both missing mechanisms.

5.3 Real data application: Landsat Thematic Mapper

In this application the target matrix is an image from a Landsat Thematic Mapper data set publicly available at `http://ternauscover.science.uq.edu.au/`. This data set contains 149 multiband images of 100×100 pixels, with each image consists of six bands (blue, green and red with three infrared bands). The scene is centered on the Tumbarumba flux tower on the western slopes of the Snowy Mountains in Australia. Due to wild fires or related reasons, some pixels are of value zero which can be treated as missing. Also, due to detector malfunctioning, some isolated pixels have values much higher than the remaining pixels, which can be treated as outliers. We selected an image band with a high missing rate (27.6%) to test our procedure.

To evaluate the recovered matrix, the observed pixels were split into training, validation and testing sets consisting 80%, 10% and 10% of the observed (nonzero) entries respectively. We used the validation set to tune γ. The validation errors are computed in two ways: mean squared error (MSE) $\sqrt{\sum_{(i,j)\in\mathcal{V}}(X_{ij}-\hat{X}_{ij})^2/|\mathcal{V}|}$ and mean absolute deviation (MAD) $\text{median}\{|X_{ij}-\hat{X}_{ij}| : (i,j)\in\mathcal{V}\}$, where $\mathcal{V}$ represents the validation set. Similarly, we compute the testing errors in terms of MSE and MAD. Note that the validation and testing sets may contain outliers and therefore MAD serves as a robust and reliable performance measure. The corresponding results are shown in Table 2. From this table it can be seen that with the presence of outliers, Robust-Imputeprovided better results.

1. The solution path (formed by the pre-specified set of γ's) may not contain any solution of rank 50, 75, 100 and 125. Thus, the average errors were computed over those fittings that contained the corresponding fitted ranks. At most 2% of these fittings were discarded due to this reason.

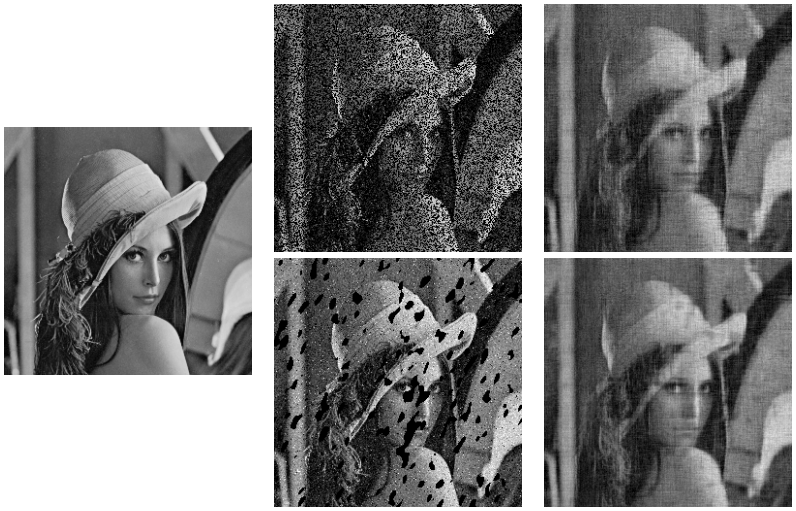


Figure 2: Left: the Lena image. Middle: degraded Lena images by the independent missing mechanism (Top) and the clustered missing mechanism (Down). Right: corresponding recovered images of rank 100 via Robust-Impute.

	tuning by MSE			tuning by MAD		
	rank	MSE	MAD	rank	MSE	MAD
Soft-Impute	24	45.20	31.15	21	45.23	31.15
Robust-Impute	24	44.63	29.00	29	44.57	28.76

Table 2: Rank and testing errors of the real data application.

6. Concluding remarks

In this paper a classical idea from robust statistics has been brought to the matrix completion problem. The result is a new matrix completion method that can handle noisy and outlying entries. This method uses the Huber function to downweigh the effects of outliers. A new algorithm is developed to solve the corresponding optimization problem. This algorithm is relatively fast, easy to implement and monotonic convergent. It can be paired with any existing (non-robust) matrix completion methods to make such methods robust against outliers. We also developed a specialized version of this algorithm, called Robust-Impute. Its promising empirical performance has been illustrated via numerical experiments. Lastly, we have shown that the proposed method is stable; that is, with high probability, the error of recovered matrix is bounded by a constant proportional to the noise level.

Acknowledgments

The authors are most grateful to the referees and the action editor for their many constructive and useful comments, which led to a much improved version of the paper. The work of Wong was partially supported by the National Science Foundation under Grants DMS-1612985 and DMS-1711952 (subcontract). The work of Lee was partially supported by the National Science Foundation under Grants DMS-1512945 and DMS-1513484.

Appendix A. Technical Details

In this section, we provide technical details of our theoretical results.

A.1 Proofs of Proposition 1

Proof By rewriting

$$\begin{aligned}&\|\mathcal{P}_{\Omega_{\text{obs}}}Z^{(k+1)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\|_F^2\\&=\|\mathcal{P}_{\Omega_{\text{obs}}}Z^{(k+1)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}\|_F^2+\|\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\|_F^2\\&\quad 2\times\text{trace}\left[\{\mathcal{P}_{\Omega_{\text{obs}}}Z^{(k+1)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}\}\{\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\}^\intercal\right],\end{aligned}$$

and using $f(Y^{(k+1)}|Z^{(k+1)})\le f(Y^{(k)}|Z^{(k+1)})$, we have

$$\begin{aligned}&\frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\|_F^2+\text{trace}\left[\{\mathcal{P}_{\Omega_{\text{obs}}}Z^{(k+1)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}\}\{\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\}^\intercal\right]\\&\quad+\gamma\|Y^{(k+1)}\|_*\le\gamma\|Y^{(k)}\|_*.\end{aligned}$$

Thus, by substituting $Z^{(k+1)}=\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}+\frac{1}{2}\rho_c'(\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)})$,

$$\begin{aligned}&\frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\|_F^2+\frac{1}{2}\text{trace}\left[\rho_c'(\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)})\{\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\}^\intercal\right]\\&\quad+\gamma\|Y^{(k+1)}\|_*\le\gamma\|Y^{(k)}\|_*.\end{aligned}\tag{10}$$

Here we abuse the notation slightly so that ρ_c' of a matrix simply means the matrix formed by applying ρ_c' to its entries. Note that for each $(i,j)\in\Omega_{\text{obs}}$, by Taylor's expansion,

$$\begin{aligned}\rho_c(X_{ij}-Y_{ij}^{(k+1)})&=\rho(X_{ij}-Y_{ij}^{(k)})+(Y_{ij}^{(k)}-Y_{ij}^{(k)})\rho_c'(X_{ij}-Y_{ij}^{(k)})\\&\quad+\int_{X_{ij}-Y_{ij}^{(k)}}^{X_{ij}-Y_{ij}^{(k+1)}}(X_{ij}-Y_{ij}^{(k+1)}-t)\rho_c''(t)dt,\end{aligned}$$

and the last integral term is less than or equal to $(Y_{ij}^{(k)}-Y_{ij}^{(k+1)})^2$ due to $\rho_c''\le 2$ almost everywhere. Thus,

$$\begin{aligned}\sum_{(i,j)\in\Omega_{\text{obs}}}\rho_c(X_{ij}-Y_{ij}^{(k+1)})&\le\sum_{(i,j)\in\Omega_{\text{obs}}}\rho_c(X_{ij}-Y_{ij}^{(k)})\\&\quad+\text{trace}\left[\rho_c'(\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)})\{\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\}^\intercal\right]\\&\quad+\|\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\|_F^2.\end{aligned}$$

Now, plugging it into (10), we have $g(Y^{(k+1)})\le g(Y^{(k)})$. ■

Proof (Alternative proof of Proposition 1) Similar to the proof of Proposition 3 in Section A.3, one can show that

$$g(Y)=\min_S\frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y-\mathcal{P}_{\Omega_{\text{obs}}}S\|_F^2+\gamma\|Y\|_*+c\|S\|_1,\tag{11}$$

where the minimizer is $S(Y)=(1/2)\psi_c(\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y)$. Now, one can show that

$$\begin{aligned}Z^{(k+1)}&=\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}+(1/2)\psi_c(\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)})\\&=\mathcal{P}_{\Omega_{\text{obs}}}X+(1/2)\psi_c(\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}X)\\&=\mathcal{P}_{\Omega_{\text{obs}}}X-(1/2)\psi_c(\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)})\\&=\mathcal{P}_{\Omega_{\text{obs}}}X-S(Y^{(k)}).\end{aligned}$$

Now, due to (11),

$$\begin{aligned}g(Y^{(k+1)})&=\min_S\frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}-\mathcal{P}_{\Omega_{\text{obs}}}S\|_F^2+\gamma\|Y^{(k+1)}\|_*+c\|S\|_1\\&\le\frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}-\mathcal{P}_{\Omega_{\text{obs}}}S(Y^{(k)})\|_F^2+\gamma\|Y^{(k+1)}\|_*+c\|S(Y^{(k)})\|_1\\&=f(Y^{(k+1)}|X-S(Y^{(k)}))+c\|S(Y^{(k)})\|_1\\&=f(Y^{(k+1)}|\mathcal{P}_{\Omega_{\text{obs}}}X-S(Y^{(k)}))+c\|S(Y^{(k)})\|_1\\&=f(Y^{(k+1)}|Z^{(k+1)})+c\|S(Y^{(k)})\|_1\\&\le f(Y^{(k)}|Z^{(k+1)})+c\|S(Y^{(k)})\|_1\\&=\frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}X-\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}-\mathcal{P}_{\Omega_{\text{obs}}}S(Y^{(k)})\|_F^2+\gamma\|Y^{(k)}\|_*+c\|S(Y^{(k)})\|_1\\&=g(Y^{(k)}).\end{aligned}$$

■

A.2 Proof of Theorem 2 for Algorithm 1

Proof This proof closely follows the proofs of Lemma 2.3 and Theorem 3.1 in Beck and Teboulle (2009) by modifying their approximation model to

$$\begin{aligned}\zeta(Y,\tilde{Y}) &= g_1(\tilde{Y}) + \langle Y-\tilde{Y}, \nabla g_1(\tilde{Y})\rangle + \frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}Y - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y}\|_F^2 + g_2(Y)\\ &= g_1(\tilde{Y}) - \frac{1}{2}\langle Y-\tilde{Y}, \psi_c(\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y})\rangle + \frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}Y - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y}\|_F^2 + g_2(Y)\\ &= g_1(\tilde{Y}) - \frac{1}{2}\langle \mathcal{P}_{\Omega_{\text{obs}}}Y - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y}, \psi_c(\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y})\rangle + \frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}Y - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y}\|_F^2\\ &\quad + g_2(Y),\end{aligned}$$

where $\langle X, Y\rangle = \sum_{i,j} X_{ij}Y_{ij}$. Note that $\arg\min_Y \zeta(Y,\tilde{Y})$ is the same as $\arg\min_Y f(Y|Z)$, where $Z = \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y} + (1/2)\psi_c(\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y})$, in Steps 2(a)-(c) of Algorithm 1. Let $\Pi(\tilde{Y}) = \arg\min_Y \zeta(Y,\tilde{Y})$. Therefore $Y^{(k+1)} = \Pi(Y^{(k)})$. Moreover,

$$g_1(Y) \le g_1(\tilde{Y}) + \langle Y-\tilde{Y}, \nabla g_1(\tilde{Y})\rangle + \frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}Y - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y}\|_F^2,$$

for any Y and $\tilde{Y}$. Therefore, $g(\Pi(\tilde{Y})) \le \zeta(\Pi(\tilde{Y}),\tilde{Y})$ for any $\tilde{Y} \in \mathbb{R}^{n_1\times n_2}$.

To proceed, we need a modified version of Lemma 2.3 in Beck and Teboulle (2009).

Lemma 5 *For any $\tilde{Y}, Y \in \mathbb{R}^{n_1\times n_2}$,*

$$g(Y) - g(\Pi(\tilde{Y})) \ge \frac{1}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}\Pi(\tilde{Y}) - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y}\|_F^2 + \langle \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y} - \mathcal{P}_{\Omega_{\text{obs}}}Y, \mathcal{P}_{\Omega_{\text{obs}}}\Pi(\tilde{Y}) - \mathcal{P}_{\Omega_{\text{obs}}}Y\rangle.$$

This lemma is proved as follows. Since $\Pi(\tilde{Y})$ is the minimizer of the convex function $\zeta(\cdot,\tilde{Y})$, there exists a $b(\tilde{Y}) \in \partial g_2(\Pi(\tilde{Y}))$, the subdifferential of g_2 at $\Pi(\tilde{Y})$, such that $\nabla g_1(\tilde{Y}) + \mathcal{P}_{\Omega_{\text{obs}}}\Pi(\tilde{Y}) - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y} + b(\tilde{Y}) = 0$. By the convexity of g_1 and g_2,

$$\begin{aligned} g_1(Y) &\ge g_1(\tilde{Y}) - \frac{1}{2}\langle Y-\tilde{Y}, \psi_c(\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y})\rangle\\ g_2(Y) &\ge g_2(\Pi(\tilde{Y})) - \langle Y - \Pi(\tilde{Y}), b(\tilde{Y})\rangle.\end{aligned}$$

Therefore,

$$g(Y) \ge g_1(\tilde{Y}) - \frac{1}{2}\langle Y-\tilde{Y}, \psi_c(\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}\tilde{Y})\rangle + g_2(\Pi(\tilde{Y})) - \langle Y - \Pi(\tilde{Y}), b(\tilde{Y})\rangle. \quad (12)$$

Since $g(\Pi(\tilde{Y})) \le \zeta(\Pi(\tilde{Y}),\tilde{Y})$, we have $g(Y) - g(\Pi(\tilde{Y})) \ge g(Y) - \zeta(\Pi(\tilde{Y}),\tilde{Y})$. Plugging in (12), the definition of ζ and the condition for b, the conclusion of the lemma follows.

Using Lemma 5 with $Y = Y^*$ and $\tilde{Y} = Y^{(k)}$, we have

$$2\{g(Y^*) - g(Y^{(k)})\} \ge \|\mathcal{P}_{\Omega_{\text{obs}}}Y^* - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)}\|_F^2 - \|\mathcal{P}_{\Omega_{\text{obs}}}Y^* - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}\|_F^2.$$

Summing it over $k = 0,\ldots,m-1$,

$$2\left\{mg(Y^*) - \sum_{k=0}^{m-1} g(Y^{(k)})\right\} \ge \|\mathcal{P}_{\Omega_{\text{obs}}}Y^* - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(m)}\|_F^2 - \|\mathcal{P}_{\Omega_{\text{obs}}}Y^* - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(0)}\|_F^2. \quad (13)$$

Applying Lemma 5 with $Y = \tilde{Y} = Y^{(k)}$,

$$2\left\{g(Y^{(k)}) - g(Y^{(k+1)})\right\} \ge \|\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)} - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}\|_F^2.$$

Multiplying it by k and summing over $k = 0,\ldots,m-1$,

$$2\left\{-mg(Y^{(m)}) + \sum_{k=0}^{m-1} g(Y^{(k+1)})\right\} \ge \sum_{k=0}^{m-1} k\|\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)} - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}\|_F^2. \quad (14)$$

Adding (13) and (14),

$$\begin{aligned}2\left\{g(Y^*) - g(Y^{(m)})\right\} &\ge \|\mathcal{P}_{\Omega_{\text{obs}}}Y^* - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(m)}\|_F^2 - \|\mathcal{P}_{\Omega_{\text{obs}}}Y^* - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(0)}\|_F^2\\ &\quad + \sum_{k=0}^{m-1} k\|\mathcal{P}_{\Omega_{\text{obs}}}Y^{(k+1)} - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(k)}\|_F^2.\end{aligned}$$

Therefore,

$$g(Y^{(m)}) - g(Y^*) \le \frac{\|\mathcal{P}_{\Omega_{\text{obs}}}Y^* - \mathcal{P}_{\Omega_{\text{obs}}}Y^{(0)}\|_F^2}{2m}.$$

■

A.3 Proof of Proposition 3

Proof Since both (2) and (7) are convex, we only need to consider the sub-gradients. The sub-gradient conditions for minimizier of (2) are given as follows:

$$0 \in -\frac{1}{2}\rho_c'(\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}Y) + \gamma\partial\|Y\|_*, \quad (15)$$

where $\partial\|\cdot\|_*$ represents the set of subgradients of the nuclear norm. The sub-gradient conditions for minimizier of (7) are given as follows:

$$0 \in -\mathcal{P}_{\Omega_{\text{obs}}}(X - L - S) + \gamma\partial\|L\|_* \quad (16)$$

$$0 \in -\mathcal{P}_{\Omega_{\text{obs}}}(X - L - S) + c\partial\|S\|_1, \quad (17)$$

where $\partial\|\cdot\|_1$ represents the set of subgradients of $\|\cdot\|_1$. Here (17) implies, for $(i,j) \in \Omega_{\text{obs}}$,

$$S_{ij} = \begin{cases} X_{ij} - L_{ij} - c, & X_{ij} - L_{ij} > c\\ 0, & |X_{ij} - L_{ij}| \le c\\ X_{ij} - L_{ij} + c, & X_{ij} - L_{ij} < -c\end{cases} \quad (18)$$

and $S_{ij} = 0$ for $(i,j) \in \Omega_{\text{obs}}^{\perp}$. Note, for $(i,j) \in \Omega_{\text{obs}}$, $X_{ij} - L_{ij} - S_{ij} = \rho_c'(X_{ij} - L_{ij})/2$. Plugging it into (16), we have (15) and thus this proves the proposition. ■

A.4 Proof of Theorem 4

To prove Theorem 4, we first show three lemmas and one proposition.

Lemma 6 (Modified Lemma A.2 in Candès et al. 2011) *Assume that for any matrix Q, $\|\mathcal{P}_T\mathcal{P}_{\Gamma^\perp}Q\|_F \le n\|\mathcal{P}_{T^\perp}\mathcal{P}_{\Gamma^\perp}Q\|_F$. Suppose there is a pair (W,F) obeying*

$$\begin{cases}\mathcal{P}_T W = 0, \quad \|W\| < 1/2,\\ \mathcal{P}_{\Gamma^\perp}F = 0, \quad \|F\|_\infty < 1/2,\\ UV^\intercal + W + \mathcal{P}_T D = \lambda(\mathrm{sgn}(S_0') + F) \quad \text{with} \quad \|\mathcal{P}_T D\|_F \le n^{-2}.\end{cases} \tag{19}$$

Then for any perturbation $H = (H_L, H_S)$ satisfying $\mathcal{P}_{\Omega_{\text{obs}}}H_L + \mathcal{P}_{\Omega_{\text{obs}}}H_S = 0$,

$$\|M_0 - H\|_\Diamond \ge \|M_0\|_\Diamond + \left(\frac{1}{2} - \frac{1}{n}\right)\|\mathcal{P}_{T^\perp}H_L\|_* + \left(\frac{\lambda}{2} - \frac{n+1}{n^2}\right)\|\mathcal{P}_\Gamma H_L\|_1.$$

The proof of this lemma can be found in Candès et al. (2011). To procced, we write $\|M\|_{F,\lambda}^2 = \|L\|_F^2 + \lambda^2\|S\|_F^2$ for any pair of matrices $M = (L,S)$.

Lemma 7 *Let $M = (M_L, M_S)$ be any pair of matrices. Suppose $\|\mathcal{P}_T\mathcal{P}_\Omega\|^2 \le p_0/8$ and $\|\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T M_L\|_F^2 \ge p_0\|\mathcal{P}_T M_L\|_F^2/2$. Then*

$$\|\mathcal{P}_\Psi(\mathcal{P}_T \times \mathcal{P}_\Omega)M\|_{F,\lambda}^2 \ge \frac{(1+\lambda^2)p_0}{16}\|(\mathcal{P}_T \times \mathcal{P}_\Omega)M\|_F^2.$$

Proof (Proof of Lemma 7) Note that for any $M' = (M_L', M_S')$,

$$\mathcal{P}_\Psi M' = \left(\frac{\mathcal{P}_{\Omega_{\text{obs}}}(M_L' + M_S')}{2}, \frac{\mathcal{P}_{\Omega_{\text{obs}}}(M_L' + M_S')}{2}\right).$$

Thus

$$\begin{aligned}\|\mathcal{P}_\Psi(\mathcal{P}_T \times \mathcal{P}_\Omega)M\|_{F,\lambda}^2 &= \frac{1+\lambda^2}{4}\|\mathcal{P}_{\Omega_{\text{obs}}}(\mathcal{P}_T M_L + \mathcal{P}_\Omega M_S)\|_F^2\\ &= \frac{1+\lambda^2}{4}\left(\|\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T M_L\|_F^2 + \|\mathcal{P}_\Omega M_S\|_F^2 + 2\langle\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T M_L, \mathcal{P}_\Omega M_S\rangle\right),\end{aligned}$$

where the last equality is due to $\Omega \subset \Omega_{\text{obs}}$. By $\|\mathcal{P}_T\mathcal{P}_\Omega\|^2 \le p_0/8$,

$$\begin{aligned}\langle\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T M_L, \mathcal{P}_\Omega M_S\rangle &= \langle\mathcal{P}_T M_L, \mathcal{P}_\Omega M_S\rangle\\ &= \langle\mathcal{P}_T M_L, (\mathcal{P}_T\mathcal{P}_\Omega)\mathcal{P}_\Omega M_S\rangle\\ &\ge -\|\mathcal{P}_T\mathcal{P}_\Omega\|\|\mathcal{P}_T M_L\|_F\|\mathcal{P}_\Omega M_S\|_F\\ &\ge -\frac{\sqrt{p_0}}{2\sqrt{2}}\|\mathcal{P}_T M_L\|_F\|\mathcal{P}_\Omega M_S\|_F.\end{aligned}$$

Combining with $\|\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T M_L\|_F^2 \ge p_0\|\mathcal{P}_T M_L\|_F^2/2$, we have

$$\|\mathcal{P}_\Psi(\mathcal{P}_T \times \mathcal{P}_\Omega)M\|_{F,\lambda}^2 \ge \frac{1+\lambda^2}{4}\left(\frac{p_0}{2}\|\mathcal{P}_T M_L\|_F^2 + \|\mathcal{P}_\Omega M_S\|_F^2 - \sqrt{\frac{p_0}{2}}\|\mathcal{P}_T M_L\|_F\|\mathcal{P}_\Omega M_S\|_F\right).$$

As $2(x^2+y^2-xy) \ge x^2+y^2$ for $x, y \ge 0$,

$$\|\mathcal{P}_\Psi(\mathcal{P}_T \times \mathcal{P}_\Omega)M\|_{F,\lambda}^2 \ge \frac{1+\lambda^2}{8}\left(\frac{p_0}{2}\|\mathcal{P}_T M_L\|_F^2 + \|\mathcal{P}_\Omega M_S\|_F^2\right) \ge \frac{(1+\lambda^2)p_0}{16}\|(\mathcal{P}_T \times \mathcal{P}_\Omega)M\|_F^2.$$

■

Lemma 8 *Let $M = (M_L, M_S)$ be any pair of matrices. Then $\|\mathcal{P}_\Psi M\|_{F,\lambda}^2 \le \|M\|_{F,\lambda}^2/2$.*

Proof (Proof of Lemma 8) Write $M^\Psi = (M_L^\Psi, M_S^\Psi) = \mathcal{P}_\Psi M$. Since $\|M_L^\Psi\|_F^2 = \|M_S^\Psi\|_F^2$,

$$\begin{aligned}\|\mathcal{P}_\Psi M\|_{F,\lambda}^2 &= \|M_L^\Psi\|_F^2 + \lambda^2\|M_S^\Psi\|_F^2\\ &= \frac{1}{2}(\|M_L^\Psi\|_F^2 + \|M_S^\Psi\|_F^2) + \frac{\lambda^2}{2}(\|M_L^\Psi\|_F^2 + \|M_S^\Psi\|_F^2)\\ &= \frac{1}{2}\|M^\Psi\|_F^2 + \frac{\lambda^2}{2}\|M^\Psi\|_F^2\\ &\le \frac{1}{2}\|M\|_F^2 + \frac{\lambda^2}{2}\|M\|_F^2 = \frac{1}{2}\|M\|_{F,\lambda}^2.\end{aligned}$$

■

Proposition 9 *Assume that for any matrix Q,*

$$\|\mathcal{P}_T\mathcal{P}_{\Gamma^\perp}Q\|_F \le n\|\mathcal{P}_{T^\perp}\mathcal{P}_{\Gamma^\perp}Q\|_F \quad \text{and} \quad \|\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T Q\|_F \ge p_0\|\mathcal{P}_T Q\|_F/2.$$

Further suppose $4/n < \lambda \le 1$, $n \ge 3$, $p_0 > 0$, $\|\mathcal{P}_T\mathcal{P}_\Omega\|^2 \le p_0/8$ and that there exists a pair (W,F) obeying (19). Then the solution $\hat{M} = (\hat{L}, \hat{S})$ to (7) satisfies

$$\|\hat{M} - M_0\|_{F,\lambda} \le \left[\sqrt{1+\lambda^2} + 4\left(1 + \sqrt{\frac{8}{p_0}}\right)(\sqrt{n} + n\lambda\sqrt{p_0})\right]\delta.$$

where $M_0 = (L_0, S_0')$ such that $\|\mathcal{P}_{\Omega_{\text{obs}}}X - \mathcal{P}_{\Omega_{\text{obs}}}(L_0 + S_0))\|_F^2 \le \delta$ and $S_0' = \mathcal{P}_{\Omega_{\text{obs}}}S_0$. Further, if $\lambda = 1/\sqrt{np_0}$ (which implies $1/n < p_0 < n/16$), we obtain

$$\|\hat{L} - L\|_F \le \left\{2 + 8\sqrt{n}\left(1 + \sqrt{\frac{8}{p_0}}\right)\right\}\delta \quad \text{and} \quad \|\hat{S} - S_0'\|_F \le \left\{2 + 8\sqrt{n}\left(1 + \sqrt{\frac{8}{p_0}}\right)\right\}\sqrt{np_0}\delta.$$

Proof (Proof of Proposition 9) Write $\hat{M} = M_0 + H$, where $H = (H_L, H_S)$, and $H^\Psi = (H_L^\Psi, H_S^\Psi) = \mathcal{P}_\Psi H$ and $H^{\Psi^\perp} = (H_L^{\Psi^\perp}, H_S^{\Psi^\perp}) = \mathcal{P}_{\Psi^\perp}H$. We want to bound

$$\begin{aligned}\|H\|_{F,\lambda} &= \|H^\Psi + H^{\Psi^\perp}\|_{F,\lambda}\\ &\le \|H^\Psi\|_{F,\lambda} + \|H^{\Psi^\perp}\|_{F,\lambda}\\ &\le \|H^\Psi\|_{F,\lambda} + \|(\mathcal{P}_{T^\perp} \times \mathcal{P}_\Gamma)H^{\Psi^\perp}\|_{F,\lambda} + \|(\mathcal{P}_T \times \mathcal{P}_{\Gamma^\perp})H^{\Psi^\perp}\|_{F,\lambda}.\end{aligned} \tag{20}$$

We start with the first term of (20). Since $H_L^\Psi = H_S^\Psi = (1/2)\mathcal{P}_{\Omega_{\text{obs}}}(H_L + H_S)$,

$$\begin{aligned}\|H^\Psi\|_{F,\lambda} &= \frac{\sqrt{1+\lambda^2}}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}(H_L + H_S)\|_F \\ &= \frac{\sqrt{1+\lambda^2}}{2}\|\mathcal{P}_{\Omega_{\text{obs}}}(\hat{L} + \hat{S} - L_0 - S_0')\|_F \\ &\le \frac{\sqrt{1+\lambda^2}}{2}\left(\|\mathcal{P}_{\Omega_{\text{obs}}}(\hat{L} + \hat{S} - X)\|_F + \|\mathcal{P}_{\Omega_{\text{obs}}}(L_0 + S_0' - X)\|_F\right) \\ &\le \delta\sqrt{1+\lambda^2},\end{aligned}$$

where the last inequality is due to the fact that both M_0 and $\hat{M}$ are feasible.

Then we focus on the second term of (20). First, we have

$$\|M_0\|_\diamond \ge \|\hat{M}\|_\diamond = \|M_0 + H\|_\diamond \ge \|M_0 + H^{\Psi^\perp}\|_\diamond - \|H^\Psi\|_\diamond.$$

By Lemma 6,

$$\|M_0 + H^{\Psi^\perp}\|_\diamond \ge \|M_0\|_\diamond + a(n)\|\mathcal{P}_{T^\perp}H_L^{\Psi^\perp}\|_* + b(n,\lambda)\|\mathcal{P}_\Gamma H_L^{\Psi^\perp}\|_1,$$

where

$$a(n) = \frac{1}{2} - \frac{1}{n} \quad \text{and} \quad b(n,\lambda) = \frac{\lambda}{2} - \frac{n+1}{n^2}.$$

Now, combining the above inequalities,

$$\|H^\Psi\|_\diamond \ge a(n)\|\mathcal{P}_{T^\perp}H_L^{\Psi^\perp}\|_* + b(n,\lambda)\|\mathcal{P}_\Gamma H_L^{\Psi^\perp}\|_1. \tag{21}$$

By the assumption that $\lambda > 4/n$ and $n \ge 3$,

$$a(n) = \frac{1}{2} - \frac{1}{n} > 0 \quad \text{and} \quad b(n,\lambda) = \frac{\lambda}{2} - \frac{n+1}{n^2} > \frac{2}{n} - \frac{1}{n} - \frac{1}{n^2} = \frac{1}{n} - \frac{1}{n^2} > 0.$$

Therefore (21) implies $\|H^\Psi\|_\diamond \ge a(n)\|\mathcal{P}_{T^\perp}H_L^{\Psi^\perp}\|_*$ and $\|H^\Psi\|_\diamond \ge b(n,\lambda)\|\mathcal{P}_\Gamma H_L^{\Psi^\perp}\|_1$.

Now, we are ready to establish a bound for the second term of (20).

$$\begin{aligned}\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_\Gamma)H^{\Psi^\perp}\|_{F,\lambda} &\le \|\mathcal{P}_{T^\perp}H_L^{\Psi^\perp}\|_F + \lambda\|\mathcal{P}_\Gamma H_S^{\Psi^\perp}\|_F \\ &\le \|\mathcal{P}_{T^\perp}H_L^{\Psi^\perp}\|_* + \lambda\|\mathcal{P}_\Gamma H_S^{\Psi^\perp}\|_1 \\ &\le \left\{\frac{1}{a(n)} + \frac{\lambda}{b(n,\lambda)}\right\}\|H^\Psi\|_\diamond \\ &\le 4(\|H_L^\Psi\|_* + \lambda\|H_S^\Psi\|_1).\end{aligned}$$

As for the third term of (20), we apply Lemma 7 and the bound of the second term in (20). As $\mathcal{P}_\Psi H^{\Psi^\perp} = 0$, $\mathcal{P}_\Psi(\mathcal{P}_T \times \mathcal{P}_{\Gamma^\perp})H^{\Psi^\perp} + \mathcal{P}_\Psi(\mathcal{P}_{T^\perp} \times \mathcal{P}_\Gamma)H^{\Psi^\perp} = 0$. Therefore, due to Lemma 8,

$$\|\mathcal{P}_\Psi(\mathcal{P}_T \times \mathcal{P}_{\Gamma^\perp})H^{\Psi^\perp}\|_{F,\lambda} = \|\mathcal{P}_\Psi(\mathcal{P}_{T^\perp} \times \mathcal{P}_\Gamma)H^{\Psi^\perp}\|_{F,\lambda} \le \frac{1}{\sqrt{2}}\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_\Gamma)H^{\Psi^\perp}\|_{F,\lambda}.$$

As $\mathcal{P}_{\Omega_{\text{obs}}^\perp}H_S$ does not affect the feasibility of $M+H$ and H is chosen such that $\|M+H\|_\diamond$ is minimized, thus $\mathcal{P}_{\Omega_{\text{obs}}^\perp}H_S^{\Psi^\perp} = \mathcal{P}_{\Omega_{\text{obs}}^\perp}H_S = 0$ which implies $(\mathcal{P}_T \times \mathcal{P}_{\Gamma^\perp})H^{\Psi^\perp} = (\mathcal{P}_T \times \mathcal{P}_\Omega)H^{\Psi^\perp}$. Thus, by Lemma 7,

$$\|(\mathcal{P}_T \times \mathcal{P}_{\Gamma^\perp})H^{\Psi^\perp}\|_F \le \sqrt{\frac{8}{(1+\lambda^2)p_0}}\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_\Gamma)H^{\Psi^\perp}\|_{F,\lambda} \le \sqrt{\frac{8}{p_0}}\|(\mathcal{P}_{T^\perp} \times \mathcal{P}_\Gamma)H^{\Psi^\perp}\|_{F,\lambda}.$$

And $\|(\mathcal{P}_T \times \mathcal{P}_{\Gamma^\perp})H^{\Psi^\perp}\|_{F,\lambda} \le \|(\mathcal{P}_T \times \mathcal{P}_{\Gamma^\perp})H^{\Psi^\perp}\|_F$ as $\lambda \le 1$.

Collecting all the above bounds for the three terms, we derive the bound for $\|H\|_{F,\lambda}$:

$$\|H\|_{F,\lambda} \le \delta\sqrt{1+\lambda^2} + 4\left(1 + \sqrt{\frac{8}{p_0}}\right)(\|H_L^\Psi\|_* + \lambda\|H_S^\Psi\|_1).$$

Finally, $\|H_L^\Psi\|_* \le \sqrt{n}\|H_L^\Psi\|_F$, $\|H_S^\Psi\|_1 = \sqrt{p_0 n^2}\|H_S^\Psi\|_F$ (since H_S^Ψ is supported on Ω_{obs}) and $\|H_L^\Psi\|_F = \|H_S^\Psi\| = \|\mathcal{P}_{\Omega_{\text{obs}}}(H_L + H_S)\|_F/2 \le \delta$. Therefore,

$$\|H\|_{F,\lambda} \le \delta\left[\sqrt{1+\lambda^2} + 4\left(1 + \sqrt{\frac{8}{p_0}}\right)(\sqrt{n} + n\lambda\sqrt{p_0})\right].$$

Assume that $\lambda = 1/\sqrt{np_0}$. First we note that, due to $\lambda > 4/n$, this condition imposes a reasonable coverage of p_0: $1/n < p_0 < n/16$. Now we focus on simplifying the bound for $\|H\|_{F,\lambda}$.

$$\sqrt{1+\lambda^2} + 4\left(1 + \sqrt{\frac{8}{p_0}}\right)(\sqrt{n} + n\lambda\sqrt{p_0}) \le 2 + 8\sqrt{n}\left(1 + \sqrt{\frac{8}{p_0}}\right).$$

This implies

$$\|H_L\|_F \le \left\{2 + 8\sqrt{n}\left(1 + \sqrt{\frac{8}{p_0}}\right)\right\}\delta \qquad \text{and} \qquad \|H_S\|_F \le \left\{2 + 8\sqrt{n}\left(1 + \sqrt{\frac{8}{p_0}}\right)\right\}\sqrt{np_0}\delta.$$

■

To prove Theorem 4, we establish one additional lemma.

Lemma 10 *Suppose* $\|\mathcal{P}_T - p_0^{-1}\mathcal{P}_T\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T\| \le 1/2$. *Then for any matrix* Q,

$$\|\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T Q\|_F^2 \ge \frac{p_0}{2}\|\mathcal{P}_T Q\|_F^2.$$

Proof (Proof of Lemma 10) By the assumptions, for any matrix Q,

$$\begin{aligned}\|\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T Q\|_F^2 &= \langle \mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T Q, \mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T Q\rangle \\ &= \langle \mathcal{P}_T Q, \mathcal{P}_T\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T Q\rangle \\ &= p_0\langle \mathcal{P}_T Q, p_0^{-1}\mathcal{P}_T\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T Q\rangle \\ &= p_0\left[\|\mathcal{P}_T Q\|_F^2 + \langle \mathcal{P}_T Q, (p_0^{-1}\mathcal{P}_T\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T - \mathcal{P}_T)Q\rangle\right] \\ &\le p_0\left(\|\mathcal{P}_T Q\|_F^2 - \frac{1}{2}\|\mathcal{P}_T Q\|_F^2\right) \\ &= \frac{p_0}{2}\|\mathcal{P}_T Q\|_F^2.\end{aligned}$$

■

Proof (Proof of Theorem 4) Recall that we write that an event occurs with high probability if it holds with probability at least $1 - \mathcal{O}(n^{-10})$. Due to the asymptotic nature of Theorem 4, we only require the conditions of Proposition 9 to hold asymptotically with large probability. By Lemma A.3 of Candès et al. (2011), $\|\mathcal{P}_T\mathcal{P}_{\Gamma^\perp}Q\|_F \leq n\|\mathcal{P}_{T^\perp}\mathcal{P}_{\Gamma^\perp}Q\|_F$ for all Q, with high probability. By Lemma 10 and Theorem 2.6 of Candès et al. (2011) (see also Candès and Recht, 2009, Theorem 4.1), $\|\mathcal{P}_{\Omega_{\text{obs}}}\mathcal{P}_T Q\|_F^2 \geq \frac{p_0}{2}\|\mathcal{P}_T Q\|_F^2$ for all Q, with high probability. Further, by Candès and Recht (2009), $\|\mathcal{P}_T\mathcal{P}_\Omega\|^2 \leq p_0/8$ occurs with high probability. Candès et al. (2011, pp. 33-35) show that there exist dual certificates (W, F) obeying (19) with high probability. For sufficiently large n, the conditions of λ and p_0 in Proposition 9 are fulfilled. Therefore, Theorem 4 follows from Proposition 9. ■

References

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

James Bennett and Stan Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35, 2007.

Jian-Feng Cai, Emmanuel J Candés, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

Emmanuel J Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):Article 11, 2011.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2): 572–596, 2011.

Yudong Chen, Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust matrix completion and corrupted columns. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 873–880, 2011.

David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

T. Hastie, R. Mazumder, J. Lee, and R Zadeh. Matrix completion and low-rank SVD via fast alternating least squares. Unpublished manuscript, 2014.

Peter J Huber and Elvezio M Ronchetti. *Robust Statistics*, volume 693. John Wiley & Sons, New Jersey, second edition, 2011.

David R Hunter and Kenneth Lange. A tutorial on mm algorithms. *The American Statistician*, 58(1):30–37, 2004.

Juha Karhunen. Robust pca methods for complete and missing data. *Neural Network World*, 21(5):357, 2011.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(1):2057–2078, 2010a.

Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010b.

Vladimir Koltchinskii, Karim Lounici, Alexandre B Tsybakov, et al. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.

Kenneth Lange. *Numerical Analysis for Statisticians*. Springer, New York, 2010.

Kenneth Lange, David R Hunter, and Ilsoon Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.

Jaakko Luttinen, Alexander Ilin, and Juha Karhunen. Bayesian robust pca of incomplete data. *Neural processing letters*, 36(2):189–202, 2012.

Shiqian Ma, Donald Goldfarb, and Lifeng Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2):321–353, 2011.

Goran Marjanovic and Victor Solo. On l_q optimization and matrix completion. *IEEE Transactions on Signal Processing*, 60(11):5714–5724, 2012.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11: 2287–2322, 2010.

Andrea Montanari and Sewoong Oh. On positioning via distributed matrix completion. In *Sensor Array and Multichannel Signal Processing Workshop (SAM), 2010 IEEE*, pages 197–200, 2010.

Yurii Nesterov. Gradient methods for minimizing composite objective function. Technical report, CORE, 2007.

Hee-Seok Oh, Douglas W. Nychka, and Thomas C. M. Lee. The role of pseudo data for robust smoothing with application to wavelet regression. *Biometrika*, 94(4):893–904, 2007.

Benjamin Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 12:3413–3430, 2011.

Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719, 2005.

Yiyuan She and Art B Owen. Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, 106(494):626–639, 2011.

Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *ICML*, volume 3, pages 720–727, 2003.

Kilian Q Weinberger and Lawrence K Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.

Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candes, and Yi Ma. Stable principal component pursuit. In *2010 IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 1518–1522, 2010.

Journal of Machine Learning Research 18 (2017) 1-29 Submitted 5/17; Revised 11/17; Published 12/17

Faithfulness of Probability Distributions and Graphs

Kayvan Sadeghi K.SADEGHI@STATSLAB.CAM.AC.UK
Statistical Laboratory
University of Cambridge
Centre for Mathematical Sciences, Wilberforce Road
Cambridge, CB3 0WB, United Kingdom

Editor: Peter Spirtes

Abstract

A main question in graphical models and causal inference is whether, given a probability distribution P (which is usually an underlying distribution of data), there is a graph (or graphs) to which P is faithful. The main goal of this paper is to provide a theoretical answer to this problem. We work with general independence models, which contain probabilistic independence models as a special case. We exploit a generalization of ordering, called preordering, of the nodes of (mixed) graphs. This allows us to provide sufficient conditions for a given independence model to be Markov to a graph with the minimum possible number of edges, and more importantly, necessary and sufficient conditions for a given probability distribution to be faithful to a graph. We present our results for the general case of mixed graphs, but specialize the definitions and results to the better-known subclasses of undirected (concentration) and bidirected (covariance) graphs as well as directed acyclic graphs.

Keywords: causal discovery, compositional graphoid, directed acyclic graph, faithfulness, graphical model selection, independence model, Markov property, mixed graph, structural learning

1. Introduction

Graphs have been used in graphical models in order to capture the conditional independence structure of probability distributions. Generally speaking, nodes of the graph correspond to random variables and missing edges to conditional independencies (Lauritzen, 1996). The connection between graphs and probability distributions is usually established in the literature by the concept of Markov property (Clifford, 1990; Pearl, 1988; Studený, 1989), which ensures that if there is a specific type of separation between node subsets A and B of the graph "given the node subset C" then random vectors X_A and X_B are conditionally independent given the random vector X_C in the probability distribution. However, the "ultimate" connection between probability distributions and graphs requires the other direction to hold, namely for every conditional independence in the probability distribution to correspond to a separation in the graph. This connection has been called faithfulness of the probability distribution and the graph in Spirtes et al. (2000), and the graph has been called the perfect map of such a distribution in Pearl (1988).

However, "given a probability distribution P whether there is a graph (or graphs) to which P is faithful" is an open problem, and consequently so is the problem of finding these graphs. This problem can be raised for any type of graph existing in the literature of graphical models ranging from different types of mixed graphs with three types of edges (Richardson and Spirtes, 2002; Wermuth, 2011; Sadeghi, 2016) and different types of chain graphs (Lauritzen and Wermuth, 1989; Frydenberg, 1990; Andersson et al., 2001; Cox and Wermuth, 1993; Wermuth, 2011) to better-known classes of undirected (concentration) (Darroch et al., 1980) and bidirected (covariance) (Kauermann, 1996) graphs as well as directed acyclic graphs (Kiiveri et al., 1984; Pearl, 1988). Our goal is to provide an answer to this problem. A similar problem of "given a graph whether there is a family of distributions faithful to it" has been answered for very specific types of graphs and specific types of distributions; for example, for Gaussian distributions and undirected graphs in Lněnička and Matúš (2007), DAGs in Pearl (1988); Geiger et al. (1990); Spirtes et al. (2000), ancestral graphs in Richardson and Spirtes (2002), and LWF chain graphs in Peña (2011); and discrete distributions and DAGs in Geiger et al. (1990); Meek (1995) and LWF chain graphs in Peña (2009).

The concept of faithfulness was originally defined for the purpose of causal inference (Spirtes et al., 2000; Pearl, 1988), and the theory developed in this paper can be interpreted in causal language. A main approach to causal inference is based on graphical representations of causal structures, usually represented by *causal graphs* that are directed acyclic with nodes as random variables (that is a Bayesian network). Causal graphs are assumed to capture the true causal structure; see, for example, Neapolitan (2004). A main assumption made is called the *causal faithfulness condition* (CFC) stating that the true probability distribution is faithful to the true causal graph. Although the distribution being faithful to the "true underlying causal graph" is a much stronger assumption, it necessarily means that it must be faithful to "some graph" in order for one to be able to use graphical methods for causal inference. For an extensive discussion on the CFC, see Zhang and Spirtes (2008), and for related philosophical discussions, see, for example, Woodward (1998); Steel (2006). Although the results in this paper cover those of causal Bayesian networks, we have developed our theory to a much more general case of simultaneous representation of "direct effects", "confounding", and "non-causal symmetric dependence structures"; see, for example, Pearl (2009).

In this paper, we work with the general independence model $\mathcal{J}$. Independence models $\mathcal{J}(P)$ induced by probability distributions P are a special case. We provide necessary and sufficient conditions for $\mathcal{J}$ to be graphical, that is to be faithful to a graph. In short, as proved in Theorem 17, $\mathcal{J}$ is graphical if and only if it satisfies the so-called compositional graphoid axioms as well as singleton-transitivity, and what we call ordered upward- and downward-stability.

As apparent from their names, ordered upward- and downward-stability depend on a generalization of ordering of variables, and consequently the nodes of the graph (called preordering). We provide our results for the most general case of mixed graphs, which contains almost all classes of graphs used in graphical models as subclasses with the exception of the chain graphs with the AMP Markov property (Andersson et al., 2001) and its generalizations (Peña, 2014). However, based on the preordering w.r.t. which ordered upward- and downward-stability are satisfied, one can deduce to what type (or types) of graph P is faithful.

©2017 Kayvan Sadeghi.

License: CC-BY 4.0, see https://creativecommons.org/licenses/by/4.0/. Attribution requirements are provided at http://jmlr.org/papers/v18/17-275.html.

Since many of the subclasses of mixed graphs are not as well-known or used as some simpler classes of graphs, we provide a specialization of definitions and the main results for undirected and bidirected graphs as well as directed acyclic graphs at the end of the paper. We advise the readers that are only familiar with or interested in these simpler subclasses to skip the general definitions and results for mixed graphs, and focus on the specialization.

The structure of the paper is as follows: In the next section, we provide the basic definitions for graphs as well as different classes of graphs in graphical models, and independence models. In Section 3, we define and provide basic results for Markovness and faithfulness of independence models and graphs. In Section 4, we define and exploit the preordering for (so-called anterial) graphs, and then we define the concept of upward- and downward-stability. In Section 5, we provide sufficient conditions for an independence model to be so-called minimally Markov to a graph, and then we provide necessary and sufficient conditions for an independence model to be faithful to a graph. In Section 6, we specialize the definitions and results to the classes of undirected and bidirected graphs as well as directed acyclic graphs. In Section 7, we show how other results in the literature concerning faithfulness of certain probability distributions and certain classes of graphs are corollaries of our results. This also provides a nice set of examples for the theory presented in the paper. In Section 8, we end the paper with a short summary, a discussion on the statistical implications of the results, and an outline of future work.

2. Definitions and Concepts

In this section, we provide the basic definitions and concepts needed for the paper.

2.1 Graph Theoretical Definitions and Concepts

A *graph* G is a triple consisting of a *node* set or *vertex* set V, an *edge* set E, and a relation that with each edge associates two nodes (not necessarily distinct), called its *endpoints*. When nodes i and j are the endpoints of an edge, they are *adjacent*. We call a node adjacent to the node i, a *neighbour* of i, and denote the set of all neighbours of i by $\mathrm{ne}(i)$. We say that an edge is *between* its two endpoints. We usually refer to a graph as an ordered pair $G = (V, E)$. Graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are called *equal* if $(V_1, E_1) = (V_2, E_2)$. In this case we write $G_1 = G_2$.

Notice that our graphs are *labeled*, that is every node is considered a different object. Hence, for example, graph $i \text{——} j \text{——} k$ is not equal to $j \text{——} i \text{——} k$.

A *loop* is an edge with the same endpoints. Here we only discuss graphs without loops. *Multiple edges* are edges with the same pair of endpoints. A *simple graph* has neither loops nor multiple edges.

Graphs in this paper are so-called *mixed graphs*, which contain three types of edges: *lines* ($i \text{——} j$), *arrows* ($i \text{——>} j$), and *arcs* ($i \text{<—>} j$), where in the two latter cases, we say that there is an *arrowhead* at j. If we remove all arrowheads from edges of the graph G, the obtained undirected graph is called the *skeleton* of G. The simple graph whose edge ij indicates whether there is an edge (or multiple edges) between i and j in G is called the *adjacency graph* of G. It is clear that if a graph is simple then its adjacency graph is the same as its skeleton. In this paper, only skeleton of anterial graphs or its subclasses are used, which, as will be defined later, are simple graphs; hence, we denote the simple skeleton by $\mathrm{sk}(G)$. We say that we *direct* the edges of a skeleton by putting arrowheads at the edges in order to obtain mixed graphs.

A *subgraph* of a graph G_1 is graph G_2 such that $V(G_2) \subseteq V(G_1)$ and $E(G_2) \subseteq E(G_1)$ and the assignment of endpoints to edges in G_2 is the same as in G_1. We define a subgraph *induced by edges* $A \subseteq E$ of $G = (V, E)$ to be a subgraph that contains V as the node set and all and only edges in A as the edge set.

A *walk* is a list $\langle v_0, e_1, v_1, \ldots, e_k, v_k \rangle$ of nodes and edges such that for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i. A *path* is a walk with no repeated node or edge. A maximal set of nodes in a graph whose members are connected by some paths constitutes a *connected component* of the graph. A *cycle* is a walk with no repeated nodes or edges except for $v_0 = v_k$.

We say a walk is *between* the first and the last nodes of the list in G. We call the first and the last nodes *endpoints* of the walk and all other nodes *inner nodes*.

A *subwalk* of a walk $\omega = \langle i_0, e_1, i_1, \ldots, e_n, i_n \rangle$ is a walk $\langle i_r, e_{r+1}, i_{r+1}, \ldots, e_p, i_p \rangle$ that is a subsequence of ω between two occurrences of nodes (i_r, i_p, $0 \leq r \leq p \leq n$). If a subwalk forms a path then it is called a *subpath* of ω.

A walk $\omega = \langle i = i_0, i_1, \ldots, i_n = j \rangle$ is *directed* from i to j if all edges $i_k i_{k+1}$, $0 \leq k \leq n-1$, are arrows pointing from i_k to i_{k+1}. If there is a directed walk from i to j then i is an *ancestor* of j and j is a *descendant* of i. We denote the set of ancestors of j by $\mathrm{an}(j)$.

A walk $\omega = \langle i = i_0, i_1, \ldots, i_n = j \rangle$ is *semi-directed* from i to j if it has at least one arrow, no arcs, and every arrow $i_k i_{k+1}$ is pointing from i_k to i_{k+1}. A walk between i and j is *anterior* from i to j if it is semi-directed from i to j or if it is undirected. If there is an anterior walk from i to j then we also say that i is an *anterior* of j. We use the notation $\mathrm{ant}(j)$ for the set of all anteriors of j. For a set A, we define $\mathrm{ant}(A) = \bigcup_{j \in A} \mathrm{ant}(j) \setminus A$.

Notice that, unlike in most places in the literature (for example Richardson and Spirtes 2002), we use walks instead of paths to define ancestors and anteriors. Because of this and the fact that ancestral graphs have no arrowheads pointing to lines, our definition of anterior extends the notion of anterior for ancestral graphs in Richardson and Spirtes (2002) with the modification that in this paper, a node is not an anterior of itself. Using walks instead of paths is immaterial, as shown in Lauritzen and Sadeghi (2017).

A *section* ρ of a walk is a maximal subwalk consisting only of lines, meaning that there is no other subwalk that only consists of lines and includes ρ. Thus, any walk decomposes uniquely into sections; these are not necessarily edge-disjoint and sections may also be single nodes. A section ρ on a walk ω is called a *collider section* if one of the following walks is a subwalk of ω: $i \text{——>} \rho \text{<——} j$, $i \text{<—>} \rho \text{<——} j$, $i \text{<—>} \rho \text{<—>} j$. All other sections on ω are called *non-collider* sections. Notice that a section may be a collider on one walk and a non-collider on another, but we may speak of collider or non-collider sections without mentioning the relevant walk when this is apparent from the context.

2.2 Different Classes of Graphs

All the subclasses of graphs included here are *acyclic* in the sense that they do not contain semi-directed cycles. The most general class of graphs discussed here is the class of *chain mixed graphs* (CMGs) (Sadeghi, 2016) that contains all mixed graphs without semi-directed cycles. They may have multiple edges containing an arc and a line or an arc and an

arrow but not a combination of an arrow and a line or arrows in opposite directions, as such combinations would constitute semi-directed cycles. In this paper, in using the term "graph" we mean a CMG unless otherwise stated.

A general class of graphs that plays an important role in this paper is the class of *anterial graphs* (AnGs) (Sadeghi, 2016). AnGs are CMGs in which an endpoint of an arc cannot be an anterior of the other endpoint.

CMGs include *summary graphs* (SGs) (Wermuth, 2011) and *acyclic directed mixed graphs* (ADMGs) (Richardson, 2003) as a subclass, but not include *AMP chain graphs* (Andersson et al., 2001), and AnGs include *ancestral graphs* (AGs) (Richardson and Spirtes, 2002) but not SGs or ADMGs. These have typically been introduced to describe independence structures obtained by marginalization and conditioning in DAG independence models; see for example Sadeghi (2013).

AnGs also contain *undirected and bidirected chain graphs* (CGs). A chain graph is an acyclic graph so that if we remove all arrows, all connected components of the resulting graph—called *chain components*—contain one type of edge only. If all chain components contain lines, the chain graph is an undirected chain graph (UCG) (known as *LWF chain graphs*); if all chain components contain arcs it is a bidirected chain graph (BCG) (known as *multivariate regression chain graphs*). AnGs also contain *Regression graphs* (Wermuth and Sadeghi, 2012), which are chain graphs consisting of lines and arcs (although dashed undirected edges have mostly been used instead of arcs in the literature), where there is no arrowhead pointing to lines.

These also contain graphs with only one type of edge; namely *undirected graphs* (UGs), containing only lines; *bidirected graphs* (BGs), containing only bidirected edges; and *directed acyclic graphs* (DAGs), containing only arrows and being acyclic. Clearly, a graph without arrows has no semi-directed cycles, and a semi-directed cycle in a graph with only arrows is a directed cycle. Cox and Wermuth (1993); Kauermann (1996); Wermuth and Cox (1998); Drton and Richardson (2008) used the terms *concentration graphs* and *covariance graphs* for UGs and BGs, referring to their independence interpretation associated with covariance and concentration matrices for Gaussian graphical models. DAGs have been particularly useful to describe causal Markov relations; see for example Kiiveri et al. (1984); Pearl (1988); Lauritzen and Spiegelhalter (1988); Geiger et al. (1990); Spirtes et al. (2000).

For an extensive discussion on the subclasses of acyclic graphs and their relationships and hierarchy, see Lauritzen and Sadeghi (2017).

2.3 Independence Models and Their Properties

An *independence model* $\mathcal{J}$ over a finite set V is a set of triples $\langle X, Y \,|\, Z\rangle$ (called *independence statements*), where X, Y, and Z are disjoint subsets of V; Z may be empty, but $\langle \varnothing, Y \,|\, Z\rangle$ and $\langle X, \varnothing \,|\, Z\rangle$ are always included in $\mathcal{J}$. The independence statement $\langle X, Y \,|\, Z\rangle$ is read as "X is independent of Y given Z". Independence models may in general have a probabilistic interpretation, but not necessarily. Similarly, not all independence models can be easily represented by graphs. For further discussion on general independence models, see Studený (2005).

In order to define probabilistic independence models, consider a set V and a collection of random variables $\{X_\alpha\}_{\alpha\in V}$ with state spaces $\mathcal{X}_\alpha, \alpha \in V$ and joint distribution P. We let

$X_A = \{X_v\}_{v\in A}$ etc. for each subset A of V. For disjoint subsets A, B, and C of V we use the short notation $A \perp\!\!\!\perp B \,|\, C$ to denote that X_A is *conditionally independent of* X_B *given* X_C (Dawid, 1979; Lauritzen, 1996), that is that for any measurable $\Omega \subseteq \mathcal{X}_A$ and P-almost all x_B and x_C,

$$P(X_A \in \Omega \,|\, X_B = x_B, X_C = x_C) = P(X_A \in \Omega \,|\, X_C = x_C).$$

We can now induce an independence model $\mathcal{J}(P)$ by letting

$$\langle A, B \,|\, C\rangle \in \mathcal{J}(P) \text{ if and only if } A \perp\!\!\!\perp B \,|\, C \text{ w.r.t. } P.$$

Similarly we use the notation $A \not\perp\!\!\!\perp B \,|\, C$ for $\langle A, B \,|\, C\rangle \notin \mathcal{J}(P)$.

In order to use graphs to represent independence models, the notion of *separation* in a graph is fundamental. For three disjoint subsets A, B, and C, we use the notation $A \perp B \,|\, C$ if A and B are *separated* given C, and $A \not\perp B \,|\, C$ for A and B not separated given C.

The separation can be unified under one definition for all graphs by using walks instead of paths: A walk π is *connecting* given C if every collider section of π has a node in C and all non-collider sections are disjoint from C. For pairwise disjoint subsets (A, B, C), $A \perp B \,|\, C$ if there are no connecting walks between A and B given C. This is in wording the same as the definition in Studený and Bouckaert (1998) for undirected (LWF) chain graphs (although it is in fact a generalization as collider sections are generalized), and a generalization of the *m-separation* used with different wordings in Richardson and Spirtes (2002); Wermuth (2011); Wermuth and Sadeghi (2012) for AGs and SGs, and of the *d-separation* of Pearl (1988). For UGs, the notion has a direct intuitive meaning, so that $A \perp B \,|\, C$ if all paths from A to B pass through C.

If A, B, or C has only one member $\{i\}$, $\{j\}$, or $\{k\}$, for better readability, we write $\langle i, j \,|\, k\rangle \in \mathcal{J}$ instead of $\langle\{i\}, \{j\} \,|\, \{k\}\rangle \in \mathcal{J}$; and similarly for $i \perp j \,|\, k$ and $i \perp\!\!\!\perp j \,|\, k$. We also write $A \perp B$ when $C = \varnothing$; and similarly $A \perp\!\!\!\perp B$.

A graph G induces an independence model $\mathcal{J}(G)$ by separation, letting

$$\langle A, B \,|\, C\rangle \in \mathcal{J}(G) \iff A \perp B \,|\, C \text{ in } G.$$

An independence model $\mathcal{J}$ over a set V is a *semi-graphoid* if it satisfies the four following properties for disjoint subsets A, B, C, and D of V:

1. $\langle A, B \,|\, C\rangle \in \mathcal{J}$ if and only if $\langle B, A \,|\, C\rangle \in \mathcal{J}$ (*symmetry*);
2. if $\langle A, B \cup D \,|\, C\rangle \in \mathcal{J}$ then $\langle A, B \,|\, C\rangle \in \mathcal{J}$ and $\langle A, D \,|\, C\rangle \in \mathcal{J}$ (*decomposition*);
3. if $\langle A, B \cup D \,|\, C\rangle \in \mathcal{J}$ then $\langle A, B \,|\, C \cup D\rangle \in \mathcal{J}$ and $\langle A, D \,|\, C \cup B\rangle \in \mathcal{J}$ (*weak union*);
4. if $\langle A, B \,|\, C \cup D\rangle \in \mathcal{J}$ and $\langle A, D \,|\, C\rangle \in \mathcal{J}$ then $\langle A, B \cup D \,|\, C\rangle \in \mathcal{J}$ (*contraction*).

Notice that the reverse implication of contraction clearly holds by decomposition and weak union. A semi-graphoid for which the reverse implication of the weak union property holds is said to be a *graphoid*; that is, it also satisfies

5. if $\langle A, B \,|\, C \cup D\rangle \in \mathcal{J}$ and $\langle A, D \,|\, C \cup B\rangle \in \mathcal{J}$ then $\langle A, B \cup D \,|\, C\rangle \in \mathcal{J}$ (*intersection*).

Furthermore, a graphoid or semi-graphoid for which the reverse implication of the decomposition property holds is said to be *compositional*, that is, it also satisfies

6. if $\langle A, B \,|\, C\rangle \in \mathcal{J}$ and $\langle A, D \,|\, C\rangle \in \mathcal{J}$ then $\langle A, B \cup D \,|\, C\rangle \in \mathcal{J}$ (*composition*).

Separation in graphs satisfies all these properties; see Theorem 1 in Lauritzen and Sadeghi (2017):

Proposition 1 *For any graph G, the independence model $\mathcal{J}(G)$ is a compositional graphoid.*

On the other hand, probabilistic independence models are always semi-graphoids (Pearl, 1988), whereas the converse is not necessarily true; see Studený (1989). If, for example, P has strictly positive density, the induced independence model is always a graphoid; see, for example, Proposition 3.1 in Lauritzen (1996). See also Peters (2015) for a necessary and sufficient condition for the intersection property to hold. If the distribution P is a regular multivariate Gaussian distribution, $\mathcal{J}(P)$ is a compositional graphoid; for example see Studený (2005). Probabilistic independence models with positive densities are not in general compositional; this only holds for special types of multivariate distributions such as, for example, Gaussian distributions and the symmetric binary distributions used in Wermuth et al. (2009).

Another important property that is satisfied by separation in all graphs, but not necessarily for probabilistic independence models, is *singleton-transitivity* (also called *weak transitivity* in Pearl (1988), where it is shown that for Gaussian and binary distributions P, $\mathcal{J}(P)$ always satisfies it). For i, j, and k, single elements in V,

7. if $\langle i, j \,|\, C\rangle \in \mathcal{J}$ and $\langle i, j \,|\, C \cup \{k\}\rangle \in \mathcal{J}$ then $\langle i, k \,|\, C\rangle \in \mathcal{J}$ or $\langle j, k \,|\, C\rangle \in \mathcal{J}$ (singleton-transitivity).

A singleton-transitive compositional graphoid is equivalent to what is called a *Gaussoid* in Lněnička and Matúš (2007) with a rather different axiomatization. The name reminds one that these are the axioms satisfied by the independence model of a regular Gaussian distribution (Pearl, 1988).

Proposition 2 *For a graph G, $\mathcal{J}(G)$ satisfies singleton-transitivity.*

Proof If there is a walk between i and k given C and a walk between j and k given C then by connecting these two walks, we obtain a walk π between i and j. Except the section ρ that contains k, all collider sections on π have a node in C and all non-collider sections are outside C. Hence depending on ρ being a collider or non-collider, π is connecting given $C \cup \{k\}$ or C respectively. ■

3. Markov and Faithful Independence Models

In this section, we define and discuss the concepts of Markovness and faithfulness for probability distributions, independence models, and graphs.

3.1 Markov Properties

For a graph $G = (V, E)$, an independence model $\mathcal{J}$ defined over V satisfies the *global Markov property* w.r.t. G, or is simply *Markov* to G, if for disjoint subsets A, B, and C of V it holds that

$$A \perp B \,|\, C \implies \langle A, B \,|\, C\rangle \in \mathcal{J},$$

or equivalently $\mathcal{J}(G) \subseteq \mathcal{J}$. In particular, the graphical independence $\mathcal{J}(G)$ is trivially Markov to its graph G.

For a probability distribution P, we simply say P is *Markov to* a graph G if $\mathcal{J}(P)$ is Markov to G, that is if $\mathcal{J}(G) \subseteq \mathcal{J}(P)$. Notice that every independence model over V is Markov to the complete graph with the node set V.

For a graph $G = (V, E)$, an independence model $\mathcal{J}$ defined over V satisfies a *pairwise Markov property* w.r.t. a graph G, or is simply *pairwise Markov to* G, if for every non-adjacent pair of nodes i, j, it holds that $\langle i, j \,|\, C(i,j)\rangle \in \mathcal{J}$, for some $C(i,j)$, where $C(i,j)$ is *the conditioning set* of the pairwise Markov property and does not include i and j.

The independence model $\mathcal{J}(G)$ only satisfies pairwise Markov properties w.r.t. what is called *maximal graphs*, which are graphs where the lack of an edge between i and j corresponds to a conditional separation statement for i and j. For a probability distribution P, we say P is *pairwise Markov to* a graph G if $\mathcal{J}(P)$ is pairwise Markov to G.

For maximal graphs, a pairwise Markov property is defined by letting $C(i,j) = \text{ant}(i) \cup \text{ant}(j) \setminus \{i, j\}$ (Lauritzen and Sadeghi, 2017), which we henceforth use as "the" pairwise Markov property. The conditioning set of the pairwise Markov property simplifies for DAGs, $C(i,j) = an(i) \cup \text{an}(j) \setminus \{i, j\}$ (Richardson and Spirtes, 2002; Sadeghi and Lauritzen, 2014); for BGs, $C(i,j) = \varnothing$ (as defined in Wermuth and Cox 1998); and for connected UGs, $C(i,j) = V \setminus \{i, j\}$ (as defined in Lauritzen 1996).

This pairwise Markov property, is in fact equivalent to the global Markov property under compositional graphoids; see Theorem 4 of Lauritzen and Sadeghi (2017):

Proposition 3 *Let G be a maximal graph. If the independence model $\mathcal{J}$ is a compositional graphoid, then $\mathcal{J}$ is pairwise Markov to G if and only if $\mathcal{J}$ is Markov to G.*

In particular, for UGs, the equivalence of the global and pairwise Markov properties holds under graphoids, and for BGs under compositional semi-graphoids.

Two undirected or two bidirected graphs G and H, where $G \neq H$, induce different independence models; that is $\mathcal{J}(G) \neq \mathcal{J}(H)$. This is not necessarily true for larger subclasses. We call two graphs G and H such that $\mathcal{J}(G) = \mathcal{J}(H)$ *Markov equivalent*. Conditions for Markov equivalence for most subclasses of graphs are known; see Verma and Pearl (1990); Ali et al. (2009); Wermuth and Sadeghi (2012). Notice that two Markov equivalent maximal graphs have the same skeleton.

3.2 Faithfulness and Minimal Markovness

We say that an independence model $\mathcal{J}$ and a probability distribution P are *faithful* if $\mathcal{J} = \mathcal{J}(P)$. Similarly, we say that $\mathcal{J}$ and a graph G are faithful if $\mathcal{J} = \mathcal{J}(G)$. We also say that P and G are faithful if $\mathcal{J}(P) = \mathcal{J}(G)$. If P and G are faithful then we may sometimes also say that P is *faithful to* G or vice versa, although in principle faithfulness is a symmetric relation; the same holds for saying that $\mathcal{J}$ is faithful to G or P. Notice that

we are extending the definition of faithfulness to include the relation between independence models and graphs as well as independence models and probability distributions rather than only that between graphs and probability distributions as originally used in Spirtes et al. (2000).

Thus, if $\mathcal{J}$ and G are faithful then $\mathcal{J}$ is Markov to G, but it also requires every independence statement to correspond to a separation in G. We say that $\mathcal{J}$ or G is *probabilistic* if there is a distribution P that is faithful to $\mathcal{J}$ or G respectively. If there is a graph that is faithful to $\mathcal{J}$ or P then we say that $\mathcal{J}$ or P is *graphical*.

Our main goal is to characterize graphical probability distributions, and in addition, if existent, to provide graphs that are faithful to a given P. We solve this problem for the general case of independence models $\mathcal{J}$.

For a given independence model $\mathcal{J}$, we define the *skeleton* of $\mathcal{J}$, denoted by $\mathrm{sk}(\mathcal{J})$, to be the undirected graph that is obtained from $\mathcal{J}$ as follows: we define the node set of $\mathrm{sk}(\mathcal{J})$ to be V, and for every pair of nodes i, j, we check whether $\langle i, j \,|\, C\rangle \in \mathcal{J}$ holds for some $C \subseteq V \setminus \{i, j\}$; if it does not then we draw an edge between i and j. One can similarly define $\mathrm{sk}(P)$ for a probability distribution P by checking whether $i \perp\!\!\!\perp j \,|\, C$. This is the same as the graph obtained by the first step of the SGS algorithm (Glymour et al., 1987). Notice that this is not necessarily the same as the undirected graph obtained by the pairwise Markov property for UGs; see Section 6.1.

For UGs and BGs, the skeleton of $\mathcal{J}$ uniquely determines the graph, whereas for other subclasses, there are several graphs with the same skeleton $\mathrm{sk}(\mathcal{J})$. As will be seen, the preordering, defined in the next section, enables us to direct the edges of the skeleton.

Proposition 4 *If an independence model $\mathcal{J}$ is Markov to a graph G then* $\mathrm{sk}(\mathcal{J})$ *is a subgraph of* $\mathrm{sk}(G)$.

Proof Suppose that there is no edge between i and j in $\mathrm{sk}(G)$. Since the global Markov property w.r.t. G implies the pairwise Markov property, it holds that $\langle i, j \,|\, C(i, j)\rangle \in \mathcal{J}$, which implies i is not adjacent to j in $\mathrm{sk}(\mathcal{J})$. ■

Hence, if $\mathcal{J}$ is Markov to a graph G such that $\mathrm{sk}(G) = \mathrm{sk}(\mathcal{J})$ then G has the fewest number of edges among those to which $\mathcal{J}$ is Markov. We say that $\mathcal{J}$ is *minimally Markov* to a graph G if $\mathcal{J}$ is Markov to G and $\mathrm{sk}(G) = \mathrm{sk}(\mathcal{J})$. The same can be defined for probability distributions, and has been used in the literature under the name of *(causal) minimality assumption* (Pearl, 2009; Spirtes et al., 2000; Zhang and Spirtes, 2008; Neapolitan, 2004). Minimally Markov independence models to a graph are important since only these can also be faithful to the graph:

Proposition 5 *If an independence model $\mathcal{J}$ and a graph G are faithful then $\mathcal{J}$ is minimally Markov to G.*

Proof Since $\mathcal{J}$ is Markov to G, we need to prove that $\mathrm{sk}(G) = \mathrm{sk}(\mathcal{J})$. By Proposition 4, $\mathrm{sk}(\mathcal{J})$ is a subgraph of $\mathrm{sk}(G)$. Now, suppose that there is no edge between i and j in $\mathrm{sk}(\mathcal{J})$. By the construction of $\mathrm{sk}(\mathcal{J})$, it holds that $\langle i, j \,|\, C\rangle \in \mathcal{J}$ for some C. Since $\mathcal{J}$ and G are faithful, $i \perp j \,|\, C$ in G. This implies that i is not adjacent to j in G. ■

Hence, we need to discuss conditions under which $\mathcal{J}$ is minimally Markov to a graph as well as conditions for faithfulness of minimally Markov independence models and graphs. These will be presented in Section 5.

4. Preordering in Graphs and Ordered Stabilities

In this section, we first define the preordering for sets and its validity for nodes of the graphs, and then use preordering to define some new properties of conditional independence.

4.1 Preordering of the Nodes of a Graph

Over a set V, a *partial order* is a binary relation $\leq$ that satisfies the following properties for all members $a, b, c \in V$:

- $a \leq a$ (*reflexivity*);
- if $a \leq b$ and $b \leq a$ then $a = b$ (*antisymmetry*);
- if $a \leq b$ and $b \leq c$ then $a \leq c$ (*transitivity*).

If $a \leq b$ or $b \leq a$ then a and b are *comparable*; otherwise they are *incomparable*. A partial order under which every pair of elements is comparable is called a *total order* or a *linear order*.

A *preorder* $\lesssim$ over V, on the other hand, is a binary relation that is only reflexive and transitive. Given a preorder $\lesssim$ over V, one may define an equivalence relation $\sim$ on V such that $a \sim b$ if and only if $a \lesssim b$ and $b \lesssim a$. This explains the use of the notation $\lesssim$. It then holds that

- if $a \lesssim b$ and $b \sim c$ then $a \lesssim c$; and
- if $a \lesssim b$ and $a \sim c$ then $c \lesssim b$.

We also use the notation $a < b$ indicating that $a \lesssim b$ and $a \not\sim b$.

We first provide the following result; see Section 5.2.1 of Schröder (2003):

Proposition 6 *Let $\lesssim$ be a preorder over the set V, and $\sim$ the equivalence relation on V as defined above. Let also Q be the set of all equivalence classes of V w.r.t. $\sim$. Then the relation $\leq$ defined over Q by $\big[[a] \leq [b] \iff a \lesssim b\big]$ is a partial order over Q.*

We say that a graph $G = (V, E)$ admits a *valid preorder* $\lesssim$ if, for nodes i and j of G, the following holds:

- if $i \text{ —— } j$ then $i \sim j$;
- if $i \longrightarrow j$ then $j < i$;
- if $i \dashleftarrow\!\dashrightarrow j$ then i and j are incomparable.

The global interpretation of a valid preorder on graphs is as follows:

Proposition 7 *Let $\lesssim$ be a valid preorder for a graph G. It then holds for nodes i and j of G that if $i \in \mathrm{ant}(j)$ then $j \lesssim i$; in particular,*

1. *if there is a semi-directed path from i to j then $j < i$; and*
2. *if i and j are connected by a path consisting only of lines then $i \sim j$.*

Proof The proof follows from transitivity of both preorders and anterior paths. ■

Corollary 8 *Let $\lesssim$ be a valid preorder for a graph G and G_- its subgraph induced by lines. The equivalence classes of G based on $\lesssim$ are the connected components of G_-. In addition, $\leq$ defines a partial order over the connected components of G_- by [if $i \longrightarrow j$, $i \in \tau$ and $j \in \delta$ then $\tau > \delta$], for τ and δ two connected components of G_-.*

Proof The results follows from Propositions 7 and 6. ■

For example, consider the graph in Fig. 1. The graph admits a valid preorder with the preorder provided over the nodes, or equivalently over the connected components of G_- (with the abuse of the notation for numbering). The notation implies that two nodes with label 2 are in the same equivalence class, $2 > 1$ and $2' > 1'$, but a, b', and c'', $a, b, c \in \{1, 2\}$, are not comparable.

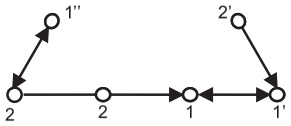


Figure 1: A preordered graph.

There may be many different preorders that are valid for a graph. If there is a valid preordering and we expect the other direction of Proposition 7 to hold then we obtain a unique preordering for the graph: Given a graph G, if $i \notin \text{ant}(j)$ and $j \notin \text{ant}(i)$ then set i and j to be incomparable. Otherwise, let $i \sim j$ when $i \multimap\!\!-\!\!\multimap j$, and $i > j$ when $i \longrightarrow j$. It is easy to see that this, in fact, is a preorder for the nodes of G. We call this preordering the *minimal preorder* for G since it gives the fewest possible comparable pairs of nodes. For example, the preordering in the graph of Fig. 1 is minimal. In this paper, we mostly deal with this type of preorderings for graphs. It is easy to observe the following:

Proposition 9 *If G is anterial then the minimal preorder for G is a valid preorder for G.*

In fact, in general, we have the following:

Proposition 10 *A graph admits a valid preorder if and only if it is anterial.*

Proof If a graph G admits a valid preorder then by Proposition 7, there cannot be a directed cycle, nor can there be an arc with one endpoint anterior of the other. The converse is Proposition 9. ■

Therefore, in addition to CMGs, SGs (and ADMGs) do not admit a valid preorder, but AGs do. However, notice that for every CMG, there exists a Markov equivalent AnG (see Sadeghi 2016); and the same for SGs and AG. Hence, the question of whether there is a CMG that is faithful to $\mathcal{J}$ is the same as whether there is an AnG that is faithful to $\mathcal{J}$; and the same holds for SG and AG.

As discussed above, there is a one-to-one correspondence between AnGs and the minimal preorder for the AnGs. When the skeleton of the concerned graph is known, we have the following trivial result:

Proposition 11 *Given the skeleton of a graph G and a preorder $\lesssim$ of its nodes, G can be constructed uniquely by directing the edges of* $\text{sk}(G)$. *In addition, $\lesssim$ is a valid preorder for G.*

Let $\mathcal{J}$ be defined over V. We introduce preordering in this paper in order to stress that, in principle, a preordering of members of V (which are possibly random variables) is defined irrespective of graphs to which they are Markov or faithful. However, for our purposes, knowing the skeleton of such graphs, one can directly discover the directions of the edges of the graph, which correspond to the minimal preordering for the graph, rather than working directly on the set V.

Suppose that there exists a preorder $\lesssim$ over V. Proposition 11 implies that we can direct the edges of $\text{sk}(\mathcal{J})$ based on $\lesssim$ in order to define a *dependence graph* $G(\mathcal{J}, \lesssim)$ induced by $\mathcal{J}$ and $\lesssim$. Notice that, by Proposition 10, $G(\mathcal{J}, \lesssim)$ is anterial. Similarly, for P, one can define $G(P, \lesssim) = G(\mathcal{J}(P), \lesssim)$.

As mentioned, we are only interested in preorderings that are minimal preorders of graphs. Given $\mathcal{J}$, we define a preordering $\lesssim$ to be *$\mathcal{J}$-compatible* if $\lesssim$ is the minimal preorder for $G(\mathcal{J}, \lesssim)$. Similarly, one can define a P-compatible preorder.

4.2 Ordered Upward- and Downward-stabilities

We now exploit the preodering for independence models in order to define two other properties in addition to the seven properties defined in Section 2.3 (namely singleton-transitive compositional graphoid axioms). We say that an independence model $\mathcal{J}$ over the set V respectively satisfies *ordered upward-* and *downward-stability* w.r.t. a preorder $\lesssim$ of V if the following hold:

8. if $\langle i, j \,|\, C\rangle \in \mathcal{J}$ then $\langle i, j \,|\, C \cup \{k\}\rangle \in \mathcal{J}$ for every $k \in V \setminus \{i, j\}$ such that $l \lesssim k$ for some $l \in \{i, j\}$ or $l \sim k$ for some $l \in C$ (ordered upward-stability);

9. if $\langle i, j \,|\, C\rangle \in \mathcal{J}$ then $\langle i, j \,|\, C \setminus \{k\}\rangle \in \mathcal{J}$ for every $k \in V \setminus \{i, j\}$ such that $l \not\lesssim k$ for every $l \in \{i, j\}$ and $l \not\sim k$ for every $l \in C \setminus \{k\}$ (ordered downward-stability).

Ordered upward-stability is a generalization of a modification of *upward stability*, defined in Fallat et al. (2017), and *strong union*, defined in Pearl and Paz (1985), for undirected graphs, with singletons instead of node subsets.

Proposition 12 *Let G be an AnG and $\lesssim$ the minimal preorder for G. Then $\mathcal{J}(G)$ satisfies ordered upward- and downward-stability w.r.t. $\lesssim$.*

Proof First we prove that $\mathcal{J}(G)$ satisfies ordered upward-stability w.r.t. the minimal preorder: If $k \in C$, the result is obvious, thus suppose that $k \notin C$. Suppose that there is a connecting walk π between i and j given $C \cup \{k\}$. If k is not on π then we are done; otherwise, k is on a collider section with no other node in C on π. Notice that $l \lesssim k$ means here that $k \in \text{ant}(l)$, and $l \sim k$ that k and l are connected by a path consisting of lines.

If $l \in C$ then denote the path of lines between k and l by ω. The walk consisting of the subwalk of π from i to k, ω, ω in the reverse direction (from l to k), and the subwalk of π from k to j is connecting given C. This is because on this walk k and l are in the same collider section with l in C.

Now denote an anterior path from k to $l \in \{i, j\}$ by ϖ. Without loss of generality we can assume that $l = j$. We now consider two cases: 1) If ϖ has no nodes in C then the walk containing the subwalk of π between i and k and ϖ is connecting given C between i and j since k and all nodes on ϖ are on non-collider sections on this walk. 2) If ϖ has a node in C then consider the one closest to k and call it h. Then the walk consisting of the subwalk of π from i to k, the subpath of ϖ from k to h, the reverse of the subpath from h to k, and the subwalk of π from k to j is connecting given C as proven as follows: 1) if there is an arrow on ϖ then k is on a non-collider section in both instances in which it appears on this walk, and h is on a collider section and in C. 2) if ϖ only consists of lines then k and h are in the same collider section with a node (h) in C.

Now we prove that $\mathcal{J}(G)$ satisfies ordered downward-stability: If $k \notin C$ then the result is obvious, thus suppose that $k \in C$. Suppose that there is a connecting walk π between i and j given $C \setminus \{k\}$. If k is not on π then we are done. Otherwise, first it is clear that since π is connecting given $C \setminus \{k\}$, k cannot be the only node on a collider section that is in C.

If k is on the same section as another member of C then π is clearly connecting given C. The only case that is left is when k is on a non-collider section on π; but this is impossible: there is no arrowhead at the section containing k from one side on π, say from the j side. By moving towards j from k, it can be implied that k is either an anterior of a node in a collider section that is in C, or an anterior of j, but both are impossible since the former implies $l < k$ for $l \in C$, and the latter implies $j \lesssim k$. ■

5. Characterization of Graphical Independence Models

In this section, we first provide sufficient conditions for minimal Markovness, and then necessary and sufficient conditions for faithfulness.

5.1 Sufficient Conditions for Minimal Markovness

Here we provide sufficient conditions for an independence model to be minimally Markov to a graph. First, we have the following trivial result:

Proposition 13 *If $\mathcal{J}$ is minimally Markov to an anterial graph G then $G = G(\mathcal{J}, \lesssim)$ for the minimal preorder $\lesssim$ for G.*

By Proposition 5, the above statement also holds when $\mathcal{J}$ and G are faithful.

Proposition 14 *Suppose that there exists a $\mathcal{J}$-compatible preorder $\lesssim$ over V w.r.t. which $\mathcal{J}$ satisfies ordered downward- and upward-stability. It then holds that $\mathcal{J}$ is pairwise Markov to $G(\mathcal{J}, \lesssim)$.*

Proof Suppose that there are non-adjacent nodes i and j in $G(\mathcal{J}, \lesssim)$. These are non-adjacent in $\mathrm{sk}(\mathcal{J})$ too. By definition of $\mathrm{sk}(\mathcal{J})$, it holds that $\langle i, j \,|\, C\rangle \in \mathcal{J}$ for some $C \subseteq V \setminus \{i, j\}$.

Since $\mathcal{J}$ satisfies ordered downward-stability and since the preorder is $\mathcal{J}$-compatible, as long as a node $k \in C$ is not an anterior of $\{i, j\}$ or there is no semi-directed path from it to $C \setminus \{k\}$, it can be removed from the conditioning set, that is we have that $\langle i, j \,|\, C \setminus k\rangle \in \mathcal{J}$. Since $G(\mathcal{J}, \lesssim)$ is acyclic, as long as $\mathrm{ant}(\{i, j\}) \subset C$, such a k exists. Now consider a node $k' \in C \setminus \{k\}$ that is not an anterior of $\{i, j\}$ or there is no semi-directed path from it to $C \setminus \{k, k'\}$ and apply downward-stability again. By an inductive argument we imply that $\langle i, j \,|\, C \cap \mathrm{ant}(\{i, j\})\rangle \in \mathcal{J}$. Now since P satisfies ordered upward-stability, nodes outside C that are in $\mathrm{ant}(\{i, j\})$ can be added to the conditioning set. Hence, it holds that $\langle i, j \,|\, \mathrm{ant}(\{i, j\}) \setminus \{i, j\}\rangle \in \mathcal{J}$. This completes the proof. ■

In principle, a compatible preorder w.r.t. which $\mathcal{J}$ satisfies ordered downward- and upward-stability can be found by going through all such preorderings. Finding such a preorder efficiently is a structural learning task. We believe we have an efficient way to do this when the skeleton of $\mathcal{J}$ has been learned, but this is beyond the scope of this paper.

Theorem 15 *Suppose, for an independence model $\mathcal{J}$ over V, that*

1. *$\mathcal{J}$ is a compositional graphoid; and*
2. *there exists a $\mathcal{J}$-compatible preorder $\lesssim$ over V w.r.t. which $\mathcal{J}$ satisfies ordered downward- and upward-stability.*

It then holds that $\mathcal{J}$ is minimally Markov to $G(\mathcal{J}, \lesssim)$.

Proof The proof follows from Proposition 14, the fact that $\mathrm{sk}(\mathcal{J}) = \mathrm{sk}(G(\mathcal{J}, \lesssim))$, and Proposition 3. ■

5.2 Conditions for Faithfulness

In this section, we present the main result of this paper, which provides necessary and sufficient conditions for faithfulness of independence models (and probability distributions) and a (not necessarily known) graph in the general case. In Section 6, we specialize the result to the more well-known subclasses as corollaries.

Proposition 16 *Let $\mathcal{J}$ be an independence model over V. Suppose that $\mathcal{J}$ is minimally Markov to an anterial graph G. It then holds that $\mathcal{J}$ and G are faithful if and only if*

1. *$\mathcal{J}$ satisfies singleton-transitivity; and*
2. *$\mathcal{J}$ satisfies ordered downward- and upward-stability w.r.t. the minimal preorder for G.*

Proof Suppose that $\mathcal{J}$ and G are faithful. This is equivalent to $\mathcal{J} = \mathcal{J}(G)$. By Proposition 2, $\mathcal{J}$ satisfies singleton-transitivity, and by Proposition 12, the result follows.

Conversely, suppose that $\mathcal{J}$ satisfies singleton-transitivity as well as ordered downward- and upward-stability w.r.t. the minimal preorder $\lesssim$ for G. By Proposition 13, $G = G(\mathcal{J}, \lesssim)$. We show that $\mathcal{J}$ and G are faithful. We need to show that if $\langle A, B \,|\, C\rangle \in \mathcal{J}$ then $A \perp B \,|\, C$ in G. Consider $i \in A$ and $j \in B$. By decomposition, we have that $\langle i, j \,|\, C\rangle \in \mathcal{J}$, which implies that i and j are not adjacent in G.

If, for contradiction, C does not separate nodes i and j then there exists a connecting walk $\pi = \langle i = i_1, i_2, \ldots, i_r = j \rangle$, on which all collider sections have a node in C and all non-collider sections are outside C. In addition, for every node i_q, $2 \leq q \leq r-1$, on π define C_{i_q} as follows: If $i_q \notin C$ then $C_{i_q} = \varnothing$; if $i_q \in C$ then C_{i_q} is the set of all $l \in C \setminus \pi$ such that there is a semi-directed path from i_q to l but not from any $i_p \in C$ to l, $p \in \{q+1, \ldots, r-1\}$. Let $C_\pi = \bigcup_{q=2}^{r-1} C_{i_q}$.

We prove by induction along nodes of π that for every q, $2 \leq q \leq r$, it holds that $\langle i_1, i_q \,|\, (\cup_{p=2}^{q-1} C_{i_p}) \cup [C \setminus (\{i_q, \ldots, i_r\} \cup C_\pi)] \rangle \notin \mathcal{J}$. The base for $q = 2$ is trivial since by the construction of $G(\mathcal{J}, \lesssim)$, adjacent nodes are dependent given anything. Suppose now that the result holds for q and we prove it for $q+1$.

First, again by the construction of $G(\mathcal{J}, \lesssim)$, it holds that $\langle i_q, i_{q+1} \,|\, (\cup_{p=2}^{q-1} C_{i_p}) \cup [C \setminus (\{i_q, \ldots, i_r\} \cup C_\pi)] \rangle \notin \mathcal{J}$. By singleton-transitivity, this and the induction hypothesis imply (a): $\langle i_1, i_{q+1} \,|\, (\cup_{p=2}^{q-1} C_{i_p}) \cup [C \setminus (\{i_q, \ldots, i_r\} \cup C_\pi)] \rangle \notin \mathcal{J}$; or (b): $\langle i_1, i_{q+1} \,|\, \{i_q\} \cup (\cup_{p=2}^{q-1} C_{i_p}) \cup [C \setminus (\{i_q, \ldots, i_r\} \cup C_\pi)] \rangle \notin \mathcal{J}$. We now have two cases:

1) If i_q is on a collider section: The section containing i_q has a node in C. We again consider two subcases:

1.i) If i_q is in C: If (a) holds then first notice that i_q is not an anterior of $\{i_1, i_{q+1}\}$ nor is there a semi-directed path from i_q to $(\cup_{p=2}^{q-1} C_{i_p}) \cup [C \setminus (\{i_q, \ldots, i_r\} \cup C_\pi)]$. We apply downward-stability to (a) to obtain (b). If $C_{i_q} \neq \varnothing$ then consider a highest order node l in C_{i_q}. Again by downward-stability, we obtain $\langle i_1, i_{q+1} \,|\, \{l\} \cup (\cup_{p=2}^{q-1} C_{i_p}) \cup [C \setminus (\{i_{q+1}, \ldots, i_r\} \cup C_\pi)] \rangle \notin \mathcal{J}$. By an inductive argument along the members of C_{i_q} (by, at each step, choosing a highest order node in C_{i_q} that has not yet been chosen), we eventually obtain $\langle i_1, i_{q+1} \,|\, (\cup_{p=2}^{q} C_{i_p}) \cup [C \setminus (\{i_{q+1}, \ldots, i_r\} \cup C_\pi)] \rangle \notin \mathcal{J}$. Notice that i_q is in the conditioning set of this dependency.

1.ii) If i_q is not in C: If (b) holds then observe that i_q is in the same equivalence class as that of either i_{q+1} or a node $i_s \in C$, $s < q$. Hence we can apply upward-stability to obtain (a). Since $C_{i_2} = \varnothing$, (a) is clearly the same as $\langle i_1, i_{q+1} \,|\, (\cup_{p=2}^{q} C_{i_p}) \cup [C \setminus (\{i_{q+1}, \ldots, i_r\} \cup C_\pi)] \rangle \notin \mathcal{J}$. Notice that i_q is not in the conditioning set of this dependency.

2) If i_q is on a non-collider section: We have that $i_q \notin C$, and $i_q \in \text{ant}(\{i_1, i_{q+1}\})$. Hence, by upward-stability, from (b) we obtain (a). Again, since $C_{i_q} = \varnothing$, we obtain the result.

Therefore, we established that $\langle i_1, i_q \,|\, (\cup_{p=2}^{q-1} C_{i_p}) \cup [C \setminus (\{i_q, \ldots, i_r\} \cup C_\pi)] \rangle \notin \mathcal{J}$, for all q, where depending on whether $i_q \in C$ or not, the conditioning set contains or does not contain i_q. By letting $q = r$, we obtain $\langle i_1, i_r \,|\, C \rangle \notin \mathcal{J}$, a contradiction. Therefore, it holds that $i \perp j \,|\, C$ in G. By the composition property for separation for graphs (Proposition 1), we obtain the result. ■

We follow the arguments in the proof for the following example.

Example 1 *Consider the undirected (LWF) chain graph G in Fig. 2. Suppose that a probability distribution P is minimally Markov to G, and $\mathcal{J}(P)$ satisfies singleton-transitivity and ordered downward- and upward-stability w.r.t. the minimal preorder for G. In order to prove faithfulness, one needs to show that if two sets of nodes are not separated given a third set then they are not independent either.*

For example, we have that $l \not\perp j \,|\, l_2$. Notice that this is a case where there is no path that connects l and j. In order to show that $l \not\perp j \,|\, l_2$, (as in the proof of the above theorem) we start from $l = l_1$ and move towards $j = l_7$ via the connecting walk $\langle l_1, l_2, l_3, l_4, l_5, l_6, l_7 \rangle$ given l_2.

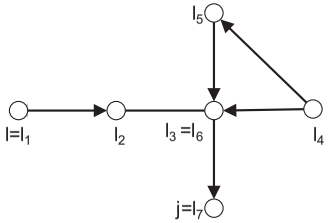


Figure 2: An undirected (LWF) chain graph.

We abbreviate singleton-transitivity, ordered upward-stability, and ordered downward stability by "st", "ous", and "ods" respectively. We write the arguments in a condensed from. For example, the first implication (on the left hand side) says that $l_1 \not\perp l_2$ and $l_2 \not\perp l_3$, using singleton-transitivity, imply that either $l_1 \not\perp l_3$ or $l_1 \not\perp l_3 \,|\, l_2$; but now ordered downward-stability implies that we must have $l_1 \not\perp l_3 \,|\, l_2$.

$$
\begin{array}{ccccccccc}
l_1 \not\perp l_2 & & l_1 \not\perp l_3 & & l_4 \not\perp l_5 \,|\, l_2 & & l_1 \not\perp l_5 \,|\, \{l_2, l_4\} & & l_6 \not\perp l_7 \,|\, l_2 & & l_1 \not\perp l_7 \,|\, \{l_2, l_6\} \\
& \overset{st}{\Rightarrow} or \;\; \Downarrow ods & & & & \overset{st}{\Rightarrow} or \;\; \Downarrow ous & & & & \overset{st}{\Rightarrow} or \;\; \Downarrow ous & \\
l_2 \not\perp l_3 & l_1 \not\perp l_3 \,|\, l_2 & & & l_1 \not\perp l_4 \,|\, l_2 & & l_1 \not\perp l_5 \,|\, l_2 & & l_1 \not\perp l_6 \,|\, l_2 & & l_1 \not\perp l_7 \,|\, l_2. \\
& & & \overset{st}{\Rightarrow} or \;\; \Uparrow ous & & & & \overset{st}{\Rightarrow} or \;\; \Uparrow ous & & & \\
& & l_3 \not\perp l_4 \,|\, l_2 & l_1 \not\perp l_3 \,|\, \{l_2, l_3\} & & l_5 \not\perp l_6 \,|\, l_2 & & l_1 \not\perp l_6 \,|\, \{l_2, l_5\} & & &
\end{array}
$$

The following example shows how singleton-transitivity and ordered stabilities are necessary for faithfulness.

Example 2 *Consider the DAG G in Fig. 3. If an independence model $\mathcal{J}$ is minimally Markov to G then we must have $\langle h, k \,|\, \{j, l\} \rangle \in \mathcal{J}$ and $\langle j, l \,|\, k \rangle \in \mathcal{J}$. Now, in violation of faithfulness, if only $\langle h, k \,|\, \varnothing \rangle \in \mathcal{J}$ in addition to these holds then upward-stability is violated since for $j > h$, $\langle h, k \,|\, j \rangle \in \mathcal{J}$ must hold. If, in violation of faithfulness, only $\langle h, k \,|\, j \rangle \in \mathcal{J}$ in addition to the original statements holds then singleton-transitivity is violated since $\langle h, k \,|\, j \rangle \in \mathcal{J}$ and $\langle h, k \,|\, \{j, l\} \rangle \in \mathcal{J}$ must imply either $\langle h, l \,|\, j \rangle \in \mathcal{J}$ or $\langle k, l \,|\, j \rangle \in \mathcal{J}$, which are both impossible due to minimality since h, l and k, l are adjacent in G.*

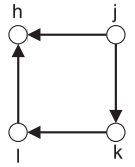


Figure 3: A DAG.

Without assuming the Markov assumption we have the following:

Theorem 17 *Let $\mathcal{J}$ be an independence model defined over V. It then holds that $\mathcal{J}$ is graphical if and only if*

1. *$\mathcal{J}$ is a singleton-transitive compositional graphoid; and*
2. *there exists a $\mathcal{J}$-compatible preorder $\lesssim$ over V w.r.t. which $\mathcal{J}$ satisfies ordered downward- and upward-stability.*

In addition, if 1 and 2 hold then $\mathcal{J}$ is faithful to $G(\mathcal{J}, \lesssim)$.

Proof ($\Rightarrow$) follows from Propositions 1, 2, and 12, and the fact that there is always an AnG that is Markov equivalent to a given CMG. To prove ($\Leftarrow$) and the last statement of the theorem, consider $G(\mathcal{J}, \lesssim)$. Theorem 15 implies that $\mathcal{J}$ is minimally Markov to $G(\mathcal{J}, \lesssim)$. This together with the $\mathcal{J}$-compatibility of the preorder, using Proposition 16, implies that $\mathcal{J}$ and $G(\mathcal{J}, \lesssim)$ are faithful. ■

Hence, for probability distributions, we have the following characterization:

Corollary 18 *Let P be a probability distribution defined over $\{X_\alpha\}_{\alpha \in V}$. It then holds that P is graphical if and only if*

1. *$\mathcal{J}(P)$ satisfies intersection, composition, and singleton-transitivity; and*
2. *there exists a P-compatible preorder $\lesssim$ over V w.r.t. which $\mathcal{J}(P)$ satisfies ordered downward- and upward-stability.*

In addition, if 1 and 2 hold then P is faithful to $G(P, \lesssim)$.

In fact, we have shown in Theorem 17 that if $\mathcal{J}$ is graphical then for every $\mathcal{J}$-compatible preorder w.r.t. which $\mathcal{J}$ satisfies ordered downward- and upward-stability, $G(\mathcal{J}, \lesssim)$ is a graph that is faithful to $\mathcal{J}$. A consequence of Proposition 13 implies that such graphs (that is $G = G(\mathcal{J}, \lesssim)$ for some preorder $\lesssim$ w.r.t. which $\mathcal{J}$ satisfies ordered downward- and upward-stability) constitute the set of all graphs that are faithful to $\mathcal{J}$, that is a Markov equivalence class of graphs whose members are faithful to $\mathcal{J}$. Every member of the equivalence class corresponds to a different preorder w.r.t. which $\mathcal{J}$ satisfies ordered downward- and upward-stability. This equivalence set can be represented by one member (a graph with the least number of arrowheads) similar to the idea of a CPDAG for the Markov equivalence class of DAGs (see Spirtes et al. 2000), but we do not go through the details of this here.

Of course, if the goal is to verify whether $\mathcal{J}$ and a given G are faithful then we can use the following corollary:

Corollary 19 *Let $\mathcal{J}$ be an independence model defined over V, and G an AnG with node set V. It then holds that if*

1. *$\mathcal{J}$ is a singleton-transitive compositional graphoid; and*
2. *$\mathcal{J}$ satisfies ordered downward- and upward-stability w.r.t. the minimal preorder for G,*

then $\mathcal{J}$ and G are faithful.

It is also important to note that singleton-transitivity and ordered upward or downward-stability are necessary in the sense that they are not implied by one another, although there are different ways to axiomatize singleton-transitive compositional graphoids that satisfy ordered upward- and downward-stability.

If P is Gaussian then we have the following:

Corollary 20 *Let P be regular Gaussian distribution. It then holds that P is graphical if and only if there exists a P-compatible preorder $\lesssim$ w.r.t. which $\mathcal{J}(P)$ satisfies ordered downward- and upward-stability.*

Proof The proof follows from the fact that regular Gaussian distributions satisfy intersection, composition, and singleton-transitivity. ■

6. Specialization to Subclasses

The definitions and results presented in this paper can be specialized to any subclass of CMGs including ancestral graphs and LWF chain graphs. However, here we only focus on the three of the most used classes of undirected (concentration), bidirected (covariance), and directed acyclic graphs.

6.1 Specialization to Undirected and Bidirected Graphs

For connected UGs, every two nodes are in the same equivalence class, and for BGs every two nodes are incomparable; hence, the minimal (pre)ordering is trivial and uninteresting in these cases. More precisely, ordered upward- and downward-stabilities can be specialized for these two types of trivial preordering:

8a. if $\langle i, j \,|\, C\rangle \in \mathcal{J}$ then $\langle i, j \,|\, C \cup \{k\}\rangle \in \mathcal{J}$ for every $k \in V \setminus \{i, j\}$ (*upward-stability*);

9a. if $\langle i, j \,|\, C\rangle \in \mathcal{J}$ then $\langle i, j \,|\, C \setminus \{k\}\rangle \in \mathcal{J}$ for every $k \in V \setminus \{i, j\}$ (*downward-stability*).

Proposition 21 *An independence model $\mathcal{J}$ satisfies ordered upward- and downward-stability (8) and (9) w.r.t. the minimal preorder if and only if,*

1. *for connected undirected graphs, it satisfies upward-stability (8a);*
2. *for bidirected graphs, it satisfies downward-stability (9a).*

In addition, if, for undirected graphs, $\mathcal{J}$ satisfies (8a) then it satisfies (8).

Proof The proof of 1 and 2 are straightforward since in 1 all nodes are in the same equivalence class, and in 2 all nodes are incomparable. The second part is trivial. ■

Analogous to Proposition 12, it is also straightforward to show the following:

Proposition 22 *It holds that*

1. *for an undirected graph G, $\mathcal{J}(G)$ satisfies upward-stability;*

2. for a bidirected graph G, $\mathcal{J}(G)$ satisfies downward-stability.

For UGs and BGs, denote the unique and trivial valid preorders by $\lesssim_u^*$ and $\lesssim_b^*$ respectively. For UGs, $G_u^*(\mathcal{J}) := G(\mathcal{J}, \lesssim_u^*) = \mathrm{sk}(\mathcal{J})$, whereas for BGs, $G_b^*(\mathcal{J}) := G(\mathcal{J}, \lesssim_b^*)$ is $\mathrm{sk}(\mathcal{J})$ with all edges directed to be bidirected. Another way to construct induced UGs and BGs by $\mathcal{J}$ is to construct them based on their corresponding pairwise Markov property, that is to connect every pair of nodes i and j if $\langle i, j \,|\, V \setminus \{i, j\}\rangle \notin \mathcal{J}$ for UGs, and if $\langle i, j \,|\, \varnothing\rangle \notin \mathcal{J}$ for BGs. Denote these graphs by $G_u(\mathcal{J})$ and $G_b(\mathcal{J})$ respectively. We then have the following:

Proposition 23 *It holds that,*

1. *for undirected graphs, $G_u^*(\mathcal{J}) = G_u(\mathcal{J})$ if $\mathcal{J}$ satisfies upward-stability;*
2. *for bidirected graphs, $G_b^*(\mathcal{J}) = G_b(\mathcal{J})$ if $\mathcal{J}$ satisfies downward-stability.*

For upward- and downward-stability, we also have the following observation:

Lemma 24 *For an independence model $\mathcal{J}$ the following holds:*

1. *If $\mathcal{J}$ is a semi-graphoid and satisfies upward-stability then $\mathcal{J}$ satisfies composition.*
2. *If $\mathcal{J}$ is a semi-graphoid and satisfies downward-stability then $\mathcal{J}$ satisfies intersection.*

Proof 1. Suppose that $\langle A, B \,|\, C\rangle \in \mathcal{J}$ and $\langle A, D \,|\, C\rangle \in \mathcal{J}$. By decomposition, it holds, for $i \in A$, $j \in B$, and $k \in D$, that $\langle i, j \,|\, C\rangle \in \mathcal{J}$ and $\langle i, k \,|\, C\rangle \in \mathcal{J}$. By upward-stability, we obtain $\langle i, j \,|\, C \cup \{k\}\rangle \in \mathcal{J}$. Contraction implies $\langle i, \{j, k\} \,|\, C\rangle \in \mathcal{J}$. By an inductive argument, we obtain the result.

2. Suppose that $\langle A, B \,|\, C \cup D\rangle \in \mathcal{J}$ and $\langle A, D \,|\, C \cup B\rangle \in \mathcal{J}$. By decomposition, it holds, for $i \in A$, $j \in B$, and $k \in D$, that $\langle i, j \,|\, C \cup \{k\}\rangle \in \mathcal{J}$ and $\langle i, k \,|\, C \cup \{j\}\rangle \in \mathcal{J}$. By downward-stability, we obtain $\langle i, k \,|\, C\rangle \in \mathcal{J}$. Contraction implies $\langle i, \{j, k\} \,|\, C\rangle \in \mathcal{J}$. By an inductive argument, we obtain the result. ■

Lemma 25 *Let $\mathcal{J}$ be an independence model, and suppose that $\mathcal{J}$ is a graphoid. It then holds that $\mathcal{J}$ satisfies ordered downward-stability (9) w.r.t. the minimal preorder for $G_u(\mathcal{J})$.*

Proof By definition, we know that $\mathcal{J}$ is pairwise Markov to $G_u(\mathcal{J})$. It is known that under graphoid, this implies that $\mathcal{J}$ is Markov to $G_u(\mathcal{J})$; see Pearl (1988). Now suppose that $\langle i, j \,|\, C\rangle \in \mathcal{J}$. The only applicable k in the definition of (9) is not in the same connected component as that of i or j. Hence, since $i \perp j \,|\, C \setminus \{k\}$, by the global Markov property, it holds that $\langle i, j \,|\, C \setminus \{k\}\rangle \in \mathcal{J}$. ■

As a consequence of Theorem 15, we have the following for UGs and BGs:

Corollary 26 *It holds that*

1. *if $\mathcal{J}$ is a graphoid and satisfies upward-stability then $\mathcal{J}$ is minimally Markov to the undirected graph $G_u(\mathcal{J})$;*
2. *if $\mathcal{J}$ is a compositional semi-graphoid and satisfies downward-stability then $\mathcal{J}$ is minimally Markov to the bidirected graph $G_b(\mathcal{J})$.*

Proof The proof follows from Propositions 23 and 21, and Lemmas 24 and 25. ■

For other classes of graphs, the preordering (or ordering) of nodes (and consequently random variables) is necessary since the conditioning set $C(i, j)$ for the pairwise Markov property depends on the preordering of the variables. This implies that in general one cannot construct more than the skeleton of graphs based on the pairwise Markov property if there is no given preordering.

Proposition 16 specializes to the following two corollaries:

Corollary 27 *Suppose that an independence model $\mathcal{J}$ is Markov to $G_u(\mathcal{J})$. Then $\mathcal{J}$ and $G_u(\mathcal{J})$ are faithful if and only if $\mathcal{J}$ satisfies singleton-transitivity and upward-stability.*

Proof The proof follows from Propositions 21, 22, and 23, and Lemma 25. ■

Similarly, for BGs, we have the following:

Corollary 28 *Suppose that an independence model $\mathcal{J}$ is Markov to $G_b(\mathcal{J})$. Then $\mathcal{J}$ and $G_b(\mathcal{J})$ are faithful if and only if $\mathcal{J}$ satisfies singleton-transitivity and downward-stability.*

The main conditions for an independence model to be faithful to a UG or BG (consequence of Theorem 17) are as follows; see also Theorem 3 of Pearl and Paz (1985) for a similar result for UGs:

Corollary 29 *Let $\mathcal{J}$ be an independence model defined over V. It then holds that $\mathcal{J}$ is faithful to an undirected graph if and only if it is an upward-stable singleton-transitive graphoid. In addition, if the conditions hold then $\mathcal{J}$ is faithful to $G_u(\mathcal{J})$.*

Proof The proof follows from Propositions 21, 22, and 23, and Lemmas 24 and 25. ■

Corollary 30 *Let $\mathcal{J}$ be an independence model defined over V. It then holds that $\mathcal{J}$ is faithful to a bidirected graph if and only if it is a downward-stable singleton-transitive compositional semi-graphoid. In addition, if the conditions hold then $\mathcal{J}$ is faithful to $G_b(\mathcal{J})$.*

Therefore, a probability distribution is faithful to an undirected graph if and only if it satisfies intersection, singleton-transitivity, and upward-stability; and to a bidirected graph if and only if it satisfies composition, singleton-transitivity, and downward-stability.

Finally, as a consequence of Corollary 20, we have the following:

Corollary 31 *Let P be a regular Gaussian distribution. It then holds that P being faithful to an undirected graph is equivalent to it satisfying upward-stability; and P being faithful to a bidirected graph is equivalent to it satisfying downward-stability.*

6.2 Specialization to Directed Acyclic Graphs

For DAGs, the valid preorder specializes to a valid (partial) order since there are no lines in the graph: If $i \longrightarrow\!\!\!\!\!\succ j$ then $i > j$. In the minimal order for a DAG, two nodes are incomparable if and only if neither is an ancestor of the other. Therefore, we define ordered stabilities w.r.t. ordering $\leq$ for DAGs:

8b. if $\langle i, j \,|\, C\rangle \in \mathcal{J}$ then $\langle i, j \,|\, C \cup \{k\}\rangle \in \mathcal{J}$ for every $k \in V \setminus \{i, j\}$ such that $l < k$ for some $l \in \{i, j\}$ (*ordered upward-stability*);

9b. if $\langle i, j \,|\, C\rangle \in \mathcal{J}$ then $\langle i, j \,|\, C \setminus \{k\}\rangle \in \mathcal{J}$ for every $k \in V \setminus \{i, j\}$ such that $l \not< k$ for every $l \in \{i, j\} \cup C \setminus \{k\}$ (*ordered downward-stability*).

As a consequence of Proposition 12, we have the following:

Corollary 32 *For a directed acyclic graph G, the induced independence model by d-separation $\mathcal{J}(G)$ satisfies ordered upward- and downward-stability (8b) and (9b) w.r.t. the minimal order for G.*

As a consequence of Theorem 15, and for $G(\mathcal{J}, \leq)$, the DAG obtained by directing the edges of $\mathrm{sk}(\mathcal{J})$ based on the order $\leq$, we have the following:

Corollary 33 *Suppose, for an independence model $\mathcal{J}$ over V, that*

1. *$\mathcal{J}$ is a compositional graphoid; and*
2. *there exists a $\mathcal{J}$-compatible order $\leq$ over V w.r.t. which $\mathcal{J}$ satisfies ordered downward- and upward-stability, (8b) and (9b).*

Then $\mathcal{J}$ is minimally Markov to the directed acyclic graph $G(\mathcal{J}, \leq)$.

The main conditions for an independence model to be faithful to a DAG (consequence of Theorem 17) is as follows:

Corollary 34 *Let $\mathcal{J}$ be an independence model defined over V. It then holds that $\mathcal{J}$ is faithful to a directed acyclic graph G if and only if*

1. *$\mathcal{J}$ is a singleton-transitive compositional graphoid; and*
2. *there exists a $\mathcal{J}$-compatible order $\leq$ over V w.r.t. which $\mathcal{J}$ satisfies ordered downward- and upward-stability, (8b) and (9b).*

In addition, if the conditions hold then $\mathcal{J}$ is faithful to $G(\mathcal{J}, \leq)$.

For a probability distribution P, the first condition of the above corollary simplifies to that P satisfies intersection, composition, and singleton-transitivity.

7. Comparison to the Results in the Literature

All results in the literature concerning faithfulness of certain probability distributions and certain types of graphs can be considered corollaries (or examples) of Theorem 17. Here we verify this for some of the more well-known or interesting results. For brevity, we leave out some interesting results such as faithfulness of Gaussian and discrete distributions and LWF chain graphs; see Peña (2011) and Peña (2009); Studený and Bouckaert (1998) respectively.

7.1 Faithfulness of MTP_2 Distributions and Undirected Graphs

Multivariate totally positive of order 2 (MTP_2) distributions (Karlin and Rinott, 1980) are distributions whose density f is MTP_2, that is $f(x)f(y) \leq f(x \wedge y)f(x \vee y)$, for all vectors x, y, where $x \wedge y = (\min(x_1, y_1), \ldots, \min(x_m, y_m))$ and $x \vee y = (\max(x_1, y_1), \ldots, \max(x_m, y_m))$. In Fallat et al. (2017), it was shown that when an MTP_2 distribution P is a graphoid, it holds that P and what we denote here by $G_u(P)$ are faithful. It is known that MTP_2 distributions satisfy both singleton-transitivity and upward-stability, but not necessarily the intersection property; see Fallat et al. (2017). Hence, this result is a direct implication of Corollary 29.

7.2 Faithfulness of Gaussian Distributions and Undirected Graphs

Lněnička and Matúš (2007) provides regular Gaussian distributions that are faithful to any undirected graph; see Corollary 3 of the mentioned paper. In order to do so, given an undirected graph $G = (V, E)$, they provide the following matrix $A^{G,\epsilon}$, which acts as the covariance matrix of a Gaussian distribution that is faithful to G:

$$A^{G,\epsilon} = \begin{cases} 1 & \text{if } i = j, \\ \epsilon & \text{if } i \neq j \text{ and } i \text{ and } j \text{ are adjacent in } G, \\ 0 & \text{otherwise,} \end{cases}$$

where $i, j \in V$.

It is known that, for sufficiently small $\epsilon > 0$, the corresponding Gaussian distribution is regular, and by Corollary 31, we only need to show that it satisfies upward-stability. It is straightforward to show that for sufficiently small $\epsilon > 0$, the inverse of $A^{G,\epsilon}$, which is the concentration matrix of the Gaussian distribution, is an *M-matrix*, that is all its on-diagonal elements are positive and all its off-diagonal elements are non-positive. This is equivalent to the Gaussian distribution being MTP_2; see Karlin and Rinott (1983). Hence, upward-stability is satisfied as discussed above.

7.3 Unfaithfulness of Certain Gaussian Distributions and Undirected Graphs and DAGs

There are many examples in the literature for, and in fact it is easy to construct examples of, probability distributions that are not faithful to undirected graphs; for one of the first observations of this behaviour in a data set, see Example 4 of Cox and Wermuth (1993). As a constructed example, let us discuss Example 1 of Soh and Tatikonda (2014), which provided the Gaussian distribution for $X = (X_1, X_2, X_3, X_4)$ with zero mean and positive definite covariance matrix

$$\Sigma = \begin{bmatrix} 3 & 2 & 1 & 2 \\ 2 & 4 & 2 & 1 \\ 1 & 2 & 7 & 1 \\ 2 & 1 & 1 & 6 \end{bmatrix}.$$

It can be seen that the concentration matrix Σ^{-1} has no zero entries, but $X_1 \perp\!\!\!\perp X_3 \,|\, X_2$ since $\Sigma_{13} = \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{23}$. Hence, the distribution and its complete skeleton are not faithful. This is clearly a consequence of the violation of upward-stability.

For DAGs too, it is easy to construct or observe examples where faithfulness is violated. One of the simplest of such examples, is what is related to the so-called "transitivity of causation" in philosophical causality; see, for example, Ramsey et al. (2006), where the goal is to show how the PC algorithm (Spirtes et al., 2000) errs. Consider the graph $a \dashrightarrow b \dashrightarrow c$, where $a \perp\!\!\!\perp c$ and $a \not\perp\!\!\!\perp c \,|\, b$. This, in our language, is a clear violation of singleton-transitivity since the graph implies that a, b and b, c are always dependent.

7.4 Faithfulness of Gaussian and Discrete Distributions and DAGs

It was shown in Meek (1995) that, in certain measure-theoretic senses, almost all the regular Gaussian and discrete distributions that are Markov to a given DAG are faithful to it. It is known, for DAGs, that the global Markov property is equivalent to the *factorization property* (see Lauritzen 1996), which holds if the joint density can factorize as follows:

$$f_V(x) = \prod_{v \in V} f_V(x_v \,|\, x_{\mathrm{pa}(v)}).$$

Since Gaussian and discrete distributions satisfy singleton-transitivity, in order to prove the result, one needs to show that ordered upward- and downward-stabilities, (8b) and (9b), are satisfied for almost all distributions that factorize w.r.t. the DAG. We refrain from showing this explicitly here due to technicality of the proof, but the general idea is similar to the proofs in Appendix B of Meek (1995): For (9b), we have that the probabilistic conditional independence holds for the marginal probability over (i, j, C, k), and we need to exploit the factorization to show that the same conditional independence formula holds for the marginal probability over (i, j, C) when $k \notin \mathrm{an}(\{i, j\} \cup C$. In this case the corresponding resulting polynomials are much simpler than those of Meek (1995) since one needs to consider extra marginalization over only one variable k. For (8b), we will need to prove the other direction in a similar fashion.

7.5 Faithfulness of Gaussian Distributions and Ancestral Graphs

The main goal of Richardson and Spirtes (2002) is to capture the conditional independence structure of a DAG model after marginalization and conditioning; see also Sadeghi (2013). This was done by introducing the class of AGs, and it was shown that every AG is probabilistic. In fact, there is a Gaussian distribution faithful to any AG, which is indeed the Gaussian distribution faithful to a DAG after marginalization and conditioning; see Theorem 7.5 of this paper.

However, by using our results, without going through the standard theory of ancestral graphs, it is possible to show that distributions that are faithful to a DAG are graphical after marginalization and conditioning: In order to do so, we use the notation in Sadeghi (2013), for an independence model $\mathcal{J}$, to define the *independence model after marginalization over M and conditioning on C*, denoted by $\alpha(\mathcal{J}; M, C)$, for disjoint subsets M and C of V:

$$\alpha(\mathcal{J}; M, C) = \{\langle A, B \,|\, D\rangle : \langle A, B \,|\, D \cup C\rangle \in \mathcal{J} \text{ and } (A \cup B \cup D) \cap (M \cup C) = \varnothing\}.$$

First, we show that $\alpha(\mathcal{J}; M, C)$ satisfies condition 1 of Theorem 17.

Proposition 35 *If $\mathcal{J}$ satisfies singleton-transitivity, intersection, and composition then so does $\alpha(\mathcal{J}; M, C)$.*

Proof For composition, assume that $\langle A, B \,|\, C'\rangle \in \alpha(\mathcal{J}; M, C)$ and $\langle A, D \,|\, C'\rangle \in \alpha(\mathcal{J}; M, C)$. By definition, we have $\langle A, B \,|\, C' \cup C\rangle \in \mathcal{J}$ and $\langle A, D \,|\, C' \cup C\rangle \in \mathcal{J}$. Composition for $\mathcal{J}$ implies $\langle A, B \cup D \,|\, C' \cup C\rangle \in \mathcal{J}$. Therefore, $\langle A, B \cup D \,|\, C'\rangle \in \alpha(\mathcal{J}; M, C)$. Proof of the others is similar to this one and as straightforward. ■

Denote by $\alpha(G; M, C)$, the generated AG from the DAG G and M and C two disjoint subsets of its node set that are marginalized over and conditioned on respectively. Here we do not go through the generating process of AGs from DAGs; for more details, see Richardson and Spirtes (2002); Sadeghi (2013). Without explicitly using the theory of stability under marginalization and conditioning for AGs, we show the following.

Proposition 36 *If $\mathcal{J}$ and a DAG G are faithful then $\alpha(\mathcal{J}; M, C)$ and the ancestral graph $\alpha(G; M, C)$ are faithful.*

Proof By Proposition 35, we only need to show that w.r.t. the ordering associated with $\alpha(G; M, C)$, condition 2 of Theorem 17 is also satisfied by $\alpha(\mathcal{J}; M, C)$.

To prove ordered upward-stability, assume that $\langle i, j \,|\, C' \cup \{k\}\rangle \notin \alpha(\mathcal{J}; M, C)$, where k is an anterior of a node l, $l \in \{i, j\}$ or connected by lines to l, $l \in C'$ in $H = \alpha(G; M, C)$. We have that $\langle i, j \,|\, C \cup C' \cup \{k\}\rangle \notin \mathcal{J}$. Since $\mathcal{J}$ and G are faithful, there is a connecting walk π between i and j given $C \cup C' \cup \{k\}$. If k is not on π, π is connecting given $C \cup C'$. Otherwise, it holds that k must be a collider on π. By Lemma 2 in Sadeghi (2013), k is in ancestor of a node $l' \in \{i, j\} \cup C \cup C'$. By replacing the subwalk of π from l' to k by the directed path from k to l' if $l' \in \{i, j\}$, or by adding the directed path from k to l' plus the reverse from l' to k to π if $l' \in C \cup C'$, we conclude that $\langle i, j \,|\, C \cup C'\rangle \notin \mathcal{J}$. Hence, $\langle i, j \,|\, C'\rangle \notin \alpha(\mathcal{J}; M, C)$.

To prove ordered downward-stability, assume that $\langle i, j \,|\, C' \setminus \{k\}\rangle \notin \alpha(\mathcal{J}; M, C)$, where $k \in C'$ is not an anterior of any node $l \in \{i, j\} \cup C'$ in H. We have that $\langle i, j \,|\, C \cup C' \setminus \{k\}\rangle \notin \mathcal{J}$. Since $\mathcal{J}$ and G are faithful, there is a connecting walk π between i and j given $C \cup C' \setminus \{k\}$. Node k cannot be on π since k must be a non-collider and consequently an ancestor of a node in $\{i, j\} \cup C \cup C'$ in G, which by Lemma 3 in Sadeghi (2013), implies that $k \in \mathrm{an}(l)$ in H. Hence, k is not on π, and consequently π is connecting given $C \cup C'$. We conclude that $\langle i, j \,|\, C \cup C'\rangle \notin \mathcal{J}$. Hence, $\langle i, j \,|\, C'\rangle \notin \alpha(\mathcal{J}; M, C)$. ■

This shows that, for every AG, there is a distribution faithful to it since, for every AG, there is a DAG that, after marginalization and conditioning, results in the given AG.

8. Summary and Future Work

We provided sufficient and necessary conditions for an independence model, and consequently a probability distribution, to be faithful to a chain mixed graph. All the definitions and results concerning independence models are true for probabilistic independence models. The conditions can be divided into two main categories: 1) Those that ensure that the distribution is Markov to a graph whose skeleton is the same as the skeleton of the distribution—these properties are namely intersection, composition, and ordered upward- and downward-stabilities. If the distribution is already pairwise Markov to the graph then intersection and composition suffice. Ordered upward- and downward-stabilities direct the edges of the skeleton of the distribution so that the pairwise Markov property is satisfied.

2) Those that ensure that a Markov distribution is also faithful to the graph. These are singleton-transitivity and ordered upward- and downward-stabilities.

The type of preordering or preorderings w.r.t. which the distribution satisfies ordered upward- and downward-stabilities implies the type or types of graphs that are faithful to the distribution. In order to only deal with simpler classes of graphs, it is enough to search through a more refined set of preorderings determined by the conditions that graphs of a specific class must satisfy. It is indeed true that when the skeleton of the graph or distribution is known, the preordering is equivalent to different ways the edges of the skeleton can be directed, but in principle the preordering of the variables is unrelated to the skeleton of the distribution.

The conditions of faithfulness provided in this paper for probability distributions are sufficient and necessary. Thus, based on these conditions many families of distributions can be shown not to be suitable for graphical modeling. This also shows the importance of Gaussian distributions in graphical models as the provided conditions (other than ordered upward- and downward-stabilities) are clearly satisfied by the regular Gaussian distributions. In addition, these conditions help those who devise new parametric models to ensure that their models satisfy the provided conditions for faithfulness.

It is clearly important to study the characteristics of distributions that satisfy these conditions. The intersection property is well-studied and, in addition to positivity of the density that has been known to be a sufficient condition, necessary and sufficient conditions for a probabilistic independence to satisfy this property are known; see Peters (2015). For composition and singleton-transitivity, more studies are needed; in particular, it would be worthwhile to find simple sufficient conditions for probability distributions in order to satisfy these properties.

Another approach is to devise algorithms that test whether these conditions are satisfied given data or an independence model. Indeed one can argue for an alternative approach to determine whether a distribution is faithful to some graph: one first applies the structure learning algorithm (taking as input the distribution and outputting a graph) and then one tests whether the separation relationships in the found graph correspond to the conditional independencies in the distribution. However, testing our conditions are much more efficient. As an example, in the case of undirected graphs and Gaussian distributions, by our results, one only needs to test upward-stability for non-adjacent single elements i and j (starting with a minimal node cut-set as the conditioning set) rather than testing all global conditional independencies implied by the graph.

As another algorithmic consequence that the results in this paper may prompt, they lead to a *constraint-based* structural learning algorithm (as opposed to score-based, for example, Chickering 2003) most similar to the Fast Causal Inference (FCI) algorithm (Spirtes et al., 2000)—it finds the skeleton of the graph and then explores directions of the edges of the graph on the skeleton. Directing the edges of the skeleton is performed based on an exploitation of the preordering of the variables (and consequently the nodes of the graph) so that w.r.t. such a preordering ordered upward- and downward-stabilities are satisfied. This algorithm would have two advantages over the known constraint-based algorithms in the literature: Firstly, it is devised for a larger class of AnGs, which can be thought of as learning the LWF chain graphs with hidden and selection variables; see Sadeghi (2016). Secondly, the exploitation of ordered upward- and downward-stability for structural learning can be combined with actually testing whether these conditions are satisfied, and the theory in the paper ensures the faithfulness of the generated graphs. We plan on providing the details of this algorithm elsewhere.

When dealing with data, another issue is that obtaining a faithful graph is very sensitive to hypothesis testing errors for inferring the independence statements from data; see Uhler et al. (2013). The concept of *strong faithfulness* (Zhang and Spirtes, 2003) has been suggested for the Gaussian case in order to deal with this issue. Although strong faithfulness, by nature, does not seem to be characterizable in this way, it is an interesting problem to study which of the equivalent conditions to faithfulness are more prone to such sensitivity.

Acknowledgments

The original idea of this paper was inspired by discussions at the American Institute of Mathematics workshop “Positivity, Graphical Models, and the Modeling of Complex Multivariate Dependencies” in October 2014. The author thanks the AIM and NSF for their support of the workshop and is indebted to other participants of the discussion group, especially to Shaun Fallat, Steffen Lauritzen, Nanny Wermuth, and Piotr Zwiernik for follow-up discussions. The author is also grateful to Rasmus Petersen for valuable discussions. This work was partly supported by grant #FA9550-12-1-0392 from the U.S. Air Force Office of Scientific Research (AFOSR) and the Defense Advanced Research Projects Agency (DARPA).

References

R. Ayesha Ali, Thomas S. Richardson, and Peter Spirtes. Markov equivalence for ancestral graphs. *Annals of Statistics*, 37(5B):2808–2837, 2009.

Steen A. Andersson, David Madigan, and Michael D. Perlman. Alternative Markov properties for chain graphs. *Scandinavian Journal of Statistics*, 28:33–85, 2001.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2003.

Peter Clifford. Markov random fields in statistics. In G. R. Grimmett and D. J. A. Welsh, editors, *Disorder in Physical Systems: A Volume in Honour of John M. Hammersley*, pages 19–32. Oxford University Press, 1990.

David R. Cox and Nanny Wermuth. Linear dependencies represented by chain graphs (with discussion). *Statistical Science*, 8(3):204–218; 247–277, 1993.

John N. Darroch, Steffen L. Lauritzen, and Terry P. Speed. Markov fields and log-linear interaction models for contingency tables. *Annals of Statistics*, 8:522–539, 1980.

A. Philip Dawid. Conditional independence in statistical theory (with discussion). *Journal of the Royal Statistical Society Series B*, 41(1):1–31, 1979.

Mathias Drton and Thomas S. Richardson. Binary models for maginal independence. *Journal of the Royal Statistical Society Series B*, 41(2):287–309, 2008.

Shaun Fallat, Steffen Lauritzen, Kayvan Sadeghi, Caroline Uhler, Nanny Wermuth, and Piotr Zwiernik. Total positivity in Markov structures. *Annals of Statistics.*, 45(3), 2017.

Morten Frydenberg. The chain graph Markov property. *Scandinavian Journal of Statistics*, 17:333–353, 1990.

Dan Geiger, Thomas S. Verma, and Judea Pearl. Identifying independence in Bayesian networks. *Networks*, 20:507–534, 1990.

Clark Glymour, Richard Scheines, Peter Spirtes, and Kevin Kelly. *Discovering Causal Structure*. Academic Press, 1987.

Samuel Karlin and Yosef Rinott. Classes of orderings of measures and related correlation inequalities. I. Multivariate totally positive distributions. *Journal of Multivariate Analysis*, 10(4):467–498, 1980.

Samuel Karlin and Yosef Rinott. M-matrices as covariance matrices of multinormal distributions. *Linear Algebra and its Applications*, 52:419 – 438, 1983.

Göran Kauermann. On a dualization of graphical Gaussian models. *Scandinavian Journal of Statistics*, 23(1):105–116, 1996.

Harri Kiiveri, Terry P. Speed, and J. B. Carlin. Recursive causal models. *Journal of the Australian Mathematical Society, Series A*, 36:30–52, 1984.

Steffen L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, United Kingdom, 1996.

Steffen L. Lauritzen and Kayvan Sadeghi. Unifying Markov properties for graphical models. *Annals of Statistics*, To appear, 2017. URL `https://arxiv.org/abs/1608.05810`.

Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50:157–224, 1988.

Steffen L. Lauritzen and Nanny Wermuth. Graphical models for association between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, 17:31–57, 1989.

Radim Lněnička and František Matúš. On Gaussian conditional independence structures. *Kybernetika*, 43(3):327–342, 2007.

Christopher Meek. Strong completeness and faithfulness in Bayesian networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, pages 411–418, San Francisco, CA, USA, 1995.

Richard E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, Upper Saddle River, NJ, USA, 2004.

Jose M. Peña. Faithfulness in chain graphs: The discrete case. *International Journal of Approximate Reasoning*, 50:1306–1313, 2009.

Jose M. Peña. Faithfulness in chain graphs: The Gaussian case. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, volume 15, pages 588–599. JMLR.org, 2011.

Jose M. Peña. Marginal AMP chain graphs. *International Journal of Approximate Reasoning*, 55:1185–1206, 2014.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : networks of plausible inference*. Morgan Kaufmann Publishers, San Mateo, CA, USA, 1988.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, New York, NY, USA, 2nd edition, 2009.

Judea Pearl and Azaria Paz. *Graphoids: A Graph-based Logic for Reasoning about Relevance Relations*. University of California (Los Angeles). Computer Science Department, 1985.

Jonas Peters. On the intersection property of conditional independence and its application to causal discovery. *Journal of Causal Inference*, 3:97–108, 2015.

Joseph Ramsey, Jiji Zhang, and Peter Spirtes. Adjacency-faithfulness and conservative causal inference. In *UAI*. AUAI Press, 2006.

Thomas Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1):145–157, 2003.

Thomas S. Richardson and Peter Spirtes. Ancestral graph Markov models. *Annals of Statistics*, 30(4):962–1030, 2002.

Kayvan Sadeghi. Stable mixed graphs. *Bernoulli*, 19(5(B)):2330–2358, 2013.

Kayvan Sadeghi. Marginalization and conditioning for LWF chain graphs. *Annals of Statistics*, 44(4):1792–1816, 2016.

Kayvan Sadeghi and Steffen L. Lauritzen. Markov properties for mixed graphs. *Bernoulli,*, 20(2):676–696, 2014.

Bernd S.W. Schröder. *Ordered Sets: An Introduction*. Birkhäuser, Boston, 2003.

De W. Soh and Sekhar C. Tatikonda. Testing unfaithful Gaussian graphical models. In Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2681–2689. Curran Associates, Inc., 2014.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT press, 2nd edition, 2000.

Daniel Steel. Homogeneity, selection, and the faithfulness condition. *Minds and Machines*, 16(3):303–317, 2006.

Milan Studený. Multiinformation and the problem of characterization of conditional independence relations. *Problems of Control and Information Theory*, 18:3–16, 1989.

Milan Studený. *Probabilistic Conditional Independence Structures*. Springer-Verlag, London, United Kingdom, 2005.

Milan Studený and Remco R. Bouckaert. On chain graph models for description of conditional independence structures. *Annals of Statistics*, 26(4):1434–1495, 1998.

Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *Annals of Statistics*, 41(2):436–463, 2013.

Thomas Verma and Judea Pearl. On the equivalence of causal models. In *Proceedings of the Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-90)*, pages 220–227, New York, NY, 1990.

Nanny Wermuth. Probability distributions with summary graph structure. *Bernoulli*, 17 (3):845–879, 2011.

Nanny Wermuth and David R. Cox. On association models defined over independence graphs. *Bernoulli*, 4:477–495, 1998.

Nanny Wermuth and Kayvan Sadeghi. Sequences of regressions and their independences. *TEST*, 21(2):215–252 and 274–279, 2012.

Nanny Wermuth, Giovanni M. Marchetti, and David R. Cox. Triangular systems for symmetric binary variables. *Electronic Journal of Statistics*, 3:932–955, 2009.

James Woodward. Causal independence and faithfulness. *Multivariate Behavioral Research*, 33(1):129–148, 1998.

Jiji Zhang and Peter Spirtes. Strong faithfulness and uniform consistency in causal inference. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, UAI'03, 2003.

Jiji Zhang and Peter Spirtes. Detection of unfaithfulness and robust causal inference. *Minds and Machines*, 18(2):239–271, 2008.

Journal of Machine Learning Research 18 (2017) 1-49 Submitted 12/16; Revised 9/17; Published 11/17

Community Extraction in Multilayer Networks with Heterogeneous Community Structure

James D. Wilson* JDWILSON4@USFCA.EDU
Department of Mathematics and Statistics
University of San Francisco
San Francisco, CA 94117-1080

John Palowitch* PALOJJ@EMAIL.UNC.EDU
Shankar Bhamidi BHAMIDI@EMAIL.UNC.EDU
Andrew B. Nobel NOBEL@EMAIL.UNC.EDU
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599

Editor: Edo Airoldi

Abstract

Multilayer networks are a useful way to capture and model multiple, binary or weighted relationships among a fixed group of objects. While community detection has proven to be a useful exploratory technique for the analysis of single-layer networks, the development of community detection methods for multilayer networks is still in its infancy. We propose and investigate a procedure, called Multilayer Extraction, that identifies densely connected vertex-layer sets in multilayer networks. Multilayer Extraction makes use of a significance based score that quantifies the connectivity of an observed vertex-layer set through comparison with a fixed degree random graph model. Multilayer Extraction directly handles networks with heterogeneous layers where community structure may be different from layer to layer. The procedure can capture overlapping communities, as well as background vertex-layer pairs that do not belong to any community. We establish consistency of the vertex-layer set optimizer of our proposed multilayer score under the multilayer stochastic block model. We investigate the performance of Multilayer Extraction on three applications and a test bed of simulations. Our theoretical and numerical evaluations suggest that Multilayer Extraction is an effective exploratory tool for analyzing complex multilayer networks. Publicly available code is available at `https://github.com/jdwilson4/MultilayerExtraction`.

Keywords: community detection, clustering, multiplex networks, score based methods, modularity

*JDW is corresponding author. JDW and JP contributed equally to the writing of this paper.

©2017 James D. Wilson, John Palowitch, Shankar Bhamidi and Andrew B. Nobel.

License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v18/16-645.html`.

1. Introduction

Networks are widely used to represent and analyze the relational structure among interacting units of a complex system. In the simplest case, a network model is an unweighted, undirected graph $G = (V, E)$, where V is a set of vertices that represent the units, or *actors*, of the modeled system, and E is an edge set containing all pairs of vertices $\{u, v\}$ such that actors u and v share a physical or functional relationship. Networks have been successfully applied in a wide array of fields, including the social sciences to study social relationships among individuals (Wasserman and Galaskiewicz, 1994), biology to study interactions among genes and proteins (Bader and Hogue, 2003), and neuroscience to study the structure and function of the brain (Sporns, 2011).

In many cases, the vertices of a network can be divided into groups (often disjoint) with the property that there are many edges between vertices in the same group, but relatively few edges between vertices in different groups. Vertex groups of this sort are commonly referred to as *communities*. The unsupervised search for communities in a network is known as *community detection*. Community structure has been used to identify functionally relevant groups in gene and protein interaction systems (Lewis et al., 2010; Parker et al., 2015), structural brain networks (Bassett et al., 2011), and social networks (Onnela et al., 2011; Greene et al., 2010). As communities are often associated with important structural characteristics of a complex system, community detection is a common first step in the understanding and analysis of networks. The search for communities that optimize a given quantitative performance criterion is typically an NP-hard problem, so in most cases one must rely on approximate algorithms to identify community structure.

The focus of this paper is community detection in *multilayer* networks. Formally, an (m, n)-multilayer network is a collection $\mathbf{G}(m, n) = (G_1, \ldots, G_m)$ of m simple graphs $G_\ell = ([n], E_\ell)$ having common vertex set $[n] = \{1, \ldots, n\}$, where the edge sets E_ℓ may vary from layer to layer. The graph G_ℓ will be referred to as the ℓth *layer* of the network. We assume that the vertices of the multilayer network are registered, in the sense that a fixed vertex $u \in [n]$ refers to the same actor across all layers. Thus the graph G_ℓ reflects the relationships between identified actors $1, \ldots, n$ in circumstance ℓ. There are no edges between vertices in different layers, and the layers are regarded as unordered so that the indices $\ell \in [m]$ do not reflect an underlying spatial or temporal order among the layers.

In general, the actors of a multilayer network may not exhibit the same community structure across all layers. For example in social networks, a group of individuals may be well connected via friendships on Facebook; however, this common group of actors will likely, for example, not work at the same company. In realistic situations such as these, a given vertex community will only be present in a subset of the layers, and different communities may be present in different subsets of layers. We refer to such multilayer systems as heterogeneous as each layer may exhibit noticeably different community structure. Complex and differential relationships between actors will be reflected in the heterogenous behavior of different layers of a multilayer network. In spite of this heterogeneity, many existing community detection methods for multilayer networks typically assume that the community structure is the same across all, or a substantial fraction of, the layers (Berlingerio et al., 2011; Rocklin and Pinar, 2013; Barigozzi et al., 2011; Berlingerio et al., 2013; Holland et al., 1983; Paul and Chen, 2015).

We develop and investigate a multilayer community detection method called Multilayer Extraction, which efficiently handles multilayer networks with heterogeneous layers. Theoretical and numerical evaluation of our method reveals that Multilayer Extraction is an effective exploratory tool for analyzing complex multilayer networks. Our contributions to the current literature of statistical analysis of multilayer networks are threefold

1. We develop a testing-based algorithm for identifying densely connected vertex-layer sets (B, L), where $B \subseteq [n]$ is a set of vertices and $L \subseteq [m]$ is a set of layers such that the vertices in B are densely connected across the layers in L. The strength of the connections in (B, L) is measured by a local modularity score derived from a null random network model that is based on the degree sequence of the observed multilayer network. Identified communities can have overlapping vertex or layer sets, and some vertex-layer pairs may not belong to any community. Vertex-layer pairs that are not assigned to any community are interpreted as background as they are not strongly connected to any other. Overlap and background are common features of real networks that can have deleterious effects on partition based methods (Lancichinetti et al., 2011; Wilson et al., 2013, 2014). The Multilayer Extraction procedure directly addresses community heterogeneity in multilayer networks.

2. We assess the consistency of the global optimizer of the aforementioned local modularity score under a generative model for multilayer networks with communities. The generative model studied is a multilayer generalization of the stochastic 2 block model from Snijders and Nowicki (1997); Wang and Wong (1987) for multilayer networks, which we call the Multilayer Stochastic Block Model (MSBM). The MSBM is a generative model that characterizes preferential (or a-preferential) behavior of pre-specified vertex-layer communities in a multilayer network, via specification of inter- and intra-community probabilities. We are able to show that under the MSBM, the number of mis-clustered vertices and layers from the vertex-layer community that maximizes our proposed significance score vanishes to zero, with high probability, as the number of vertices tends to infinity. There has been considerable work in the area of consistency analysis for single-layer networks (e.g. Zhao et al., 2012); however, to the best of our knowledge, we are the first to address the joint optimality properties for *both* vertices and layers in multilayer networks. Furthermore, we provide complete and explicit expressions of all error bounds in the proof, as we anticipate future analyses where the number of layers is allowed to grow with the size of the network. Our proof involves a novel inductive argument, which, to our knowledge, has not been employed elsewhere.

3. We apply Multilayer Extraction to three diverse and important multilayer network types, including a multilayer social network, a transportation network, and a collaboration network. We compare and contrast Multilayer Extraction with contemporary methods, and highlight the advantages of our approach over single layer and aggregate alternatives. Our findings reveal important insights about these three complex relational systems.

1.1 Related Work

Multilayer network models have been applied to a variety of problems, including modeling and analysis of air transportation routes (Cardillo et al., 2013), studying individuals with multiple sociometric relations (Fienberg et al., 1980, 1985), and analyzing relationships between social interactions and economic exchange (Ferriani et al., 2013). Kivelä et al. (2014) and Boccaletti et al. (2014) provide two recent reviews of multilayer networks. We note that $\mathbf{G}(m, n)$ is also sometimes referred to as a *multiplex* network.

While there is a large and growing literature concerning community detection in standard, single-layer, networks (Fortunato, 2010; Newman, 2004; Porter et al., 2009), the development of community detection methods for multilayer networks is still relatively new. One common approach to multilayer community detection is to project the multilayer network in some fashion onto a single-layer network and then identify communities in the single layer network (Berlingerio et al., 2011; Rocklin and Pinar, 2013). A second common approach to multilayer community detection is to apply a standard detection method to each layer of the observed network separately (Barigozzi et al., 2011; Berlingerio et al., 2013). The first approach fails to account for layer-specific community structure and may give an oversimplified or incomplete summary of the community structure of the multilayer network; the second approach does not enable one to leverage or identify common structure between layers. In addition to the methods above, there have also been several generalizations of single-layer methods to multilayer networks. For example, Holland et al. (1983) and Paul and Chen (2015) introduce multilayer generalizations of the stochastic block model from Wang and Wong (1987) and Snijders and Nowicki (1997). Notably, these generative models require the community structure to be the same across layers.

Peixoto (2015) considers a multilayer generalization of the stochastic block model for weighted networks that models hierarchical community structure as well as the degree distribution of an observed network. Paul and Chen (2016) describe a class of null models for multilayer community detection based on the configuration and expected degree model. We utilize a similar model in our consideration. Stanley et al. (2016) considered the clustering of layers of multilayer networks based on recurring community structure throughout the network. Peixoto (2015) and Mucha et al. (2010) generalized the notion of modularity to multilayer networks, and De Domenico et al. (2014) generalized the map equation, which measures the description length of a random walk on a partition of vertices, to multilayer networks. De Domenico et al. (2013) discuss a generalization of the multilayer method in Mucha et al. (2010) using tensor decompositions. Approximate optimization of either multilayer modularity or the map equation results in vertex-layer communities that form a partition of $[n] \times [m]$. By contrast, the communities identified by Multilayer Extraction need not form a partition of $[n] \times [m]$ as some vertices and or layers may be considered background and not significantly associated with any community.

1.2 Overview of the Paper

In the next section we describe the null multilayer random graph model and the scoring of vertex-layer sets. In Section 3 we present and prove theoretical results regarding the asymptotic consistency properties of our proposed score for multilayer networks. Section 5 provides a detailed description of the proposed Multilayer Extraction procedure. We apply

Multilayer Extraction to three real-world multilayer networks and compare and contrast its performance with existing community detection methods in Section 6. In Section 7 we evaluate the performance of Multilayer Extraction on a test bed of simulated multilayer networks. We conclude the main paper with a discussion of future research directions in Section 8. The Appendix is divided into three sections. In Appendix A, we prove supporting lemmas contributing to the results given in Section 3. In Appendix B, we discuss competing methods to Multilayer Extraction. In Appendix C, we give the complete details of our simulation framework.

2. Scoring a Vertex-Layer Group

Seeking a vertex partition that optimizes, or approximately optimizes, an appropriate score function is a standard approach to single layer community detection. Prominent examples of score-based approaches include modularity maximization (Newman, 2006b), likelihood maximization for a stochastic block model (Wang and Wong, 1987), as well as minimization of the conductance of a partition (Chung, 1997). Rather than scoring a partition of the available network, Multilayer Extraction makes use of a significance based score that is applicable to individual vertex-layer sets. Below, we describe the multilayer null model, and then the proposed score.

2.1 The Null Model

Our significance-based score for vertex-layer sets in multilayer networks relies on the comparison of an observed multilayer network with a null multilayer network model. Let $\mathbf{G}(m,n)$ be an observed (m,n)-multilayer network. For each layer $\ell \in [m]$ and pair of vertices $u, v \in [n]$, let

$$x_\ell(u,v) = \mathbb{I}(\{u,v\} \in E_\ell)$$

indicate the presence or absence of an edge between u and v in layer ℓ of $\mathbf{G}(m,n)$. The *degree* of a vertex $u \in [n]$ in layer ℓ, denoted by $d_\ell(u)$, is the number of edges incident on u in G_ℓ. Formally,

$$d_\ell(u) \;=\; \sum_{v\in[n]} x_\ell(u,v).$$

The *degree sequence* of layer ℓ is the vector $\boldsymbol{d}_\ell = (d_\ell(1), \ldots, d_\ell(n))$ of degrees in that layer; the degree sequence of $\mathbf{G}(m,n)$ is the list $\boldsymbol{d} = (\boldsymbol{d}_1, \ldots, \boldsymbol{d}_m)$ containing the degree sequence of each layer in the network.

Let $\mathcal{G}(m,n)$ denote the family of all (m,n)-multilayer networks. Given the degree sequence $\boldsymbol{d}$ of the observed network $\mathbf{G}(m,n)$, we define a multilayer configuration model and an associated probability measure $\mathbb{P}_{\boldsymbol{d}}$ on $\mathcal{G}(m,n)$, as follows. In layer G_1, each node is given $d_1(u)$ half-stubs. Pairs of these edge stubs are then chosen uniformly at random, to form edges until all half-stubs are exhausted (disallowing self-loops and multiple edges). This process is done for every subsequent layer $G_2, \ldots, G_m$ independently, using the corresponding degree sequence from each layer.

In the multilayer network model described above, each layer is distributed according to the configuration model, first introduced by Bollobás and Universitet (1979) and Bender and Canfield (1978). The probability of an edge between nodes u and v in layer ℓ depends

only on the degree sequence $\boldsymbol{d}_\ell$ of the observed graph G_ℓ. The distribution $\mathbb{P}_{\boldsymbol{d}}$ has two complementary properties that make it useful for identifying communities in an observed multilayer network: (i) it preserves the degree structure of the observed network; and (ii) subject to this restriction, edges are assigned at random, without regard to the higher order connectivity structure of the network. Because of these characteristics, the configuration model has long been taken as the appropriate null model against which to judge the quality of a proposed community partition.

The configuration model is the null model which motivates the modularity score of a partition in a network (Newman, 2004, 2006b). Consider a single-layer observed network $\mathbf{G}(n)$ with n nodes and degree sequence $\boldsymbol{d}$. For fixed $K > 0$, let $c_u \in [K]$ be the community assignment of node u. The modularity score of the partition associated with the assignment $c_1, \ldots, c_n$ is defined as

$$M(c_1, \ldots, c_n;\ \mathbf{G}(n)) := \frac{1}{2|E|} \sum_{i\in[K]} \sum_{u<v\in[n]} \left(x(u,v) - \frac{d(u)d(v)}{\sum_{w\in[n]} d(w)} \right) \mathbb{I}(c_u = c_v = i). \quad (1)$$

Above, the ratio $\frac{d(u)d(v)}{\sum_{w\in[n]} d(w)}$ is the approximate expected number of edges between u and v under the configuration model. If the partition $c_1, \ldots, c_n$ represents communities with a large observed intra-edge count relative to what is expected under the configuration model, it receives a high modularity score. The identification of the communities that (approximately) maximize the modularity of a partition is among the most common techniques for community detection in networks.

2.2 Multilayer Extraction Score

Rather than scoring a partition, the Multilayer Extraction method scores individual vertex-layer sets. We define a multilayer node score that is based on the single-layer modularity score (1) and amenable to iterative maximization. We first define a local *set* modularity for a collection of vertices $B \subseteq [n]$ in the layer $\ell \in [m]$:

$$Q_\ell(B) := \frac{1}{n\binom{|B|}{2}^{1/2}} \sum_{u,v\in B: u<v} \left(x_\ell(u,v) - \frac{d_\ell(u)d_\ell(v)}{\sum_{w\in[n]} d_\ell(w)} \right). \quad (2)$$

The scaling term in the equation above is related to the total number of vertices in the network and the total number of possible edges between the vertices in B. This score is one version of the various set-modularities considered in Fasino and Tudisco (2016), and is reminiscent of the *local* modularity score introduced in Clauset (2005).

Our Multilayer Extraction procedure seeks communities that are *assortative* across layers, in the sense that $Q_\ell(B)$ is large and positive for each $\ell \in L$. In light of this, we define the *multilayer set score* as

$$H(B,L) := \frac{1}{|L|} \left(\sum_{\ell\in L} Q_\ell(B)_+ \right)^2, \quad (3)$$

where Q_+ denotes the positive part of Q. Generally speaking, the score acts as a yardstick with which one can measure the connection strength of a vertex-layer set. Large values of the score signify densely connected communities.

We note that the multilayer score $H(B, L)$ is reminiscent of a chi-squared test-statistic computed from $|L|$ samples. That is, under appropriate regularity assumptions on $Q_\ell(B)$, the score in (3) will be approximately chi-squared with one degree of freedom.

3. Consistency Analysis

In this section we give theoretical results establishing the asymptotic consistency of the Multilayer Extraction score introduced in Section 2.2. Explicitly, we show that the global optimizer of the score has bounded misclassification error under a 2-community multilayer stochastic block model.

We note that our results are particular to the performance of the score, rather than the performance of the algorithm to maximize the score. This approach to consistency analysis is common in the community detection literature, particularly for modularity-based methods, as maximizing modularity is NP-hard (Brandes et al., 2006). For instance, Bickel and Chen (2009) showed that under the standard (single-layer) stochastic block model, the global optimizer of the modularity partition score converges (in classification error) to the partition given by the model. Zhao et al. (2012) extended this result to a generalized class of partition scores, which includes both modularity and the SBM likelihood, and established conditions under which global optimizers of those scores are consistent under the *degree-corrected* stochastic block model (an extension of the standard version, see Coja-Oghlan and Lanka, 2009). In Zhao et al. (2011), a global-optimizer consistency result for a *local* set score, one *not* based on the modularity metric, was established in similar fashion. Many other analyses assess the theoretical "detectability" of the community structure in the stochastic block model (e.g. Decelle, Krzakala, Moore, and Zdeborová, 2011; Mossel, Neeman, and Sly, 2012; Nadakuditi and Newman, 2012).

As with the theorems we present in this section, most results from the aforementioned publications establish, or help to establish, asymptotic "with high probability" statements, and apply only to the *global* maximizers of the scores under study. In other words, they do not directly regard in-practice approaches to local maximization of the scores. Regardless, the demonstration of global optimizer consistency for a given score provides the measure of comfort that any local optimization method has a sensible target. The following analyses will serve our method in accordance with this intuition.

3.1 The Multilayer Stochastic Block Model

We assess the consistency of the Multilayer Extraction score under the multilayer stochastic block model (MSBM) with two vertex communities, defined as a probability distribution $\mathbb{P}_{m,n} = \mathbb{P}_{m,n}(\cdot|\boldsymbol{P}, \pi_1, \pi_2)$ on the family of undirected multilayer networks with m layers, n vertices and 2 communities. The distribution is fully characterized by (i) containment probabilities $\pi_1, \pi_2 > 0$, which satisfy $\pi_1 + \pi_2 = 1$, and (ii) a sequence of symmetric 2×2 matrices $\boldsymbol{P} = \{P_1, \ldots, P_m\}$ where $P_\ell = \{P_\ell(i, j)\}$ with entries $P_\ell(i, j) \in (0, 1)$. Under the distribution $\mathbb{P}_{m,n}$, a random multilayer network $\widehat{\boldsymbol{G}}(m, n)$ is generated using two simple steps:

1. A subset of $\lceil \pi_1 n \rceil$ vertices are placed in community 1, and remaining vertices are placed in community 2. Each vertex u in community j is assigned a community label $c_u = j$.

2. An edge is placed between nodes $u, v \in [n]$ in layer $\ell \in [m]$ with probability $P_\ell(c_u, c_v)$, independently from pair to pair and across layers, and no self-loops are allowed.

For a fixed n and m, the community labels $\mathbf{c}_n = (c_1, \ldots, c_n)$ are chosen once and only once, and the community labels are the same across each layer of $\widehat{\boldsymbol{G}}(m, n)$. On the other hand, the inner and intra community connection probabilities (and hence the assortativity) can be different from layer to layer, introducing heterogeneity among the layers. Note that when $m = 1$, the MSBM reduces to the (single-layer) stochastic block model from Wang and Wong (1987).

3.2 Consistency of the Score

We evaluate the consistency of the Multilayer Extraction score under the MSBM described above. Our first result addresses the vertex set maximizer of the score given a fixed layer set $L \subseteq [m]$. Our second result (Theorem 3 in Section 3.2.1) leverages the former to analyze the global maximizer of the score across layers and vertex sets. Explicitly, consider a multilayer network $\widehat{\boldsymbol{G}}(m, n)$ with distribution under the multilayer stochastic block model $\mathbb{P}_{m,n} = \mathbb{P}_{m,n}(\boldsymbol{P}, \pi_1, \pi_2)$. For a fixed vertex set $B \subseteq [n]$ and layer set $L \subseteq [m]$, define the random score by

$$\widehat{H}(B, L) := \frac{1}{|L|} \left(\sum_{\ell \in L} \widehat{Q}_\ell(B)_+ \right)^2,$$

where $\widehat{Q}_\ell(B)$ is the set-modularity of B in layer ℓ under $\mathbb{P}_{m,n}$. Our main results address the behavior of $\widehat{H}(B, L)$ under various assuptions on the parameters of the MSBM.

Toward the first result, for a fixed layer set $L \subseteq [m]$, let $\widehat{B}_{opt}(n)$ denote the node set that maximizes $\widehat{H}(B, L)$ (if more than one set does, any may be chosen arbitrarily). To define the notion of a "misclassified" node, for any two sets $B_1, B_2 \subseteq [n]$ let $d_h(B_1, B_2)$ denote the Hamming distance (rigorously defined as the cardinality of the symmetric difference between B_1 and B_2). We then define the number of misclassified nodes by a set B by

$$\texttt{Error}(B) := \min\{d_h(B, C_1),\ d_h(B, C_2)\}.$$

Note that this definition accounts for arbitrary labeling of the two communities. As the nodes and community assignments are registered across layers, neither d_h nor `Error` depend on the choice of L. Before stating the main theorem, we define a few quantities that will be used throughout its statement and proof:

Definition 1 *Let "*det*" denote matrix determinant. For a fixed layer set $L \subseteq [m]$, define*

$$\delta_\ell := \det P_\ell \quad \delta(L) := \min_{\ell \in [L]} \delta_\ell \quad \pi := (\pi_1, \pi_2)^t \quad \kappa_\ell := \pi^T P_\ell \pi \quad \kappa(L) := \min_{\ell \in [L]} \kappa_\ell \tag{4}$$

We now state the fixed-layer-set consistency result:

Theorem 2 *Fix m and let $\{\widehat{\boldsymbol{G}}(m,n)\}_{n>1}$ be a sequence of multilayer stochastic 2 block models where $\widehat{\boldsymbol{G}}(m,n)$ is a random graph with m layers and n nodes generated under $\mathbb{P}_{m,n}(\cdot|\boldsymbol{P},\pi_1,\pi_2)$. Assume $\pi_1 \leqslant \pi_2$, and that π_1, π_2, and $\mathbb{P}$ do not change with n. Fix a layer set $L \subseteq [m]$. If $\delta(L) > 0$ then there exist constants $A, \eta > 0$ depending on π_1 and $\delta(L)$ such that for all fixed $\varepsilon \in (0,\eta)$,*

$$\mathbb{P}_{m,n}\left(\texttt{Error}\left(\widehat{B}_{opt}(n)\right) < An^{\varepsilon}\log n\right) \geqslant 1 - \exp\left\{-\frac{\kappa(L)^2\varepsilon}{32}n^{\varepsilon}(\log n)^{2-\varepsilon} + \log 4|L|\right\} \quad (5)$$

for large enough n.

Note that the right-hand-side of (5) converges to 1 for all $\varepsilon \in (0,1)$, regardless of η. Furthermore, if $\varepsilon \in (0,\eta)$, we have $n^{\varepsilon} < n^{\varepsilon'}$ for all $\varepsilon' \geqslant \eta$. Therefore, a corollary of Theorem 2 is that for any $\varepsilon \in (0,1)$,

$$\mathbb{P}_{m,n}\left(\texttt{Error}\left(\widehat{B}_{opt}(n)\right) < n^{\varepsilon}\log n\right) \to 1 \ \text{ as } \ n \to \infty.$$

The above statement is perhaps a more illustrative version of the Theorem 2, and shows that the constants A and η play a role only in bounding the convergence rate of the probability.

The proof of Theorem 2 is given in Section 4.1. We note that the assumption that $\pi_1 \leqslant \pi_2$ is made without loss of generality, since the community labels are arbitrary. When $m = 1$, Theorem 2 implies asymptotic $n \to \infty$ consistency in the (single-layer) stochastic block model. In this case, the condition that $\delta_\ell = P_\ell(1,1)P_\ell(2,2) - P_\ell(1,2)^2 > 0$ is a natural requirement on the inner community edge density of a block model. This condition appears in a variety of consistency analyses, including the evaluation of modularity (Zhao et al., 2012). When $m > 1$, Theorem 2 implies the vertex set that maximizes $H(B,L)$ will have asymptotically vanishing error with high probability, given that L is a fixed layer set with *all* layers satisfying $\delta_\ell > 0$.

3.2.1 Consistency of the joint optimizer

Theorem 2 does not address the *joint* optimizer of the score across all vertex-layer pairs. First, we point out that for a fixed $B \subseteq [n]$, the limiting behavior of the score $\widehat{H}(B,L)$ depends on $L \subseteq [m]$ through the layer-wise determinants $\{\delta_\ell : \ell \in [n]\}$ and the scaling constant $\frac{1}{|L|}$ inherent to $H(B,L)$, as defined in equation (3). Let $\gamma : \mathbb{N}^+ \mapsto \mathbb{R}^+$ be a non-decreasing function of $|L|$. Define

$$H_\gamma(B,L) := \frac{1}{\gamma(|L|)}\left(\sum_{\ell\in L} Q_\ell(B)_+\right)^2. \quad (6)$$

and let $\widehat{H}_\gamma(B,L)$ be the corresponding random version of this score under an MSBM. We analyze the joint node-set optimizer of H_γ under some representative choices of γ, an analysis which will ultimately motivate the choice $\gamma(|L|) = |L|$.

We first provide an illustrative example. Consider a MSBM with $m > 1$ layers having the following structure: the first layer has positive determinant, and all $m-1$ remaining layers have determinant equal to 0. Note that $\delta_1 > 0$ implies that the first layer has

ground-truth assortative community structure, and that $\delta_\ell = 0$ for all $\ell > 1$ implies that the remaining layers are (independent) Erdos-Renyi random graphs. In this case, the desired global optimizer of $H_\gamma(B,L)$ is community 1 (or 2) and the first layer. However, setting $\gamma(|L|) \equiv 1$ (effectively ignoring the scaling of H) will ensure that, in fact, the *entire* layer set is optimal, since $Q_\ell(B)_+ \geqslant 0$ by definition. It follows that setting $\gamma(|L|)$ to increase (strictly) in $|L|$, which introduces a penalty on the size of the layer set, is desirable.

For a fixed scaling function γ, define the global joint optimizer of $\widehat{H}(B,L)$ by

$$\left(\widehat{B}_{opt}^{(n)}, \widehat{L}_{opt}^{(n)}\right) := \underset{2^{[n]}\times 2^{[m]}}{\arg\max}\ \widehat{H}_\gamma(B,L). \quad (7)$$

Note that $\left(\widehat{B}_{opt}^{(n)}, \widehat{L}_{opt}^{(n)}\right)$ is random, and may contain multiple elements of $2^{[m]} \times 2^{[n]}$. The next theorem addresses the behavior of $\left(\widehat{B}_{opt}^{(n)}, \widehat{L}_{opt}^{(n)}\right)$ under the MSBM for various choices of $\gamma(|L|)$, and shows that setting $\gamma(|L|) = |L|$ is desirable for consistency.

Theorem 3 *Fix m and let $\{\widehat{\boldsymbol{G}}(m,n)\}_{n>1}$ be a sequence of multilayer stochastic 2 block models where $\widehat{\boldsymbol{G}}(m,n)$ is a random graph with m layers and n nodes generated under $\mathbb{P}_{m,n}(\cdot|\boldsymbol{P},\pi_1,\pi_2)$. Assume $\pi_1 \leqslant \pi_2$, and that π_1, π_2, and $\mathbb{P}$ do not change with n. Fix $0 = \delta^{(0)} < \delta^{(1)} < 1$. Suppose the layer set $[m]$ is split according to $[m] = \cup_{i=0,1} L_i$, where $\delta_\ell = \delta^{(i)}$ for all $\ell \in L_i$. Then for any $\varepsilon > 0$, the following hold:*

(a) Let $\widehat{L}^+ := \{\ell : \widehat{Q}_\ell(\widehat{B}_{opt}^{(n)}) > 0\}$. If $\gamma(|L|) \equiv 1$, then for all $n > 1$, $\widehat{L}_{opt}^{(n)} = \widehat{L}^+$, and

$$\mathbb{P}_{m,n}\left(\texttt{Error}\left(\widehat{B}_{opt}^{(n)}\right) < n^{\varepsilon}\log n\right) \to 1 \ \text{ as } \ n \to \infty.$$

(b) If $\gamma(|L|) = |L|$,

$$\mathbb{P}_{m,n}\left(\widehat{L}_{opt}^{(n)} = L_1,\ \texttt{Error}\left(\widehat{B}_{opt}^{(n)}\right) < n^{\varepsilon}\log n\right) \to 1 \ \text{ as } \ n \to \infty.$$

(c) If $\gamma(|L|) = |L|^2$,

$$\mathbb{P}_{m,n}\left(\widehat{L}_{opt}^{(n)} \subseteq 2^{L_1},\ \texttt{Error}\left(\widehat{B}_{opt}^{(n)}\right) < n^{\varepsilon}\log n\right) \to 1 \ \text{ as } \ n \to \infty.$$

The proof of Theorem 3 is given in Section 4.2. Part (a) implies that setting $\gamma(|L|) \equiv 1$ ensures that the optimal layer set will be, simply, all layers with positive modularity, thereby making this an undesirable choice for the function γ. Part (c) says that if $\gamma(|L|) = |L|^2$, the layer set with the highest *average* layer-wise modularity will be optimal (with high probability as $n \to \infty$), which means that all subsets of L_1 are asymptotically equivalent with respect to $\widehat{H}(B,L)$ (with high probability). By part (b), if $\gamma(|L|) = |L|$, then L_1 is the unique asymptotic maximizer of the population score (with high probability). Therefore, $\gamma(|L|) = |L|$ is the most desirable choice of scaling.

3.3 Discussion of theoretical results

As mentioned previously, the theoretical results in Section 3.2 regard the *global* optimizer of the score. The results have the following in-practice implication: if one simulates from an MSBM satisfying the assumptions of the theorem, and subsequently finds the *global* optimizer of the score, the classification error of the optimizer will be vanishingly small with increasingly high probability (as $n \to \infty$). To illustrate this point, consider an MSBM with $m > 1$ layers, each having community structure $\pi_1 = 0.4$ and for $r \in (0, .95)$, $P(1,1) = P(2,2) = 0.05 + r$ and $P(1,2) = 0.05$. Under this model, $\delta_\ell = r(r + 0.1) > 0$ for all $\ell \in [m]$, and the assumptions of Theorem 3 are satisfied. Therefore, if we were to simulate this MSBM and find the global optimizer of H, with high probability (increasingly, $n \to \infty$) we would recover (1) the correct layer set $[m]$, and (2) the optimal node set will have small classification error (vanishingly, $n \to \infty$).

Of course, in practice it is computationally infeasible to find the global optimizer, so we employ the method laid out in Section 5 to find local maxima. We find through simulation (see Section 7) that our method achieves extremely low error rates for relatively small networks, including on the MSBM described above, for many values of r. These results reflect the intuition that theoretical results for global optimizers should have practical implications when local optimization methods are sufficiently effective.

A limitation of our theoretical results is that they assume an MSBM with only two communities. Furthermore, the model does not allow for any notion of overlap between communities. Nevertheless, in the setting we consider we demonstrate that there is a concise condition on the parameters of the model that guarantees consistency, namely, that $\delta(L) > 0$ for a target layer set L. We expect a similar condition to exist in more complicated settings. In particular, at the outset of Section 4.1, we sketch how the $\delta(L) > 0$ condition relates to the maxima of the *population* version of the score, which roughly speaking is the deterministic, limiting form of the score under the MSBM. We expect that, in more complex settings with (say) more than two communities or overlapping nodes, similar conditions on the model parameters would guarantee that the population version of the score be maximized at the correct community/layer partitions, which (as we show) would entail that the classification error of the global maximizer converges in probability to zero. Though the proofs in such settings would undoubtedly be more complicated, the analyses in this paper should serve as a theoretical foundation. Furthermore, we have empirically analyzed those settings via simulation in Section 7.1 (for more than two communities) and Appendix D (for overlapping communities).

4. Proofs

In this section we prove the theoretical results given in Section 3. The majority of the section is devoted to a detailed proof of Theorem 2 and supporting lemmas. This is followed by the proof of Theorem 3, of which we give only a sketch, as many of the results and techniques contributing to the proof of Theorem 2 can be re-used.

4.1 Proof of Theorem 2, and Supporting Lemmas

We prove Theorem 2 via a number of supporting lemmas. We begin with some notation:

Definition 4 *For a fixed vertex set $B \subseteq [n]$ define*

$$\rho_n(B) = \frac{|B \cap C_{1,n}|}{|B|}, \quad s_n(B) = \frac{|B|}{n}, \quad v_n(B) := (\rho_n(B), 1 - \rho_n(B)). \tag{8}$$

We will at times suppress dependence on n and B in the above expressions.

Definition 5 *Define the **population** normalized modularity of a set B in layer ℓ by*

$$q_\ell(B) := \frac{s_n(B)}{\sqrt{2}} \left(v_n(B)^t P_\ell v_n(B) - \frac{(v_n(B)^t P_\ell \pi)^2}{\kappa_\ell} \right). \tag{9}$$

*Define the **population** score function $H_*(\cdot, L) : 2^{[n]} \mapsto \mathbb{R}$ by*

$$H_*(B, L) = |L|^{-1} \left(\sum_{\ell \in [L]} q_\ell(B) \right)^2. \tag{10}$$

Throughout the results in this section, we assume that $L \subseteq [m]$ is a fixed layer set (as in the statement of Theorem 2). We will therefore, at times, suppress the dependence on L from $\delta(L)$ and $\kappa(L)$ (from Definition 1).

4.1.1 Sketch of the Proof of Theorem 2

The proof of Theorem 2 is involved and broken into many lemmas. In this section, we give a rough sketch of the argument, as follows. The lemmas in this section establish that:

1. $C_{1,n}$ maximizes the population score $H_*(\cdot, L)$ (Lemmas 6 and 7).
2. For large enough sets $B \subseteq [n]$, the random score $\widehat{H}(B, L)$ is bounded in probability around the population score $H_*(B, L)$ (Lemmas 9 and 12).
3. **Inductive Step**: For fixed $k > 1$, assume that $d_h(\widehat{B}_{opt}(n), C_{1,n})/n = O_p(b_{n,k})$, where larger k makes $b_{n,k}$ of smaller order. Then, based on concentration properties of the score, in fact $d_h(\widehat{B}_{opt}(n), C_{1,n})/n = O_p(b_{n,k+1})$ (Lemma 13).
4. There exists a constant η such that for any $\varepsilon \in (0, \eta)$, $d_h(\widehat{B}_{opt}(n), C_{1,n})/n = O_p(n^\varepsilon \log n)$ (Theorem 2). This result is shown using the Inductive Step.

4.1.2 Supporting lemmas for the Proof of Theorem 2

Lemma 6 *Define $\phi(L) := (|L|^{-1} \sum_\ell \frac{\det P_\ell}{2\kappa_\ell})^2$. Then:*

1. *For any $B \subseteq [n]$, $q_\ell(B) = \frac{s_n(B)}{\sqrt{2}} (\pi_1 - \rho_n(B))^2 \cdot \frac{\det P_\ell}{2\kappa_\ell}$, and therefore*

$$H_*(B, L) = |L| \phi(L) \frac{s_n(B)^2}{2} (\pi_1 - \rho_n(B))^4.$$

2. *$\delta(L)^2 \leqslant \phi(L) \leqslant \frac{1}{\pi_1^2 \delta(L)^2}$ and therefore $H_*(C_{1,n}, L) \geqslant |L| \frac{\pi_1^2}{2} (1 - \pi_1^4) \delta(L)^2$.*

Lemma 7 *Fix any $n > 1$. Define*

$$\mathcal{R}(t) := \begin{cases} \{B \subseteq [n] : \max\{|s(B) - \pi_1|,\ 1 - \rho(B)\} \leqslant t\}, & \pi_1 < \pi_2 \\ \{B \subseteq [n] : \max\{|s(B) - \pi_1|,\ \rho(B)\} \leqslant 1 - \rho(B) \leqslant t\}, & \pi_1 = \pi_2 \end{cases}$$

Then there exists a constant $a > 0$ depending just on π_1 such that for sufficiently small t, $B \notin \mathcal{R}(t)$ implies $H_*(B, L) < H_*(C_{1,n}, L) - a|L|\phi(L)t$.

The proofs of Lemmas 6-7 are given in Appendix A. We now give a general concentration inequality for $\widehat{H}(B, L)$, which shows that for sufficiently large sets $B \subseteq [n]$, $\widehat{H}(B, L)$ is close to the population score $H_*(B, L)$ with high probability. This result is used in the proof of Lemma 12, and its proof is given in Appendix A. We first give the following definition:

Definition 8 *For fixed $\varepsilon > 0$ and $n > 1$, define $\mathcal{B}_n(\varepsilon) := \{B \subseteq [n] : |B| \geqslant n\varepsilon\}$.*

Lemma 9 *Fix $\varepsilon \in (0, 1)$. Let κ be as in Definition 1. For each $n > 1$ suppose a collection of node sets $\mathcal{B}_n$ is contained in $\mathcal{B}_n(\varepsilon)$. Then for large enough n,*

$$\mathbb{P}_n\left(\sup_{\mathcal{B}_n}\left(\left|\widehat{H}(B, L) - H_*(B, L)\right|\right) > \frac{4|L|t}{n^2} + \frac{52|L|}{\kappa n}\right) \leqslant 4|L||\mathcal{B}_n| \exp\left(-\kappa^2 \frac{\varepsilon t^2}{16n^2}\right)$$

for all $t > 0$.

We now define new notation that will serve the remaining lemmas:

Definition 10 *Let $\gamma_n := \log n/n$, and for any integer $k > 0$, define $b_{n,k} := \gamma_n^{1-\frac{1}{2^k}}$.*

Definition 11 *For any $r \in [0, 1]$ and $C \subseteq [n]$, define the r-neighborhood of C by $N(C, r) := \{B \subseteq [n] : d_h(B, C)/n \leqslant r\}$. For all $n > 1$, any constant $A > 0$, and fixed $k > 1$, define*

$$N_{n,k}(A) := \begin{cases} N\left(C_1, A \cdot b_{n,k-1}\right) \cup N\left(C_2, A \cdot b_{n,k-1}\right), & k > 1 \\ \mathcal{B}_n(A), & k = 1 \end{cases}$$

Lemma 12, stated below, is a concentration inequality for the random variable $\widehat{H}(B, L)$ on particular neighborhoods of C_1:

Lemma 12 *Fix $\varepsilon \in (0, \pi_1)$ and any constant $A > 0$. For $k > 1$ satisfying $1/2^{k-1} < \varepsilon$, we have for sufficiently large n that*

$$\sup_{B \in N_{n,k}(A)} \left|\widehat{H}(B, L) - H_*(B, L)\right| \leqslant 5|L|b_{n,k}. \tag{11}$$

with probability greater than $1 - 2\exp\{-\frac{\kappa^2\varepsilon}{32} n\gamma_n^{1-\varepsilon} \log(n) + \log 4|L|\}$. The conclusion holds with $k = 1$ if $A = \varepsilon$.

The proof of Lemma 12 is given in Appendix A. We now state and prove the key lemma used to drive the induction step in the proof of Theorem 2:

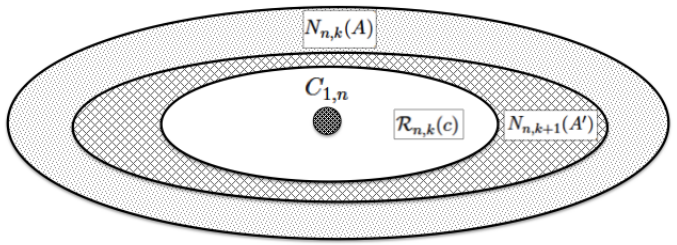


Figure 1: Illustration of relationship between collections of node sets.

Lemma 13 *Fix $\varepsilon \in (0, \pi_1)$ and an integer $k > 1$ satisfying $1/2^{k-1} < \varepsilon$. Suppose there exist constants $A, b > 0$ such that for large enough n,*

$$\mathbb{P}_n\left(\widehat{B}_{opt}(n) \in N_{n,k}(A)\right) \geqslant 1 - b\exp\left\{-\frac{\kappa^2\varepsilon}{32} n\gamma_n^{1-\varepsilon} \log n + \log 4|L|\right\} := 1 - b\beta_n(\varepsilon)$$

Then there exists a constant $A' > 0$ depending only on π_1 and δ such that for large enough n, $\mathbb{P}_n\left(\widehat{B}_{opt}(n) \in N_{n,k+1}(A')\right) \geqslant 1 - (4 + b)\beta_n(\varepsilon)$. The conclusion holds for $k = 1$ if $A = \varepsilon$.

Proof Assume $\pi_1 < \pi_2$; the following argument may be easily adapted to the case where $\pi_1 = \pi_2$, which we explain at the end. Recall $b_{n,k}$ from Definition 10. For $c > 0$, define

$$\mathcal{R}_{n,k}(c) := \left\{B \subset [n] : \max\{|s(B) - \pi_1|,\ 1 - \rho(B)\} \leqslant c \cdot b_{n,k}\right\},$$

Note that sets $B \in \mathcal{R}_{n,k}(c)$ have bounded Hamming distance from $C_{1,n}$, as shown by the following derivation. Writing $s = s(B)$ and $\rho = \rho(B)$, for all $B \in \mathcal{R}_{n,k}(c)$ we have

$$\begin{aligned} n^{-1}|d_h(B, C_{1,n})| &= n^{-1}\left(|B \setminus C_{1,n}| + |C_{1,n} \setminus B|\right) \\ &= n^{-1}\left(|B| - |B \cap C_{1,n}| + |C_{1,n}| - |B \cap C_{1,n}|\right) \\ &= s + \pi_1 - 2\rho s \\ &\leqslant s + (s + c \cdot b_{n,k}) - 2\left(1 - c \cdot b_{n,k}\right)s \\ &= c \cdot b_{n,k} + 2sc \cdot b_{n,k} \leqslant 3c \cdot b_{n,k}. \end{aligned} \tag{12}$$

Therefore, $\mathcal{R}_{n,k}(c) \subseteq N(C_{1,n}, A' \cdot b_{n,k}) \subset N_{n,k+1}(A')$ with $A' = 3c$.

We have assumed $\widehat{B}_{opt}(n) \in N_{n,k}(A)$ with high probability; our aim is to show $\widehat{B}_{opt}(n) \in N_{n,k+1}(A')$. Since $\mathcal{R}_{n,k}(c) \subseteq N_{n,k+1}(A')$, it is sufficient to show that $\widehat{B}_{opt}(n) \notin N_{n,k}(A) \cap \mathcal{R}_{n,k}(c)^c$ with high probability. This is illustrated by figure 1: since $\widehat{B}_{opt}(n)$ is inside the outer oval (with high probability), it is sufficient to show that it cannot be outside the inner

oval. To this end, it is enough to show that, with high probability, $\widehat{H}(B,L) < \widehat{H}(C_{1,n},L)$ for all sets B in $N_{n,k}(A) \cap \mathcal{R}_{n,k}(c)^c$. Note that by Lemma 12,

$$\sup_{B \in N_{n,k}(A)} \widehat{H}(B,L) < H_*(B,L) + 5|L|b_{n,k} \tag{13}$$

for large enough n, with probability at least $1-2\beta_n(\varepsilon)$. Next, since $cb_{n,k} \to 0$ as $n \to \infty$, by Lemma 7 there exists a constant $a > 0$ depending just on π_1 such that for large enough n, $B \in \mathcal{R}_{n,k}(c)^c$ implies $H_*(B,L) < H_*(C_{1,n}) - a|L|\phi(L)cb_{n,k}$. Applying Lemma 12 again, we also have $H_*(C_{1,n},L) < \widehat{H}(C_{1,n}) + 5|L|b_{n,k}$ with probability at least $1-2\beta_n(\varepsilon)$. Furthermore, $\phi(L) \geqslant \delta^2$ by Lemma 6. Applying these inequalities to (13), we obtain

$$\sup_{B \in N_{n,k}(A) \cap \mathcal{R}_{n,k}(c)^c} \widehat{H}(B,L) < \widehat{H}(C_{1,n},L) - a|L|\delta^2 cb_{n,k} + 10|L|b_{n,k} \tag{14}$$

with probability at least $1 - 4\beta_n(\varepsilon)$. With c large enough, (14) implies that $\widehat{H}(B,L) < \widehat{H}(C_{1,n},L)$ for all $B \in N_{n,k}(A) \cap \mathcal{R}_{n,k}(c)^c$. This proves the result in the $\pi_1 < \pi_2$ case.

If $\pi_1 = \pi_2$, the argument is almost identical. We instead define $\mathcal{R}_{n,k}(c)$ as

$$\mathcal{R}_{n,k}(c) := \big\{B \subseteq [n] : \max\{|s(B) - \pi_1|,\ \rho(B),\ 1-\rho(B)\} \leqslant c \cdot b_{n,k}\big\}.$$

A derivation analogous to that giving inequality (12) yields

$$n^{-1}\max\{d_h(B,C_{1,n}),\ d_h(B,C_{2,n})\} \leqslant 3c \cdot b_{n,k},$$

which directly implies that $\mathcal{R}_{n,k}(c) \subseteq N_{n,k+1}(A')$ with $A' = 3c$. The remainder of the argument follows unaltered. ■

4.1.3 Proof of Theorem 2

Recall $Q_\ell(B)$ from Definition 2 and let $\widehat{Q}_\ell(B)$ be its random version under the MSBM, as in Section 3.2. For any $B \subseteq [n]$, we have the inequality

$$\Big\{\widehat{Q}_\ell(B)\Big\}_+ \leqslant \frac{Y_\ell(B)}{n\binom{|B|}{2}^{1/2}} \leqslant \frac{\binom{|B|}{2}}{n\binom{|B|}{2}^{1/2}} \leqslant \frac{|B|}{n}. \tag{15}$$

This yields the following inequality for $\widehat{H}(B,L)$:

$$\widehat{H}(B,L) = |L|^{-1}\left\{\left(\sum_{\ell \in [L]} Q_\ell(B)\right)_+\right\}^2 \leqslant |L|^{-1}\left\{\sum_{\ell \in [L]} Q_\ell(B)_+\right\}^2 \leqslant |L|^{-1}n^{-2}|B|^2. \tag{16}$$

Recall that $\mathcal{B}_n(\varepsilon) := \{B \in 2^{[n]} : |B| \geqslant \varepsilon n\}$. Inequality (16) implies $\widehat{H}(B,L) \leqslant |L|\varepsilon^2$ for all $B \in \mathcal{B}_n(\varepsilon)^c$. By part 2 of Lemma 6, $\phi(L) \geqslant \delta^2$. Therefore, defining $\tau := \frac{\pi_1^2}{2}(1-\pi_1)^4\delta^2/2$,

$$|L|\tau < |L|\phi(L)\frac{\pi_1^2}{2}(1-\pi_1)^4 = H_*(B,L).$$

Therefore, for all $B \in \mathcal{B}_n(\varepsilon)^c$, we have $\widehat{H}(B,L) \leqslant |L|\varepsilon^2 < H_*(C_{1,n},L) - |L|(\tau - \varepsilon^2)$. By Lemma 12, for large enough n we therefore have

$$\sup_{\mathcal{B}_n(\varepsilon)^c} \widehat{H}(B,L) < \widehat{H}(C_{1,n},L) - |L|(\tau - \varepsilon^2) + 5|L|\gamma_n^{1-\varepsilon} \tag{17}$$

with probability greater than $1 - 2\beta_n(\varepsilon)$, where $\beta_n(\varepsilon) := \exp\{-\frac{\kappa^2\varepsilon}{32}n\gamma_n^{1-\varepsilon}\log n + \log 4|L|\}$. For any $\varepsilon < \sqrt{\tau}$, inequality (17) implies $\widehat{H}(B,L) < \widehat{H}(C_{1,n},L)$ for all $B \in \mathcal{B}_n(\varepsilon)$, and therefore $\widehat{B}_{opt}(n) \in \mathcal{B}_n(\varepsilon)$, with probability at least $1 - 2\beta_n(\varepsilon)$. Note that $\varepsilon < \sqrt{\tau} < \pi_1$, and $N_{n,k}(\varepsilon) = \mathcal{B}_n(\varepsilon)$ by Definition 11. Therefore, the conditions for Lemma 13 with $k = 1$ (and $A = \varepsilon$) are satisfied. For any fixed $\varepsilon \in (0,\eta)$ with $\eta := \sqrt{\tau}$, we may now apply Lemma 13 recursively until $1/2^k \leqslant \varepsilon$. This establishes that for sufficiently large n,

$$\mathbb{P}_n\left(\widehat{B}_{opt}(n) \in N_{n,k}(A)\right) \geqslant 1 - (2+4k)\beta_n(\varepsilon). \tag{18}$$

By definition, $\widehat{B}_{opt}(n) \in N_{n,k}(A)$ implies that

$$\texttt{Error}(\widehat{B}_{opt}(n)) := \min_{C = C_1, C_2} d_h(\widehat{B}_{opt}(n), C) \leqslant A \cdot n \cdot b_{n,k}. \tag{19}$$

Since $1/2^k \leqslant \varepsilon$, note that

$$n \cdot b_{n,k} = n\gamma_n^{1-\frac{1}{2^k}} = n \cdot n^{\frac{1}{2^k}-1}(\log n)^{1-\frac{1}{2^k}} < n^\varepsilon \log n.$$

Combined with inequality (18), this completes the proof. ■

4.2 Proof of Theorem 3

To prove part (a), we first note that Theorem 2 implies that on the layer set L_1, for any $\varepsilon > 0$, $\texttt{Error}(\widehat{B}_{opt}^{(n)}) = O_p(n^\varepsilon \log n)$. Lemma 6 can be used to show that $H_*(B,L) = 0$ for any $L \subseteq L_0$ and any $B \subseteq [n]$. Using Lemma 9 and taking a union bound over L_0, it is then straightforward to show (using techniques from the proof of Theorem 2) that on the full layer set $[m]$, for any $\varepsilon > 0$, $\texttt{Error}(\widehat{B}_{opt}) = O_p(n^\varepsilon \log n)$. Considering now $\widehat{L}_{opt}^{(n)}$, observe that if $\widehat{Q}_\ell(B) \leqslant 0$, then $\widehat{H}(B,L) = \widehat{H}(B, L \setminus \{\ell\})$. This immediately implies that $\widehat{L}_{opt}^{(n)} = \widehat{L}^+$.

To prove part (b), we note that it is straightforward to show (using Lemma 6) that $H_*(B,L_1) \geqslant H_*(B,L)$ for any $L \subset [m]$, with equality if and only if $L = L_1$. Using Lemma 9 and a union bound over $[m]$ will show that $\widehat{L}_{opt}^{(n)} = L_1$ with high probability. Applying Theorem 2 completes the part. Part (c) is shown similarly, with the application of Lemma 6 showing that for any $L \subseteq L_1$ and $L' \subseteq [m]$, $H_*(B,L) \geqslant H_*(B,L')$, with equality if and only if $L' \subseteq L_1$. ■

5. The Multilayer Extraction Procedure

The Multilayer Extraction procedure is built around three operations: initialization, extraction, and refinement. In the initialization stage, a family of seed vertex sets is specified.

Next an iterative extraction procedure (**Extraction**) is applied to each of the seed sets. **Extraction** alternately updates the layers and vertices in a vertex-layer community in a greedy fashion, improving the score at each iteration, until no further improvement to the score is possible. The family of extracted vertex-layer communities is then reduced using the **Refinement** procedure, which ensures that the final collection of communities contains the extracted community with largest score, and that the pairwise overlap between any pair of communities is at most β, where $\beta \in [0, 1]$ is a user-defined parameter. The importance and relevance of this parameter is discussed in Section 5.3.1. We describe the Multilayer Extraction algorithm in more detail below.

5.1 Initialization

For each vertex $u \in [n]$ and layer $\ell \in [m]$ let $N(u, \ell) = \{v \in [n] : \{u, v\} \in E_\ell\}$ be the set of vertices connected to u in G_ℓ. We will refer to $N(u, \ell)$ as the neighborhood of u in layer ℓ. Let $\mathcal{B}_0 = \{N(u, \ell), u \in [n], \ell \in [m]\}$ be the family of all vertex neighborhoods in the observed multilayer network $\mathbf{G}(m, n)$. Multilayer Extraction uses the vertex sets in $\mathcal{B}_0$ as seed sets for identifying communities. Our choice of seed sets is motivated by Gleich and Seshadhri (2012), who empirically justified the use of vertex neighborhoods as good seed sets for local detection methods seeking communities with low conductance.

5.2 Extraction

Given an initial vertex set, the **Extraction** procedure seeks a vertex-layer community with large score. The algorithm iteratively conducts a *Layer Set Search* followed by a *Vertex Set Search*, and repeats these steps until a vertex-layer set, whose score is a local maximum, is reached. In each step of the procedure, the score of the candidate community strictly increases, and the procedure is stopped once no improvements to the score are possible. These steps are described next.

Layer Set Search: For a fixed vertex set $B \subseteq [n]$, **Extraction** searches for the layer set L that maximizes $H(B, \cdot)$ using a rank ordering of the layers that depends only on B. In particular let $Q_\ell(B)$ be the local set modularity of layer ℓ from (2). Let L_o be the layer set identified in the previous iteration of the algorithm. We will now update the layer set $L_0 \rightsquigarrow L$. This consists of the following three steps:

(i) Order the layers so that $Q_{\ell_1}(B) \geqslant \cdots \geqslant Q_{\ell_m}(B)$.

(ii) Identify the smallest integer k such that $H(B, \{\ell_1, \ldots, \ell_k\}) \geqslant H(B, \{\ell_1, \ldots, \ell_k, \ell_{k+1}\})$. Write $L_p := \{\ell_1, \ldots, \ell_k\}$ for the proposed change in the layer set.

(iii) *If* $H(B, L_p) > H(B, L_o)$ set $L = L_p$. *Else* set $L = L_o$

In the first iteration of the algorithm (where we take $L_o = \emptyset$), we set $L = L_p$ in step (iii) of the search. The selected layer set L_p is a local maximum for the score $H(B, \cdot)$.

Vertex Set Search: Suppose now that we are given a vertex-layer set (B, L). **Extraction** updates B, one vertex at a time, in a greedy fashion, with updates depending on the layer set L and the current vertex set. In detail, for each $u \in [n]$ let

$$\delta_u(B, L) = \begin{cases} H(B/\{u\}, L) - H(B, L) & \text{if } u \in B \\ H(B \cup \{u\}, L) - H(B, L) & \text{if } u \notin B. \end{cases} \tag{20}$$

Vertex Set Search iteratively updates B using the following steps:

(i) Calculate $\delta_u(B, L)$ for all $u \in [n]$. If $\delta_u(B, L) \leqslant 0$ for all $u \in [n]$, then stop. Otherwise, identify $u^* = \arg\max_{u \in [n]} \delta_u(B, L)$.

(ii) If $u^* \in B$, then remove u^* from B. Otherwise, add u^* to B.

At each iteration of **Extraction**, the score of the updated vertex-layer set strictly increases, and the eventual convergence of this procedure to a local maximum is guaranteed as the possible search space is finite. The resulting local maxima is returned as an extracted community.

5.3 Refinement

Beginning with the n vertex neighborhoods in each layer of the network, the **Extraction** procedure identifies a collection $\mathcal{C}_T = \{(B_t, L_t)\}_{t \in T}$ of at most $m * n$ vertex-layer communities. Given an overlap parameter $\beta \in [0, 1]$, the family $\mathcal{C}_T$ is refined in a greedy fashion, via the **Refinement** procedure, to produce a subfamily $\mathcal{C}_S$, $S \subseteq T$, of high-scoring vertex-layer sets having the property that the overlap between any pair of sets is at most β.

To quantify overlap, we specify a generalized Jaccard match score to measure overlap between two communities. We measure the overlap between two candidate communities (B_q, L_q) and (B_r, L_r) using a generalized Jaccard match score

$$J(q, r) \;=\; \frac{1}{2}\frac{|B_q \cap B_r|}{|B_q \cup B_r|} + \frac{1}{2}\frac{|L_q \cap L_r|}{|L_q \cup L_r|}. \tag{21}$$

It is easy to see that $J(q, r)$ is between 0 and 1. Moreover, $J(q, r) = 1$ if and only if $(B_q, L_q) = (B_r, L_r)$ and $J(q, r) = 0$ if and only if (B_q, L_q) and (B_r, L_r) are disjoint. Larger values of $J(\cdot, \cdot)$ indicate more overlap between communities.

In the first step of the procedure, **Refinement** identifies and retains the community (B_s, L_s) in $\mathcal{C}_T$ with the largest score and sets $S = \{s\}$. In the next step, the procedure identifies the community (B_s, L_s) with largest score that satisfies $J(s, s') \leqslant \beta$ for all $s' \in S$. The index s is then added to S. **Refinement** continues expanding S in this way until no further additions to S are possible, namely when for each $s \in T$, there exists an $s' \in S$ such that $J(s, s') > \beta$. The refined collection $C_S = \{B_s, L_s\}_{s \in S}$ is returned.

5.3.1 Choice of β

Many existing community detection algorithms have one or more tunable parameters that control the number and size of the communities they identify (Von Luxburg, 2007; Leskovec et al., 2008; Mucha et al., 2010; Lancichinetti et al., 2011; Wilson et al., 2014). The family of communities output by Multilayer Extraction depends on the overlap parameter $\beta \in [0, 1]$. In practice, the value of β plays an important role in the structure of the vertex-layer communities. For instance, setting $\beta = 0$ will provide vertex-layer communities that are fully disjoint (no overlap between vertices or layers). On the other hand, when $\beta = 1$ the procedure outputs the full set of extracted communities, many of which may be redundant. In exploratory applications, we recommend investigating the identified communities at multiple values of β, as the structure of communities at different resolutions may

provide useful insights about the network itself (see for instance Leskovec et al. (2008) or Mucha et al. (2010)).

Empirically, we observe that the number of communities identified by the Multilayer Extraction procedure is non-decreasing with β, and there is typically a long interval of β values over which the number and identity of communities remains constant. In practice we specify a default value of β by analyzing the number of communities across a grid of β between 0 and 1 in increments of size 0.01. For fixed i, let $\beta_i = (i-1) * 0.01$ and let $k_i = k(\beta_i)$ denote the number of communities identified at β_i. The default value β' is the smallest β value in the longest stable window, namely

$$\beta' \;=\; \text{smallest } \beta_i \text{ such that } k(\beta_i) = \text{mode}(k_1, \ldots, k_{101}).$$

6. Application Study

In this section, we assess the performance and potential utility of the Multilayer Extraction procedure through an empirical case study of three multilayer networks, including a multilayer social network, transportation network, and collaboration network. We compare and contrast the performance of Multilayer Extraction with six benchmark methods: Spectral Clustering (Newman, 2006a), Label Propagation (Raghavan et al., 2007), Fast and Greedy (Clauset et al., 2004), Walktrap (Pons and Latapy, 2005), multilayer Infomap (De Domenico et al., 2014), and multilayer GenLouvain (Mucha et al., 2010; Jutla et al., 2011). The multilayer Infomap and GenLouvain methods are generalized multilayer methods that can be directly applied to each network considered here. Each of the other four methods have publicly available implementations in the *igraph* package in R and in Python, and each method is a standard single-layer detection method that can handle weighted edges. We apply the first four methods to both the aggregate (weighted) network computed from the average of the layers in the analyzed multilayer network, and to each layer separately. A more detailed description of the competing methods and their parameter settings is provided in the Appendix. For this analysis and the subsequent analysis in Section 7, we set Multlayer Extraction to identify vertex-layer communities that have a large significance score as specified by equation (2).

For each method we calculate a number of quantitative features, including the number and size of the identified communities, as well as the number of identified background vertices. For all competing methods, we define a *background vertex* as a vertex that was placed in a trivial community, namely, a community of size one. We also evaluate the similarity of communities identified by each method. As aggregate and layer-by-layer methods do not provide informative layer information, we compare the vertex sets identified by each of the competing methods with those identified by Multilayer Extraction. To this end, consider two collections of vertex sets $\mathcal{B}, \mathcal{C} \subseteq 2^{[n]}$. Let size$(\mathcal{B})$ denote the number of vertex sets contained in the collection $\mathcal{B}$ and let $|A|$ represent the number of vertices in the vertex set A. Define the coverage of $\mathcal{B}$ by $\mathcal{C}$ as

$$\text{Co}(\mathcal{B};\mathcal{C}) = \frac{1}{\text{size}\,(\mathcal{B})} \sum_{B \in \mathcal{B}} \max_{C \in \mathcal{C}} \left(\frac{|B \cap C|}{|B \cup C|} \right). \tag{22}$$

The value $\text{Co}(\mathcal{B};\mathcal{C})$ quantifies the extent to which vertex sets in $\mathcal{B}$ are contained *in* $\mathcal{C}$. In general, $\text{Co}(\mathcal{B};\mathcal{C}) \neq \text{Co}(\mathcal{C};\mathcal{B})$. The coverage value $\text{Co}(\mathcal{B};\mathcal{C})$ is between 0 and 1, with $\text{Co}(\mathcal{B};\mathcal{C}) = 1$ if and only if $\mathcal{B}$ is a subset of $\mathcal{C}$.

We investigate three multilayer networks of various size, sparsity, and relational types: a social network from an Austrailian computer science department (Han et al., 2014); an air transportation network of European airlines (Cardillo et al., 2013); and a collaboration network of network science authors on arXiv.org (De Domenico et al., 2014). The size and edge density of each network is summarized in Table 1.

Network	# Layers	# Vertices	Total # Edges
AU-CS	5	61	620
EU Air Transport	36	450	3588
arXiv	13	14489	59026

Table 1: Summary of the real multilayer networks in our study.

	AU-CS Network				EU Air Transport Network				arXiv Network			
	# Nodes				# Nodes				# Nodes			
	Comm.	mean (sd)	Back.	Cov.	Comm.	mean (sd)	Back.	Cov.	Comm.	mean (sd)	Back.	Cov.
M-E	6	8.7(2.3)	11	1.00	11	13.1(5.0)	358	1.00	272	8.2(3.5)	12412	1.00
Fast A.	5	12.2(2.6)	0	0.68	8	52.1(54.6)	33	0.10	1543	9.1(55.5)	424	0.27
Spectral A.	7	8.7(5.4)	0	0.54	8	52.0(39.6)	34	0.10	1435	9.8(222.6)	424	0.15
Walktrap A.	6	10.2(3.3)	0	0.75	15	22.9(46.5)	107	0.15	2238	6.3(36.7)	424	0.50
Label A.	6	10.2(6.5)	0	0.59	4	104.3(190.5)	33	0.09	2329	6.0(14.4)	424	0.59
Fast L.	6	7.7(4.2)	16.2	0.78	4	13.1(12.4)	395	0.34	301	6.9(27.8)	12428	0.68
Spectral L.	6	8.0(5.0)	16.4	0.72	3	16.1(13.9)	398	0.30	283	7.2(38.1)	12457	0.66
Walktrap L.	7	6.6(5.1)	16.2	0.74	5	10.2(12.8)	404	0.34	353	5.8(15.4)	12428	0.76
Label L.	5	9.7(9.6)	16.2	0.76	2	27.8(26.8)	395	0.29	383	5.4(6.7)	12428	0.79
GenLouvain	6	50.8(28.4)	0	0.50	9	49.6(26.1)	422	0.64	402	9.1(4.5)	12101	0.83
Infomap	20	7.0(4.9)	0	0.74	41	2.3 (2.4)	33	0.21	3655	7.1(109.1)	423	0.29

Table 2: Quantitative summary of the identified communities in each of the three real applications. Shown are the number of communities (Comm.), the mean and standard deviation of the number of nodes in each community, the number of background nodes per layer (Back.) and the coverage of the M-E with the method (Cov.). Methods run on the aggregate network are followed A. and layer-by-layer methods are followed by L.

Table 2 provides a summary of the quantitative features of the communities identified by each method. For ease of discussion, we will abbreviate Multilayer Extraction by M-E in the next two sections of the manuscript, where we evaluate the performance of the method on real and simulated data.

6.1 AU-CS Network

Individuals interact across multiple modes: for example, two people may share a friendship, business partnership, college roommate, or a sexual relationship, to name a few. Thus multilayer network models are particularly useful for understanding social dynamics. In such

a model, vertices represent the individuals under study and layers represent the type of interaction among individuals. Correspondingly, vertex-layer communities represent groups of individuals that are closely related across a subset of interaction modes. The number and type of layers in a vertex-layer community describes the strength and type of relationships that a group of individuals share.

We demonstrate the use of M-E on a social network by investigating the AU-CS network, which describes online and offline relationships of 61 employees of a Computer Science research department in Australia. The vertices of the network represent the employees in the department. The layers of the network represent five different relationships among the employees: *Facebook*, *leisure*, *work*, *co-authorship*, and *lunch*.

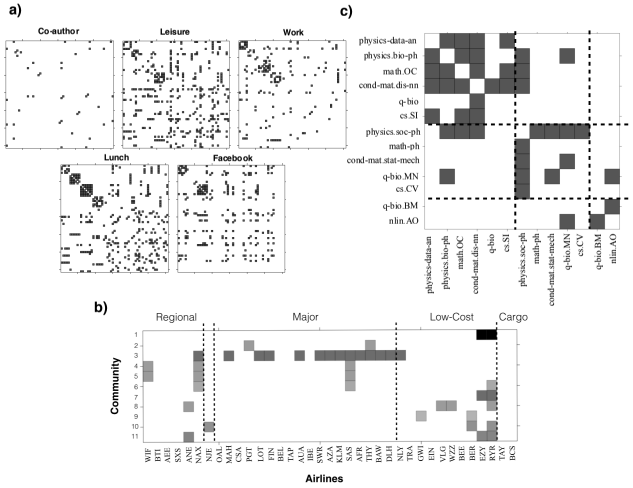


Figure 2: **a)** The AU-CS multilayer network. The vertices have been reordered based on the six communities identified by M-E. **b)** The layers of the eleven extracted significant communities identified in the EU transport network. The layers are ordered according to the type of airline. The darkness of the shaded in blocks represents the score of the identified community. **c)** Adjacency matrix of layers in the arXiv network, where edges are placed between layers that were contained in one or more of the communities identified by M-E. Dotted lines separate three communities of submission types that were identified using spectral clustering.

Results

M-E identified 6 non-overlapping vertex-layer communities, which are illustrated in Figure 2 **a**. These communities reveal several interesting patterns among the individuals in the network. Both the *work* and *lunch* layers were contained in all six of the identified

communities, reflecting a natural co-occurrence of work and lunch interactions among the employees. Of the competing methods, GenLouvain was the only other method to identify this co-occurrence. Furthermore, two of the identified communities by M-E contained the *leisure* and *Facebook* layers, both of which are social activities that sometimes extend beyond work and lunch. Thus these communities likely represent groups of employees with a stronger friendship than those that were simply colleagues. These interpretable features identified by M-E provide an example of how our method can be used to provide useful insights beyond aggregate and layer by layer methods.

With the exception of Infomap, all of the methods identify a similar number of communities (ranging from 5 to 7). Infomap, on the other hand, identified 20 small communities in the multilayer network. The 11 background vertices identified by M-E were sparsely connected, having two or fewer connections in 3 of the layers. As seen by the coverage measure in Table 2, the vertex sets identified by M-E were similar to those identified by the single-layer methods as well as by Infomap. Furthermore, the communities identified by the aggregate approaches are in fact well contained in the family identified by M-E (average coverage = 0.78). In summary, the vertex sets identified by M-E reflect both the aggregate and separate layer community structure of the network, and the layer sets reveal important features about the social relationships among the employees.

6.2 European Air Transportation Network

Multilayer networks have been widely used to analyze transportation networks (Strano et al., 2015; Cardillo et al., 2013). Here, vertices generally represent spatial locations (e.g., an intersection of two streets, a railroad crossing, GPS coordinates, or an airport) and layers represent transit among different modes of transportion (e.g., a car, a subway, an airline, or bus). The typical aim of multilayer analysis of transportation networks is to better understand the efficiency of transit in the analyzed location. Vertex-layer communities in transportation networks contain collections of vehicles (layers) that frequently travel along the same transit route (vertices). Vertex-layer communities reveal similarity, or even redundancy, in transportation among various modes of transportation and enable optimization of travel efficiency in the location.

In the present example, we use M-E to analyze the European air transportation network, where vertices represent 450 airports in Europe and layers represent 37 different airlines. An edge in layer j is present between two airports if airline j traveled a direct flight between the two airports on June 1st, 2011. Notably here, each airline belongs to one of five classes: *major* (18); *low-cost* (10); *regional* (6); *cargo* (2); and *other* (1). A multiplex visualization of this network is shown in Figure 3.

Results

The layer sets of the extracted M-E communities are illustrated in Figure 2 **b**, and a summary of M-E and the competing methods is available in the second major column of Table 2. M-E identified 11 small communities (mean number of vertices = 13.1, mean number of layers = 3.73). This suggests that the airlines generally follow distinct routes containing a small number of unique airports. Furthermore, Figure 2 illustrates that the layers of each community are closely associated with airline classes. Indeed, an average

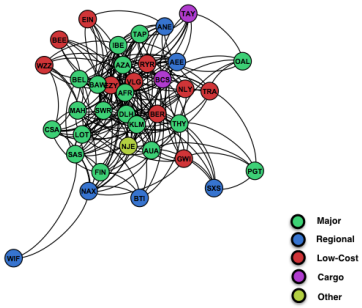


Figure 3: A one-dimensional visualization of the European Air Transportation Network. Edges are placed between airlines that share at least two routes between airports.

of 78 % of the layers in each community belonged to the same airline class. This reflects the fact that airlines of the same class tend to have direct flights between similar airports. Together these two findings suggest that in general there is little redundancy among the travel of airlines in Europe but that airlines of a similar class tend to travel the same routes.

Interestingly, the *regional* airline Norwegian Air Shuttle (NAX) and the *major* airline Scandanavian Airlines (SAS) appeared together in 4 unique communities. These airlines are in fact the top two air carriers in Scandanavia and fly primarily to airports in Norway, Sweden, Denmark, and Finland. Thus, M-E reveals that these two airlines share many of the same transportation routes despite the fact that they are different airline classes.

In comparison to the competing methods, we find that both the single-layer methods and M-E identified on the order of 400 background vertices ($\approx 89\%$), which suggests that many of the airports are not frequented by multiple airlines. This finding aligns with the fact that many airports in Europe are small and do not service multiple airline carriers. Aggregate detection approaches, as well as multilayer GenLouvain, identified a similar number of communities as M-E. In fact, the results of M-E most closely matched those of GenLouvain (Coverage = 0.64). We found that Infomap again identified the most communities (almost 4 times as many as any other method) and, like the aggregate approaches, identified few background vertices.

6.3 arXiv Network

Our final demonstration of M-E is on the multilayer collaboration arXiv network from De Domenico et al. (2014). In a multilayer representation of a collaboration network, vertices represent researchers or other possible collaborators, and layers represent scientific fields or sub-fields under which researchers collaborated. For these applications, multilayer networks provide information about the dissemination and segmentation of scientific research, including the possible overlap of collaborative work under different scientific fields.

Vertex-layer communities in collaborative networks represent groups of individuals who collaborated with one another across a subset of scientific fields. Such communities describe how differing scientific fields overlap, which collaborators are working in a similar area, as well as how to disseminate research across fields and how scientists can most easily take part in interdisciplinary collaborations.

The arXiv network that we analyze represents the authors of all arXiv submissions that contained the word "networks" in its title or abstract between the years 2010 and 2012. The network has 14489 vertices representing authors, and 13 layers representing the arXiv category under which the submission was placed. An edge is placed between two authors in layer ℓ if they co-authored a paper placed in that category. The network is sparse, with each layer having edge density less than 1.5%.

Results

M-E identified 272 multilayer communities, with an average of 2.39 layers per community. The communities were small in size, suggesting that network science collaboration groups are relatively tightly-knit, both in number of authors and number of differing fields. In Figure 2 **c**, we plot an adjacency matrix for layers whose (i, j) entry is 1 if and only if layers i and j were contained in at least one multilayer community. Using the adjacency matrix, the layers of the network were partitioned into communities using Spectral clustering. This figure identifies the existence of three active interdisciplinary working groups among the selected researchers. These results suggest two important insights. First, network researchers can identify his or her primary sub-field community and best disseminate research in this area by communicating with researchers from other fields in the same community. Second, Figure 2 **c** illustrates clear separation of three major areas of study. By promoting cross-disciplinary efforts between each of these three major areas, the dissemination of knowledge among network scientists will likely be much more efficient.

Infomap and the aggregate approaches identify on the order of 1000 small to moderately sized communities (mean number of vertices between 6.04 and 9.80) with approximately 423 background vertices. Notably, the 423 background vertices identified by each of these methods had a 0.92 match. On the other hand M-E, GenLouvain and the single layer approaches identify a smaller number of communities (between 272 and 402), and classify about 12 thousand (roughly 86%) of the vertices as background (of which share a match of 0.9). These findings suggest that the individual layers of the arXiv network have heterogeneous community structure, and that they contain many non-preferentially attached vertices. Once again, GenLouvain had the highest match with M-E (coverage = 0.83).

7. Simulation Study

As noted above, Multilayer Extraction has three key features: it allows community overlap; it identifies background; and it can identify communities that are present in a small subset of the available layers. Below we describe a simulation study that aims to evaluate the performance of M-E with respect to these features. The results of additional simulations are described in the Appendix.

In this simulation study, we are particularly interested in comparing the performance of M-E with off-the-shelf aggregate and layer-by-layer methods. We make this comparison

using the layer-by-layer and aggregate approaches using the community detection methods described in Section 6. Define the *match* between two vertex families $\mathcal{B}$ and $\mathcal{C}$ by

$$M(\mathcal{B};\mathcal{C}) \;=\; \frac{1}{2}\,\mathrm{Co}(\mathcal{B};\mathcal{C}) \;+\; \frac{1}{2}\,\mathrm{Co}(\mathcal{C};\mathcal{B}), \tag{23}$$

where $\mathrm{Co}(\mathcal{B};\mathcal{C})$ is the coverage measure for vertex families from (22). The match $M(\mathcal{B};\mathcal{C})$ is symmetric and takes values in $[0,1]$. In particular, $M(\mathcal{B};\mathcal{C}) = 1$ if and only if $\mathcal{B} = \mathcal{C}$ and $M(\mathcal{B};\mathcal{C}) = 0$ if and only if $\mathcal{B}$ and $\mathcal{C}$ are disjoint.

In our simulations, we compute the match between the family of vertex sets identified by each method and the family of true simulated communities. For the layer-by-layer methods, we evaluate the average match of the communities identified in each layer. Suppose that $\mathcal{T}$ is the true family of vertex sets in a simulation and $\mathcal{D}$ is the family of vertex sets identified by a detection procedure of interest. Note that the value $\mathrm{Co}(\mathcal{D};\mathcal{T})$ quantifies the specificity of $\mathcal{D}$, while $\mathrm{Co}(\mathcal{T};\mathcal{D})$ quantifies its sensitivity; thus, $M(\mathcal{D};\mathcal{T})$ is a quantity between 0 and 1 that summarizes both the sensitivity and specificity of the identified vertex sets $\mathcal{D}$. The results of the simulation study are summarized in Figures 4, 5, and 6 and discussed in more detail below.

7.1 Multilayer Stochastic Block Model

In the first part of the simulation study we generated multilayer stochastic block models with $m \in \{1, 5, 10, 15\}$ layers, $k \in \{2, 5\}$ blocks, and $n = 1000$ vertices such that each layer has the same community structure. In more detail, each vertex is first assigned a community label $\{1, \ldots, k\}$ according to a probability mass function $\pi = (0.4, 0.6)$ for $k = 2$ and $\pi = (0.2, 0.1, 0.2, 0.1, 0.4)$ for $k = 5$. In each layer, edges are assigned independently, based on vertex community membership, according to the probability matrix P with entries $P(i,i) = r + 0.05$ and $P(i,j) = 0.05$ for $i \neq j$. Here r is a parameter representing connectivity strength of vertices within the same community. The resulting multilayer network consists of m independent realizations of a stochastic k block model with the same communities. For each value of m and k we vary r from 0.00 to 0.10 in increments of 0.005. M-E and all other competing methods are run on ten replications of each simulation. The average match of each method to the true communities is given in Figure 4.

Results

In the single-layer ($m = 1$) setting M-E is competitive with the existing single-layer and multilayer methods for $r \geqslant 0.05$, and identifies the true communities without error for $r \geqslant 0.06$. For $m \geqslant 5$ M-E outperforms all competing single-layer methods for $r \geqslant 0.02$. As the number of layers increases, M-E and the multilayer Infomap and GenLouvain methods exhibit improved performance across all values of r. For the two community block model, M-E and the other multilayer methods have comparable performance in networks with five or more layers. M-E outperforms the single-layer and multilayer methods in the $k = 5$ block model when $m > 1$. As expected, aggregate approaches perform well in this simulation, outperforming or matching other methods when $m \leqslant 5$ (results not shown). These results suggest that in homogeneous multilayer networks M-E can outperform or match existing methods when the network contains a moderate to large number of layers.

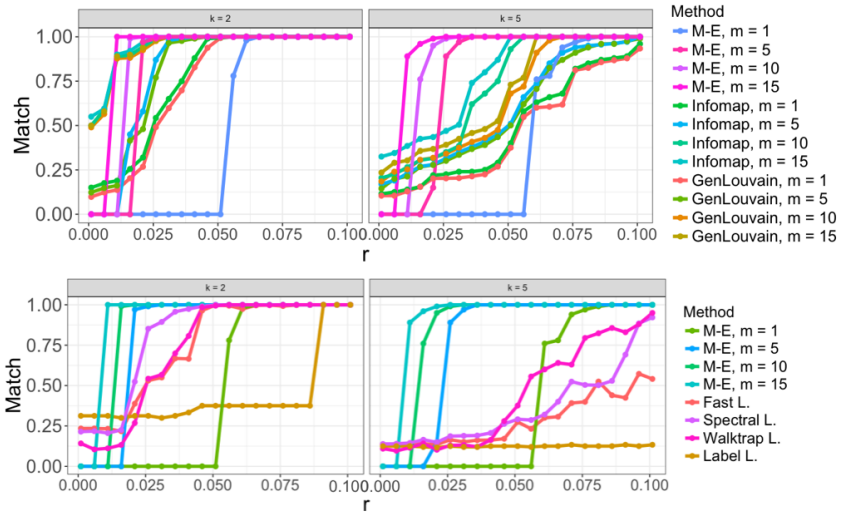


Figure 4: (Color) Simulation results for multilayer stochastic block model. In each plot, we report the match of the identified communities with the true communities where the match is calculated using the match score in (23).

7.2 Persistence

In the second part of the simulation study we consider multilayer networks with heterogeneous community structure. We simulated networks with 50 layers and 1000 vertices. The first $\tau * 50$ layers follow the stochastic block model outlined in Section 7.1 with a fixed connection probability matrix P having entries $P(i,i) = 0.15$ and $P(i,j) = 0.05$ for $i \neq j$. The remaining $(1-\tau)*50$ layers are independent Erdős-Rényi random graphs with $p = 0.10$, so that in each layer every pair of vertices is connected independently with probability 0.10. For each $k \in \{2, 5\}$ we vary the persistence parameter τ from 0.02 to 1 in increments of 0.02, and for each value of τ, we run M-E as well as the competing methods on ten replications. The average match of each method is reported in Figure 5.

Results

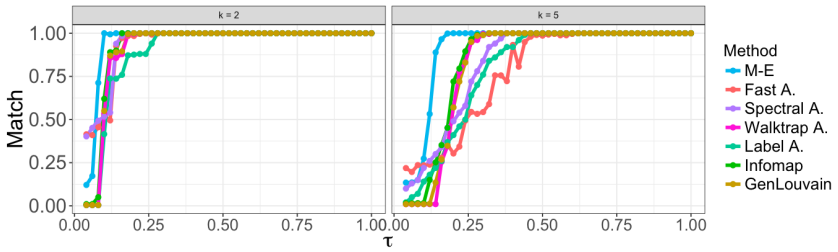


Figure 5: (Color) Simulation results persistence simulations. In each plot, we report the match of the identified communities with the true communities where the match is calculated using the match score in (23).

In both block model settings with $k = 2$ and 5 communities, M-E outperforms competing aggregate and multilayer methods for small values of τ. At these values, aggregate methods perform poorly since the community structure in the layers with signal is hidden by the noisy Erdős-Rényi layers once the layers are aggregated. Though not shown in Figure 6, the layer-by-layer methods are able to correctly identify the community structure of the layers with signal. However, these methods identify on average of 4 or more non-trivial communities in each noisy layer where there is in fact no community structure present. Multilayer GenLouvain and Infomap outperform aggregate methods in this simulation, suggesting that available multilayer methods are able to better handle multilayer networks with heterogeneous layers. Whereas the noisy Erdős-Rényi layers posed a challenge for both single-layer and aggregate methods, M-E never included any of these layers in an identified community. These results highlight M-E's ability to handle networks with noisy and heterogeneous layers.

7.3 Single Embedded Communities

We next evaluate the ability of M-E to detect a single embedded community in a multilayer network. We construct multilayer networks with $m \in \{1, 5, 10, 15\}$ layers and 1000 vertices according to the following procedure. Each layer of the network is generated by embedding a common community of size $\gamma * 1000$ in an Erdős-Rényi random graph with connection probability 0.05 in such a way that vertices within the community are connected independently with probability 0.15. The parameter γ is varied between 0.01 and 0.20 in increments of 0.005; ten independent replications of the embedded network are generated for each γ. For each method, we calculate the coverage $C(E, \mathcal{C})$ of the true embedded community E by the identified collection $\mathcal{C}$. We report the average coverage over the ten replications in Figure 6.

Results

In the single layer setting, M-E is able to correctly identify the embedded community when the embedded vertex set takes up approximately 11 percent ($\gamma = 0.11$) of the layer. As before, the performance of M-E greatly improves as the number of layers in the observed multilayer network increases. For example at $m = 5$ and $m = 10$, the algorithm correctly identifies the embedded community (with at least 90% match) once the community has size taking as little as 6 percent ($\gamma = 0.055$) of each layer. At $m = 15$, M-E correctly extracts communities with size as small as three percent of the graph in each layer.

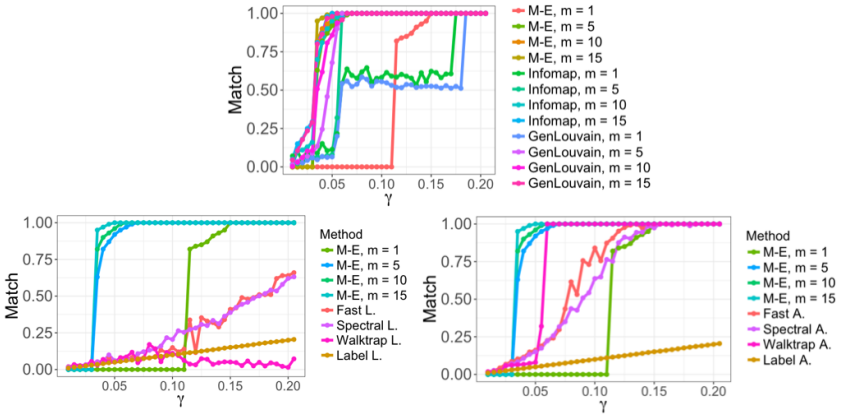


Figure 6: (Color) Simulation results for single embedded simulations. In each plot, we report the match of the identified communities with the true communities where the match is calculated using the match score in (23).

In the lower right plot of Figure 6, we illustrate the results for the aggregate methods applied to the simulated network with $m = 15$ layers. In the lower left plot of Figure 6, we show the results for the layer-by-layer methods in this simulation and the upper part of the figure show the results for the multilayer methods. For $m \geqslant 5$ M-E outperforms all of the competing aggregate methods. In addition, M-E outperforms every layer-by-layer method for all m. Multilayer GenLouvain and Infomap have comparable performance to M-E across all m and γ. These results emphasize the extraction cabilities of M-E and show that the procedure, contrary to aggregate and single-layer competing methods, is able to detect small embedded communities in the presence of background vertices.

8. Discussion

Multilayer networks have been profitably applied to a number of complex systems, and community detection is a valuable exploratory technique to analyze and understand networks. In many applications, the community structure of a multilayer network will differ from layer to layer due to heterogeneity. In such networks, actors interact in tightly connected groups that persist across only a subset of layers in the network. In this paper we have introduced and evaluated the first community detection method to address multilayer networks with heterogeneous communities, Multilayer Extraction. The core of Multilayer Extraction is a significance score that quantifies the connection strength of a vertex-layer set by comparing connectivity in the observed network to that of a fixed degree random network.

Empirically, we showed that Multilayer Extraction is able to successfully identify communities in multilayer networks with overlapping, disjoint, and heterogeneous community structure. Our numerical applications revealed that Multilayer Extraction can identify relevant insights about complex relational systems beyond the capabilities of existing detection methods. We also established asymptotic consistency of the global maximizer of the Multilayer Extraction score under the multilayer stochastic block model. We note that in practice the Multilayer Extraction procedure can identify overlapping community structure in multilayer networks; however, our consistency results apply to a multilayer model with non-overlapping communities. Future work should investigate consistency results like the ones derived here on a multilayer model with overlapping communities, such as a multilayer generalization of the mixed membership block model introduced in Airoldi et al. (2008). We expect that similar theory will hold in an overlapping model and that the theoretical techniques utilized here can be used to prove such results.

The Multilayer Extraction method provides a first step in understanding and analyzing multilayer networks with heterogeneous community structure. This work encourages several interesting areas of future research. For instance, the techniques used in this paper could be applied, with suitable models, to networks having ordered layers (e.g. temporal networks), as well as to networks with weighted edges such as the recent work done in Palowitch et al. (2016); Wilson et al. (2017). Furthermore, one could incorporate both node- and layer-based covariates in the null model to handle exogenous features of the multilayer network. Finally, it would be interesting to evaluate the consistency of Multilayer Extraction in multilayer networks in the high dimensional setting where the number of vertex-layer communities grows with the number of vertices.

Acknowledgements

The authors gratefully acknowledge Peter Mucha for helpful discussions and suggestions for this work. We also thank the associate editor and the two anonymous referees whose comments greatly improved the content and exposition of the paper. The work of JDW was supported in part by NSF grants DMS-1105581, DMS-1310002, and SES grant 1357622. The work of JP was supported in part by NIH/NIMH grant R01-MH101819-01. The work of SB was supported in part by NSF grants DMS-1105581, DMS-1310002, DMS-160683, DMS-161307, SES grant 1357622, and ARO grant W911NF-17-1-0010. The work of ABN

was supported in part by NSF DMS-1310002, NSF DMS-1613072, as well as NIH HG009125-01, NIH MH101819-01. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

Appendix A. Proofs of Lemmas from Section 3.2

A.1 Proof of Lemma 6

It is easy to show that for any 2×2 symmetric matrix A and 2-vectors $\mathbf{x}$, $\mathbf{y}$,

$$(\mathbf{x}^T A\mathbf{x})(\mathbf{y}^T A\mathbf{y}) - (\mathbf{x}^T A\mathbf{y})^2 = (\mathbf{x}_1\mathbf{y}_2 - \mathbf{x}_2\mathbf{y}_1)^2 \det(A).$$

Fix $B \subseteq [n]$ and let s, ρ, and v correspond to B, as in Definition 4. Then for any $\ell \in [L]$, using the fact that $\kappa_\ell := \pi^T P_\ell \pi$ and the identity above, we have

$$v^t P_\ell v - \frac{(\pi^t P_\ell \pi)^2}{\kappa_\ell} = \frac{\kappa_\ell v^t P_\ell v}{\kappa_\ell} - \frac{(v^t P_\ell \pi)^2}{\kappa_\ell} = \frac{(\pi^t P_\ell \pi)(v^t P_\ell v) - (v^t P_\ell \pi)^2}{\kappa_\ell}$$

$$= (\pi_1(1-\rho) - \pi_2\rho)^2 \frac{\det P_\ell}{\kappa_\ell} = (\pi_1 - \rho)^2 \frac{\det P_\ell}{\kappa_\ell}.$$

Recall that $q_\ell(B) := \frac{s}{\sqrt{2}}\left(v^t P_\ell v - (\pi^t P_\ell \pi)^2/\kappa_\ell\right)$ and $H_*(B, L) = |L|^{-1}\left(\sum_\ell q_\ell(B)\right)^2$. Part 1 follows by summation over L. For part 2, note that $\pi_1^2 P_\ell(1,1) + \pi_2^2 P(2,2) \geqslant 2\pi_1\pi_2\sqrt{P_\ell(1,1)P_\ell(2,2)}$. Therefore,

$$\begin{aligned}
\kappa_\ell &= \pi_1^2 P_\ell(1,1) + 2\pi_1\pi_2 P_\ell(1,2) + \pi_2^2 P(2,2) \\
&\geqslant 2\pi_1\pi_2\left(\sqrt{P_\ell(1,1)P_\ell(2,2)} + P_\ell(1,2)\right) \\
&\geqslant 2\pi_1\pi_2\left(\sqrt{P_\ell(1,1)P_\ell(2,2)} + P_\ell(1,2)\right)\left(\sqrt{P_\ell(1,1)P_\ell(2,2)} - P_\ell(1,2)\right) \\
&= 2\pi_1\pi_2\delta \geqslant \pi_1\delta.
\end{aligned}$$

Thus $\delta \leqslant \frac{\det P_\ell}{\kappa_\ell} \leqslant \frac{1}{\pi_1\delta}$. Part 2 follows. ■

A.2 Proof of Lemma 7

Define $g : 2^{[n]} \mapsto \mathbb{R}$ by $g(B) := \frac{s(B)}{2}(\rho(B) - \pi_1)^4$. Recall the function $\phi(L)$ defined in Lemma 6. Note that part 1 of Lemma 6 implies $H_*(B, L) = |L|\phi(L)g(B)$. It is therefore sufficient to show that there exists a constant $a > 0$ such that for sufficiently small t, $B \in \mathcal{R}(t)^c$ implies $g(B) < g(C_{1,n}) - at$. We will show this separately for the $\pi_1 < \pi_2$ and $\pi_1 = \pi_2$ cases.

Part 1 ($\pi_1 < \pi_2$): Define the intervals $I_1 := [0, \pi_1]$, $I_2 := (\pi_1, \pi_2]$, and $I_3 := (\pi_2, 1]$. We trisect $2^{[n]}$, the domain of g, with the collections $\mathcal{D}_{i,n} := \{B \subseteq [n] : s(B) \in I_i\}$, for $i = 1, 2, 3$. We will prove that the inequality $g(B) < g(C_{1,n}) - at$ holds for all $B \in \mathcal{R}(t)$ on each of those collections. We will continually rely on the fact that $B \in \mathcal{R}(t)$ implies at least one of the inequalities (I) $|s(B) - \pi_1| > t$ or (II) $1 - \rho(B) < t$ is true.

Suppose $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{1,n}$ and inequality (I) is true. Then $s(B) < \pi_1 - t$, and

$$\begin{aligned} g(B) := \frac{s(B)^2}{2}(\rho(B) - \pi_1)^4 &\leqslant \frac{s(B)^2}{2}(1-\pi_1)^4 \text{ (since } \pi_1 \leqslant 1/2) \\ &< \frac{(\pi_1 - t)^2}{2}(1-\pi_1)^4 = \frac{\pi_1^2}{2}(1-\pi_1)^4 - 2t(1-\pi_1)^4 + o(t) \\ &< \frac{\pi_1^2}{2}(1-\pi_1)^4 - t(1-\pi_1)^4 = g(C_{1,n}) - t(1-\pi_1)^4 \end{aligned} \tag{24}$$

for sufficiently small t. If inequality (II) is true, then

$$(\rho(B) - \pi_1)^4 \leqslant \max\{(1 - t - \pi_1)^4, \pi_1^4\} = \max\{(\pi_2 - t)^4, \pi_1^4\} = (\pi_2 - t)^4$$

for sufficiently small t, as $\pi_1 < \pi_2$. Therefore,

$$g(B) \leqslant \frac{\pi_1^4}{2}(\pi_2 - t)^4 = \frac{\pi_1^4}{2}\pi_2^4 - 4\pi_2^3 t + o(t) < g(C_{1,n}) - 2\pi_2^3 t \tag{25}$$

for sufficiently small t. Thus for all $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{1,n}$, $g(B) < g(C_{1,n}) - a_1 t$ with $a_1 = \min\{(1-\pi_1)^4, 2\pi_2^3\}$.

Suppose $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{2,n}$ and inequality (I) is true. Then $s(B) > \pi_1 + t$. Note that $0 \leqslant \rho(B)|B| \leqslant |C_{1,n}|$, yielding the useful inequality

$$0 \leqslant \rho(B) \leqslant \pi_1 / s(B). \tag{26}$$

Subtracting through by π_1 gives

$$(\rho(B) - \pi_1)^4 \leqslant \max\{\pi_1^4, \pi_1^4(1/s(B) - 1)^4\} = \pi_1^4(1/s(B) - 1)^4.$$

Therefore,

$$\begin{aligned} g(B) \leqslant \frac{s(B)^2}{2}\pi_1^4(1/s(B) - 1)^4 &= \frac{\pi_1^4}{2}(1/\sqrt{s(B)} - \sqrt{s(B)})^4 \\ &< \frac{\pi_1^4}{2}(1/\sqrt{\pi_1 + t} - \sqrt{\pi_1 + t})^4, \end{aligned} \tag{27}$$

since $F(x) := (1/\sqrt{x} - \sqrt{x})^4$ is decreasing on $(0, 1]$, and $s(B) > \pi_1 + t$. Note that

$$\frac{d}{dt}\left(\frac{1}{\sqrt{\pi_1 + t}} - \sqrt{\pi_1 + t}\right)^4 = -3\left(\frac{1}{\sqrt{\pi_1 + t}} - \sqrt{\pi_1 + t}\right)^3 \left(\frac{1}{2(\pi_1 + t)^{3/2}} + \frac{1}{2\sqrt{\pi_1 + t}}\right). \tag{28}$$

By Taylor's theorem, this implies that

$$(1/\sqrt{\pi_1 + t} - \sqrt{\pi_1 + t})^4 = (1/\sqrt{\pi_1} - \sqrt{\pi_1})^4 - a_2 t + o(t) < (1/\sqrt{\pi_1} - \sqrt{\pi_1})^4 - a_2 t/2$$

for sufficiently small t, where a_2 is the right-hand-side of (28) at $t = 0$. Note further that $(1/\sqrt{\pi_1} - \sqrt{\pi_1})^4 = (\pi_2/\sqrt{\pi_1})^4 = \pi_2^4/\pi_1^2$. Putting these facts together with inequality (27), we obtain

$$g(B) < \frac{\pi_1^4}{2}\frac{\pi_2^4}{\pi_1^2} - a_2 t/2 = \frac{\pi_1}{2}\pi_2^4 - a_2 t/2 = g(C_{1,n}) - a_2 t/2. \tag{29}$$

If inequality (II) is true, $\rho(B) < 1 - t$. If $\rho(B) \leqslant \pi_1$, $(\rho(B) - \pi_1)^4$ is maximized when $\rho(B) = 0$, so that, since $s(B) \leqslant \pi_2$,

$$g(B) \leqslant \frac{\pi_2^2}{2}\pi_1^4 = g(C_{1,n}) + \frac{\pi_2^2}{2}\pi_1^4 - \frac{\pi_1^2}{2}\pi_2^4 = g(C_{1,n}) + \frac{\pi_1^2\pi_2^2}{2}(\pi_2^2 - \pi_1^2) < g(C_{1,n}) - t \tag{30}$$

for sufficiently small t, since π_1 is fixed. If $\rho(B) > \pi_1$, note that inequality (26) implies $s(B) \leqslant \pi_1/s(B)$. Therefore,

$$\begin{aligned} g(B) \leqslant \frac{\pi_1^2}{2\rho(B)^2}(\rho(B) - \pi_1)^4 &= \frac{\pi_1^2}{2}\left(\sqrt{\rho(B)} - \pi_1/\sqrt{\rho(B)}\right)^4 \\ &< \frac{\pi_1^2}{2}\left(\sqrt{1-t} - \pi_1/\sqrt{1-t}\right)^4 \end{aligned} \tag{31}$$

since $G(x) := (\sqrt{x} - \pi_1/\sqrt{x})^4$ is increasing on $(\pi_1, 1]$. A similar Taylor expansion argument to that yielding inequality (29) yields, for a constant a_3 depending only on π_1,

$$g(B) < \frac{\pi_1^2}{2}(1-\pi_1)^4 - a_3 t/2 = g(C_{1,n}) - a_3 t/2, \tag{32}$$

for sufficiently small t. Pulling together inequalities (29), (30), and (32), we have that for all $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{1,n}$, $g(B) < g(C_{1,n}) - a_4$ with $a_4 := \min\{a_2/2, 1, a_3/2\}$.

Suppose $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{3,n}$. Note that $|B| - |C_{2,n}| \leqslant |B \cap C_{1,n}| \leqslant |C_{1,n}|$. Dividing through by $|B|$ yields the useful inequality

$$1 - \pi_2/s(B) \leqslant \rho(B) \leqslant \pi_1/s(B). \tag{33}$$

Subtracting inequality (33) by π_1 gives

$$\pi_2(1 - 1/s(B)) \leqslant \rho(B) - \pi_1^4 \leqslant \pi_1(1/s(B) - 1).$$

Since $\pi_1 < \pi_2$, this implies that $(\rho(B) - \pi_1)^4 \leqslant \pi_2^4(1 - 1/s(B))^4$. Therefore,

$$g(B) \leqslant \frac{s(B)^2}{2}\pi_2^4(1/s(B) - 1)^4 = \frac{\pi_2^4}{2}\left(1/\sqrt{s(B)} - \sqrt{s(B)}\right)^4 < \frac{\pi_2^4}{2}(1/\sqrt{\pi_2} - \sqrt{\pi_2})^4, \tag{34}$$

since $F(x) := (1\sqrt{x} - \sqrt{x})^4$ is decreasing on $I_3 := (\pi_2, 1]$ and $s(B) \in I_3$. Note that $\sqrt{\pi_2} - 1/\sqrt{\pi_2} = -\pi_1/\sqrt{\pi_2}$. Therefore,

$$g(B) < \frac{\pi_2^4}{2}\frac{\pi_1^4}{\pi_2^2} = \frac{\pi_2^2}{2}(0 - \pi_1)^4 = g(C_{2,n}) < g(C_{1,n}) - t$$

for t sufficiently small. Thus, for $a := \min\{a_1, a_4, 1\}$, for sufficiently small t we have $g(B) < g(C_{1,n}) - at$ whenever $B \in \mathcal{R}(t)$. This completes the proof in the case $\pi_1 < \pi_2$.

Part 2 ($\pi_1 = \pi_2$): Recall that when $\pi_1 = \pi_2$ we define $\mathcal{R}(t)$ by

$$\mathcal{R}(t) := \left\{B \subseteq [n] : \max\{|s(B) - \pi_1|,\ \rho(B),\ 1 - \rho(B)\} \leqslant t\right\}.$$

Hence we will use the fact that $B \in \mathcal{R}(t)$ implies at least one of the inequalities (I) $|s(B) - \pi_1| > t$ or (II) $t < \rho(B) < 1 - t$ is true. Define the intervals $I_1 := [0, \pi_1]$, $I_2 := (\pi_1, 1]$. We bisect $2^{[n]}$, the domain of g, with the collections $\mathcal{D}_{i,n} := \{B \subseteq [n] : s(B) \in I_i\}$, for $i = 1, 2$. We will prove that the inequality $g(B) < g(C_{1,n}) - at$ holds for all $B \in \mathcal{R}(t)$ on each of those collections.

Suppose $B \in \mathcal{R}(t)^c \cap \mathcal{D}_{1,n}$ and inequality (I) is true. Then the same derivation yielding inequality (24) gives $g(B) < g(C_{1,n}) - t(1 - \pi_1)^4$ for sufficiently small t. If inequality (II) is true, then

$$(\rho(B) - \pi_1)^4 \leqslant \max\{(1 - t - \pi_1)^4, (\pi_1 - t)^4\} = \max\{(\pi_2 - t)^4, (\pi_1 - t)^4\} = (\pi_2 - t)^4,$$

since $\pi_1 = \pi_2$. Therefore, inequality (25) remains intact. Both inequalities hold on I_2 as well, for the roles of π_1 and π_2 may be interchanged, and the derivations treated symmetrically. This completes the proof in the case $\pi_1 = \pi_2$. ■

A.3 Proof of Lemma 9

Recall the definitions of set modularity and population set modularity from Definitions 2 and 9. Define $W := \sum_{\ell \in [L]} \widehat{Q}_\ell(B)$ and $w := \sum_{\ell \in [L]} q_\ell(B)$. Note that by Part 1 of Lemma 6, $q_\ell(B) \geqslant 0$ regardless of B, a fact which will allow the application of Lemma 16 in what follows. We have $\widehat{H}(B, L) = |L|^{-1} W_+^2$, $H_*(B, L) = |L|^{-1} w^2$, and for any B such that $|B| \geqslant n\varepsilon$,

$$\mathbb{P}_n\left(\left|\widehat{H}(B, L) - H_*(B, L)\right| > \frac{4|L|t}{n^2} + \frac{52|L|}{\kappa n}\right) = \mathbb{P}_n\left(|W_+^2 - w^2| > \frac{4|L|^2 t}{n^2} + \frac{52|L|^2}{\kappa n}\right)$$

$$\leqslant \mathbb{P}_n\left(\max_{\ell \in [L]} \left|\widehat{Q}_\ell(B) - q_\ell(B)\right| > \frac{t}{n^2} + \frac{13}{\kappa n}\right) \leqslant 4|L| \exp\left(-\kappa^2 \frac{\varepsilon t^2}{16 n^2}\right)$$

for large enough $t > 0$, where the first inequality follows from Lemma 16 for large enough n, and the second inequality follows from Lemma 19 and a union bound. Applying a union bound over sets $B \in \mathcal{B}_n$ yields the result. ■

A.4 Proof of Lemma 12

Assume first that $k > 1$. By definition, $B \in N_{n,k}(A)$ implies that at least one of $d_h(B, C_1) \leqslant A \cdot n \cdot b_{n,k-1}$ or $d_h(B, C_2) \leqslant A \cdot n \cdot b_{n,k-1}$ is true. Suppose the first inequality holds. Since $d_h(B, C_1) = |B \setminus C_1| + |C_1 \setminus B|$, we have the inequality

$$\big||B| - n\pi_1\big| = \big||B| - |C_1|\big| \leqslant \big||B| - |B \cap C_1| - |C_1| + |B \cap C_1|\big| \leqslant \big||B| - |B \cap C_1|\big|$$

$$+ \big||C_1| - |B \cap C_1|\big| = |B \setminus C_1| + |C_1 \setminus B| \leqslant A \cdot n \cdot b_{n,k-1}.$$

Alternatively, if $d_h(B, C_2) \leqslant A \cdot n \cdot b_{n,k-1}$, we have the same bound for $\big||B| - n\pi_2\big|$. Therefore, since $\pi_1 \leqslant \pi_2$, $B \in N_{n,k}(A)$ implies that $|B| \geqslant n\pi_1 - A \cdot n \cdot b_{n,k-1}$. Since $b_{n,k-1} = o(1)$ as $n \to \infty$ and $\varepsilon < \pi_1$, this implies that for large enough n, $N_{n,k}(A) \subseteq \mathcal{B}_n(\varepsilon)$. By Lemma 9,

therefore, for large enough n, we have

$$\mathbb{P}_n\left(\sup_{N_{n,k}(A)} \left|\widehat{H}(B, L) - H_*(B, L)\right| > \frac{4|L|t}{n^2} + \frac{52|L|}{\kappa n}\right) \leqslant 4|L||N_{n,k}(A)| \exp\left(-\kappa^2 \frac{\varepsilon t^2}{16 n^2}\right) \tag{35}$$

for all $t > 0$. We now bound the right-hand side of inequality (35) with t replaced by $t_n := n^{1 + \frac{1}{2^k}} (\log n)^{1 - \frac{1}{2^k}}$. Note that

$$\frac{t_n^2}{n^2} = \frac{1}{n^2} n^{2 + \frac{1}{2^{k-1}}} (\log n)^{2 - \frac{1}{2^{k-1}}} = n \cdot n^{\frac{1}{2^{k-1}} - 1} (\log n)^{1 - \frac{1}{2^{k-1}}} \log n = n \cdot b_{n,k-1} \log n.$$

Furthermore, by Corollary 15 (see Appendix B) we have $|N_{n,k}(A)| \leqslant 2 \exp\left[3A \cdot n \cdot b_{n,k-1} \log\left(1/b_{n,k-1}\right)\right]$. These facts yield the bound

$$|N_{n,k}(A)| \exp\left(-\kappa^2 \frac{\varepsilon t_n^2}{16 n^2}\right) \leqslant 2 \exp\left\{-\kappa^2 \frac{\varepsilon}{16} n \cdot b_{n,k-1} \left(\log n - \frac{16}{\kappa^2 \varepsilon} 3A \log\left(1/b_{n,k-1}\right)\right)\right\}$$

$$\leqslant 2 \exp\left(-\kappa^2 \frac{\varepsilon}{32} n \cdot b_{n,k-1} \log n\right) \text{ (for large } n\text{, since } 1/b_{n,k-1} = o(n))$$

$$< 2 \exp\left(-\kappa^2 \frac{\varepsilon}{32} n \gamma_n^{1-\varepsilon} \log n\right),$$

where the final inequality follows from the choice of k satisfying $\frac{1}{2^{k-1}} < \varepsilon$. Therefore,

$$4|L||N_{n,k}(A)| \exp\left(-\kappa^2 \frac{\varepsilon t_n^2}{16 n^2}\right) \leqslant 2 \exp\left\{-\frac{\kappa^2 \varepsilon}{32} n \gamma_n^{1-\varepsilon} \log n + O(\log |L|)\right\} \tag{36}$$

for large enough n. Notice now that $t_n / n^2 = b_{n,k}$ vanishes slower than $1/n$, and is therefore the leading order term in the expression $\frac{4|L|t_n}{n^2} + \frac{52|L|}{\kappa n}$ (see equation 35). Hence for large enough n we have $\frac{4|L|t_n}{n^2} + \frac{52|L|}{\kappa n} \leqslant 5|L| b_{n,k}$. Combining this observation with lines (35) and (36) proves the result in the case $k > 1$.

If $k = 1$, assume $A = \varepsilon$. By definition, then (see Definition 11), $N_{n,k}(A) = \mathcal{B}_n(\varepsilon)$. Returning to inequality (35), we note that $\log |\mathcal{B}_n(\varepsilon)| = O(n)$, and thus we can derive the bound (36) with the same choice of $t_n := n^{1 + \frac{1}{2^k}} (\log n)^{1 - \frac{1}{2^k}} = n\sqrt{n \log n}$. The rest of the argument goes through unaltered. ■

Appendix B. Technical Results

Lemma 14 *Fix $\pi_1 \in [0, 1]$. For each n, let $C_1 \subseteq [n]$ be an index set of size $\lfloor n\pi_1 \rfloor$. Let $C_2 := [n] \setminus C_1$. Let $\gamma_n \in [0, 1]$ be a sequence such that $\gamma_n \to 0$ and $n\gamma_n \to \infty$. Then for large enough n,*

$$|N(C_1, \gamma_n)| \leqslant \exp\{3n\gamma_n \log(1/\gamma_n)\}.$$

Proof Define the boundary of a neighborhood of $C \subseteq [n]$ by

$$\partial N(C, r) := \{B \subseteq [n] : d_h(B, C) = \lfloor nr \rfloor\}.$$

Note that any $B \subseteq [n]$ may be written as the disjoint union $B = \{C_2 \cap B\} \cup \{C_1 \cap B\}$. Since $C_1 \cap B = C_1 \setminus \{C_1 \setminus B\}$, for fixed $k \in [n]$ it follows that each set $B \in \partial N(C, k/n)$ is uniquely identified with choices of $|C_2 \cap B|$ indices from C_2 and $|C_1 \setminus B|$ indices from C_1 such that

$$|B \cap C_2| + |C_1 \setminus B| = |B \setminus C_1| + |C_1 \setminus B| = d_h(B, C_1) = k.$$

Therefore, we have the equality

$$|\partial N(C_1, k)| \;=\; \sum_{m=0}^{k} \left\{ \binom{|C_2|}{m} + \binom{|C_1|}{k-m} \right\}. \tag{37}$$

Note that for positive integers K, N with $K < N/2$, properties of the geometric series yield the following bound:

$$\begin{aligned} \binom{N}{K}^{-1} \sum_{m=0}^{K} \binom{N}{m} &= \sum_{m=0}^{K} \frac{(N-K)!K!}{(N-m)!m!} = \sum_{m=0}^{K} \prod_{j=m+1}^{K} \frac{j}{N-j+1} \\ &< \sum_{m=0}^{K} \left(\frac{K}{N-K+1} \right)^m < \frac{N-(K-1)}{N-(2K-1)}. \end{aligned} \tag{38}$$

For sufficiently small K/N, the right-hand side of inequality (38) is less than 2, and thus $\sum_{m=0}^{K} \binom{N}{m} < 2\binom{N}{K}$ if $K \ll N$. We apply this inequality to equation (37). Choose n large enough so that $n\gamma_n < \frac{1}{2}\min\{|C_1|, |C_2|\}$, which is possible since $\gamma_n \to 0$. Then for fixed $k \leqslant n\gamma_n$, we have that $|\partial N(C_1, k)| < 2\left[\binom{|C_2|}{k} + \binom{|C_1|}{k}\right]$ for large enough n. By another application of the inequality derived from (38), using the fact that $n\gamma_n = o(n)$, we therefore obtain

$$\begin{aligned} |N(C_1, \gamma_n)| = \sum_{k=0}^{\lfloor n\gamma_n \rfloor} |\partial N(C_1, k)| &< \sum_{k=0}^{\lfloor n\gamma_n \rfloor} 2\left\{ \binom{|C_2|}{k} + \binom{|C_1|}{k} \right\} \\ &< 4\left\{ \binom{|C_2|}{\lfloor n\gamma_n \rfloor} + \binom{|C_1|}{\lfloor n\gamma_n \rfloor} \right\} \leqslant 8\binom{n}{\lfloor n\gamma_n \rfloor}. \end{aligned}$$

As $\binom{N}{K} \leqslant \left(\frac{N \cdot e}{K}\right)^K$, we have

$$|N(C_1, \gamma_n)| \leqslant \exp\left(\log(8) + n\gamma_n \left\{\log(e) + \log(1/\gamma_n)\right\}\right) \leqslant \exp(3n\gamma_n \log(1/\gamma_n))$$

for large enough n, since $1/\gamma_n \to \infty$. ■

Here we give a short Corollary to Lemma 14 which directly serves the proof of Lemma 12. Recall $N_{n,k}(A)$ from Definition 11 in Section 4.1.2.

Corollary 15 *Fix an integer $k > 1$. For large enough n,*

$$|N_{n,k}(A)| \leqslant 2\exp\left[3A \cdot n \cdot b_{n,k-1} \log\left(1/b_{n,k-1}\right)\right].$$

Proof The corollary follows from a direct application of Lemma 14 to $N(C_1, A \cdot b_{n,k-1})$ and $N(C_2, A \cdot b_{n,k-1})$. ■

Lemma 16 *Let $x_1, \ldots, x_k \in (0,1)$ be fixed and let $X_1, \ldots, X_k$ be arbitrary random variables. Define $W := \sum_i X_i$ and $w := \sum_i x_i$. Then for t sufficiently small, $\mathbb{P}(|W_+^2 - w^2| > 4k^2 t) \leqslant \mathbb{P}(\max_i |X_i - x_i| > t)$.*

Proof Define $D_i := |X_i - x_i|$ and fix $t < \min_i x_i$. Then if $\max_i D_i \leqslant t$, all X_i's will be positive, and thus $W_+ = W$ and $|W - w| \leqslant kt$, by the triangle inequality. Therefore $\max_i D_i \leqslant t$ implies that

$$|W_+^2 - w^2| = |(W-w)^2 + 2w(W-w)| \leqslant k^2t^2 + 2wkt \leqslant k^2t^2 + 2k^2t. \tag{39}$$

Thus by the law of total probability, we have

$$\mathbb{P}(|W_+^2 - w^2| > 4k^2t) \leqslant \mathbb{P}(\{|W_+^2 - w^2| > 4k^2t\} \cap \{\max_i D_i \leqslant t\}) + \mathbb{P}(\max_i D_i > t).$$

Inequality (39) implies that for sufficiently small t, the first probability on the right-hand side above is equal to 0. The result follows. ■

In what follows we state and prove Lemma 19, a concentration inequality for the modularity of a node set (see Definition 2) from a single-layer SBM with n nodes and two communities. We first give a few short facts about the 2-community SBM. For for all results that follow, let s, ρ, and v (see Definition 4) correspond to the fixed set $B \subseteq [n]$ in each result (though sometimes we will make explicit the dependence on B). Define a matrix V by $V(i,j) := P(i,j)(1 - P(i,j))$ for $i = 1, 2$, where P is the probability matrix associated with the 2-block SBM.

Lemma 17 *Consider a single-layer SBM with $n > 1$ nodes, two communities, and parameters P and π_1. Fix a node set $B \subseteq [n]$ with $|B| \geqslant \alpha n$ for some $\alpha \in (0,1)$. Then*

1. $\left| \mathbb{E}(Y(B)) - \binom{|B|}{2} v^t P v \right| \leqslant 3|B|/2$
2. $\left| \mathbb{E}\left(\sum_{u \in B} \widehat{d}(u) \right) - |B| n v^t P \pi \right| \leqslant |B|$
3. $Var\left(\sum_{u \in B} \widehat{d}(u) \right) \leqslant 9|B|n$

Proof For part 1, note that by definition, $\mathbb{E}\left(Y(B)\right) = \sum_{u,v \in B: u<v} \mathbb{P}\left((u,v) \in \widehat{E}\right) = \frac{1}{2}\sum_{u \neq v: u,v \in B} \mathbb{P}\left((u,v) \in \widehat{E}\right)$ The right-hand sum can be expressed the sum of the entries of a 2×2 symmetric block matrix with zeroes on the diagonal. In this matrix, the upper diagonal block is of size $|B \cap C_1|$ with off-diagonal entries equal to $P(1,1)$. The lower diagonal block is of size $|B \cap C_2|$ with off-diagonal entries equal to $P(2,2)$. The off-diagonal blocks

have entries equal to $P(1,2)$. Therefore, summing over blocks and accounting for the zero diagonal, we have

$$\mathbb{E}\left(Y(B)\right) = \frac{1}{2}\Big(|B\cap C_1|^2 P(1,1) + |B\cap C_1||B\cap C_2|P(1,2) + |B\cap C_2|^2 P(2,2)\Big) - \frac{1}{2}\Big(|B\cap C_1|P(1,1) + |B\cap C_2|P(2,2)\Big).$$

By dividing and multiplying by $|B|^2$ and collapsing cross-products, we get

$$\begin{aligned}\mathbb{E}\left(Y(B)\right) &= \frac{|B|^2}{2}\left(v^tPv - \frac{\rho P(1,1)+(1-\rho)P(2,2)}{|B|}\right)\\ &= \binom{|B|}{2}\left(1+\frac{1}{|B|-1}\right)\left(v^tPv - \frac{\rho P(1,1)+(1-\rho)P(2,2)}{|B|}\right)\\ &= \binom{|B|}{2}\left(v^tPv - \frac{\rho P(1,1)+(1-\rho)P(1,2)}{|B|} + \frac{v^tPv}{|B|-1} - \frac{\rho P(1,1)+(1-\rho)P(2,2)}{|B|(|B|-1)}\right).\end{aligned}$$

Part 1 follows by carrying out the multiplication by $\binom{|B|}{2}$ in the last expression.

For part 2, let $P(\cdot,i)$ denote the i-th column of P. Note that $\mathbb{E}\left(\widehat{d}(u)\right) = n\pi^TP(\cdot,c_u) - P(c_u,c_u)$, with $c_u\in\{1,2\}$ denoting the community index of u. Therefore,

$$\begin{aligned}\mathbb{E}\left(\sum_{u\in B}\widehat{d}(u)\right) &= \sum_{u\in B}\mathbb{E}\left(\widehat{d}(u)\right) = \sum_{u\in B\cap C_1}\mathbb{E}\left(\widehat{d}(u)\right) + \sum_{u\in B\cap C_2}\mathbb{E}\left(\widehat{d}(u)\right)\\ &= |B\cap C_1|\left(n\pi^TP(\cdot,1) - P(1,1)\right) + |B\cap C_2|\left(n\pi^TP(\cdot,2) - P(2,2)\right)\\ &= |B|\left(n\rho\pi^TP(\cdot,1) + n(1-\rho)\pi^TP(\cdot,2) - \rho P(1,1) - (1-\rho)P(2,2)\right)\\ &= |B|nv^tP\pi - |B|\big(\rho P(1,1) + (1-\rho)P(2,2)\big),\end{aligned}$$

which completes part 2.

Finally, for part 3, we have

$$\mathrm{Var}\left(\sum_{u\in B}\widehat{d}(u)\right) = \mathrm{Var}\left(2Y(B)\right) + \sum_{u,v:u\in B,v\in B^C}\mathrm{Var}\left(\widehat{X}(u,v)\right). \tag{40}$$

We address these two terms separately. For the first term, a calculation analogous to that from part 1 yields that $|\mathrm{Var}\left(Y(B)\right) - \binom{|B|}{2}v^tVv| \leqslant 3|B|/2$. Defining $\bar{v} := (\rho(B^C), 1-\rho(B^C))^t$, it is easy to show that $\sum_{u,v:u\in B,v\in B^C}\mathrm{Var}\left(\widehat{X}(u,v)\right) = |B||B^C|v^tV\bar{v}$, which is simply the sum of variances of all edge indicators for edges from B to B^C. Applying these

observations to equation (40), we have

$$\begin{aligned}\mathrm{Var}\left(\sum_{u\in B}\widehat{d}(u)\right) &\leqslant 4\binom{|B|}{2}v^tVv + 12|B|/2 + |B||B^C|v^TV\bar{v}\\ &\leqslant |B|\big(2(|B|-1)v^tVv + 6 + |B^C|v^tV\bar{v}\big) \leqslant 9|B|n.\end{aligned}$$

■

Lemma 18 *Under a single-layer SBM with $n>1$ nodes, two communities, and parameters P and π_1, define $\kappa := \pi^TP\pi$. Then for large enough n, $\mathbb{P}\left(\left|2|\widehat{E}| - n^2\kappa\right| > t+4n\right) \leqslant 2\exp\left\{-\frac{t^2}{n^2}\right\}$ for any $t>0$.*

Proof Note that $|\widehat{E}| = Y([n])$. Thus part 1 of Lemma 17 with $B=[n]$ yields $\left|\mathbb{E}(|\widehat{E}|) - \binom{n}{2}\kappa\right| \leqslant 3n/2$ for large enough n. As $n^2/2 = \binom{n}{2} + n/2$, by the triangle inequality, $\left|\mathbb{E}(|\widehat{E}|) - \frac{n^2}{2}\kappa\right| \leqslant \left|\mathbb{E}(|\widehat{E}|) - \binom{n}{2}\kappa\right| + \frac{n}{2} \leqslant 2n$ Thus for any $t>0$, Hoeffding's inequality gives

$$\begin{aligned}\mathbb{P}\left(\left|2\widehat{E} - n^2\kappa\right| > t+4n\right) &\leqslant \mathbb{P}\left(\left|2\widehat{E} - n^2\kappa\right| > t + 2\left|\mathbb{E}(|\widehat{E}|) - \frac{n^2}{2}\kappa\right|\right)\\ &\leqslant \mathbb{P}\left(\left|2\widehat{E} - 2\mathbb{E}(|\widehat{E}|)\right| > t\right) \leqslant 2\exp\left\{-2\frac{t^2}{4\binom{n}{2}}\right\} \leqslant 2\exp\{-t^2/n^2\}.\end{aligned}$$

■

Lemma 19 *Consider a single-layer 2-block SBM having $n>1$ nodes and parameters P and π. Fix $\alpha\in(0,1)$ and $B\subseteq[n]$ such that $|B|\geqslant \alpha n$. Then for large enough n we have*

$$\mathbb{P}_n\left(\left|\widehat{Q}(B) - q(B)\right| > \frac{t}{n^2} + \frac{8}{\kappa n}\right) \leqslant 4\exp\left(-\frac{\kappa^2\alpha t^2}{16n^2}\right) \tag{41}$$

for any $t>0$.

Proof With notation laid out in Section 2.2, define

$$\widetilde{Q}(B) := n^{-1}\binom{|B|}{2}^{-1/2}\left(Y(B) - \widetilde{\mu}(B)\right), \tag{42}$$

where

$$\widetilde{\mu}(B) := \frac{\sum_{u,v\in B:u<v}\widehat{d}(u)\widehat{d}(v)}{n^2\kappa}. \tag{43}$$

We will prove the inequality in three steps: *Step 1*: bounding $\left|\widehat{Q}(B) - \widetilde{Q}(B)\right|$ in probability; *Step 2*: deriving a concentration inequality for $\widetilde{Q}(B)$; and *Step 3*: showing that $\left|\mathbb{E}\left(\widetilde{Q}(B)\right) - q(B)\right|$ is eventually bounded by a constant.

Step 1. As $\sum_{u,v\in B;u<v} \widehat{d}(u)\widehat{d}(v) \leqslant \sum_{u\in B} \widehat{d}(u)^2$, we have

$$\sum_{u,v\in B;u<v} \widehat{d}(u)\widehat{d}(v) \leqslant \sqrt{\sum_{u,v\in B;u<v} \widehat{d}(u)\widehat{d}(v)}\sqrt{\sum_{u\in B} \widehat{d}(u)^2} \leqslant n\binom{|B|}{2}^{1/2}|\widehat{E}|.$$

Therefore,

$$\left|\widehat{Q}(B) - \widetilde{Q}(B)\right| = n^{-1}\binom{|B|}{2}^{-1/2}\left|\frac{(2|\widehat{E}| - n^2\kappa)\sum_{u,v\in B;u<v} \widehat{d}(u)\widehat{d}(v)}{2|\widehat{E}|n^2\kappa}\right| \leqslant \frac{\left|2|\widehat{E}| - n^2\kappa\right|}{2n^2\kappa}. \quad (44)$$

Combining the inequality in (44) with Lemma 18, for any $t > 0$,

$$\mathbb{P}\left(\left|\widehat{Q}(B) - \widetilde{Q}(B)\right| > \frac{t}{2n^2} + \frac{2}{\kappa n}\right) \leqslant \mathbb{P}\left(\left|2|\widehat{E}| - n^2\kappa\right| > \kappa t + 4n\right)$$

$$\leqslant 2\exp\left(-\frac{\kappa^2 t^2}{n^2}\right). \quad (45)$$

Step 2. This step relies on McDiarmid's concentration inequality. Recall from Section 2.1 that $\widehat{X}(u,v)$ denotes the indicator of edge presence between nodes u and v. Note that node pairs have a natural, unique ordering along the upper-diagonal of the adjacency matrix. Define $ord\{u,v\} = 2(u-1) + (v-1)$, for $\{u,v\} \in [n]^2$ with $u < v$ (e.g. $ord\{1,2\} = 1$, $ord\{1,3\} = 2$, etc.). For all $n > 1$ and $i \leqslant n(n-1)/2$, define $\widehat{Z}(i) := \widehat{X}(u,v)$ such that $ord\{u,v\} = i$. If $ord\{u,v\} = i$, we call $\{u,v\}$ the "i-th ordered node pair". Define the set $\mathcal{I}(B) := \{i : \text{the } i\text{-th ordered node pair has at least one node in } B\}$ and let $\widehat{\mathcal{Z}}(B) := \{\widehat{Z}(i) : i \in I(B)\}$. Note that the proxy score $\widetilde{Q}(B)$ is a function $f(z_1, z_2, \ldots)$ of the indicators $\widehat{\mathcal{Z}}(B)$.

Consider a *fixed* indicator set $\mathcal{Z}(B)$. For each $j \in I(B)$, define $\mathcal{Z}^j(B) := \{Z^j(i) : i \in I(B)\}$ with

$$Z^j(B) := \begin{cases} Z^j(i) = 1 - Z(i), & i = j \\ Z^j(i) = Z(i), i \neq j \end{cases} \quad (46)$$

To apply McDiarmid's inequality, we must bound $\Delta(j) := |f(\mathcal{Z}(B)) - f(\mathcal{Z}^j(B))|$ uniformly over $j \in \mathcal{I}(B)$. Fix $j \in \mathcal{I}(B)$ and let $\{u', v'\}$ be the j-th ordered edge. Without loss of generality, we assume $Z(j) = 1$. Since $f(\mathcal{Z}(B)) = Q(B)$, $f(\mathcal{Z}(B))$ has a representation in terms of $Y(B)$ and $\widetilde{\mu}(B)$. We let $Y^j(B)$ and $\widetilde{\mu}^j(B)$ correspond to $f(\mathcal{Z}(B)^j)$. Notice that

$$n\binom{|B|}{2}^{1/2}\Delta(j) = \left|Y(B) - Y^j(B) - \left(\widetilde{\mu}(B) - \widetilde{\mu}^j(B)\right)\right|. \quad (47)$$

We bound the right hand side of equation (47) in two cases: (i) $u', v' \in B$, and (ii) $u' \notin B, v' \in B$. In case (i), $Y(B) - Y^j(B) = 1$, and

$$\widetilde{\mu}(B) - \widetilde{\mu}^i(B) = \frac{\sum_{u,v\in B;u\neq v} d(u)d(v) - d^j(u)d^j(v)}{n^2\kappa} = \frac{d(u')d(v') - d^j(u')d^j(v')}{n^2\kappa}$$

$$= \frac{d(u')d(v') - (d(u')-1)(d(v')-1)}{n^2\kappa} = \frac{d(u') + d(v') - 1}{n^2\kappa},$$

which is bounded in the interval $(0,1)$ for large enough n. Thus in case (i), $\Delta(j) \leqslant 2\binom{|B|}{2}^{-1/2}$ by the triangle inequality, for large enough n. In case (ii), $Y(B) - Y'(B) = 0$, and

$$\widetilde{\mu}(B) - \widetilde{\mu}^j(B) = \frac{\sum_{u,v\in B;u\neq v} d(u)d(v) - d^j(u)d^j(v)}{n^2\kappa} = \frac{\sum_{u\in B;u\neq v'} d(u)\left(d(v') - d^j(v')\right)}{n^2\kappa}$$

$$= \frac{\sum_{u\in B;u\neq v'} d(u)}{n^2\kappa} \leqslant \frac{n|B|}{n^2\kappa} \leqslant \kappa^{-1}.$$

Hence due to equation (47), we have for sufficiently large n that

$$\Delta(j) \leqslant n^{-1}\binom{|B|}{2}^{-1/2} \cdot \max\{2, \kappa^{-1}\} \leqslant n^{-1}\binom{|B|}{2}^{-1/2} \cdot 2 \cdot \kappa^{-1} \quad (48)$$

for all $j \in \mathcal{I}(B)$, as $\kappa \leqslant 1$. Since $|\mathcal{I}(B)| = \binom{|B|}{2} + |B||B^C| \leqslant n|B|$, McDiarmid's bounded-difference inequality implies that for sufficiently large n,

$$\mathbb{P}\left(\left|\widetilde{Q}(B) - \mathbb{E}\left(\widetilde{Q}(B)\right)\right| > \frac{t}{n}\right) = 2\exp\left(\frac{-t^2}{n|B|\Delta(j)}\right) \leqslant 2\exp\left(-\kappa^2\frac{n^2\binom{|B|}{2}t^2}{4n^3|B|}\right)$$

$$\leqslant 2\exp\left(-\kappa^2\frac{(|B|-1)t^2}{8n}\right) \leqslant 2\exp\left(-\kappa^2\frac{\alpha t^2}{16}\right)$$

for any $t > 0$. Replacing t by t/n gives

$$\mathbb{P}\left(\left|\widetilde{Q}(B) - \mathbb{E}\left(\widetilde{Q}(B)\right)\right| > \frac{t}{n^2}\right) \leqslant 2\exp\left(-\kappa^2\frac{\alpha t^2}{16n^2}\right). \quad (49)$$

Step 3. Turning our attention to $\mathbb{E}(\widetilde{Q}(B))$, recall that $n\binom{|B|}{2}^{1/2}\widetilde{Q}(B) = Y(B) - \widetilde{\mu}(B)$ and that $\widetilde{\mu}(B) := \sum_{u,v\in B;u<v} \widehat{d}(u)\widehat{d}(v)/(n^2\kappa)$. As in previous lemmas, we will shorthand the quantities $s(B), \rho(B)$, and $v(B)$, by s, ρ, and v (respectively). Note that

$$\mathbb{E}\left(2\cdot\sum_{u,v\in B;u<v} \widehat{d}(u)\widehat{d}(v)\right) = \mathbb{E}\left\{\left(\sum_{u\in B}\widehat{d}(u)\right)^2 - \sum_{u\in B}\widehat{d}^2(u)\right\}$$

$$= \mathrm{Var}\left(\sum_{u\in B}\widehat{d}(u)\right) + \mathbb{E}\left(\sum_{u\in B}\widehat{d}(u)\right)^2 - \sum_{u\in B}\mathbb{E}\left(\widehat{d}^2(u)\right). \quad (50)$$

Part 3 of Lemma 17 gives $\text{Var}\left(\sum_{u\in B}\widehat{d}(u)\right) \leqslant 9sn^2$. Furthermore, for $u \in C_i$ we have

$$\begin{aligned}\mathbb{E}\left(\widehat{d}^2(u)\right) &= \text{Var}\left(\widehat{d}(u)\right) + \mathbb{E}\left(\widehat{d}(u)\right)^2 \\ &= n\pi^T V(\cdot,i) - V(i,i) + n^2\left(\pi^T P(\cdot,i) - P(i,i)\right)^2,\end{aligned}$$

and therefore $\sum_{u\in B}\mathbb{E}\left(\widehat{d}^2(u)\right) \leqslant 2sn^3$. Finally, Part 2 of Lemma 17 gives $\left|\mathbb{E}\left(\sum_{u\in B}\widehat{d}(u)\right) - |B|nv^T P\pi\right| \leqslant |B|$. By expansion, this implies there exists a constant a with $|a| < 3$ such that for large enough n, $\mathbb{E}\left(\sum_{u\in B}\widehat{d}(u)\right)^2 = s^2n^4(v^tP\pi)^2 + as^2n^3$. Therefore overall, line (50) implies there exists a constant b with $|b| < 6$ such that for large enough n, $\mathbb{E}\left(2\cdot\sum_{u,v\in B;u<v}\widehat{d}(u)\widehat{d}(v)\right) = s^2n^4(v^TP\pi)^2 + bsn^3$. Therefore, using the definition of $\widetilde{\mu}(B)$,

$$\begin{aligned}\mathbb{E}\left(\widetilde{\mu}(B)\right) &= s^2n^4\frac{(v^tP\pi)^2 + b(sn)^{-1}}{2n^2\kappa} = \binom{sn}{2}\left(1+\frac{1}{sn-1}\right)\left(\frac{(v^tP\pi)^2}{\kappa} + \frac{b}{\kappa sn}\right) \\ &= \binom{sn}{2}\left(\frac{(v^tP\pi)^2}{\kappa} + \frac{b}{\kappa sn} + \frac{(v^tP\pi)^2}{\kappa(sn-1)} + \frac{b}{\kappa sn(sn-1)}\right) \\ &= \binom{sn}{2}\left(\frac{(v^tP\pi)^2}{\kappa} + \frac{1}{\kappa sn}\left(b + \frac{sn(v^tP\pi)^2 + b}{sn-1}\right)\right) \\ &= \binom{sn}{2}\left(\frac{(v^tP\pi)^2}{\kappa} + \frac{c_1}{\kappa sn}\right) \qquad (51)\end{aligned}$$

for a constant c_1 with $|c_1| < 8$, for large enough n. Now, part 1 of Lemma 17 gives that $\left|\mathbb{E}(Y(B)) - \binom{|B|}{2}v^tPv\right| \leqslant 3|B|/2$ for large enough n. Thus there exists a constant c_2 with $|c_2| < 3$ such that for large enough n, $\mathbb{E}(Y(B)) = \binom{|B|}{2}\left(v^tPv + \frac{c_2}{sn}\right)$. Thus

$$\begin{aligned}n\mathbb{E}\left(\widetilde{Q}(B)\right) &= \binom{|B|}{2}^{-1/2}\left(\mathbb{E}(Y(B)) - \mathbb{E}(\tilde{\mu}(B))\right) = \frac{sn}{\sqrt{2}}\left(v^tPv - \frac{(v^tP\pi)^2}{\kappa} + \frac{1}{sn}\left(c_1/\kappa + c_2\right)\right) \\ &\quad * \left(\sqrt{1-\frac{1}{sn}}\right) = \frac{sn}{\sqrt{2}}\left(v^tPv - \frac{(v^tP\pi)^2}{\kappa}\right) + \frac{c_1/\kappa + c_2}{\sqrt{2}}\left(\sqrt{1-\frac{1}{sn}}\right).\end{aligned}$$

Thus there exists a constant c with $|c| \leqslant |c_1|/\kappa + |c_2| < 8/\kappa + 3$ such that for large enough n, $\mathbb{E}(\widetilde{Q}(B)) = q(B) + c/n$. This completes Step 3.

Completion of the proof: We now recall the results of the three steps:

(i) For large enough n, we have $\mathbb{P}\left(\left|\widehat{Q}(B) - \widetilde{Q}(B)\right| > \frac{t}{2n^2} + \frac{2}{\kappa n}\right) \leqslant 2\exp\left(-\frac{\kappa^2t^2}{n^2}\right)$

(ii) $\mathbb{P}\left(\left|\widetilde{Q}(B) - \mathbb{E}\left(\widetilde{Q}(B)\right)\right| > \frac{t}{n^2}\right) \leqslant 2\exp\left(-\kappa^2\frac{\alpha t^2}{16n^2}\right)$

(iii) There exists c with $|c| < 8/\kappa + 3$ such that for large enough n, $\mathbb{E}\left(\widehat{Q}(B)\right) = q(B) + c/n$

Noting that $\alpha/16 < 1$, we apply a union bound to the results of steps (i) and (ii):

$$\mathbb{P}\left(\left|\widehat{Q}(B) - \mathbb{E}\left(\widetilde{Q}(B)\right)\right| > \frac{t}{n^2} + \frac{2}{\kappa n}\right) \leqslant 4\exp\left(-\frac{\kappa^2\alpha t^2}{16n^2}\right) \qquad (52)$$

Applying the inequality $|x-a| \geqslant |x| - |a|$ with (iii) and some algebra gives the result. ■

Appendix C. Competing Methods

In Sections 6 and 7, we compare and contrast the performance of Multilayer Extraction with the following methods:

Spectral clustering (Newman, 2006a): an iterative algorithm based on the spectral properties of the modularity matrix of an observed network. In the first step, the modularity matrix of the observed network is calculated and its leading eigenvector is identified. The graph is divided into two disjoint communities so that each vertex is assigned according to its sign in the leading eigenvector. Next, the modularity matrix is calculated for both of the subgraphs corresponding to the previous division. If the modularity of the partition increases, these communities are once again divided into two disjoint communities, and the procedure is repeated in this fashion until the modularity no longer increases. For the desired igraph object `graph`, the call for this in `R` was:

```
cluster_leading_eigen(graph, steps = -1, weights = NULL, start = NULL,
options = arpack_defaults, callback = NULL, extra = NULL, env = parent.frame())
```

Label Propagation (Raghavan et al., 2007): an iterative algorithm based on propagation through the network. At the first step, all vertices are randomly assigned a community label. Sequentially, the algorithm chooses a single vertex and updates the labels of its neighborhood to be the majority label of the neighborhood. The algorithm continues updating labels in this way until no updates are available. For the desired igraph object `graph`, the call for this in `R` was:

```
cluster_label_prop(graph, weights = E(graph)$weight, initial = NULL,
fixed = NULL)
```

Fast and greedy (Clauset et al., 2004): an iterative and greedy algorithm that seeks a partition of vertices with maximum modularity. The algorithm is an agglomerative approach that is a modification of the Kernighan-Lin algorithm commonly used in the identification of community structure in network. For the desired igraph object `graph`, the call for this in `R` was:

```
cluster_fast_greedy(graph, merges = TRUE, weights = E(graph)$weight,
modularity = TRUE, membership = TRUE, weights = NULL)
```

Walktrap (Pons and Latapy, 2005): an agglomerative algorithm that seeks a partition of vertices that minimizes the total length of a random walk within each community. At the first stage, each vertex of the network is placed in its own community. At each subsequent stage, the two closest communities (according to walk distance) are merged. This process is continued until all vertices have been merged into one large community, and a community dendrogram is formed. The partition with the smallest random walk distance is chosen as the final partition.

```
cluster_walktrap(graph, weights = E(graph)$weight, steps = 4,
 merges = TRUE, modularity = TRUE, membership = TRUE)
```

GenLouvain (Jutla et al., 2011): a multilayer generalization of the iterative GenLouvain algorithm. This algorithm seeks a partition of the vertices and layers that maximizes the multilayer modularity of the network, as described in (Mucha et al., 2010). We use the MATLAB implementation from (Jutla et al., 2011) of GenLouvain with resolution parameter set to 1, and argument `randmove = moverandw`.

Infomap (De Domenico et al., 2014): a multilayer generalization of the Infomap algorithm from Rosvall and Bergstrom (2008). This algorithm seeks to identify a partition of the vertices and layers that minimize the generalized map equation, which measures the description length of a random walk on the partition. We use the C++ multiplex implementation of Infomap provided at `http://www.mapequation.org/code.html`. In implementation, we set the arguments of the function to permit overlapping communities, and set the algorithm to ignore self-loops.

For the first four methods, we use the default settings from the *igraph* package version 0.7.1 set in `R`.

Appendix D. Extraction Simulations

D.1 Simulation

We now investigate several intrinsic properties of Multilayer Extraction by applying the method to multilayer networks with several types of community structure, including I) disjoint, II) overlapping, III) persistent, IV) non-persistent, and V) hierarchical structure. Figure 7 illustrates six multilayer networks that we analyze for this purpose. Each simulated network contains 1000 nodes and 90 layers. Embedded communities have inner connection probability 0.15; whereas, the remaining vertices independently connected to all other vertices with probability 0.05.

D.2 Results

In the disjoint, overlapping, persistent, and non-persistent networks (I, II, III, and IV, respectively), Multilayer Extraction identifies communities that perfectly match the true embedded communities. On the other hand, in the hierarchical community setting, Multilayer Extraction is unable to identify the full set of communities. In example V, Multilayer

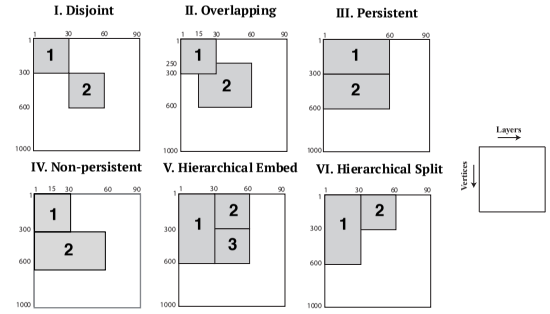


Figure 7: Simulation test bed for extraction procedures. Each graphic displays a multilayer network on 1000 nodes and 90 layers. In each plot, shaded rectangles are placed over the nodes (rows) and layers (columns) that are included in a multilayer community. Communities are labeled by number. Vertices within the same community are randomly connected with probability 0.15 while all other vertices have connection probability 0.05 to vertices in their respective layer.

Extraction does not identify community 1, and in example VI Extraction identifies a community with vertices 1 - 300 across layers 1 - 60, which combines community 1 and community 2.

Together, these results suggest two properties of the Multilayer Extraction procedure. First, the method can efficiently identify disjoint and overlapping community structure in multilayer networks with heterogeneous community structure. Second, Multilayer Extraction tends to disregard communities with a large number of vertices (e.g. communities that include over half of the vertices in a network). The inverse relationship between the score and the number of vertices in a community may provide some justification as to why this is the case. In networks with large communities, one can in principle modify the score by introducing a reward for large collections. We plan to pursue this further in future research.

References

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014, 2008.

Gary D Bader and Christopher WV Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC bioinformatics*, 4(1):2, 2003.

Matteo Barigozzi, Giorgio Fagiolo, and Giuseppe Mangioni. Identifying the community structure of the international-trade multi-network. *Physica A: statistical mechanics and its applications*, 390(11):2051–2066, 2011.

Danielle S. Bassett, Nicholas F. Wymbs, Mason A. Porter, Peter J. Mucha, Jean M. Carlson, and Scott T. Grafton. Dynamic reconfiguration of human brain networks during learning. *Proceedings of the National Academy of Sciences*, 108(18):7641 7646, May 2011. doi: 10.1073/pnas.1018985108.

E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory, Series A*, 24(3):296–307, 1978.

Michele Berlingerio, Michele Coscia, and Fosca Giannotti. Finding redundant and complementary communities in multidimensional networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2181–2184. ACM, 2011.

Michele Berlingerio, Fabio Pinelli, and Francesco Calabrese. Abacus: frequent pattern mining-based community discovery in multidimensional networks. *Data Mining and Knowledge Discovery*, 27(3):294–320, 2013.

Peter J Bickel and Aiyou Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50): 21068–21073, 2009.

Stefano Boccaletti, G Bianconi, R Criado, CI Del Genio, J Gómez-Gardeñes, M Romance, I Sendina-Nadal, Z Wang, and M Zanin. The structure and dynamics of multilayer networks. *Physics Reports*, 544(1):1–122, 2014.

B. Bollobás and A. Universitet. *A probabilistic proof of an asymptotic formula for the number of labelled regular graphs*. Aarhus Universitet, 1979.

Ulrik Brandes, Daniel Delling, Marco Gaertler, Robert Görke, Martin Hoefer, Zoran Nikoloski, and Dorothea Wagner. Maximizing modularity is hard. *arXiv preprint physics/0608255*, 2006.

Alessio Cardillo, Jesús Gómez-Gardeñes, Massimiliano Zanin, Miguel Romance, David Papo, Francisco Del Pozo, and Stefano Boccaletti. Emergence of network features from multiplexity. *Scientific reports*, 3, 2013.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

Aaron Clauset. Finding local community structure in networks. *Physical review E*, 72(2): 026132, 2005.

Amin Coja-Oghlan and André Lanka. Finding planted partitions in random graphs with general degree distributions. *SIAM Journal on Discrete Mathematics*, 23(4):1682–1714, 2009.

Manlio De Domenico, Albert Solé-Ribalta, Emanuele Cozzo, Mikko Kivelä, Yamir Moreno, Mason A Porter, Sergio Gómez, and Alex Arenas. Mathematical formulation of multilayer networks. *Physical Review X*, 3(4):041022, 2013.

Manlio De Domenico, Andrea Lancichinetti, Alex Arenas, and Martin Rosvall. Identifying modular flows on multilayer networks reveals highly overlapping organization in social systems. *arXiv preprint arXiv:1408.2925*, 2014.

Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011.

Dario Fasino and Francesco Tudisco. Modularity bounds for clusters located by leading eigenvectors of the normalized modularity matrix. *arXiv preprint arXiv:1602.05457*, 2016.

Simone Ferriani, Fabio Fonti, and Raffaele Corrado. The social and economic bases of network multiplexity: Exploring the emergence of multiplex ties. *Strategic Organization*, 11(1):7–34, 2013.

Stephen E Fienberg, Michael M Meyer, and Stanley S Wasserman. Analyzing data from multivariate directed graphs: An application to social networks. Technical report, DTIC Document, 1980.

Stephen E Fienberg, Michael M Meyer, and Stanley S Wasserman. Statistical analysis of multiple sociometric relations. *Journal of the American Statistical Association*, 80(389): 51–67, 1985.

S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

David F Gleich and C Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 597–605. ACM, 2012.

D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, page 176183, 2010. doi: 10.1109/ASONAM.2010.17.

Qiuyi Han, Kevin S Xu, and Edoardo M Airoldi. Consistent estimation of dynamic and multi-layer networks. *arXiv preprint arXiv:1410.8597*, 2014.

Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.

Inderjit S Jutla, Lucas GS Jeub, and Peter J Mucha. A generalized louvain method for community detection implemented in matlab. *URL http://netwiki. amath. unc. edu/GenLouvain*, 2011.

Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato. Finding statistically significant communities in networks. *PloS one*, 6(4):e18961, 2011.

Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008.

Anna CF Lewis, Nick S. Jones, Mason A. Porter, and Charlotte M. Deane. The function of communities in protein interaction networks at multiple scales. *BMC Systems Biology*, 4 (1):114, Dec 2010. ISSN 1752-0509. doi: 10.1186/1752-0509-4-100.

Elchanan Mossel, Joe Neeman, and Allan Sly. Stochastic block models and reconstruction. *arXiv preprint arXiv:1202.1499*, 2012.

Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

Raj Rao Nadakuditi and Mark EJ Newman. Graph spectra and the detectability of community structure in networks. *Physical review letters*, 108(18):188701, 2012.

Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006a.

Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006b.

M.E.J. Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004.

Jukka-Pekka Onnela, Samuel Arbesman, Marta C. Gonzlez, Albert-Lszl Barabsi, and Nicholas A. Christakis. Geographic constraints on social network groups. *PLoS ONE*, 6 (4):e16939, Apr 2011. doi: 10.1371/journal.pone.0016939.

John Palowitch, Shankar Bhamidi, and Andrew B Nobel. The continuous configuration model: A null for community detection on weighted networks. *arXiv preprint arXiv:1601.05630*, 2016.

Kaveri S Parker, James D Wilson, Jonas Marschall, Peter J Mucha, and Jeffrey P Henderson. Network analysis reveals sex-and antibiotic resistance-associated antivirulence targets in clinical uropathogens. *ACS Infectious Diseases*, 1(11):523–532, 2015.

Subhadeep Paul and Yuguo Chen. Community detection in multi-relational data with restricted multi-layer stochastic blockmodel. *arXiv preprint arXiv:1506.02699*, 2015.

Subhadeep Paul and Yuguo Chen. Null models and modularity based community detection in multi-layer networks. *arXiv preprint arXiv:1608.00623*, 2016.

Tiago P Peixoto. Inferring the mesoscale structure of layered, edge-valued, and time-varying networks. *Physical Review E*, 92(4):042807, 2015.

Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005*, pages 284–293. Springer, 2005.

Mason A Porter, Jukka-Pekka Onnela, and Peter J Mucha. Communities in networks. *Notices of the AMS*, 56(9):1082–1097, 2009.

Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

Matthew Rocklin and Ali Pinar. On clustering on graphs with multiple edge types. *Internet Mathematics*, 9(1):82–112, 2013.

M. Rosvall and C.T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

Tom AB Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.

Olaf Sporns. *Networks of the Brain*. MIT press, 2011.

Natalie Stanley, Saray Shai, Dane Taylor, and Peter Mucha. Clustering network layers with the strata multilayer stochastic block model. *IEEE*, 2016.

Emanuele Strano, Saray Shai, Simon Dobson, and Marc Barthelemy. Multiplex networks in metropolitan areas: generic features and local effects. *Journal of The Royal Society Interface*, 12(111):20150651, 2015.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.

Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.

Stanley Wasserman and Joseph Galaskiewicz. *Advances in social network analysis: Research in the social and behavioral sciences*, volume 171. Sage Publications, 1994.

James Wilson, Shankar Bhamidi, and Andrew Nobel. Measuring the statistical significance of local connections in directed networks. In *NIPS Workshop on Frontiers of Network Analysis: Methods, Models, and Applications.*, 2013.

James D Wilson, Simi Wang, Peter J Mucha, Shankar Bhamidi, and Andrew B Nobel. A testing based extraction algorithm for identifying significant communities in networks. *The Annals of Applied Statistics*, 8(3):1853–1891, 2014.

James D Wilson, Matthew J Denny, Shankar Bhamidi, Skyler J Cranmer, and Bruce A Desmarais. Stochastic weighted graphs: Flexible model specification and simulation. *Social Networks*, 49:37–47, 2017.

Y. Zhao, E. Levina, and J. Zhu. Community extraction for social networks. *Proceedings of the National Academy of Sciences*, 108(18):7321–7326, 2011.

Yunpeng Zhao, Elizaveta Levina, Ji Zhu, et al. Consistency of community detection in networks under degree-corrected stochastic block models. *The Annals of Statistics*, 40 (4):2266–2292, 2012.